資料處理教育用
84－21

'83鑛工業動態·生産雇傭給與·原材料

電算處理開發報告書

經濟企劃院 調査統計局

일 러 두 기

이 책은 鑛工業動態調査의 資料處理에 關한 指針書입니다. 資料處理過程이나 Program에 疑問點이 있으면 資料處理課로 問議하시기 바랍니다.

資料處理課長 崔 燉 哲

電話構內 (304) 李 慶 義

(308) 許 洪

嚴 斗 鎔

崔 鉉 哲

Ⅰ. 천 공 요 령 서

천 공 요 령 서

년 월 일 작성자

COLUMN 위치		FIELD 명	문 자	천 공 요 령
최 초	최 후		H·A·N	
		〈 1 CARD 〉		사업체에 관한 사항
1	5	조사구 및 사업체번호	N	
6	6	카드번호	N	
7	8	년		
9	13	산업분류	N	
14	14	종업원규모	N	
15	15	유고사항	※ N	
16	75	년		
76	76	조사표구분	N	
77	78	월	N	
79	79	년		
80	80	삭제코드	N	
		〈 2 CARD 〉		제품에 관한 사항 천공
1	5	조사구 및 사업체번호	N	
6	6	카드번호	N	
7	8	일련번호	N	
9	13	품목번호	N	
14	14	유고사항	※	

COLUMN위치		FIELD 명	문 자	천 공 요 령
최 초	최 후		H·A·N	
15	21	월초재고량	N	LEFT 천공
		(월초재고액)	N	
22	28	생산량	N	
		(생산액)	N	
29	35	자가소비량	N	
		(시판)	N	
36	42	시판	N	
		(수출)	N	
43	49	수출	N	
		(과부족보정액)	N	
50	56	과부족보정량	N	
		(월말재고액)	N	
57	63	월말재고량	N	
		~~b~~		
64	64	생산증감요인	※ N	
65	65	출하증감요인	※ N	
66	75	~~b~~		
76	76	조사표종류	N	
77	78	월		
79	79	~~b~~		
80	80	삭제코드	N	

COLUMN 위치		FIELD 명	문 자	천 공 요 령
최 초	최 후		H·A·N	
		〈 3 CARD 〉		원재료에 관한 사항 천공
1	5	조사구 및 사업체번호	N	
6	6	카드번호	N	
7	8	일련번호	N	
9	12	원재료번호	N	
13	13	유고사항	※	
14	14	국산수입구분	N	
15	21	월중사용량	N	LEFT 천공
22	28	월말재고량	N	
29	75	~~b~~		
76	76	조사표종류	N	
77	78	월	N	
79	79	~~b~~		
80	80	삭제코드	N	
		〈 4 CARD 〉		고용, 급여 및 조업상황
1	5	조사구 및 사업체번호	N	
6	6	카드번호	N	
7	8	~~b~~		
		월말 상용종업원수		
9	13	생산직	N	LEFT 천공
57	63	월말재고량	N	LEFT 천공

COLUMN 위치		FIELD 명	문 자	천 공 요 령
최 초	최 후		H·A·N	
		ø		
64	64	생산증감요인	※ N	
65	65	출하증감요인	※ N	
66	75	ø		
76	76	조사표종류	N	
77	78	월		
79	79	ø		
80	80	삭제코드	N	
		〈 3 CARD 〉		원재료에 관한 사항 천공
1	5	조사구 및 사업체번호	N	
6	6	카드번호	N	
7	8	일련번호	N	
9	12	원재료번호	N	
13	13	유고사항	※	
14	14	국산수입구분	N	
15	21	월중사용량	N	LEFT 천공
22	28	월말재고량	N	
29	75	ø		
76	76	조사표종류	N	
77	78	월	N	
79	79	ø		
80	80	삭제코드	N	

COLUMN 위치		FIELD 명	문 자	천 공 요 령
최 초	최 후		H·A·N	
		〈 4 CARD 〉		고용, 급여 및 조업상황
1	5	조사구 및 사업체번호	N	
6	6	카드번호	N	
7	8	∅		
		월말 상용종업원수		
9	13	생산직	N	LEFT 천공
57	63	월말재고량	N	LEFT 천공
		∅		
64	64	생산증감요인	※ N	
65	65	출하증감요인	※ N	
66	75	∅		
76	76	조사표종류	N	
77	78	월		
79	79	∅		
80	80	삭제코드	N	
		〈 3 CARD 〉		원재료에 관한 사항 천공
1	5	조사구 및 사업체번호	N	
6	6	카드번호	N	
7	8	일련번호	N	
9	12	원재료번호	N	
13	13	유고사항	※	

COLUMN 위치		FIELD 명	문 자	천 공 요 령
최 초	최 후		H·A·N	
14	14	국산수입구분	N	
15	21	월중사용량	N	LEFT 천공
22	28	월말재고량	N	
29	75	ƀ		
76	76	조사표종류	N	
77	78	월	N	
79	79	ƀ		
80	80	삭제코드	N	
		〈 4 CARD 〉		고용, 급여 및 조업상황
1	5	조사구 및 사업체번호	N	
6	6	카드번호	N	
7	8	ƀ		
		월말 상용종업원수		
9	13	생산직	N	LEFT 천공
14	18	사무직 및 기타	N	LEFT 천공
19	23	자영업주 및 무급가족 종사자	N	
24	28	합 계	N	
29	33	월중 입직자 수	N	
34	41	생산직 급여총액	N	
42	49	사무직 및 기타 급여총액	N	
50	57	합계 급여총액	N	

COLUMN 위치		FIELD 명	문 자 H·A·N	천 공 요 령
최 초	최 후			
58	75	ƀ		
76	76	조사표종류	N	
77	78	월	N	
79	79	ƀ		
80	80	삭제코드	N	
		〈 5 CARD 〉		고용, 급여 및 조업상황
1	5	조사구 및 사업체번호	N	
6	6	카드번호	N	
7	8	ƀ		
9	16	생산직 특별급여액	N	LEFT 천공
17	24	사무직 특별급여액	N	
25	32	합 계 특별급여액	N	
33	40	월중 이직자 수	N	
41	48	월중 연근무 인원수	N	
49	56	월중 연근무 시간수	N	
57	75	ƀ		
76	76	조사표종류	N	
77	78	월	N	
79	79	ƀ		
80	80	삭제코드	N	

COLUMN 위치		FIELD 명	문 자	천 공 요 령
최 초	최 후		H·A·N	
		〈 6 CARD 〉		
1	5	조사구 및 사업체번호	N	
6	6	카드번호	N	
7	8	일련번호	N	
9	13	품목번호	N	
14	14	유고사항	※ N	
15	21	생산능력	N	LEFT 천공
22	28	생산량	N	
29	75	ø		
76	76	조사표종류	N	
77	78	월	N	
79	79	ø		
80	80	삭제코드	N	

주의사항

① ※표시된 것은 수정시 'ø'로 하려면 '—'를 천공함.

② 2카드에서 (　　)내의 사항은 조사표종류가 '2'인 경우의 내용임.

③ 삭제코드는 수정시에만 삭제할 때 '1'을 천공함.

RECORD 서 식

업무명 광동지수 SMIJN12 SMIGM12 년 월 일 작성자

COLUMN 위치		FIELD 명	PROGRAM중	FIELD	문 자	비 고
최 초	최 후		약 기 호	길 이	H.A.N	
1	5	사업체 번호	SAUP	5	N	
6	6	카드 번호	CD	1	N	
1카드						
7	8	'φφ'	NUMB	2	N	
9	13	산업분류	INDUST	5	N	
14	14	종업원 규모	SIZE	1	N	
15	15	유고사항	YUGO	1	N	
16	16	조사표 종류	TAB	1	N	
17	18	월	MONTH	2	N	
19	20	승 수	RATE	2	N	
21	50	ϕ				
2카드						
7	8	일련번호	NUMB	2	N	
9	13	품목번호	PUM	5	N	
14	14	유고사항	YUGO 2	1	N	
15	18	월초재고	DAT2(7)	4	FIXED	
19	22	생 산 량			BIN(31)	
23	26	자가소비				

COLUMN 위치		FIELD 명	PROGRAM 중	FIELD	문 자	비 고
최 초	최 후		약 기 호	길 이	H.A.N	
27	30	시판				
31	34	수출				
35	38	과부족 보정				
39	42	월말재고		4	FIXED BIN (31)	
43	43	생산증감요인	GRT (2)	1	N	
44	44	출하증감요인		1	N	
45	50	ƀ				
3 카드						
17	8	일련번호	NUMB	2		
9	12	원재료번호	WONNO	4		
13	13	유고사항	YUGO3	1		
14	14	국산수입구분	GUBUN	1		
15	18	사용량	DAT3 (2)	4	FIXED BIN (31)	
19	22	재고량		4		
23	50	ƀ				
4 카드						
7	8	‘ϕϕ’	NUMB	2	-	
9	12	생산종업원수	DAT4 (10)	4	FIXED BIN (31)	
13	16	ϕ		4		
17	20	사무직 종업원수		4		
21	24	무급종사자		4		

COLUMN 위치		FIELD 명	PROGRAM 중	FIELD	문자	비고
최초	최후		약기호	길이	H.A.N	
25	28	총종업원수		4		
29	32	입직자수		4		
33	36	생산직급여액		4		
37	40	ϕ		4		
41	44	사무직급여액		4		
45	48	총급여액		4		
49	50	ϕ				
5 카드						
7	8	'ϕϕ'	NUMB	2		
9	12	생산직 특별급여액	DAT 5 (7)	4	FIXED BIN (31)	
13	16	ϕ		4		
17	20	사무직 특별급여액		4		
21	24	총특별급여액		4		
25	28	이직자수		4		
29	32	연근무인원수		4		
33	36	연근무시간수		4		
37	50	ϕ		4		
6 카드						
7	8	일련번호	NUMB	2		
9	13	품목번호	CAPAG	5		

COLUMN 위치		FIELD 명	PROGRAM 중	FIELD	문 자	비 고
최 초	최 후		약 기 호	길 이	H.A.N	
14	14	유고사항	YYGO 6	1		
15	18	생산능력	DAT 6 (2)	4	FIXED BIN (31)	
19	22	생 산 량		4		
23	50	ƀ				
1	1	카 드	CD	1		
2	2	구 분	GU	1		
3	5	산업분류	SAN	3		
6	10	품목번호	PUM	5		
11	11	특수분류	SPC	1		
12	13	년 도	YR	2		
14	15	월	MN	2		
16	19	가중치	WEIGHT	4	FLOAT	
20	23	기준물량	BASEM	4	FIXED BIN (31)	
24	27	물량 37 개	MUL	4	FIXED BIN (31)	
28	31	지수	IIJU	4	FLOAT	
320	320	ƀ				

Ⅱ. 기 계 내 검 요 령 서

기 계 내 검 요 령 서

기 계 내 용 검 사

CHECK 항목	정 CODE	카드	COLUMN 위치		ERROR 기호
			최 초	최 후	
종업원규모		1	14	14	(공 통) '✕' 표시는 금월조사표 자체에 ERROR 가 있는 것. '$' 표시는 전월조사표와 비교시에 ERROR 가 있는 것.
유고사항		1	15	15	
조사표종류		1	16	16	
일련번호		2	7	8	
		3	7	8	
		6	7	8	
품목번호, 원재료번호		2	9	13	
		3	9	12	
		6	9	13	
유고사항		2	14	14	
		3	13	13	
		6	14	14	
월초재고량		2	15	18	
생산량 및 출하량		2	19	22	
		2	합 { 27 31	30 34	

요 령 (DATA CHECK 요령)

업무명 : 광동지수　　년　월　일　작성자

ERROR CODE 처 리	비 고
① 종업원수가 200인 미만인데 '1'로 되었는가?	
② 〃 400인 이상인데 '2'로 〃 ?	
③ 전월과 규모가 다른가?	
① '2'카드, '6'카드가 하나도 없는가?	
② 전월에 '1'카드나 '4'카드가 빠졌는가?	
① 산업이 '322'나 '3520'인데 '1'로 되었는가?	
② 〃 〃 이 아닌데 '2'로 되었는가?	
③ 전월과 조사표종류가 다른가?	
① 번호순서가 맞는가?	
① 번호가 중복되었는가?	원재료번호는 원재료번호와 국산수입구분번호를 합친 것으로 비교
② 조사대상번호인가?	
③ 전월에 없던 품목이 새로 들어왔는가?	
① 후업, 폐업, 전출인데 생산, 출하가 있는가?	
② 신규품목인가?	
전월의 월말재고량과 일치하는가?	
① 대기업인 경우 $0.5 < \frac{\text{금월물량}}{\text{전월물량}} < 5\%$의 범위내에 있는가?	

CHECK 항목	정 CODE	카드	COLUMN 위치		ERROR 기호
			최 초	최 후	
월말재고량		2	39	42	
증감요인		2	43	44	
합 계		4	25	28	
		4	45	48	
		5	21	24	
입직자수		4	29	32	
총급여액		4	33	36	
		4	41	44	
		4	45	48	
연근무인원수		5	29	32	
연근무시간수		5	33	36	
월말종업원수		4	9	12	
연근무인원수		4	17	20	
		5	29	32	

ERROR CODE 처 리	비 고
② 중소기업인 경우 $0.2 < \frac{\text{금월물량}}{\text{전월물량}} < 10$ 의 범위내에 있는가? 월초재고량+생산-자가소비-시판-수출±과부족보정량 = 월말재고의 식이 맞는가? ① 증가인데 홀수가 아닌가? ② 감소인데 짝수가 아닌가? 생산직+사무직+자영 및 무급=합계의 식이 맞는가? 전월말종업원수+입직자-이직자=금월말종업원수의 식이 맞는가? $50 < \frac{\text{총급여액}-\text{특별급여액}}{\text{월말종업원수}} < 1{,}000$ 의 범위내에 있는가? $15 < \frac{\text{연근무인원수}}{\text{생산종업원수}} < \begin{matrix}28\\30\\31\end{matrix}$ 의 범위내에 있는가? $4 < \frac{\text{연근무시간수}}{\text{연근무인원수}} < 13$ 의 범위내에 있는가? ① 종업원 200인 이상인 경우 $0.8 < \frac{\text{금월}}{\text{전월}} < 1.2$ 의 범위내에 있는가? ② 종업원 100～199인 경우 $0.5 < \frac{\text{금월}}{\text{전월}} < 1.5$ 의 범위내에 있는가?	

CHECK 항목	정 CODE	카드	COLUMN 위 치		ERROR 기호
			최 초	최 후	
생 산직총급여액		4	33	36	
사무종업원수		4	17	20	
생 산능력		6	15	18	
생 산량		6	19	22	

ERROR CODE 처 리	비 고
③ 종업원 50～99인 경우 $0.3 < \frac{금월}{전월} < 2$의 범위내에 있는가? ④ 종업원 49인 이하의 경우 $0.1 < \frac{금월}{전월} < 3$의 범위내에 있는가? $0.5 < \frac{금월(총급여-특별급여)}{전월(총급여-특별급여)} < 2$의 범위내에 있는가? 사무종업원수와 자영무급종사자가 모두 '0'인가? ① $\frac{생산량}{생산능력} < 2$의 범위내에 있는가? ② $0.8 < \frac{금월생산능력}{전월생산능력} < 1.2$의 범위내에 있는가? $0.5 < \frac{금월생산량}{전월생산량} < 2$의 범위내에 있는가?	

STEP 별

RUN NAME	STEP 명	언 어	작성자	처 리 개 요	추정처리 시간·분	FILE 명
CR12	CREATION	PL/1	방윤화	천공내용을 KEY-CHECK하여 CREATION(VSAM)	10분	SMiGM12 SMiGM12
ED12	EDITING	PL/1	〃	기계내검	1시간	SMiGM12 SMiCV12 SMiOK12
OS12	OK SELECT	〃	〃	기계내검후 OK된것은 따로 SELECT함	3분	SMiOK12 SMiGM12
UP12	Update	〃	〃	ERROR 내용수정후 천공된것을 Update	10분	
CH12	카드체크	〃	〃	금월과 전월을 비교하여 카드누락분체크	3분	SMiGM12 SMiGM12 SMiJN12
OE12	OK ERROR구분	〃	〃	ERROR사업체번호만 카드천공하여 구분함	3분	SMiGM12
OU12	확정치수정	〃	〃	이미 확정된 자료를 수정하거나 잠정치를 확정치로 수정함	10분	SMiGM12

특 성 · 기 술 표

업무명　　년　월　일　작성자

I / O 구 분	RECORD FORM FB.VB	RECORD LENGTH	BLOCK SIZE	RECORD 수	DATA TA.DI.DK	DATA 보존기간	비고
I	F	80	80	41,000	TAPE	1 개월	
O	VSAM	50		〃	Disk	〃	
I	VSAM	50		〃	〃	〃	
I	VSAM	53		〃	〃	〃	
O	FB	10	3,000	1,000	〃	〃	
I	FB	10	3,000	1,000	〃	〃	
O	VSAM	50		41,000	〃	〃	
I	F	80	80		TAPE	〃	
O	VSAM	50		41,000	Disk	〃	
I	〃	〃		〃	〃	〃	
I	〃	〃		〃	〃	〃	
I / O	〃	〃		〃	〃	〃	
I / O	〃	〃		〃	〃	〃	
I	F	80	80		TAPE	〃	

RUN NAME	STEP 명	언 어	작성자	처 리 개 요	추정처리 시간·분	FILE 명
CV12	전월자료 CONVERT	PL/1	방윤화	전원자료의 KEY를 품목번호로 변경함	10분	SMiJN12 SMiCV12
JS12	승수산업삽입	〃	〃	금월자료에 승수단 산업은 삽입함	10분	SMiSA12 SMiGM12
MV12	FILE MOVE	〃	〃	SUM FILE을 한달씩 앞으로 MOVE시킴	10분	SMiSM12
SM12	물량집계	〃	〃	카드별로 물량집계 사업체별 LIST SUM FILE up date함	30분	SMiJN12 SMiGM12 SMiSM12
IX12	지수산출	〃	〃	제품의 지수산출	10분	SMiSM12
BK12	BACK UP	〃	〃	조사표내용 BACK UP	5분	SMiGM12
GY35	고용급여TABLE	〃	고성순	지수작성 및 TABLE	10분	SMiSM12
CP34	생산능력TABLE	〃	조성철	지수작성 및 TABLE	10분	〃
MC12	전월CREATION	〃	방윤화	금월을 전월로 바꿈	10분	SMiGM12 SMiJN12
FP12	FILE MERGE	〃	〃	금월 2개 FILE을 1개로 붙임	30분	SMiGM12 SMiG 212 SMiGM12
MT34	MASTER수정	〃	조성철	유고사업체를 MASTER에 수정	10분	

I / O 구 분	RECORD FORM FB.VB	RECORD LENGTH	BLOCK SIZE	RECORD 수	DATA TA.DI.DK	DATA 보존기간	비고
I	VSAM	50		41,000	Disk	1 개월	
O	〃	53		〃	〃	〃	
I	〃	80		4,500	〃	〃	
I / O	〃	50		41,000	〃	〃	
I / O	〃	320		5,850	〃	〃	
I	〃	50		41,000	〃	〃	
I	〃	50		41,000	〃	〃	
O	〃	320		5,850	〃	〃	
I / O	〃	320		〃	〃	〃	
I	〃	50		41,000	〃	〃	
O	FB	50	3,000	〃	TAPE	5 년	
I	VSAM	50		5,850	Disk	1 개월	
〃	〃	〃		〃	〃	〃	
I	〃	〃		41,000	〃	〃	
O	〃	〃		〃	〃	〃	
I	〃	〃		〃	〃	〃	
I	〃	〃		〃	〃	〃	
O	〃	〃		〃	〃	〃	
I	FB	80	3,200	5,000	TAPE	5 년	

RUN NAME	STEP 명	언 어	작성자	처 리 개 요	추정처리 시간·분	FILE 명
						SMiM34
MS12	MASTER	PL/1	방윤화	MASTER SAM	10분	SMiM34
	CREATION			CREATION		SMiSA 12
EE12	조사표체크	〃	〃	MASTER와 조사표	10분	SMiSA 12
				MATCH		SMiGM 12
T 132	제품TABLE	〃	정길섭	제품TABLE	30분	SMiSM 12
T 232	〃	〃	〃			SMiSM 12
BA12	SUMMARY BACK UP	〃	방윤화	BACK UP	5분	
BB12	MASTER BACK UP	〃	〃	〃	3분	SMiM34

I / O 구 분	RECORD FORM FB.VB	RECORD LENGTH	BLOCK SIZE	RECORD 수	DATA TA.DI.DK	DATA 보존기간	비고
O	FB	80	3,200	5,000	Disk	5 년	
I	〃	80	〃	〃	〃	〃	
O	USAM	80		〃	〃	〃	
I	〃	80		〃	〃	〃	
I	〃	50		41,000	〃	〃	
I	〃	320		5,850	〃	〃	
I	〃	320		5,850	〃	〃	
O	FB	320	〃	〃	TAPE	〃	
I	〃	80	〃	5,000	Disk	1 개월	
O	〃	〃	〃	〃	TAPE	5 년	

Ⅲ. 천공, Master Tape Layout

MULTIPLE DISKETTE LAYOUT FORM

업무명 광공업 동태지수 구분 천공 Design

담당자

경제기획원조사통계국
19 년 월 일

1. RL=80 RECFM = F

사업체번호	카드 1	blank	산업분류	규모	유고사항	blank	조사표번호	월	blank	삭제코드

2.

3.

사업체번호	카드 2	일련번호	품목번호	유고사항	월초재고량	생산량	자가소비량	시판량	수출량	과부족보전량	월말재고량	생산증감요인	출하증감요인	blank	조사표번호	월	blank	삭제코드
					월초재고액	생산액	시판액	수출액	과부족보정액	월말재고액	blank							

4.

5.

사업체번호	카드 3	일련번호	원재료번호	유고사항	가공분	시용량	재고량	blank	조사표번호	월	blank	삭제카드

MULTIPLE DISKETTE LAYOUT FORM

업무명 광공업 동태지수 구분 천공 Design

담당자

경제기획원조사통계국
19 년 월 일

1.

사업체번호	카드 4	blank	생산직종업원수	사무직종업원수	무급가족종사자	총종업원수	입직자수	생산직총급여액	사무직총급여액	총급여액	blank	조사표종류	월	blank	삭제코드

5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100 105 110 115 120 125 130

2.

5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100 105 110 115 120 125 130

3.

사업체번호	카드 5	blank	생산직특별급여액	사무직특별급여액	총특별급여액	이직자수	월중연근무인원수	월중연근무시간수	blank	조사표종류	월	blank	삭제코드

5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100 105 110 115 120 125 130

4.

5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100 105 110 115 120 125 130

5.

사업체번호	카드 6	일련번호	품목번호	유고사항	생산능력	생산량	blank	조사표종류	월	blank	삭제코드

5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100 105 110 115 120 125 130

MULTIPLE DISKETTE LAYOUT FORM

업무명 광공업 동태지수　　구분 조사표 File　　담당자 _____　　경제기획원조사통계국 19 년 월 일

1. RL=50

사업체번호	카드 1	φφ	산업분류	규모	유형	조사표	사종	월	승수	blank

2.

3.

사업체번호	카드 2	일련번호	품목번호	유고사항	월초재고량 월초재고액	생산량 생산액	자가소비량 시판액	시판량 수출액	수출량 과부족보정액	과부족보전량 월말재고액	월말재고량 blank	생산증감요인	출하증감요인	blank

4.

5.

사업체번호	카드 3	일련번호	원재료번호	유고사항	구분	사용량	재고량	blank

MULTIPLE DISKETTE LAYOUT FORM

업무명 광공업 동태지수　　구분 조사표 File　　담당자

경제기획원조사통계국
19 년 월 일

1.

사업체번호	카드 4	ϕϕ	생산직종업원수	ϕ	사무직종업원수	무급가족종사자	총종업원수	입직자수	생산직총급여액	ϕ	사무직총급여액	총급여액	blank

2.

3.

사업체번호	카드 5	ϕϕ	생산직특별급여액	ϕ	사무직특별급여액	총특별급여액	이직자수	월중연근무인원수	월중연근무시간수	blank

4.

5.

사업체번호	카드 6	일련번호	품목번호	유고사항	생산능력	생산량	blank

MULTI PLE TAPE LAYOUT FORM

19 年 月 日

經濟企劃院調査統計局

業 務 名 광공업 동태지수 符號 區分 SUM VOL LBL FILE LBL RL 320 BF BS 3200 MFS 擔當者

1. 001~100

카드분	구분	산업	품번 구분	품번 목호	특수분류	년도	월	가중치	기준물량	DATA-1 물량	DATA-1 지수

5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100

2. 101~200

5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100

3. 201~300

5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100

4. 301~400

DATA-37 물량	DATA-37 지수	blank

〈제품의 구분〉

카드 '2' → 제 품
카드 '4' → 고용급여
카드 '6' → 생산능력

'A' → 생산지수
'B' → 〃 (생산액 가중치)
'C' → 대기업 생산지수
'D' → 중소기업 생산지수

'E' → 출하지수
'F' → 내수지수
'G' → 수출지수
'H' → 재고지수

'I' → 재고율지수

※ J → O : 품목없음

5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100

5. 401~500

'J' → 특수분류 생산지수
'K' → 〃 출하지수
'S' → 〃 내수지수 } 추가
'T' → 〃 수출지수 } 추가
'L' → 〃 재고지수

'M' → 중경공업 생산지수
'N' → 〃 출하지수
'O' → 〃 재고지수

'P' → 철도선박제외 생산지수
'Q' → 〃 출하지수
'R' → 〃 재고지수

〈생산능력의 구분〉
'1' → 생산능력지수
'2' → 생 산 지 수
'3' → 가동율지 수

〈고용급여의 구분
'1' → 종업원수
'2' → 총급여액
'3' → 특별급여제외 급여액

5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100

MULTI PLE TAPE LAYOUT FORM

19 年 月 日

業 務 名 광공업동태사업체명부 符號 區分 MASTER VOL LBL FILE LBL RL 80 BF 40 BS 3200 MFS 擔當者 經濟企劃院調査統計局

1.

※ 각 항목(ITEM)별 수정(update) 가능

항목중 ƀ로 수정을 필요로 할 경우 ϕ로 항목을 채움

001~100 | 5 10 | 15 20 | 25 30 | 35 40 | 45 50 | 55 60 | 65 70 | 75 80 | 85 90 | 95 100

2.

83 년 신 번 호	82 년 구 번 호	이 전 번 호	산 업 분 류	'82 사업체 시 도	'82 사업체 일 련 번 호	승 수	능 력	중 복	종 업 원 규 모	blank	유 고 년 월	구 분	사업체명 (한글)	'80 사업체 시 도	'80 사업체 일 련 번 호	blank

101~200 | 5 10 | 15 20 | 25 30 | 35 40 | 45 50 | 55 60 | 65 70 | 75 80 | 85 90 | 95 100

3.

'ƀ' : 동태단독
'1' : 동태+능력
'2' : 능력단독

'1' → 대 기 업
'2' → 중소기업

'1' : 조사구 변경
'2' : 표본대비
'3' : 대상외
'4' : 폐 업
'5' : 신규 ----→ ϕ로 하면 안됨
'6' : 이전
'8' : 합산
'9' : 휴업

201~300 | 5 10 | 15 20 | 25 30 | 35 40 | 45 50 | 55 60 | 65 70 | 75 80 | 85 90 | 95 100

4.

301~400 | 5 10 | 15 20 | 25 30 | 35 40 | 45 50 | 55 60 | 65 70 | 75 80 | 85 90 | 95 100

5.

401~500 | 5 10 | 15 20 | 25 30 | 35 40 | 45 50 | 55 60 | 65 70 | 75 80 | 85 90 | 95 100

Ⅳ. 사전준비작업 Flowchart

사 전 준 비 작 업

INPUT	작 업 내 용	OUTPUT	비 고
master back-up CRT수정	MASTER UPOATE (SMIMWT62)	SMIM 34	* 수정사항이 많을 경우 D.T.T 처리함. (Master design 참조) * Console Key in 후 Acending sort하시오(SORT* A15)
SMIM 34	MASTER CREAT-ION (SMIWMS 62)	SMISA12	
SMISA 12	MASTER COUNT (SMIWTY 66)	도별 List	

확 정 치 (전 월) 작 업

INPUT	작 업 내 용	OUTPUT	비 고
확정 TAPE	OK UPDATE (SMIWOU 62)	JN12 List	* LIST는 심사계에서 수정 OK 될때까지 본 STEP 반복 DSN =전월→SMIJN 12 금월→SMIGM 12 ** SMIDT **R62** JN 년 φφ 월 * LIST는 심사계로 월(MONTH) CHANGE
JN 12 GM12	접 수 부 (SMIWTR 62)	LIST	
GM 12	FILE CHANGE (SMIWMC 62)	JN 12	
SM 12	SUMFILE MOVE (SMIWMV 62)		
JN 12	CONVERT (SMIWCV 62)	CV 12	

잠 정 치(금 월) 작 업			
INPUT	작 업 내 용	OUTPUT	비 고
조사표 DTT	1) CREATION (SMIWCR 62)	GM 12 (SMIG212)	1) ~4) STEP 작업시 DSN (1차 : SMIGM 12, 2차 : SMIG212)로 한다.
PM12 CV12 GM12	2) EDITING (SMIWED 62)	OK 12 LIST	* LIST는 심사계에서 조사표와 대조 수정함. 4) →2) →3) 작업은 조사표 접수 완료될 때까지 1,2차로 구분 반복 작업함.
OK 12	3) OK SELECT (SMIWOS 62)	GM12	DSN(1차 : SMIGM 12, 2차 : SMIG212
수정분 DTT	4) UPDATE (SMIWUP62)	GM 12	
CRT	OK ERROR 구분 (SMIWOE 62)	GM 12 (G 212)	
GM12 G212	FILE MERGE (SMIWFP 62)	GM 12 LIST	* 본 STEP은 조사표 완료된 이후에만 가능함.
PM12 CV12 GM12	EDITING (SMIWED 62)	OK12 LIST	* LIST는 심사계에서 조사표와 대조 수정함.
수정분 DTT	UPDATE (SMIWUP 62)	GM 12	* DSN = SMIGM12
CV 12 GM 12	CARD CHECK (SMIWCH62)	LIST	* LIST는 심사계로 이송

감 정 치(금 월) 작 업			
INPUT	작 업 내 용	OUTPUT	비 고
GM 12 SA12	MASTER CHECK (SMIWEE 62)	LIST	* LIST는 심사계로 이송
GM12	승수산업삽입 (SMIWJS 62)		* DSN (JN 12 / GM 12)로 나누어 두번 작업함.

SUMMARY (물량)작업			
INPUT	작 업 내 용	OUTPUT	비 고
GM12 JN12	SUMMARY (SMIWSM62)	SM12 LIST	* 21 : 사업체별 생산량 23 : 생산 TOTAL 41 : 사업체별 고용 43 : 고용 TOTAL 61 : 사업체별 능력 63 : 능력 TOTAL 22 : 총괄로 바꿔서 작업 * 지수 천공후 작업 ** ** *** *** *** *** 월 CARD 전월지수 금월지수 지수산출은 큰 PARTITION * 지수작업시 유의사항 ∘ WT 162 → '36**' 4회 Key in 확정월(MONTH ∘ WT 262 → '37**' 4회 Key in 잠정월 * WT 162와 WT262는 같은 PROGRAM이다. ∘ WGY 62 → 소매물가 Key in 월(MONTH) CHANGE **** **** 전월지수금월지수 '84**' 'φφ' Key in ∘ WCP 62 → '84**' '84**'
SM 12	지 수 산 출 (SMIWIX 62)		
SM12	TABLE (지수) 1. SMIWT162 2. SMIWT262 3. SMIWGY62 4. SMIWCP 62	LIST } 지수산출과 관계없음	

계 절 조 정 지 수 작 업			
INPUT	작 업 내 용	OUTPUT	비 고
DB	계절조정지수 (SMIW 2362)	DB LIST	* 월 바뀜에 유의한다. * 다른 DB 작업과 동시 수행 불가

분기 작업 (각 지수별)			
SM 12	확정분기 (동태) (SMIWT 362)	LIST	* 동태감정과 확정분기 TABLE은 같은 PROGRAM '36** **' KEY IN 월 분기 {14, 24, 34, 44} '37** **' KEY IN
SM 12	잠정분기 (동태) (SMIWT 362)	LIST	
SM 12	고 용 분 기 (SMIWB 362)	LIST	
SM 12	능 력 분 기 (SMIWC 362)	LIST	

BACK-UP			
GM 12	금월 BACK-UP (SMIWBK 62)		84 SMI BK ϕ 1R 62** GM 월 DSN = GM 12
JN 12	전월 BACK-UP (SMIWBK 62)		84 SMI BK ϕ 1R 62** JN 월 DSN = JN 12
SM12	SUM BACK-UP (SMIWBK 62)		84 SMI BK ϕ 1R 62** SM 월 DSN = SM12
SA12	MASTER BACK-UP (SMIWBK 62)		84 SMI BK ϕ 1R 62** MA 월 DSN = SA 12

광공업 동태지수(보정) SYSTEM FLOW CHART.

INPUT	작업내용	OUTPUT	비 고
BOJUNG62	SUM FILE NOVE (SMIWMV62)	BOJUNG62	1. 매월 ROUTINE 작업이 끝나면 작업시작 2. 본 PROGRAM CATAL이 되어 있으며 CATAL NAME은 SMIMV12이다.
GM12 JN12	SUMMARY (SMIWSM62)	BOJUNG62	1. CATAL NAME = SMIWN66 2. 매월 작업시 월과 CARD 확인 3. 지수 천공후 작업 4. PROGRAM LIST를 참조
BOJUNG62	지수산출 (SMIWIX62)		1. CATAL NAME = SMIBJ62 2. 37개월시산 3. 큰 PARTITION에서 작업요
BOJUNG62	Back-up (SMIWBK62)		1. 85년 1월까지는 금월 Back-up후 전월 Back-up Tape 파기요 2. TAPE의 표찰 ××SMIBKØR×× ×× BJ 년 개인번호 월

```
//SMIWSM62 JOB CLASS=3                                          SMI00010
//JOBCAT DD DSN=SMICT12,DISP=SHR                                SMI00020
//   EXEC PGM=SMISM62                                           SMI00030
//STEPLIB DD DSN=USER.LOADLIB,DISP=SHR                          SMI00040
//SYSOUT DD SYSOUT=A                                            SMI00050
//SYSPRINT DD SYSOUT=A                                          SMI00060
//SORTWK01 DD UNIT=3350,SPACE=(CYL,10)                          SMI00070
//SUMF  DD DSN=BOJUNG62,DISP=SHR                                SMI00080
//EDIT2 DD DSN=SMIJN12,DISP=SHR                                 SMI00090
//EDIT3 DD DSN=SMIGM12,DISP=SHR                                 SMI00100
//SYSIN DD *                                                    SMI00110
0823 1078 1377 1161 0967 1148 1078 1377 1161 0967 1144 0031 0031 SMI00120
/*                                                              SMI00130
//                                                              SMI00140
```

(1) (2) (3) (4) (5) (6) (7) (8) (9) (10) (11) (12)

전월지수

금월지수

"55080" 특수운동화 전월지수

"55080" 특수운동화 금월지수

월 (MONTH)

CARD

※ 21 : 사업체별 생산량
23 : 생산 TOTAL
41 : 사업체별 고용
43 : 고용 TOTAL
61 : 사업체별 능력
63 : 능력 TOTAL
22 : 총 괄

ROUTINE 업무 SMIWSM66과 DATA가 같아야 한다.

전월 금월 지수에서 (1), (2), (6), (7)은 "22010" (의약품)이며

(3), (8)은 "52060"

(4), (9)은 "82440"

(5), (10)은 "85120"

(11), (12)은 "55082"(특수운동화, 전월, 금월지수)이다.

※ 특수운동화 (55080)에 승수산출방법

55080 < 55081 (일반운동화) / 55082 (특수운동화)

"55082"의 물량에 승수를 곱해서

"55081"의 물량을 더하여 산출

V. 품 목 분 류

1980년기준 광공업동태조사지정품목분류표

품목번호	품 목 명	단위	품 질 및 규 격	비 고
	2 광 업			
	21 석 탄 광 업			
	210 석 탄 광 업			
01010	무 연 탄	M/T		
	23 금 속 광 업			
	230 금 속 광 업			
03010	철 광 석	M/T		
03020	중 석 광 석			
03030	은 광 석	kg	99.9 %은광석으로 환산	
03040	연 광 석	M/T		
03050	아 연 광 석	〃		
	29 기 타 광 업			
	290 기 타 광 업			
09010	화 강 암	M/T		
09020	석 회 석	〃		
09030	쇄 석	〃		
09040	모 래	m^3		
09050	자 갈	〃		
09060	고 령 토	M/T		

품목번호	품 목 명	단위	품 질 및 규 격	비 고
	3 제 조 업			
	31 음식료품 및 담배제조업			
	311~312 식료품 제조업			
11010	소 시 지	M/T		
11020	볼 유	〃		
11030	아 이 스 크 림	kℓ		
11040	빙 과	M/T		
11050	처 리 우 유	〃		
11060	유 산 균 발 효 유	〃		
11070	농 산 물 통 조 림	〃		
11080	수 산 물 통 조 림	〃		
11090	냉 동 해 산 물	〃		
11100	대 두 유 (콩기름)	kℓ		
11110	쇼 팅 (쇼트닝)	M/T		
11120	밀 가 루	〃		
11130	식 빵	〃		
11140	긴 과 자	〃		
11150	스 넥 과 자	〃		단위변경 kg → M/T
11160	라 면	〃		
11170	인 스 턴 트 면 류	〃		
11180	정 당	〃		
11190	설 탕 과 자	〃		

품목번호	품 목 명	단위	품 질 및 규 격	비 고
11200	껌	M/T		
11210	인 조 얼 음	〃		
11220	간 장	kℓ		
11230	글 루 타 민 산 소 다	M/T		
11240	두 부	〃		
11250	녹 말 가 루 (전분)	〃		
11260	식 용 포 도 당	〃		
11270	물 엿	〃		
11280	커 피	kg		
11290	분 말 인 삼	〃	액상은 제외	
11300	배 합 사 료	M/T		
11310	보 조 사 료	〃		
	313 음료품제조업			
13010	주 정 (메틸알콜)	kℓ		
13020	소 주	〃		
13030	위 스 키	〃	국산양주로 브랜디 꼬냑도 조사되었으나 위스키만 조사	
13040	럼 주			
13050	탁 주	kℓ		
13060	청 주	〃		
13070	맥 주	〃		
13030	사 이 다	〃		

품목번호	품 목 명	단위	품 질 및 규 격	비 고
13090	콜 라	kℓ		
13100	쥬 스	〃		
	314 담배제조업			
14010	필 터 달 린 담 배	백만본	1,000,000	
14020	필 터 없 는 담 배	〃		
	32 섬유의복 및 가죽산업			
	321 섬유제조업			
21010	생 사	M/T	옥사포함	
21020	면 사	〃		
21021	순 면 사	〃		세분조사
21022	면 혼 방 사	〃		세분조사
21030	방 모 사	〃	혼방방모사 포함	
21040	순 소 모 사	kg		
21050	혼 방 소 모 사	〃		
21060	합 성 섬 유 방 적 사	M/T		
21070	재 생 섬 유 방 적 사	〃		
21080	혼방합성섬유방적사	〃		
21090	연 사	〃	면연사, 견연사, 모연사, 인조섬유연사	
21100	담 요	㎡		
21110	천 막	매	EACH	
21120	타 월	kg		
21130	스 타 킹	천켤레	1,000	

품목번호	품 목 명	단위	품 질 및 규 격	비 고
21140	면 양 말	천켤레		
21150	인 조 섬 유 양 말	〃		
21160	메 리 야 스 내 의	천매	1,000	
21170	메 리 야 스 외 의	〃		
21180	로 프	M/T		
21190	어 망 사 (연승)	kg		
21200	어 망	M/T		
21210	면 직 물	천㎡	1,000㎡	
21220	순 소 모 직 물	〃		
21230	방 모 직 물	㎡		
21240	모 혼 방 직 물	천㎡	혼방소모직물	
21250	순 본 견 직 물	〃		
21260	순 합 성 섬 유 직 물	〃		
21270	혼 방 합 성 섬 유 직 물	〃		
21280	재 생 섬 유 직 물	〃		
21290	타 이 어 코 드 사	M/T		
21300	필 터	〃		
	322 의복제조업			
22010	의 복 류	백만원	1,000,000	
22011	남 자 용 양 복	〃		
22012	남자용셔츠, 작업복	〃		
22013	여 자 용 외 의	〃		

품목번호	품 목 명	단위	품 질 및 규 격	비 고
22014	소 아 용 외 의	백만원		
22015	여자용 및 소아용내의	〃		
22016	가 죽 및 모 피 의 복	〃		
22017	모 자	〃		
22018	장 갑	〃		
22019	달리분류되지않는의복	〃		
	323 가죽대용가죽및 모피제품제조업			
23010	소 가 죽	천㎡		
23020	재생가죽 및 인조가죽	㎡		
23030	비닐제트렁크(가죽포함)	개		
23040	핸 드 백	천개		
	324 신발제조업			
24010	남 자 용 가 죽 신	켤레	PAIR	
24020	여 자 용 가 죽 신	〃		
24030	군화 및 가죽장화	〃		
24040	등 산 화	〃		
24050	실 내 화	천켤레		
	33 나무및나무제품제조업(가구포함)			
	331 나무및나무와 콜크제품제조업(가구제외)			
31010	소 할 재	㎥	육송미송 기타 침엽수나왕 기타활엽수목재두께 2치(8㎝)이하	지침서수정
31020	각 재	〃		
31030	판 재	〃		

품목번호	품 목 명	단위	품 질 및 규 격	비 고
31040	침 목	본	EACN	
31050	장 지 문	개		
31060	합 판	㎥		
31070	재 생 목 재	〃		
31080	포 장 상 자	〃		
	332 가구및장지문 제 조 업 (금속가구제외)			
32010	의 자	개		
32020	장 농	〃		
32030	식 탁	〃		
32040	화 장 대	〃		
32050	장 식 장	〃		
32060	책 상	〃		
	34 종이및종이제품 인쇄및출판업			
	341 종이 및 종이 제품제조업			
41010	쇄 목 펄 프	M/T		
41020	화 학 펄 프	〃		
41030	신 문 용 지	〃		
41040	백 상 지 (모조지)	〃		
41050	중 질 지	〃		
41060	아 트 지	〃		
41070	크 라 프 트 지	〃		
41080	갈 포 벽 지	천㎥		

품목번호	품 목 명	단위	품 질 및 규 격	비 고
41090	위 생 용 지	M/T	화장지 나프킨지 및 생리대포함.	
41100	박 엽 지	〃		
41110	판 지	〃		
41120	골 판 지	〃		
41130	골 판 지 상 자	천㎡		
41140	크 라 프 트 지 대	천매		
41150	판 지 상 자	천㎡		
41160	벽 지	〃		
41170	목 합 지	M/T		
41180	금 속 박 지	〃		
	342 인쇄 및 출판 관련산업			
42010	일 간 신 문	천부		
42020	정 기 간 행 물	천권		
42030	서 적	〃		
42031	일 반 서 적	〃		
42032	문 고 판 서 적	〃		
42040	일 반 인 쇄 물	연		
42050	노 트	천권		
42060	앨 범	〃		
	35 화학, 석유, 석탄, 고무 및 플라스틱제조업			
	351 산업용화학물 제 조 업			
51010	에 틸 렌	M/T		

품목번호	품 목 명	단위	품 질 및 규 격	비 고
51020	프 로 필 렌	M/T		
51030	부 타 디 엔	〃		
51040	싸 이 클 로 핵 산	〃		
51050	벤 젠	〃		
51060	톨 루 엔	〃		
51070	키 실 렌	〃		
51080	알 킬 벤 젠	〃		
51090	V. C. M	〃	V.C.M의 단일체 (單量體)	
51100	스 티 렌 모 너 머	〃		
51110	메 탄 올	〃		
51120	호 루 마 린	〃		
51130	에 틸 렌 글 리 골	〃		
51140	스 테 아 린 산	〃		
51150	무 수 마 레 인 산	〃		
51160	무 스 프 탈 산	〃		단위변경 kg→M/T
51170	테 레 프 탈 산	〃		
51180	아 크 릴 로 니 트 릴	〃		
51190	카 프 로 락 담	〃		
51200	황 산	〃		
51210	가 성 소 다	〃		
51220	가 소 제	〃		
61230	소 다 회	〃		

품목번호	품 목 명	단위	품 질 및 규 격	비 고
51240	염 소	M/T		
51250	암 모 니 아	〃		
51260	카 바 이 트	〃	탄화칼슘	
51270	압 축 산 소	千㎥		
51280	아 세 틸 렌 개 스	M/T		
51290	염 료	〃		
51300	유 안 비 료	〃		
51310	요 소 비 료	〃		
51320	용 성 인 비	〃		
51330	용 과 린	〃		
51340	복 합 비 료	〃		
51350	살 충 제	〃		
51360	살 균 제	〃		
51370	제 초 제	〃		
51380	S . B . R	〃		
51390	요 소 수 지	〃		
51400	폴 리 우 레 탄	〃		
51410	폴 리 에 틸 렌	〃		
51411	저 밀 도 에 틸 렌	〃		저밀도 고밀도세분
51412	고 밀 도 에 틸 렌	〃		
51420	폴 리 프 로 필 렌	〃		
51430	폴 리 스 칠 렌	〃		

품목번호	품 목 명	단위	품 질 및 규 격	비 고
51440	P. V. C	M/T		
51450	아세테이트섬유	〃		
51460	폴리에스터섬유	〃		
51470	아크릴릭섬유	〃		
51480	셀로판지	〃		
	352 기타화학제품 제조업			
52010	바니쉬	ℓ		
52020	유성페인트	kℓ		
52030	락카	ℓ		
52040	에나멜	kℓ		
52050	수성페인트	〃		
52060	의약품	백만원		
52061	신경계용약	〃		
52062	해열진통제	〃		
52063	소화기계용약	〃		주방용제외
52064	비타민제	〃		
52065	자양강장제	〃		
52066	혈액및체액용약	〃		
52067	항생물질제	〃		
52068	기타의약품	〃	밀크로션 포함	병행예외 (8)
52070	세탁비누	〃		
52080	화장비누	〃		

품목번호	품 목 명	단위	품 질 및 규 격	비 고
52090	세탁용합성세제	백만원		주방용제외
52100	치 약	〃		
52110	샴 푸	kg		
52120	화 장 크 림	〃		
52130	화 장 백 분	〃		
52140	화 장 수	〃	밀크로션포함	
52150	글 리 세 린	M/T		
52160	다 이 나 마 이 트	〃		
52170	카 본 블 랙	〃		
52180	인 쇄 잉 크	kg		
52190	향 료	〃		
52200	계 면 활 성 제	M/T		
	353 석유정제업			
53010	젯 트 유	kℓ		
53020	휘 발 유	〃		
53030	나 프 타	〃		
53040	등 유	〃		
53050	솔 벤 트	〃		
53060	경 유	〃		
53070	중 유	〃		
53080	방 카 C 유	〃		
53090	윤 활 유	〃		

품목번호	품 목 명	단위	품 질 및 규 격	비 고
53100	액 화 석 유 개 스	kℓ		
	354 기타석유및석탄제품제조업			
54010	연 탄	M/T		
54020	코 크 스	〃		
54030	아 스 팔 트	kℓ		
	355 고무제품제조업			
55010	자 동 차 용 타 이 어	천본		
55020	자 전 거 용 타 이 어	〃		
55030	자 동 차 용 튜 브	〃		
55040	자 전 거 용 튜 브	〃		
55050	재 생 타 이 어	본		
55060	고 무 신	천켤레		
55070	고무장화및우화	〃		
55080	운 동 화	〃	케미화 포함	
55090	고 무 호 스	m		
55100	각 종 고 무 벨 트	천p/y		
55110	스 폰 지	m³		
	356 플라스틱제품제조업			
56010	플 라 스 틱 제 품	M/T		
56011	플라스틱제식기및용기	〃		세분조사
56012	플 라 스 틱 제 신 발	〃	케미화 제외	
56013	플라스틱제파이프및관	〃		

품목번호	품 목 명	단위	품 질 및 규 격	비 고
56014	플라스틱제필름	M/T		
56015	플라스틱제레저	〃		
56016	기타플라스틱제품	〃		
	36 비금속광물제품 제조업			
	361 도기자기 및 토기제조업			
61010	도자식기	천개	가정용 도자식기	토기제품제외
61020	위생도기	개	세면기 대변기 소변기 욕조등	
61109	고압애자	〃	3,300V이상	
	362 유리 및 유리 제품제조업			
62010	유리관	M/T		
62020	판유리	상자	BOX	
62030	안전유리	〃		
62040	브라운관용유리	천개		
62041	흑백브라운관용유리	〃		신규세분
62042	칼라브라운관용유리	〃		
62050	보온병	kℓ	속에 유리가 있는 제품	
62060	유리병	천개	술병+식료품병 음료수병 포함	(특히유의) 크리스탈병제외
62068	유리병	M/T		〃
62070	유리약병	〃		
62080	유리식기	천개		
62088	유리식기	M/T		병행단위
62090	유리섬유	〃		

품명번호	품 목 명	단위	품 질 및 규 격	비 고
	369 기타비금속광물 제품제조업			
69010	벽 돌 (조 성)	천매		
69020	타 일	천㎡		
69030	내 화 벽 돌	M/T		
69040	포오틀란드시멘트	천M/T		
69050	시멘트크링커	〃		
69060	소 석 회	M/T		
69070	콘크리트블록	천매		
69080	〃 벽 돌	〃		
69090	흄 관	본		
69100	콘크리트전주	〃		
69110	〃 파 일	〃		
69120	레 미 콘	천m		
69130	석 면 스 레 트	〃		
69140	활 석 분	M/T		
	37 제1차금속산업			
	371 제1차철강산업			
71010	선 철	M/T		
71020	합 금 철	〃		
71030	강 괴	〃		
71040	주 강	〃		

품명번호	품 목 명	단위	품 질 및 규 격	비 고
71050	강 반 성 품	M/T	대강 제외	
71060	봉 강	〃		
71070	스텐레스스틸판	〃		
71080	대 강	〃	열연대강, 냉연대강 포함	
71090	철 근	〃		
71100	형 강	〃		
71110	중 후 판	〃		
71120	열 연 박 판	〃		
71130	냉 연 박 판	〃		
71140	전 기 강 판	〃		
71150	선 재	〃		
71160	강 관	〃		
71170	주 철 관	〃		
71180	회 주 물	〃		구명 : 기계용 주물
71190	가 단 주 물	〃		
71200	합 금 주 물	〃		
71210	단 강 품	〃		
71220	석 도 철 판	〃		
71230	아 연 도 철 판	〃		
71240	와 이 어 로 프	〃		
71250	쇠 못	〃		

품목번호	품 목 명	단위	품 질 및 규 격	비 고
71260	흑 철 선	M/T	PC 강선제외	
71270	아 연 도 철 선	〃		
71280	경 강 선	〃		
	372 제 1 차비철 금 속 산 업			
72010	전 기 동	M/T		
72020	알 루 미 늄 괴	〃		
72030	알 루 미 늄 합 금	〃		
72040	연 괴	〃		
72050	아 연 괴	〃		
72060	은	kg		
72070	금	〃		
72080	동 봉 및 형 강	M/T		
72090	나 동 선	〃		
72100	동 판 및 띠	〃		
72110	동 관 및 중 공 봉	〃		
72120	알 루 미 늄 판	〃		
72130	알 루 미 늄 샷 시 바	〃	알미늄괴를 압축하여 만든 알미늄샷시용바	(조립품제외) (건축용샷시)
	38 조립금속제품기계 및 장비제조업			
	381 조립금속제품 제 조 업			
81010	금 속 제 식 탁 용 품	천개		

품목번호	품 목 명	단위	품 질 및 규 격	비 고
81020	공 장 용 수 공 구	천개		
81030	농 업 용 〃	〃		
81040	철 제 캐 비 넷	개		
81050	철 제 의 자	〃		
81060	철 제 책 상	〃		
81070	건 물 용 문 및 수 문	〃		
81080	건 축 용 샷 시	M/T	철제 및 알루미늄샷시	
81090	보 일 러	T/H		
81100	열 교 환 기	대		
81110	금 속 구 조 물	M/T		
81120	금 속 탱 크 및 용 기	kℓ	금속개스탱크 및 용기를 제외한 대형금속탱크	
81130	통 조 림 용 관	천개		
81140	드 럼 관	개		
81150	알루미늄프레스가공 식 기 및 용 기	M/T		
81160	주 전 자	개		
81170	스텐레스스틸제식기	M/T		
81180	커 피 포 트	개		
81190	가 스 레 인 지	〃		
81200	싱 크	〃	스텐레스제	목재싱크대는 제외
81210	방 열 기	대		
81220	자 동 차 용 방 열 기	개		

품목번호	품 목 명	단위	품 질 및 규 격	비 고
81230	석 유 스 토 브	개		
81240	볼 트 및 낫 트	M/T		
81250	금 속 관 이 음 쇠	〃		
81260	철 사 망	〃		
81270	자동차용중판스프링	천매		
81280	체 인	kg		
81290	병 마 개	천개		
81300	용 접 봉	M/T		
81310	금속가스탱크및용기	kg	병모양을 갖춘 가스용기	라이타개수주입용용기등 Can 으로 된것은 제외
	382 기계제조업			
82010	내 연 기 관	HP		
82020	동 력 탈 곡 기	대		
82030	경 운 기	HP		
82040	동 력 분 무 기	대	동력살분무기 포함	인력분무기 제 외
82050	농 업 용 트 랙 터	HP		
82060	선 반	대		
82070	밀 링 기	〃		
82080	프 레 스 기	M/T		
82090	압 출 기	대		
82100	압 연 기	〃		

품목번호	품 목 명	단위	품 질 및 규 격	비 고
82110	금 형	대		
82118	금 형	천원		
82120	로 더	대		
82130	굴 착 기	〃		
82140	불 도 저	〃	25ton 기준으로 환산(신설)	
82150	연 사 기	〃		
82160	직 기	〃	면직기, 견직기, 기타직기(통합)	
82170	편 직 기	〃	횡편직기, 환편직기, 기타현직기 통합조사	
82180	샷 틀	〃		
82190	염 색 기	〃		
82200	포 장 기	〃	식품포장기	신 규
82210	전 자 계 산 기	〃		
82211	탁상용전자계산기	〃		
82212	휴대용 〃	〃		
82220	금 전 등 록 기	〃		
82230	타 자 기	〃		
82240	동 력 펌 프	〃		
82250	공 기 압 축 기	〃		
82260	송 배 풍 기	〃		
82270	호 이 스 트	〃		
82280	크 레 인	M/T		

품목번호	품 목 명	단위	품 질 및 규 격	비 고
82290	엘 레 베 이 터	대		
82300	콘 베 이 어	〃		
82310	지 게 차	〃		
82320	가 정 용 재 봉 틀	〃		
82330	공 업 용 〃	〃		
82340	자 동 판 매 기	〃	커피, 담배 등 자동판매기	
82350	감 속 기	〃		
82360	기 어	개		자동차제동장치의 기어제외
82370	베 어 링	kg		
82371	볼 베 어 링	〃		
82372	로 울 러 베 어 링	〃		
82373	메 탈 〃	〃		
82374	기 타 〃	〃		
82380	윈도우형에어컨디셔너	대		
82390	페케이지형에어컨디셔너	R/T		
82400	상 업 용 냉 장 고	〃		
82410	자 동 차 용 피 스 톤	개		
82420	피 스 톤 링	〃		
82430	밸 브	〃	각종밸브통합조사	
	383 전기기계, 기구 제 조 업			
83010	직 류 전 동 기	HP		

품목번호	품 목 명	단위	품 질 및 규 격	비 고
83020	교 류 전 동 기	HP		
83030	변 압 기	KVA		
83040	배 전 반	대		
83050	전 력 회 로 차 단 기	천대		
83060	흑 백 TV 수 상 기	대		
83070	칼 라 TV 〃	〃		
83080	컴 비 네 이 션 TV	〃	TV와 라디오 및 기타 TV복합체	
83090	라 디 오 수 신 기	〃		
83091	일반라디오수신기	〃		
83092	시계라디오수신기	〃		
83093	카 라 디 오 수 신 기	〃		
83100	전 축	〃		
83110	녹 음 기	〃	녹음재생기포함(기존제외)	
83120	T V 튜 너	천개		
83130	확 성 기	〃		
83140	엠 프	대	튜너형, 리시버형 포함	
83150	전 화 기	〃		
83160	자동식전화교환기	회선		
83170	브 라 운 관	개		
83171	흑 백 브 라 운 관	〃		
83172	칼 라 〃	〃		

품목번호	품 목 명	단위	품 질 및 규 격	비 고
83180	다 이 오 드	천개	발광다이오드 포함	
83181	겔마니움다이오드	〃		
83182	실 리 콘 〃	〃		
83183	발 광 〃	〃		추 가
83190	집 적 회 로	〃		
83200	트 랜 지 스 트	〃		
83201	겔마니움트랜지스터	〃		
83202	실리콘트랜지스터	〃		
83210	가 변 저 항 기	〃		
83220	고 정 저 항 기	〃		
83230	가 변 콘 덴 서	〃		
83240	고 정 〃	〃		
83250	전자기기용유도자	〃		
83260	녹 음 테 이 프	km		천개 → km
83261	카 세 트 테 이 프	〃		〃
83262	릴 〃	〃		〃
83270	전 기 냉 장 고	대		
83280	선 풍 기	〃		
83290	전 기 세 탁 기	〃		
83300	전 기 밥 솥	개		
83310	전 기 보 온 밥 통	〃		

품목번호	품 목 명	단위	품 질 및 규 격	비 고
83320	통신선및통신용케이블선	M/T		
83330	전력선및전력용케이블선	〃		
83340	형 광 전 구	천개		
83350	장 식 용 전 구	〃		
83360	축 전 지	개		
83370	건 전 지	천개		
83378	〃	KV		병행단위
83380	스 위 치	천개		
83381	커 버 나 이 프 스 위 치	〃		신규세분
83382	기 타 스 위 치	〃		
83469	백 열 전 구	〃		
83509	계 수 표 시 방 전 관	〃		
	384 운수장비제조업			
84010	선 박 용 내 연 기 관	HP		
84020	철 강 유 조 선	G/T		
84030	철 강 화 물 선	〃		
84040	특 수 화 물 선	〃		
84050	철 강 어 선	〃		
84060	디 젤 기 관 차	대		
84070	여 객 차	량		
84080	철 도 차 량 륜	개		

품목번호	품 목 명	단위	품 질 및 규 격	비 고
84090	승 용 차	대		
84100	대 형 버 스	〃		
84110	소 형 버 스	〃		
84120	대 형 트 럭	〃		
84130	중 형 트 럭	〃		
84140	소 형 트 럭	〃		
84150	특 수 차	〃	일반차량을 제외한 구난차, 소방차, 살수차, 청소차 등	
84160	자동차용내연기관	〃		
84170	자 동 차 륜	개		
84180	차 축	〃		
84190	자 전 거 차 체	대		
84200	모 터 싸 이 클	〃		
84210	콘 테 이 너	〃		
	385 전문과학측정 제어장비및사진 광학제품제조업	〃		
85010	적 산 전 력 계	〃		
85020	수 도 미 터	개		
85030	광 학 용 렌 즈	〃		
85040	안 경 테	천개		
85050	사 진 기	대		
85060	사 진 성 광 기	개		
85070	쌍 안 경	〃		

품목번호	품 목 명	단위	품 질 및 규 격	비 고
85080	손 목 시 계	개		
85081	일 반 손 목 시 계	〃		
85082	전 자 〃	〃		
85090	벽 시 계	〃		
85091	일 반 벽 시 계	〃		
85092	전 자 〃	〃		
85100	탁 상 시 계	〃		
85110	손 목 시 계 케 이 스	〃		
	37 기 타 제 조 업			
	390 기 타 제 조 업			
90010	피 아 노	대		
90020	기 타	〃		
90030	전 자 현 악 기	〃		
90040	대 형 볼	개		
90050	야 구 장 갑	〃		
90060	낚 시 대	〃		
90070	샤 프 연 필	〃		구품목명 : 기계식연필
90080	볼 펜	천개		
90090	연 필	천타		
90100	철제우산및양산살대	천개		
90110	가 발	〃		

품목번호	품 목 명	단위	품 질 및 규 격	비 고
90120	라 이 타	개		
90121	일 회 용 라 이 타	〃		
90122	반 영 구 용 라 이 타	〃		
90130	인 형	천개	인형 및 봉제완구 포함	
90140	작 크	km		
	4. 전 기 업			
99990	총 발 전 량 및 판 매 량	백 만 KWH		

(1) 조사지정품목(137개)

품목번호	품 목 명	단위	설비		표준조업시간	표준조업일수
			설 비 명	단위		
11020	분유 및 연유	M/T	① 분무건조기 (분유의 경유)	대	10	30
			② 충 전 기 (연유의 경유)	대	10	25
11050	처 리 우 유	M/T	살 균 기	대	8	28
11910	통 조 림	M/T	권 체 기	대	8	25
11100	대 두 유	Kℓ	탈 산 기	대	24	25
11120	밀 가 루	M/T	Sifter, Roller	대	24	25
11160	라 면	M/T	Roller	대	20	25
11180	정 당	M/T	용당관, 결정관	대	24	25
11230	글루타민산소다	M/T	발효조, 결정관	대	24	30
11300	배 합 사 료	M/T	배합기, 마쇄기	대	10	25

산 정 기 준 표

표 준 생 산 능 력 산 식	개 념 및 포 괄 범 위
C =시간당 건조기 탈수능력 (kg) × 10 30 × 1 / 1,000	분유, 연유 포함 조사
C =시간당 기계능력 (Can) × Can Size (kg / Can) × 10 × 25 × 1 / 1,000	
C =시간당 기계능력 (kg) × 8 × 28	저온살균, 균질화, 멥톤화 또는 젖화한 우유 (시유 (市乳) 를 지칭) (비타민, 미네랄염 등이 첨가된 것 포함)
C =분당 기계능력 (Can) × 60 × Can Size (kg / Can) × 8 × 25 × 1 / 1,000	투입원재료 불문, 동태조사에서 농산물과 해산물로 국한되어 있으나 생산능력조사에서는 축산물 통조림도 포함되고 있음.
C =시간당 탈산능력 (Kℓ) × 24 × 25	
C = 1 일 공칭능력 (M/T) × 25	1 Bbl = 88 kg 관, 민수용 포함 조사
C = 1 일 공칭능력 (M/T) × 25	
C = 1 일 공칭능력 (M/T) × 25	
C = 1 일 공칭능력 (M/T) × 30	
C = 1 일 생산능력 (M/T) × 25	배합사료의 원료 (밀기울, 대두박등) 만은 조사하지 않으며, 반드시 배합된 사료를 조사한다. 화학사료는 제외한다.

품목번호	품 목 명	단위	설 비		표준조업시간	표준조업일수
			설 비 명	단위		
13010	주 정 (에틸알콜)	Kℓ	① 증 자 기	대	24	25
			② 발 효 조	대	24	25
			③ 증 류 기	대	24	25
13020	소 주	Kℓ	사 입 부	대	16	25
13070	맥 주	Kℓ	저장탱크, 당화조	대	24	25
13910	청 량 음 료	Kℓ	주입기, 혼합기	대	10	25
21010	생 사	kg	① 다조기(서)	대	16	25
		kg	② 자동조사기	Set	16	25

표 준 생 산 능 력 산 식	개 념 및 포 괄 범 위
C =총용량 (ℓ) × 사입비율 × 1 일회전수 × 증류비율 × 숙성비율 × 시료도수 × $\frac{100}{95} \times 25 \times \frac{1}{1000}$	
C =위와 같음.	
C = (요탑반경 cm)^2 × 3.14 × 유하계수 × 시류도수 × 증류비율 × $\frac{100}{95} \times 25 \times \frac{1}{1000}$	
C =사입부당 출고량 (ℓ) × 월간회전수 × $\frac{1}{1000}$	
C =저장능력 (Kℓ) × 월간회전수	
C =시간당 기계능력 (Btl) × 10 × 25 × $\frac{\text{Btl 당 } \ell}{1000}$	사이다, 콜라, 쥬스의 기타 청량음료 포함
C = 20 서 × 21 S (중수) × 물레둘레 (m) × 분당회전수 × 60 × 16 × 25 × $\frac{0.05\ (g)}{450\ m} \times \frac{1}{1000}$	① 서 = 물레 ② 옥사 (쌍잠) 포함
C =물레수 × 21 S (중수) × 물레둘레 (m) 분당회전수 × 60 × 16 × 25 × $\frac{0.05\ (g)}{450\ m} \times \frac{1}{1000}$	

품목번호	품 목 명	단위	설 비		표준조업시간	표준조업일수
			설 비 명	단위		
21020	면 방 적	추	면정방기(스프 정방기 포함)	추	24	30
21030	방 모 방 적	추	방모정방기	추	24	30
21910	소 모 방 적	추	소모정방기 (합섬소모정방기 포함)	추	24	30
21170	메리야쓰외의	천매	편 직 기	대	24	30
21210	면 직 물	대	① 역 직 기	대	24	30
			② 수동직기	대	24	30
21920	모 직 물	대	① 역 직 기	대	24	30
			② 수동직기	대		
21250	견 직 물	대	① 역 직 기	대	24	30
			② 수동직기	대		
21930	합성섬유직물	대	① 역 직 기	대	24	30
			② 수동직기	대		

표 준 생 산 능 력 산 식	개 념 및 포 괄 범 위
C＝월간 생산실적÷운전율 운전율＝실운전추수÷운전가능추수 실운전추수＝월간총연 운전추수시간÷(24×30)	순면사, 면혼방사 포함
C＝면정방기의 산식과 동일함.	수모방적사 제외
C＝면정방기의 산식과 동일함.	순소모사, 혼방소모사 포함.
C＝1일 생산능력(매)×30×$\frac{1}{1000}$	쉐타 포함
C＝월간 생산실적÷운전율(%) 운전율＝월간 평균실동대수÷월말운전가능대수 월간 평균실동대수＝월간 총연실동대수÷30	순면직물, 혼방면직물, 기타 면직물 포함
C＝면직물의 산식과 동일함.	순소모직물, 방모직물, 모혼방직물 포함
C＝면직물의 산식과 동일함.	순본견직물, 견혼방직물 포함 (홀치기원단, 시보리원단, 하오리원단, 쏘무기원단, 양단 등 포함)
C＝면직물의 산식과 동일함.	순합성섬유직물, 혼방합성섬유직물 포함(나이론, 다후다, Jersy, 앙고라, 포리텍스, 케미칼, 트리코트라고 일반적으로 호칭됨)

품목번호	품 목 명	단위	설 비		표준 조업 시간	표준 조업 일수
			설 비 명	단위		
23010	소 가 죽	천㎡	Band Knife Drum	대	8	25
31060	합 판	㎥	① 박 취 기 (Rotary Lathe)	대	20	30
			② 건 조 기	대	20	30
			③ Glue Spreader	대	20	30

표 준 생 산 능 력 산 식	개 념 및 포 괄 범 위
C = 시간당 기계능력 (㎡) × 8 × 25	Band Knife : 후도조절기 Drum : 화학반응기 (1 S / F ≒ 0.0929 ㎡)
C = π (3.14) × 직경 × 분당회전수 × 60 (분) × Cu × Cl × 0.55 × 30 × $\frac{1}{3703.7}$	Charging Unit : 원목의 중심점을 을 찾는 기계 ※ 1 ㎥ = 3703.7 S / F
주 : Cu = 20시간 : Charging Unit 가 있는 경우 18/20/24 12시간 : Charging Unit 가 없는 경우 6 / 20 / 24 Cl = 9′ : Cutting length 가 4.2′일때 5′ : Cutting length 가 8.4′일때	
C = 폭 (ft) × $\frac{1}{4.2}$ × 단 (Stage)의수 × 분당속도 (m) × $\frac{4}{8.4}$ × 60 (분) × $\frac{1}{3}$ × 20 × 30	
C = 분당속도 (m) × 60 (분) × Cl × 20 × 30	
주 : Cl = 4.2′ : 폭이 9′인 경우 8.4′ : 폭이 5′인 경우	

품목번호	품 목 명	단위	설비		표준조업시간	표준조업일수
			설 비 명	단위		
			④ 냉 압 기	대	20	30
			⑤ 열 압 기	대	20	30
41910	펄 프	M/T	○ 쇄목펄프의 경우 ① 쇄목기	대	24	30
41920	지 류	M/T	○ 화학펄프의 경우 ② 증해 솥(木釜)	기	24	30
41920	지 류	M/T	① 장망식초지기 ② 환망식 〃 ③ 단망식 〃 ④ 기 타	대	24	30
41110	판 지	M/T	지류와 같음	대	24	30

표 준 생 산 능 력 산 식	개 념 및 포 괄 범 위
C = Batch 당 냉압능력 (매) × 60 (분) × 1 냉압시간 (분) × 20 × 매당 ㎥ × 30 C = Opening 의 수 (매) × 60(분) × 1 열압시간 (분) × 20 × 30 × 매당 ㎥	
C = 쇄목기 정격출력 (kw) × 계수 × 30 계수 = $\frac{\text{시간당생산량} \times 24\text{시간}}{kw}$	쇄목펄프, 화학펄프 포함 (34111 전체)
C = 증해 솥 용적 (㎥) × 펄프수율 (kg / ㎥) × 증해회수 (회 / 일) × $\frac{1}{1000}$ × 30 C = 1 일 건조능력 (kg / ㎡) × 드라이야 총면적 (㎥) × $\frac{1}{1000}$ × 30	신문용지, 백상지, 중질지, 아트지, 크라프트지 및 박엽지를 포괄하고, 건축용지 (갈포벽지, 건축용지) 와 한지 (창호지, 장판지 등) 및 벽지제외 (34112 , 34113 , 34114 포함)
C = 평량 (gr/㎡) × 완성취득 (m) × 분속 (m / 분) × 총효율 × 60분 × 24 × 30 × $\frac{1}{1000}$ × $\frac{1}{1000}$	마닐라판지, 고형표백판지, 골판지, 원지 포함 (골판지로 착오하지 말것)

품목번호	품 목 명	단위	설 비		표준 조업 시간	표준 조업 일수
			설 비 명	단위		
41180	금 속 박 지	M/T	Coater (도금기)	대	16	30
51010	에 틸 렌	M/T	① 압축기 ② 냉동기	대	24	30
51020	프 로 필 렌	M/T	① 압축기 ② 냉동기	대	24	30
51050	벤 젠	M/T	증 유 탑	기	24	28
51060	톨 루 엔	M/T	〃	기	24	28
51070	키 실 렌	M/T	〃	기	24	28
51090	V . C . M (염화비닐모노머)	M/T	분 해 로	기	24	30
51100	스티렌모노머	M/T	반 응 기	기	24	28
51110	메 탄 올	M/T	정 제 탑	기	24	28
51130	에틸렌 글리콜	M/T	반 응 탑	기	24	28
51140	스테아린산	M/T	경 화 로	기	24	28

표 준 생 산 능 력 산 식	개 념 및 포 괄 범 위
	※ 총효율=완성감손×초효율×운전율=약 91%
	34119전체
C =분당 Coater의 속도(m)×〔원지 평당(gr / m²)+Foil의 평량(g / m²)〕×폭(mm)×$\frac{1}{1,000}$×60분×16×30×$\frac{1}{1,000}$	은박지 포함 Foil =은박
C =1일 생산능력(M/T)×30	
〃	
C =1일 생산능력(M/T)×28	벤 졸
〃	메틸벤젠
〃	디메틸벤젠
C =1일 생산능력(M/T)×30	
C =1일 생산능력(M/T)×28	
〃	
〃	N.E.G, T.E.G
C =경화관 1일 Batch당 생산능력(M/T / 일)×설비대수×제품수율×28	

품목번호	품 목 명	단위	설 비		표준조업시간	표준조업일수
			설 비 명	단위		
51170	테 레 프 탈 산	M/T	반 응 탑	기	24	28
51180	아크릴로니트릴	M/T	〃	기	24	30
51190	카 프 로 락 담	M/T	〃	기	24	30
51200	황산 (98 %)	M/T	접촉실, 연실	실	24	28
51230	소 다 회	M/T	화 소 로	기	24	30
51250	암모니아	M/T	합 성 탑	기	24	30
51270	압축산소	천 m^3	압 축 기	대	24	28
51310	요소비료	M/T	합 성 탑	기	24	30
51320	용성인비	M/T	〃	기	24	30
51340	복합비료	M/T	〃	기	24	30
51380	S . B . R (합성고무)	M/T	반 응 탑	기	24	30
51420	폴리푸로피렌 (P . P)	M/T	중 합 기	대	24	30
51430	폴리스치렌 (P . S)	M/T	〃	대	24	30
51440	폴리염화비닐 (P . V . C)	M/T	〃	대	24	30
51450	아세테이트섬유	M/T	〃	대	24	30
51910	비스코스섬유	M/T	〃	대	24	30
51460	폴리에스터섬유	M/T	〃	대	24	30

표 준 생 산 능 력 산 식	개 념 및 포 괄 범 위
C = 1일 생산능력 (M/T) × 28	T.P.A (H.T.A)
C = 1일 생산능력 (M/T) × 30	
〃	
C = 1일 생산능력 (M/T) × 28	① 98%로 환산할 것 ② 발열황산인 올레움 포함
C = 1일 생산능력 (M/T) × 30	탄산나트륨, 무수탄산소다
〃	
C = 1일 생산능력 (M/T) × 28	
C = 1일 생산능력 (M/T) × 30	
〃	
〃	
〃	
〃	
〃	
〃	
〃	아세테이트 Tow 포함
〃	※ 능력 단독조사 지정
〃	폴리에스터 Staple Fiber 포함
	〃 Staple Top 포함

품목번호	품 목 명	단위	설 비		표준조업시간	표준조업일수
			설 비 명	단위		
51920	나이론섬유	M/T	중 합 기	대	24	30
51470	아크릴릭섬유	M/T	〃	대	24	30
52910	비 누	M/T	압축기, 성형기	대	10	25
52090	합성세제	M/T	분무건조기	대	16	25
52100	치 약	M/T	Tube Line	대	24	30
52170	카아본블랙	M/T	반 응 로	기	24	25
53910	정 유	kℓ	원유 증유장치		24	30
54010	연 탄	M/T	윤 전 기	대	8	25
54020	코 크 스	M/T	Coke Oven (爐)	기	24	28
55010	자동차용 타이어	천본	① Extruder (압축기)	대	24	30

표 준 생 산 능 력 산 식	개 념 및 포 괄 범 위
C = 1일 생산능력 (M/T) × 30	나이론 Staple Fiber 포함
〃	나이론 Staple Top 포함 아크릴릭 Staple Top 포함 아크릴릭 Staple Fiber 포함
C = 시간당 기계능력 (M/T) × 10 × 25	화장비누, 세탁비누 포함
C = 시간당 기계능력 (M/T) × 16 × 25	연성합성세제, 경성합성세제 포함
C = 1일 생산능력 (kg) × 30	
C = 1일 생산능력 (M/T) × 25	
C = 1일 원유처리능력 (kℓ) × 30	동태 포괄품목들의 단순한 물량 합산이 아니고 원유 증유장치의 석유(원유) 정제능력과 실 원유 정제량으로 조사할 것. 젯트유, 휘발유, 나프타, 등유, 솔벤트, 경유, 중유, 방카C유, 윤활유, 액화석유가스 포함.
C = 시간당능력 (M/T) × 8 × 25	
C = 탄화실수 × 문당(門當) 원료장입량 × 가동율(문당 생산회수) × Coke 회수율 × 28	
$C = \frac{\text{시간당 기계능력 (kg)} \times 24 \times 30}{\text{본당무게 (12.3 kg)}} \times \frac{1}{1,000}$	

품목번호	품 목 명	단위	설 비		표준조업시간	표준조업일수
			설 비 명	단위		
			② 가 류 기	대	24	30
55020	자전거용 타이어	천본	가 류 기	대	24	25
55030	자동차용 튜 브	천본	① 가류기 ② 압출기 ③ 경련기	대	24	25
55070	고무장화 및 우화	천족	가 류 기	대	24	25
55080	운 동 화	천족	〃	대	10	25
62020	판 유 리	상자	인상기, 용융로	대	24	30

종 류	환 산 율	종 류	환 산 율
보통판 2 ㎜	1	형 판 2 ㎜	1.0
〃 3 ㎜	1.5	〃 4 ㎜	2.0
〃 5 ㎜	2.5	〃 5 ㎜	2.5
〃 6 ㎜	3.0	강입판	3.0

표 준 생 산 능 력 산 식	개 념 및 포 괄 범 위
C = 1회 가류능력 (본) × $\frac{24}{\text{가류시간}}$ × 30 × $\frac{1}{1,000}$	
C = 1회 가류능력 (본) × 24 / 가류시간 × 25 × 1 / 1,000	
C = 1회 가류능력 (본) × 24 / 가류시간 × 25 × 1 / 1,000	
C = 1회 가류능력 (족) × 24 / 가류시간 × 30 × 1 / 1,000	
C = 1회 가류능력 (족) × 10 / 가류시간 × 25 × 1 / 1,000	특수운동화 + 일반운동화
C = 1일 생산능력 (상자) × 30	

품목번호	품 목 명	단위	설 비		표준 조업 시간	표준 조업 일수
			설 비 명	단위		
69020	타 일	천㎡	Kiln	대	24	25
69030	내화벽돌	M/T	Kiln	대	24	25
69050	시멘트 크링커	천M/T	Kiln	대	24	30
69090	흄 관	본	원 심 대	대	16	25
69910	콘크리트 전주 및 파일	본	회전대, Mould	대	16	25
69130	석면스레트	천㎡	Wet M / C		24	25

표 준 생 산 능 력 산 식	개 념 및 포 괄 범 위
C = 1회 소성능력 (m^3) × 월간회전수 × $\frac{1}{1,000}$ ※ 1회 소성능력 = 대차당 적재량 × 1회 소성에 필요한 대차수	
C = 타일과 동일	330매 = 1M/T
C = Kiln의 시산소출량 (M/T) × 24 × 30 × $\frac{1}{1,000}$	1. 백 시멘트 포함 2. 반드시 크링카 생산능력과 생산량을 조사하고 완제품 상태의 시멘트와 혼동하지 말 것. (시멘트 조사 불요)
C = 시간당 회전대능력 (본) × 16 (시간) × 조업율 (0.8) × 25 (일)	조업율 : 1일 총조업시간에 대한 휴식시간, 운전정지시간등을 제외한 주공정인 회전대의 실운전시간의 비율 일반적으로 0.8이나 개별사업체 실정을 고려하여 탄력적으로 적용할 것.
C = 흄관과 같음	
C = 1일 공칭능력 (M/T) × 25 × 환산율 × $\frac{1}{1,000}$	

품목번호	품 목 명	단위	설 비		표준조업시간	표준조업일수
			설 비 명	단위		
71010	선 철	M/T	① 고 로	기	24	30
			② 전 기 제 선 로	기	22	30
			③ 합금철용 아크식 전기로	기	22	30

표 준 생 산 능 력 산 식	개 념 및 포 괄 범 위
1) 고로인 경우 C = 내용적(㎥) ÷ 0.8 (M/T / ㎥) × 30 주 : ① 내용적은 유효 내용적을 말함. ② 0.8은 로내 이용계수, 즉 출선비 2) 전기 제선로인 경우 C = 변압기 정격용량 (KVA) × 0.9 × 0.85 × 22 × 30 ÷ 전력원단위 (Kwh / t) 주 : ① 0.9 = 역율 ② 0.85 = 전기용량에 대하여 85%의 부하로 조업함을 말함. ③ 전력원단위 : 사업체의 과거 실적에 따라 산출 3) 합금철용 아크식 전기로인 경우 C = 변압기 정격용량 (KVA) × 0.9 × 0.85 × 22 × 30 ÷ 전력원단위	재생선철 제외

품목번호	품 목 명	단위	설 비		표준조업시간	표준조업일수
			설 비 명	단위		
71020	합 금 철	M/T	전 기 로	기	22	28
71910	조 강	M/T	① 평 로	기	24	30
		M/T	② 전 로	기	24	30
		M/T	③ 아크식전기제강로	기	24	30
		M/T	④ 유도식 전기로	기	24	30

표 준 생 산 능 력 산 식	개 념 및 포 괄 범 위
C＝변압기 정격용량(KVA)×0.9×0.85×22×28÷전력원단위(Kwh/M/T 주:① 0.9:역율 ② 0.85:부하율 ③ 전력원단위:사업체의 과거실적에 따라 산출	경철, 망간철, 규소철, 크롬철, 몰리브데늄철, 니켈철, 규소망간철, 티타늄철, 텅스텐철, 기타 합금철 등을 포함
① 용선 사용시 C＝강용중량(M/T)×4회×30 ② 냉선 사용시 C＝강용중량(M/T)×4회×30	한국표준산업분류중 세세분류 37102(제강업) 전체 포괄 (압연용강괴, 단조용강괴, 주강, 주입강, 연속주조강 및 강반성품 ―스라브, 블룸, 빌렛)
① 2기인 경우 C＝양괴(t/회)×35회×30×0.5 ② 3기인 경우 C＝양괴(t/회)×35회×30×$\frac{2}{3}$ C＝1회전당 공칭능력×1일회전수×30 C＝발전기용량(KVA)×0.00083×8760÷12	※ 동태:강괴, 주강, 강반성품을 포함.

품목번호	품 목 명	단위	설비		표준조업시간	표준조업일수
			설 비 명	단위		
71920	압 연	M/T	① 압 연 기	대	20	25
			② 신 선 기	대	20	25
			③ 조 관 기	대	20	25
71160	강 관	M/T	조 관 기	대	10	25
71170	주 철 관	M/T	① 용 선 로	기	10	25
			② 전 기 로	기	10	25
71220	석 도 강 판	M/T	① 전기도금법		20	25
			② 용융도금법		24	30
71230	아연도강판	M/T	① 전기도금법		20	25
			② 용융도금법		24	30
71240	와이어로프	M/T	연 선 기	대	22	25
72010	전 기 동	M/T	전 해 조	기	24	30

표 준 생 산 능 력 산 식	개 념 및 포 괄 범 위
C =시간당 공칭능력 (M/T) × 20 × 25 〃 〃	한국표준산업분류 37103 (철강, 압연업), 전체포괄, (봉강, 철근, 형강, 강판, 대강, 궤조, 선재 등) ※ 동태포괄품목 (봉강, 스텐레스스틸관, 열연대강, 철근, 형강, 중후판, 열연박판, 냉연박판, 전기강판, 선재)
C =조관기 Speed (m /분) × 60 × × 25 ×조관기당 평균가동율×기준규격당 무게 (M/T / m)	한국표준산업분류 37104 (강관 제조업) 전체 포괄. "무계목관, 전기용접강관, 단접강관, 아크용접강관, 리벳트관, 관부착물, 고탄소강관, 합금강강관 등 " (주철관 제외)
C =시간당 능력 (M/T) × 10 × 25 〃	한국표준산업분류 37105 (주철관제조업) 전체포괄
C = 1일 생산능력 (M/T) × 25 C = 1일 생산능력 (M/T) × 30 C = 1일 생산능력 (M/T) × 25 C = 1일 생산능력 (M/T) × 30 C =시간당 연선능력 (M/T) × 22 × 25	전기도금석도강판, 열지도금석도강판 (평판, 파판 포함) 전기도금, 열지도금, 착색도금
① 전기분해에 의하는 경우	동쇼트, 해감봉 등 제외

품목번호	품 목 명	단위	설 비		표준 조업 시간	표준 조업 일수
			설 비 명	단위		
72020	알미늄 괴	M/T	전 해 조	기	24	30
72040	연 괴	M/T	〃	기	24	30
72050	아 연 괴	M/T	〃	기	24	30
72910	동 압연품	M/T	압 연 기	대	10	25
72120	알미늄 판	M/T	압 연 기	대	10	25

표 준 생 신 능 력 산 식	개 념 및 포 괄 범 위
C＝전해당량×전류효율×전류×24×30 ×전해조수×조업율×제품율 주 : 전류효율＝$\frac{\text{실 석출량}}{\text{이론석출량}}$ 전류 : 실통전전류 조업율＝통전율×전해조 조업율 ＝ $\frac{\text{실통전시간}}{\text{역시간}}$ × $\frac{\text{연가동가능전해조수}}{\text{연전해조수}}$ 제품율＝$\frac{\text{실전착량－조반량}}{\text{실천착량}}$ × 주조율 ② 전기분해 이외의 경우 C＝광석장입량 (t /로당) ×장입물품 위×24×30×로수 ×조업율 × 제품율 C＝전기동과 같음 〃 〃	
C＝시간당 공칭능력 (M/T) × 10×25	한국표준산업분류 세세분류 37202 (제 1 차 비철금속압연업) 중 동가공제품에 국한함. (동봉 및 형강, 나동선, 동판 및 띠, 동관 및 중공봉)
C＝시간당 공칭능력 (M/T) × 10×25	

품목번호	품 목 명	단위	설 비		표준조업시간	표준조업일수
			설 비 명	단위		
72130	알미늄샷시	M/T	용해로 압출기	대	24	25
81130	통조림용관	천개	제동기 (Body Maker)	대	8	25
81300	용 접 봉	M/T	도 장 기	대	22	28
82030	경 운 기	대				
82060	선 반	대				
82130	굴 착 기	대				
82160	직 기	대				
82210	전자계산기	대				
82240	동력펌프	대				
82250	공기압축기	대				
82280	크 레 인	M/T	인발중량을 기준			
82290	엘리베이터	대				
82910	재 봉 틀	대				

표 준 생 산 능 력 산 식	개 념 및 포 괄 범 위
C = 1일 생산능력 (M/T) × 25	
C = 시간당 기계능력 (천개) × 8 × 25	미술관 (과자캔, 설탕캔, 우유캔 등) 제외
C = 시간당 피복 (被覆) 능력 (M/T) × 22 × 28	
※ 기계공업의 능력산정에 있어서는 월말현재의 보유설비에 표준적인 생산제조건 (원재료, 동력, 자금, 노동력 등)이 주어지고 실적을 고려한 표준적인 월간 조업일수 및 조업시간으로 그 제품을 생산할 경우의 능력을 지정된 단위에 따라 기입.	면직기, 견직기, 모직기까지 (편직기 제외)
① 보유설비라 함은 진부화 또는 기타의 이유로 능률이 떨어져 사용할 수 없는 설비와 갱신에 가까운 정도로 많은 개조를 하여야 가동 가능한 설비를 제외한 설비를 말한다.	공업용, 주유소용, 가정용 (수동식 제외) 펌프 및 양수기, 공기펌프, 진공펌프 포함.
② 하나의 설비로 2개이상의 제품을 생산하고 있는 경우에는 각각의 제품을 집중적으로 생산하는 능	가정용 (손틀, 발틀, 전기식재봉기) 재봉틀, 공업용제봉틀 포함.

품목번호	품 목 명	단위	설 비		표준조업시간	표준조업일수
			설 비 명	단위		
82370	베 아 링	kg	연 마 능 력			
82380	윈도우형 에어콘	대				
83910	전 동 기	HP				
83030	변 압 기	KVA				
83920	TV수상기	대	최종조립능력			

표 준 생 산 능 력 산 식	개 념 및 포 괄 범 위
력이 아니고 과거의 생산실적을 감안한 비율로 그 제품을 생산할 때의 능력을 말한다.	
③ 설비(시설명)난에 특기되어 있는 제품에 대해서는 각각의 제품에 특기되어 있는 단계에 착안하여 생산능력을 산정하고 기타의 제품에 있어서도 그 제품의 전 생산공정을 총합적으로 판단하여 생산능력을 산정한다.	볼베아링, 로울러베아링, 스러스트베아링을 포함한다. (메탈베아링 및 유니버샬 조인트 제외) 패케이지형 제외시킬 것 직류전동기, 교류전동기 포함 (내연기관용 전동기 및 수리품 제외) 송배전용 및 기계용 등 강전용 변압기만 조사 (약전 초인종 등과 같은 기계용변압기의 소형 및 수리품은 제외) 흑백TV, 칼라TV, 컴비네이숀TV 포함조사 TV게임 및 공업용(1TV), 학술조사용등 특수분야 유선TV 제외

품목번호	품 목 명	단위	설 비		표준조업시간	표준조업일수
			설 비 명	단위		
83090	라디오 수신기	대	최종조립능력			
83100	전 축	대	〃			
83110	녹 음 기	대	〃			
83120	T V 튜 너	천개	〃			
83130	확 성 기	천개	〃			
83150	전 화 기	대	〃			
83160	자동식 전화교환기	회선	〃			
83170	브라운관	개				

표 준 생 산 능 력 산 식	개 념 및 포 괄 범 위
	라디오, 시계, 결합체는 포함하나 기타 소리장비와 결합된 라디오는 제외 앰프, 턴테이블, 플레이어, 스피커 등이 조립되어 있는 것. 앰프셑트 포함조사 카셋트녹음기, 라디오부착 녹음기 포함 TV 및 음향기기 등에 부착되는 스피커와 옥내외 장치하는 스피커 포함. (트란지스터 메가폰, 학교·야외 연락용으로 쓰이는 나팔형, 대형 및 헤드폰, 전화기구용 등 소형은 제외) 자동식, 수동식, 공전식 크로스바교환기, 전자교환기 등 조사 흑백, 칼라 포함 TV수상기의 영상신호를 화면으로 변화시키는 음극선관

품목번호	품 목 명	단위	설 비		표준조업시간	표준조업일수
			설 비 명	단위		
83190	집적회로(IC)	천개				
83200	트랜지스터 (T.R)	천개	실가공장치의 능력			
83930	콘 덴 사 (전자기기용)	천개			10	25
83270	전기냉장고	대	최종조립능력			
83280	선 풍 기	대	〃			
83290	전기세탁기	대	〃			
83940	통신용 및 전력용 케블	M/T				
83340	형광전구	천개				
83360	축 전 지	개				

표 준 생 산 능 력 산 식	개 념 및 포 괄 범 위
	(관측용, 브라운관, 절환관, 프린트관, 축전관 및 TV용 브라운관중 수리품은 제외)
	후막집적회로, 피막집적회로, 반도체집적회로, 혼성집적회로 포함 조사
	전자기기용만 조사, 고정, 가변구분 불요(전기기기용 제외) (상업용 냉장고 포함)
	각종 절연 전화전신선 및 전자선, 전력용선 및 전력용 케이블선 조사
	직관형광등, 환형형광등 포함 조사(네온사인, 가스방전관 및 구(수은등), 헬륨, 알곤가스, 크세논 등 기타 방전관 제외)
	전류를 흐르게 하면 화학적 변화가 역행 충전 재생되는 것으로 건전지와 구분되는 2차 전지임.

품목번호	품 목 명	단위	설 비		표준조업시간	표준조업일수
			설 비 명	단위		
83370	건 전 지	천개				
84910	철강선박	G/T				
84070	객 화 차	량				
84090	승 용 차	대				
84920	버 스	대				
84930	트 럭	대				
84150	특 수 차	대				
84160	자동차용 내연기관	대				
84200	모터싸이클	대	최종조립능력			

표 준 생 산 능 력 산 식	개 념 및 포 괄 범 위
	(납산축전지와 알카리축전지 조사) 충전 재생할 수 없는 1차 전지(불항생전지, 습전지, 농축 전지, 카본전극봉 제외) 철강유조선, 철강화물선, 특수화물선, 철강어선 포함) 수하물차, 침대차, 식당차, 화차 포함 조사 (병원차, 죄수차, 검사차 및 기타 특수목적 차량 제외) 지프 포함 대형·소형버스 포함 대형·중형·소형트럭 포함 구난차, 소방차, 살수차, 청소차, 은행차, 진료차, 방송차 등을 포함 불꽃점화기관, 압축점화기관 포함 51cc이상 이륜 및 삼륜 포함 모타스쿠터, 오싸이클 골프차, 눈차 및 기타 특수목적차 제외

품목번호	품 목 명	단위	설 비		표준조업시간	표준조업일수
			설 비 명	단위		
85080	손목시계	개	Move Ment			
			최종조립능력			
85090	벽 시 계	개	최종조립능력			
	간 장	kℓ	① 숙성탱크	대	9	25
			② 제 품 기	대	9	30
	식용포도당	M/T	결 정 관	대	20	30
	물 엿	M/T	가 마	대	10	25
	이 스 트	kg	발 효 조	대	8	25
	청 주	kℓ	사 입 부	대	8	25
	화섬사 양말	천족	양 말 기	대	24	25
	메리야스 내의	천매	편 직 기	대	24	30
	로 우 프	kg	제 강 기	대	8	25
	각재 및 판재	m^3	제 재 기	대	10	30

표 준 생 산 능 력 산 식	개 념 및 포 괄 범 위
	일반 손목시계 또는 전자손목시계 포함. 전자벽시계 포함 〔계기판시계(옥외설치용 전기시계), 특수용시계(예, 집수기록용) 제외〕
C=총용량(석)× $\frac{4}{5}$ × 월간회전수× $\frac{180}{1,000}$	1석=180 ℓ
C=시간당 기계능력(Bbl)×9×25 ×$\frac{\text{Btl 당 ℓ}}{1,000}$	
C=1일 생산능력(M/T)×30	
C=1일 공칭능력(M/T)×25	
C=1일 생산능력(kg)×25	
C=사입부당 출고량(ℓ)×월간회전수 ×$\frac{1}{1,000}$	
C=시간당 기계능력(족)×24×25 ×$\frac{1}{1,000}$	
C=1일 생산능력(매)×30	
C=1일 생산능력(kg)×25	
C=1시간당능력(m^3)×10×30	

품목번호	품 목 명	단위	설 비		표준조업시간	표준조업일수
			설 비 명	단위		
	하 드 보 드	m^3	① Digester	대	8	25
			② 조 형 기	대	8	25
			③ 열 압 기	대	8	25
	포 르 말 린	M/T			24	30
	염산(35%)	M/T	전 해 조	조	24	30
	가 성 소 사 (97%)	〃	전 해 조	조	24	30
	카 바 이 트	〃	전 기 로	기	24	30
	아세칠렌개스	m^3	압 출 기	대	24	30
	요 소 수 지	M/T	중 합 기	대	24	25
	메 라 닌	kg	〃	대	24	30
	폴리에치렌 (P.E)	kg	〃	대	24	30
	세 로 판 지	M/T	제 막 기	대	24	30
	포리푸로피렌섬유	M/T	중 합 기	대	24	30
	다이나마이트	kg			24	25
	도 화 선	km			8	25

표 준 생 산 능 력 산 식	개 념 및 포 괄 범 위
C = Batching 능력 (사이) / Cooking time (시간) × 8 × 25 × $\frac{1}{9.2}$ × 환산율 (0.01 ㎥)	
C = 60 (분) / 조형시간 (분) × 8 × 25 × 환산율 (0.01㎥)	
C = Opening 의 수 (매) / 열압시간(분) × 60 (분) × 8 × 25 × 환산율 (0.01 ㎥)	
C = 1 일 생산능력 (M/T) × 30	
〃	
〃	
〃	
〃	
〃	아미노수지 포함
〃	
〃	
C = 1 일 생산능력 (M/T) × 30	
C = 1 일 생산능력 (M/T) × 30	
C = 1 일 생산능력 (kg) × 25	
C = 1 일 생산능력 (km) × 25	

품목번호	품 목 명	단위	설 비		표준조업시간	표준조업일수
			설 비 명	단위		
	뇌 관	천개			8	25
	인 쇄 잉 크	kg	혼합기, 증류기	대	10	25
	활 성 탄 소	kg	건 류 기	대	24	30
	고 무 신	천족	가 류 기	대	24	30
	스 폰 지	m^3	압출기, Roller	대	8	25
	유 리 병	M/T	용 해 로	기	10	30
		개	제 병 기	대	10	30
	콘크리트관	본	회전대, Mould	대	16	25
	기계용주물	M/T	① 용선로	기	10	25
			② 전기로	기	10	25
			③ 용선로	기	24	29
	보 일 러	t/h대				
	볼트 및 낫트	M/T	절삭기, 프레스	대	10	25
	금속관이음쇠	kg	선 반	대	10	25
	철 사 망	kg	제 망 기	대	10	25
	내 연 기 관	HP/대				
	동력탈곡기	대				
	농업용트랙타	대				
	불 도 저	대				
	연 사 기	대				
	표크리프트	대				

표 준 생 산 능 력 산 식	개 념 및 포 괄 범 위
C = 1일 생산능력 (천개) × 25	
C =시간당 기계능력 (kg) × 10 × 25	
C = 1일 생산능력 (kg) × 30	
C = 1회 가류능력 (족) × 24/가류시간 × 30 × 1 / 1,000	
C =시간당 기계능력 (kg) × 8 × 25	
C =용해로 용량 (M/T) ×월간회전수	
C =시간당 능력 (m) × 10 × 30	
C =회전대능력 (본) ×조업율 (0.8)×25	
C =시간당 능력 (M/T) × 10 × 25	
〃	
C =용성도 1일능력 (M/T) × 29	
일반기계 참조	
C =시간당 기계능력 (M/T) × 10 × 25	
C =시간당 기계능력 (M/T) × 10 × 25	
〃	
※ 기계공업의 능력산정은 " 표준생산능력 산정기준표 " 일반기계란 참조	

품목번호	품 목 명	단위	설 비		표준조업시간	표준조업일수
			설 비 명	단위		
	발 브	개				
	정 류 기	대				
	저압축전지	개				
	음성증폭기	개				
	백열전구	천개				
	완 충 기	개				
	자전거차체	대				
	자전거링	대				
	피 아 노	대	최종조립능력			
	올 겐	대	〃			

표 준 생 산 능 력 산 식	개 념 및 포 괄 범 위

광공업동태조사 지정품목분류표

품목번호	품 목 명	단 위
	2. 광 업	
	21 석 탄 광 업	
	210 석 탄 광 업	
01010	무 연 탄	M/T
	23 금 속 광 업	
	230 금 속 광 업	
03010	철 광 석	M/T
03020	중 석 광 석	〃
03030	은 광 석	kg
03040	연 광 석	M/T
03050	아 연 광 석	〃
	29 기 타 광 업	
	290 기 타 광 업	
09010	화 강 암	M/T
09020	석 회 석	〃
09030	쇄 석	〃
09040	모 래	m^3
09050	자 갈	m^3
09060	고 령 토	M/T
	3. 제 조 업	
	31. 음식료품및 담배제조업	

품목번호	품 목 명	단 위
	311-312. 식료품제조업	
11010	소 시 지	M/T
11020	분 유	〃
11030	아 이 스 크 림	kℓ
11040	빙 과	M/T
11050	처 리 우 유	〃
11060	유 산 균 발 효 유	〃
11070	농 산 물 통 조 림	〃
11080	수 산 물 통 조 림	〃
11090	냉 동 물 고 기	〃
11100	대 두 유 (콩 기 름)	kℓ
11110	쇼 팅 (쇼 트 닝)	M/T
11120	밀 가 루	〃
11130	식 빵	〃
11140	건 과 자	〃
11150	스 낵 과 자	〃
11160	라 면	〃
11170	인 스 턴 트 면 류	〃
11180	정 당	〃
11190	설 탕 과 자	〃
11200	껌	〃

품목번호	품 목 명	단 위
11210	인 조 얼 음	M/T
11220	간 장	kℓ
11230	글 루 타 민 산 소 다	M/T
11240	두 부	〃
11250	녹 말 가 루 (전 분)	〃
11260	식 용 포 도 당	〃
11270	물 엿	〃
11280	커 피	kg
11290	분 말 인 삼	〃
11300	배 합 사 료	M/T
11310	보 조 사 료	〃
	313. 음료품제조업	
13010	주 정 (메 틸 알 콜)	kℓ
13020	소 주	〃
13030	위 스 키	〃
13040	럼 주	〃
13050	탁 주	〃
13060	청 주	〃
13070	맥 주	〃
13080	사 이 다	〃
13090	콜 라	〃
13100	쥬 스	〃

품목번호	품 목 명	단 위
	314. 담 배 제 조 업	
14010	필 터 담 배	백만본
14020	필 터 없 는 담 배	〃
	32. 섬유, 의복및 가 죽 산 업	
	321. 섬 유 제 조 업	
21010	생 사	kg
21020	면 사	M/T
21021	순 면 사	〃
21022	면 혼 방 사	〃
21030	방 모 사	〃
21040	순 소 모 사	kg
21050	혼 방 소 모 사	〃
21060	합 성 섬 유 사	M/T
21070	재 생 섬 유 방 적 사	〃
21080	혼방합성섬유방적사	〃
21090	연 사	〃
21100	담 요	㎡
21110	천 막	매
21120	타 월	kg
21130	스 타 킹	천켤레
21140	면 양 말	〃
21150	인 조 섬 유 양 말	〃

품목번호	품 목 명	단 위
21160	메리야스내의	천 매
21170	메리야스외의	〃
21180	로 프	M/T
21190	이망사(연승)	kg
21200	어 망	M/T
21210	면 직 물	천 ㎡
21220	순소모직물	〃
21230	방모직물	㎡
21240	모혼방직물	천 ㎡
21250	순본견직물	㎡
21260	순합성섬유직물	천 ㎡
21270	혼방합성섬유직물	〃
21280	재생섬유직물	〃
21290	타이어코드사	M/T
21300	필 터	〃
	322. 의복제조업	
22010	의 복 류	백만원
22011	남자용양복	〃
22012	남자용셔츠, 작업복	〃
22013	여자용외의	〃
22014	소아용외의	〃
22015	여자용및소아용내의	〃
22016	가죽및모피의복	백만원
22017	모 자	〃
22018	장 갑	〃
22019	달리분류되지않는의복	〃
	323. 가죽,대용가죽및 모피제품제조업	
23010	소 가 죽	천 ㎡
23020	재생및인조가죽	㎡
23030	트렁크(비닐제포함)	개
23040	핸드백	천 개
	324. 신발제조업	
24010	가죽신(남자용)	켤레
24020	가죽신(여자용)	〃
24030	군화및가죽장화	〃
24040	등 산 화	〃
24050	실 내 화	천켤레
	33. 나무 및 나무제품제조업(가구포함)	
	331. 나무 및 나무와 콜크제품제조업(가구제외)	
31010	소 할 재	㎥
31020	각 재	〃

품목번호	품 목 명	단 위
31030	판 재	㎥
31040	침 목	본
31050	장 지 문	개
31060	합 판	㎥
31070	재 생 목 재	〃
31080	포 장 상 자	〃
	332. 가구 및 장지문 제조업 (금속 가구제외)	
32010	의 자	개
32020	장 농	〃
32030	식 탁	〃
32040	화 장 대	〃
32050	장 식 장	〃
32060	책 상	〃
	34. 종이 및 종이제품 제조업, 인쇄 및 출판업	
	341. 종이 및 종이 제품제조업	
41010	쇄 목 펄 프	M/T
41020	화 학 펄 프	〃
41030	신 문 용 지	〃
41040	백상지 (모조지)	〃
41050	중 질 지	〃

품목번호	품 목 명	단 위
41060	아 트 지	M/T
41070	크 라 프 트 지	〃
41080	갈 포 벽 지	천 ㎡
41088	〃	M/T
41090	위 생 용 지	〃
41100	박 엽 지	〃
41110	판 지	〃
41120	골 판 지	〃
41130	골 판 지 상 자	천 ㎡
41138	〃	M/T
41140	크라프트지포대	천 매
41148	〃	M/T
41150	판 지 상 자	천 ㎡
41158	〃	M/T
41160	벽 지	〃
41168	〃	〃
41170	복 합 지	〃
41180	금 속 박 지	〃
	342. 인쇄 및 출판 관련산업	
42010	일 간 신 문	천 부
42020	정 기 간 행 물	천 권
42030	서 적	〃

품목번호	품 목 명	단 위
42031	일 반 서 적	천 권
42032	35 문 고 판 서 적	〃
42040	일 반 인 쇄 물	연
42050	노 트	천 권
42060	앨 범	〃
	35. 화학, 석유, 석탄, 고무 및 플라스틱 제조업	
	351. 산업용 화학물제조업	
51010	에 틸 렌	M/T
51020	프 로 필 렌	〃
51030	부 타 디 엔	〃
51040	싸 이 클 로 핵 산	〃
51050	벤 젠	〃
51060	톨 루 엔	〃
51070	키 실 렌	〃
51080	알 킬 벤 젠	〃
51090	염 화 비 닐 모 너 머 (V. C. M)	〃
51100	스티렌모너머 (S.M)	〃
51110	메 탄 올	〃
51120	호 루 말 린	〃
51130	에 틸 렌 글 리 콜	〃
51140	스 테 아 린 산	〃

품목번호	품 목 명	단 위
51150	무 산	M/T
51160	무 스 프 탈 산	〃
51170	테 레 프 탈 산	〃
51180	아 크 릴 로 니 트 릴	〃
51190	카 프 로 락 담	〃
51200	황 산	〃
51210	가 성 소 다	〃
51220	가 소 제	〃
51230	소 다 회	〃
51240	염 소	〃
51250	암 모 니 아	〃
51260	카 바 이 트	〃
51270	산 소	천 ㎥
51278	〃	M/T
51280	아 세 틸 렌 개 스	〃
51290	염 료	〃
51300	유 안 비 료	〃
51310	요 소 비 료	〃
51320	용 성 인 비	〃
51330	용 과 린	〃
51340	복 합 비 료	〃
51350	살 충 제	〃

품목번호	품 목 명	단 위	품목번호	품 목 명	단 위
51360	살 균 제	M/T	52061	신 경 계 용 약	백만원
51370	제 초 제	〃	52062	해 열 진 통 제	〃
51380	S . B . R	〃	52063	소 화 기 계 용 약	〃
51390	요 소 수 지	〃	52064	비 타 민 제	〃
51400	폴 리 우 레 탄	〃	52065	자 양 강 장 제	〃
51410	폴 리 에 틸 렌	〃	52066	혈 액 및 체 액 용 약	〃
51411	저 밀 도 폴 리 에 틸 렌	〃	52067	항 감 염 제	〃
51412	고 밀 도 폴 리 에 틸 렌	〃	52068	기 타 의 약 품	〃
51420	폴 리 프 로 필 렌	〃	52070	세 탁 비 누	M/T
51430	폴 리 스 티 렌	〃	52080	화 장 비 누	〃
51440	P . V . C	〃	52090	세 탁 용 합 성 세 제	〃
51450	아 세 데 이 트 섬 유	〃	52100	치 약	〃
51460	폴 리 에 스 터 섬 유	〃	52110	샴 푸	kg
51470	아 크 릴 릭 섬 유	〃	52120	화 장 크 림	〃
51480	셀 로 판 지	〃	52130	화 장 백 분	〃
	352. 기타화학제품 제 조 업		52140	화 장 수	〃
52010	바 니 스	ℓ	52150	글 리 세 린	M/T
52020	유 성 페 인 트	kℓ	52160	다 이 나 마 이 트	〃
52030	락 카	ℓ	52170	카 본 블 랙	〃
52040	에 나 멜	kℓ	52180	인 쇄 잉 크	kg
52050	수 성 페 인 트	〃	52190	향 료	〃
52060	의 약 품	백만원	52200	계 면 활 성 제	M/T

품목번호	품 목 명	단 위
	353. 석유정제업	
53010	젯 트 유	kℓ
53020	휘 발 유	〃
53030	나 프 타	〃
53040	등 유	〃
53050	솔 벤 트	〃
53060	경 유	〃
53070	중 유	〃
53080	방 카 C 유	〃
53090	윤 활 유	〃
53100	L P G	〃
	354. 기타석유및 석탄제품제조업	
54010	연 탄	M/T
54020	코 크 스	〃
54030	아 스 팔 트	kℓ
	355. 고무제품제조업	
55010	자동차용타이어	천 본
55020	자전거용타이어	〃
55030	자동차용튜브	〃
55040	자전거용튜브	〃
55050	재생타이어	본
55060	고무신	천켤레
55070	고무장화및우화	천켤레
55080	운동화	〃
55081	운동화	〃
55082	특수운동화	〃
55090	고무호스	m
55098	〃	M/T
55110	고무벨트	천 ply
55108	〃	M/T
55110	스폰지	m³
	356. 플라스틱제품 제조업	
56010	플라스틱제품	M/T
56011	플라스틱식기및용기	〃
56012	플라스틱신발	〃
56013	플라스틱필름	〃
56014	플라스틱파이프및관	〃
56015	플라스틱레저	〃
56016	기타플라스틱제품	〃
	36. 비금속광물제품 제조업	
	361. 도기, 자기및 토기제조업	
61010	도자식기	천 개
61020	위생도기	개
	362. 유리 및 유리제품 제조업	

품목번호	품 목 명	단 위
62010	유 리 관	M/T
62020	판 유 리	상 자
62030	안 전 유 리	〃
62040	브 라 운 관 용 유 리	천 개
62041	흑백브라운관용유리	〃
62042	칼라브라운관용유리	〃
62050	보 온 병 속 유 리	kℓ
62060	유 리 병	천 개
62068	유 리 병	M/T
62070	유 리 약 병	〃
62080	유 리 식 기	천 개
62088	〃	M/T
62090	유 리 섬 유	〃
	369. 기 타 비 금 속 광물제품제조업	
69010	벽 돌 (소 성)	천 개
69020	타 일	천 ㎡
69030	내 화 벽 돌	M/T
69040	포 오 틀 란 드 시 멘 트	천 M/T
69050	시 멘 트 크 링 커	〃
69060	소 석 회	M/T
69070	콘 크 리 트 블 록	천 매
69080	콘 크 리 트 벽 돌	〃

품목번호	품 목 명	단 위
69090	흄 관	본
69098	〃	M/T
69100	콘 크 리 트 전 주	본
69108	〃	M/T
69110	콘 크 리 트 파 일	본
69118	〃	M/T
69120	레 미 콘	천 ㎡
69130	석 면 스 레 트	〃
69140	활 석 분	M/T
	37. 제 1 차금속산업	
	371. 제 1 차철강산업	
71010	선 철	M/T
71020	합 금 철	〃
71030	강 괴	〃
71040	주 강	〃
71050	강 반 성 품	〃
71060	봉 강	〃
71070	스 텐 레 스 스 틸 판	〃
71080	대 강	〃
71090	철 근	〃
71100	형 강	〃
71110	중 후 판	〃

품목번호	품 목 명	단 위
71120	열 연 박 판	M/T
71130	냉 연 박 판	〃
71140	전 기 강 판	〃
71150	선 재	〃
71160	강 관	〃
71170	주 철 관	〃
71180	회 주 물	〃
71190	가 단 주 물	〃
71200	합 금 주 물	〃
71210	단 강 품	〃
71220	석 도 강 판	〃
71230	아 연 도 강 판	〃
71240	와 이 어 로 프	〃
71250	쇠 못	〃
71260	흑 철 선	〃
71270	아 연 도 철 선	〃
71280	경 강 선	〃
	372. 제 1 차 비 철 금 속 산 업	
72010	전 기 동	M/T
72020	알 루 미 늄 괴	〃
72030	알 루 미 늄 합 금	〃
72040	연 괴	〃
72050	아 연 괴	M/T
72060	은 괴	kg
72070	금 괴	〃
72080	동 봉 및 형 강	M/T
72090	나 동 선	〃
72100	동 판 및 띠	〃
72110	동 관 및 중 공 봉	〃
72120	알 루 미 늄 판	〃
72130	알 루 미 늄 샷 시 바	〃
	38. 조립금속제품 기계 및 장비 제 조 업	
	381. 조립금속제품 제 조 업	
81010	금 속 제 식 탁 용 품	천 개
81020	공 장 용 수 공 구	〃
81030	농 업 용 수 공 구	〃
81040	철 제 캐 비 넷	개
81050	철 제 의 자	〃
81060	철 제 책 상	〃
81070	건 물 용 문 및 수 문	〃
81078	〃	M/T
81080	건 축 용 샷 시	M/T
81090	보 일 러	T/H

품목번호	품 목 명	단 위	품목번호	품 목 명	단 위
81100	열 교 환 기	대	81300	용 접 봉	M/T
81108	〃	M/T	81310	금속가스탱크및용기	kg
81110	금 속 구 조 물	〃		382. 기계제조업	
81120	금속탱크및용기	kℓ	82010	내 연 기 관	HP
81130	통 조 림 용 관	천 개	82020	동 력 탈 곡 기	대
81138	〃	M/T	82030	경 운 기	HP
81140	드 럼 관	개	82040	동 력 분 무 기	대
81150	알루미늄프레스 가공식기및용기	M/T	82050	농 업 용 트 랙 터	HP
81160	주 전 자	개	82060	선 반	대
81170	스텐레스스틸제식기	M/T	82068	〃	M/T
81180	커 피 포 트	개	82070	밀 링 기	대
81190	가 스 레 인 지	〃	82078	〃	M/T
81200	싱 크	〃	82080	프 레 스 기	M/T
81210	방 열 기	매	82090	압 출 기	대
81220	자동차용방열기	개	82098	〃	M/T
81230	석 유 스 토 브	〃	82100	압 연 기	대
81240	볼 트 및 너 트	M/T	82108	〃	M/T
81250	금 속 관 이 음 쇠	〃	82110	금 형	대
81260	철 사 망	〃	82118	〃	백만원
81270	자동차용중판스프링	천 매	82120	로 디	대
81280	체 인	kg	82130	굴 착 기	백만원
81290	병 마 개	천 개	82138	〃	M/T

품목번호	품 목 명	단 위	품목번호	품 목 명	단 위
82140	불 도 저	M/T	82310	지 게 차	대
82150	연 사 기	〃	82320	가 정 용 재 봉 틀	〃
82160	직 기	〃	82330	공 업 용 재 봉 틀	〃
82170	편 직 기	〃	82340	자 동 판 매 기	〃
82180	샷 틀	〃	82350	감 속 기	〃
82190	염 색 기	〃	82358	〃	HP
82200	포 장 기	대	82351	감 속 기 (50HP이상)	대
82208	〃	M/T	82352	감 속 기 (50HP미만)	대
82210	전 자 계 산 기	대	82360	기 어	개
82211	탁 상 용 전 자 계 산 기	〃	82368	〃	M/T
82212	휴 대 용 전 자 계 산 기	〃	82370	베 어 링	kg
82220	금 전 등 록 기	〃	82371	볼 베 어 링	〃
82230	타 자 기	〃	82372	로 울 러 베 어 링	〃
82240	동 력 펌 프	〃	82373	메 탈 베 어 링	〃
82248	〃	HP	82374	기 타 베 어 링	〃
82250	공 기 압 축 기	대	82380	윈도우형에어콘디셔너	대
82260	송 배 풍 기	〃	82390	페케지형에어콘디셔너	R/T
82270	호 이 스 트	〃	82400	상 업 용 냉 장 고	대
82280	크 레 인	M/T	82410	자 동 차 용 피 스 톤	개
82290	엘 리 베 이 터	대	82420	피 스 톤 링	〃
82300	콘 베 이 어	〃	82430	밸 브	〃
82308	〃	M/T	82438	밸 브	M/T

품목번호	품 목 명	단 위	품목번호	품 목 명	단 위
	383. 전기기계,기구 제조업		83170	브라운관	개
83010	직류전동기	HP	83171	흑백브라운관	〃
83020	교류전동기	〃	83172	칼라브라운관	〃
83030	변압기	KVA	83180	다이오드	천 개
83040	배전반	대	83181	겔마니움다이오드	〃
83050	전력회로차단기	〃	83182	실리콘다이오드	〃
83060	흑백T·V수상기	〃	83183	발광다이오드	〃
83070	칼라T·V수상기	〃	83190	집적회로	〃
83080	컴비네이션T·V	〃	83191	반도체집적회로	〃
83090	라디오수신기	〃	83192	고밀도집적회로	〃
83091	일반라디오수신기	〃	83200	트랜지스터	〃
83092	시계라디오수신기	〃	83201	겔마니움트랜지스터	〃
83093	카라디오수신기	〃	83202	실리콘트랜지스터	〃
83100	전축	〃	83210	가변저항기	〃
83110	녹음기	〃	83220	고정저항기	〃
83120	T·V튜너	천 개	83230	가변콘덴서	〃
83130	확성기	〃	83240	고정콘덴서	〃
83140	앰프	대	83250	전자기기용유도자	〃
83150	전화기	〃	83260	녹음테이프	km
83151	유선전화기	〃	83261	카세트테이프	〃
83152	무선전화기	〃	83262	릴테이프	〃
83160	자동식전화교환기	회선	83270	전기냉장고	대

품목번호	품 목 명	단 위
83280	선 풍 기	대
83290	전 기 세 탁 기	〃
83300	전 기 밥 솥	〃
83310	전 기 보 온 밥 통	〃
83320	통신선 및 통신용케이블선	M/T
83330	전력선 및 전력용케이블선	〃
83340	형 광 전 구	천 개
83350	장 식 용 전 구	〃
83360	축 전 지	개
83370	건 전 지	천 개
83378	건 전 지	KV
83380	스 위 치	천 개
83381	거 버 나 이 프 스 위 치	〃
83382	기 타 스 위 치	〃

384. 운수장비제조업

품목번호	품 목 명	단 위
84010	선 박 용 내 연 기 관	HP
84020	철 강 유 조 선	G/T
84030	철 강 화 물 선	〃
84040	특 수 화 물 선	〃
84050	철 강 어 선	〃
84060	디 젤 기 관 차	대
84070	여 객 차	량
84080	철 도 차 량 륜	개
84090	승 용 차	대
84100	대 형 버 스	〃
84110	소 형 버 스	〃
84120	대 형 트 럭	〃
84130	중 형 트 럭	〃
84140	소 형 트 럭	〃
84150	특 수 차	〃
84160	자 동 차 용 내 연 기 관	〃
84170	자 동 차 륜	개
84180	차 축	〃
84190	자 전 거	대
84200	모 터 싸 이 클	〃
84210	콘 테 이 너	〃

385. 전문, 과학측정 제어장비 및 사진광학제품제조업

품목번호	품 목 명	단 위
85010	적 산 전 력 계	대
85020	수 도 미 터	개
85030	광 학 용 렌 즈	〃
85040	안 경 테	천 개
85050	사 신 기	대
85060	사 진 섬 광 기	개

품목번호	품 목 명	단 위
85070	쌍 안 경	개
85080	손 목 시 계	〃
85081	일 반 손 목 시 계	〃
85082	전 자 손 목 시 계	〃
85090	벽 시 계	〃
85091	일 반 벽 시 계	〃
85092	전 자 벽 시 계	〃
85100	탁 상 시 계	〃
85110	손 목 시 계 케 이 스	〃
	39. 기 타 제 조 업	
	390. 기 타 제 조 업	
90010	피 아 노	대
90020	기 타	〃
90030	전 자 현 악 기	〃
90040	대 형 볼	개
90050	야 구 장 갑	〃
90060	낚 시 대	〃
90070	샤 프 연 필	천 개
90080	볼 펜	〃
90090	연 필	천 타
90100	철 제 우 산 및 양 산	천 개
90110	가 발	〃

품목번호	품 목 명	단 위
90120	라 이 타	개
90121	일 회 용 라 이 타	〃
90122	반 영 구 용 라 이 타	〃
90130	인 형	천 개
90140	작 크	km
	4. 전 기 업	
99990	총 발 전 량 및 판 매 량	백 G W

5. 사업체별 통계 Print out

5.1 제품통계표

5.1.1 품목별 사업체별 통계표

품목번호	품 목 명	조사구 및 사업체번호	추출간격	변동사항 (유고)	종업원 규모	(전월) (금월)	생산량 〃	자가 소비량 〃	시판량 〃	수출량 〃	과부족 〃	재고량 〃	전월비 증감율 생산량 수출량	전월비 증감율 시판량 재고량	증감요인 생산	증감요인 출하
01010	무 연 탄	001-01 · · · 소계 (전수) 001-02 · · · 소계 (표본) 소계 (표본) ×추출율 (i 계층의) Total ($\sum_{i=1}^{n} yi$)			(1) (2)	＊ $yi = \sum_{j=1}^{n} fj \times xj$ [i =계층수 j =사업체수 fi =사업체별 추출간격 xj =사업체별 물량 $yi = i$ 층의 소계] ＊ 종업원 규모 (1)=대기업 (2)=중소기업									＊ 부호 1 =계절적 수요증가 2 =계절적 수요감소 3 =수출증가 4 =수출감소 5 =시설증설 6 =시설보수 및 노후 7 =원재료 구입원활 8 =원재료 구입부진 9 =수요대비 0 =기 타	
99990	전 력															

5.1.2 총괄표 (산업별, 품목별 통계표)

(5.1.1)표중 품목별 합계만 Print out

5.2 원재료 통계표

5.2.1 품목별 사업체별 통계표

산업분류 (3 dig)	원재료번호	원재료명	단위	조사구 및 사업체번호	변동사항 (유고)	월중사용량	월말재고량	전월비증감율 (%)	
								사용량	재고량
210	522	갱목	㎥	001-01					
⋮	⋮	⋮	⋮	081-03					
				⋮					
400	907	전력	천KWT	소계					

5.2.2 산업별 품목별 통계표

(5.2.1)표중 원재료별 합계 부분만 Print out

5.3 고용 및 급여 통계표

5.3.1 산업별 사업체별 통계표

산업분류	조사구 및 사업체번호	추출간격	변동사항			월말종업원수	급여총액	특별급여액	월중입식자수	월중이직자수	월중 연근무인원수	월중 연근무시간수	금월	증감율(%)
				생산직	상용고				×	×	×	×	좌동	좌동
					임시및일고				×	×	×	×		
				사무직 및 기타					×	×	×	×		
				지역업주 및 무급가족종사자					×	×	×	×		
				합계										
210	〈전수계층〉 001-01 ⋮ 소계 〈표본계층〉 021-03 ⋮ 소계×i계층의 추출간격 Total ($\sum_{i=1}^{n} yi$)													

5.3.2 산업별 통계표

(5.3.1)표중 산업분류별 (3 dig)이상 부분만 Print out

5.4 생산능력 통계표

품목번호	품목명	조사구 및 사업체번호	변동사항 (유고)	(전월) (금월)	생산능력 〃	생산량 〃	전월비 증감율	
							생산능력	생산량
1101	분유 및 연유	001-01						
⋮	⋮	⋮						
8502	벽시계							

5.5 사업체별 통계 최종확인(산업과)

5.6 Error 발견시 사업체 **M/F**정정(정정방법은 **4.3**, **4.4**과정과 동일)

5.7 사업체별 통계표 작성과정에서 산출된 제품, 원재료, 고용급여, 생산능력의 산업별, 품목별 합계는 제품(**6.1**), 원재료(**6.3**), 고용급여(**6.4**), 생산능력(**6.5**)의 **M/F**에 각각수록

6. 품목별, 산업별 통계 M/F 및 결과표 작성

6.1 제품통계 M/F 및 결과표 작성

6.1.1 M/F 구성원소 및 형식

품목 및 산업분류 (산업분류상)	품목 및 산업분류 (기계집계용)	품목 및 산업분류명	구 분	기준물량(Mi) (or 기준금액, 가격)	가 중 치	년 도	1 월		2 ····· 12
							물 량 (금액·가격)	지 수	·····
〈품목별 산업별 통계〉			대 기 업 생 산						
— 품목별(8 dig)			중 소 기 업 생 산						
— 세세분류(5 dig)			생 산 (대기업+중소기업)						
— 세분류(4 dig)			시 판						
— 소분류(3 dig)			수 출 (시판+수출)						
— 중분류(2 dig)									
— 대분류(1 dig)			재 고						
Total(대분류계)			재 고 율						
〈용도별 특수분류별 지수〉									
— 세세분류별(5dig)									
1) 생산재									
가) 내구생산재									
나) 비내구생산재									
① 원 재 료									
② 건설자재									

품목 및 산업분류 (산업분류상)	품목 및 산업분류 (기계집계용)	품목 및 산업분류명	구 분	기준물량(Mi) (or 기준금액, 가격)	가 중 치	년 도	1 월		2 ····· 12
							물 량 (금액·가격)	지 수	·····
2) 소 비 재									
가) 단 용 재									
나) 반내구소비재									
다) 내 구 소 비 재									
— 세분류 (4 dig)									
⋮									
— 소분류 (3 dig)									
⋮									
— 중분류 (2 dig)									
⋮									
— 대분류 (1 dig)									
⋮									
⋮									
Total (대분류계)									
〈공업구조별 특수분류지수〉									
— 경공업지수									
산업별지수 (3 dig)									
⋮									
소 계									
— 중공업지수									
산업별지수 (3 dig)									
⋮									
소 계									
— Total (경공업+중공업)									

6.1.2 제품별 물량과 가격은 사업체 M/F (1.1) 로부터 (5.7) 에서 계산된 사업체별 합계 또는 평균량 (가격) 을 해당월의 물량란에 각각 수록

[생산 (대기업+중소기업), 대기업생산, 중소기업생산
출하 (시판+수출), 시판, 수출
재고]

6.1.3 가격지수산출 (금액조사품목)

6.1.3.1 불변생산액 산출 (대기업, 중소기업, 시판, 수출, 출하, 재고도 동일함)

6.1.3.1-1 공식

① $$\text{불변생산액} = \frac{\text{경 상 생 산 액}}{\{(3.5\times\text{도매물가지수})+(6.5\times\text{수출물가지수})\}} \times 100$$

..... 의복류

② $$\text{불변생산액} = \frac{\text{경상생산액}}{\text{도매물가지수}}$$ ..... 의약품

6.1.3.1-2 수록방법

(6.1.3.1-1)에서 계산된 불변생산액은 (6.1.2) 에서 수록된 경상생산액과 대체수록

6.1.3.2 가격지수 산출

6.1.3.2-1 공식 $$\text{가격지수} = \frac{\text{t 시점의 불변생산액}}{\text{기준생산가격}} \times 100$$

6.1.3.2-2 수록방법

(6.1.3.2-1)에서 계산된 가격지수는 해당월의 해당품

목 가격지수란에 수록

6.1.4 대기업, 중소기업, 생산, 시판, 수출, 출하, 재고, 재고율지수산출

6.1.4.1 품목별 지수

6.1.4.1-1 공 식

$$I_t = \frac{Q_t}{Q_o} \times 100$$

「주」 I_t : t 시점의 품목지수
Q_o : 기준물량
Q_t : t 시점의 물량

6.1.4.1-2 수록방법

(6.1.4.1-1)에서 계산된 지수는 해당월의 품목별 해당지수란에 수록

6.1.4.2 산업별지수, 특수분류별지수

6.1.4.2-1 공 식

$$I_t = \frac{\Sigma W_i \cdot I_{it}}{\Sigma W_i}$$

「주」 W_i : 품목가중치
I_{it} : t 시점의 i 품목지수

6.1.4.2-2 수록방법

(6.1.4.2-1)에서 계산된 지수는 해당월의 해당산업란에 수록

6.1.4.3 제품재고율지수 산출

6.1.4.3-1 공 식

$$\text{재고율지수} = \frac{\text{재고지수}}{\text{출하지수}} \times 100$$

6.1.4.3-2 수록방법

(6.1.4.3-1)에서 계산된 지수는 해당품목별, 산업별 지수란에 수록

6.1.5 통계표작성 (별첨)

6.2 증감율 및 기여도산출

6.2.1 증감율 (대분류, 중분류, 소분류, 품목)

6.2.1.1 공 식

6.2.1.1-1 비 전 월 $= \dfrac{\text{PM 지수}}{\text{PM-1 지수}} \times 100 - 100$

or $= \dfrac{(\text{PM지수} - (\text{PM-1})\text{지수})}{\text{PM-1 지수}} \times 100$

6.2.1.1-2 비전년동월 $= \dfrac{\text{PMY 지수}}{\text{PMY-1 지수}} \times 100 - 100$

or $= \dfrac{(\text{PMY 지수} - (\text{PMY-1})\text{지수})}{\text{PMY-1 지 수}} \times 100$

6.2.1.1-3 비전년동기 $= \dfrac{\sum_{i=1}^{n} \text{PMY 평균지수}}{\sum_{i=1}^{n} \text{PMY-1 평균지수}} \times 100 - 100$

or $= \dfrac{\sum_{i=1}^{n} \text{PMY평균지수} - (\sum_{i=1}^{n} \text{PMY-1})\text{평균지수}}{\sum_{i=1}^{n} \text{PMY-1 평균지수}} \times 100$

6.2.2.2-3 비전년동기 $= \dfrac{\{\sum_{i=1}^{n} \text{PMY 품목평균지수} - (\sum_{i=1}^{n} \text{PMY-1})\text{품목평균지수}\}}{(\sum_{i=1}^{n} \text{PMY-1 중분류평균지수})}$

$\dfrac{\times \text{ 품목가중치}}{\times \text{ 중분류가중치}} \times 100$

6.2.2.3 품목의 대대기여도

6.2.2.3-1 비 전 월 $= \dfrac{\{\text{PM품목지수} - (\text{PM}-1)\text{품목지수}\} \times \text{품목가중치}}{(\text{PM}-1\text{대분류지수}) \times \text{대분류가중치}}$

$\times 100$

6.2.2.3-2 비전년동월 $= \dfrac{\{\text{PMY 품목지수} - (\text{PMY}-1)\text{ 품목지수}\}}{(\text{PMY}-1\text{ 대분류지수})}$

$\dfrac{\times \text{ 품목가중치}}{\times \text{ 대분류가중치}} \times 100$

6.2.2.3-3 비전년동기 $= \dfrac{\{\sum_{i=1}^{n} \text{PMY품목평균지수} - (\sum_{i=1}^{n} \text{PMY}-1)\text{품목평균지수}\}}{(\sum_{i=1}^{n} \text{PMY}-1\text{ 대분류평균지수})}$

$\dfrac{\times \text{ 품목가중치}}{\times \text{ 대분류가중치}} \times 100$

6.2.2.4 품목의 대총기여도

6.2.2.4-1 비 전 월 $= \dfrac{\{\text{PM품목지수} - (\text{PM}-1)\text{품목지수}\} \times \text{품목가중치}}{(\text{PM}-1\text{총지수}) \times 100}$

6.2.2.4-2 비전년동월 $= \dfrac{\{\text{PMY 품목지수} - (\text{PMY}-1)\text{품목지수}\}}{(\text{PMY}-1\text{ 총지수})}$

$\dfrac{\times \text{ 품목가중치}}{\times 100}$

6.2.1.2 수록방법

(6.2.1.1)에서 계산된 증감율은 해당품목별, 산업별 증감율란에 수록

6.2.2 기여도(증감율에 대한)

6.2.2.1 품목의 대소기여도

6.2.2.1-1 비 전 월 $= \dfrac{\{\text{PM 품목지수} - (\text{PM}-1)\text{품목지수} \times \text{품목가중치}\}}{(\text{PM}-1\text{소분류지수}) \times \text{소분류가중치}}$

$$\times 100$$

6.2.2.1-2 비전년동월 $= \dfrac{\{\text{PMY 품목지수} - (\text{PMY}-1)\text{품목지수}\}}{(\text{PMY}-1\ \text{소분류지수})}$

$$\frac{\times \text{품목가중치}}{\times \text{소분류가중치}} \times 100$$

6.2.2.1-3 비전년동기 $= \dfrac{\{\sum_{i=1}^{n} \text{PMY 품목평균지수} - (\sum_{i=1}^{n} \text{PMY}-1)\text{품목평균지수}\}}{(\sum_{i=1}^{n} \text{PMY}-1\ \text{소분류평균지수})}$

$$\frac{\times \text{품목가중치}}{\times \text{소분류가중치}} \times 100$$

6.2.2.2 품목의 대중기여도

6.2.2.2-1 비 전 월 $= \dfrac{\{\text{PM 품목지수} - (\text{PM}-1)\text{품목지수}\} \times \text{품목가중치}}{(\text{PM}-1\ \text{중분류지수}) \times \text{중분류가중치}}$

$$\times 100$$

6.2.2.2-2 비전년동월 $= \dfrac{\{\text{PMY 품목지수} - (\text{PMY}-1)\ \text{품목지수}\}}{(\text{PMY}-1\ \text{중분류지수})}$

$$\frac{\times \text{품목가중치}}{\times \text{중분류가중치}} \times 100$$

6.2.2.4-3 비전년동기 $= \dfrac{\{\sum_{i=1}^{n} \text{PMY 품목평균지수} - (\sum_{i=1}^{n} \text{PMY}-1)\text{품목평균지수}\}}{(\sum_{i=1}^{n} \text{PMY}-1\ \text{총평균지수})}$

$$\frac{\times \text{품목가중치}}{\times 100}$$

6.2.2.5 소분류의 대중기여도

6.2.2.5-1 비 전 월 $= \dfrac{\{\text{PM 소분류지수} - (\text{PM}-1)\text{소분류지수}\}}{(\text{PM}-1\ \text{중분류지수})}$

$$\frac{\times \text{소분류가중치}}{\times \text{중분류가중치}} \times 100$$

6.2.2.5-2 비전년동월 $= \dfrac{\{\text{PMY 소분류지수} - (\text{PMY}-1)\text{ 소분류지수}\}}{(\text{PMY}-1\text{중분류지수})}$

$$\frac{\times \text{소분류가중치}}{\times \text{중분류가중치}} \times 100$$

6.2.2.5-3 비전년동기 $= \dfrac{\{\sum_{i=1}^{n} \text{PMY 소분류평균지수} - (\sum_{i=1}^{n} \text{PMY}-1)}{(\sum_{i=1}^{n} \text{PMY}-1\text{ 중분류평균지수})}$

$$\frac{\text{소분류평균지수}\} \times \text{소분류가중치}}{\times \text{중분류가중치}} \times 100$$

6.2.2.6 소분류의 대대기여도

6.2.2.6-1 비 전 월 $= \dfrac{\{\text{PM 소분류지수} - (\text{PM}-1)\text{소분류지수}\}}{(\text{PM}-1\text{ 대분류지수})}$

$$\frac{\times \text{소분류가중치}}{\times \text{대분류가중치}} \times 100$$

6.2.2.6-2 비전년동월 $= \dfrac{\{\text{PMY 소분류지수} - (\text{PMY}-1)\text{ 소분류지수}\}}{(\text{PMY}-1\text{ 대분류지수})}$

$$\frac{\times \text{소분류가중치}}{\times \text{대분류가중치}} \times 100$$

6.2.2.6-3 비전년동기 $= \dfrac{\{\sum_{i=1}^{n} \text{PMY 소분류평균지수} - (\sum_{i=1}^{n} \text{PMY}-1)}{(\sum_{i=1}^{n} \text{PMY}-1\text{ 대분류평균지수})}$

$$\frac{\times \text{소분류평균지수}\} \times \text{소분류가중치}}{\times \text{대분류가중치}} \times 100$$

6.2.2.7 소분류의 대총기여도

6.2.2.7-1 비 전 월 $= \dfrac{\{\text{PM 소분류지수} - (\text{PM}-1)\text{소분류지수}\}}{(\text{PM}-1\text{ 총지수})}$

$$\frac{\times \text{ 소분류가중치}}{\times \ 100}$$

6.2.2.7-2 비전년동월 $= \dfrac{\{\text{PMY 소분류지수} - (\text{PMY} - 1)\text{소분류지수}\}}{(\text{PMY} - 1 \text{ 총지수})}$

$$\frac{\times \text{ 소분류가중치}}{\times \ 100}$$

6.2.2.7-3 비전년동기 $= \dfrac{\{\sum_{i=1}^{n} \text{PMY 소분류평균지수} - (\sum_{i=1}^{n} \text{PMY} - 1)}{(\sum_{i=1}^{n} \text{PMY} - 1 \text{ 총평균지수})}$

$$\frac{\text{소분류평균지수}\} \times \text{소분류가중치}}{\times \ 100}$$

6.2.2.8 중분류의 대대기여도

6.2.2.8-1 비 전 월 $= \dfrac{\{\text{PM 중분류지수} - (\text{PM} - 1)\text{중분류지수}\}}{(\text{PM} - 1 \text{ 대분류지수})}$

$$\frac{\times \text{ 중분류가중치}}{\times \text{ 대분류가중치}} \times 100$$

6.2.2.8-2 비전년동월 $= \dfrac{\{\text{PMY 중분류지수} - (\text{PMY} - 1)\text{중분류지수}\}}{(\text{PMY} - 1 \text{ 대분류지수})}$

$$\frac{\times \text{ 중분류가중치}}{\times \text{ 대분류가중치}} \times 100$$

6.2.2.8-3 비전년동기 $= \dfrac{\{\sum_{i=1}^{n} \text{PMY 중분류평균지수} - (\sum_{i=1}^{n} \text{PMY} - 1)}{(\sum_{i=1}^{n} \text{PMY} - 1 \text{ 대분류평균지수})}$

$$\frac{\text{중분류평균지수}\} \times \text{중분류가중치}}{\times \text{ 대분류가중치}} \times 100$$

6.2.2.9 중분류의 대총기여도

6.2.2.9-1 비 전 월 $= \dfrac{\{\text{PM 중분류지수} - (\text{PM} - 1)\text{중분류지수}\}}{(\text{PM} - 1 \text{ 총지수})}$

$$\frac{\times \text{ 중분류가중치}}{\times \ 100}$$

6.2.2.9-2 비전년동월 $= \dfrac{\{\text{PMY 중분류지수} - (\text{PMY} - 1)\text{ 중분류지수}\}}{(\text{PMY} - 1 \text{ 총지수})}$

$$\frac{\times \text{ 중분류가중치}}{\times \ 100}$$

6.2.2.9-3 비전년동기 $= \dfrac{\{\sum_{i=1}^{n} \text{PMY 중분류평균지수} - (\sum_{i=1}^{n} \text{PMY} - 1)}{(\sum_{i=1}^{n} \text{PMY} - 1 \text{ 총평균지수})}$

$$\frac{\text{중분류평균지수}\} \times \text{중분류가중치}}{\times \ 100}$$

6.2.2.10 대분류의 대총기여도

6.2.2.10-1 비 전 월 $= \dfrac{\{\text{PM 대분류지수} - (\text{PM} - 1)\text{ 대분류지수}\}}{(\text{PM} - 1 \text{ 총지수})}$

$$\frac{\times \text{ 대분류가중치}}{\times \ 100}$$

6.2.2.10-2 비전년동월 $= \dfrac{\{\text{PMY 대분류지수} - (\text{PMY} - 1)\text{ 대분류지수}\}}{(\text{PMY} - 1 \text{ 총지수})}$

$$\frac{\times \text{ 대분류가중치}}{\times \ 100}$$

6.2.2.10-3 비전년동기 $= \dfrac{\{\sum_{i=1}^{n} \text{PMY 대분류평균지수} - (\sum_{i=1}^{n} \text{PMY} - 1)}{(\sum_{i=1}^{n} \text{PMY} - 1 \text{ 총평균지수})}$

$$\frac{\text{대분류평균지수}\} \times \text{대분류가중치}}{\times \ 100}$$

6.3 원재료 통계 M/F 및 결과표 작성

6.3.1 M/F 구성요소 및 형식

품목 및 산업분류 (산업분류상)	품목 및 산업분류 (기계집계용)	품목 및 산업분류명	구분1 (Mij)	구 분 2	단 위 (Mij)	기준물량	가 중 치	년 도	1 월		2 ····· 12
									물량 (Mij)	지수 (Mij)	·····
〈산업별, 원재료별통계〉			국 산 소 비	수 입							
소 분 류 (3 dig)			수 입 재 고	의 존 율							
1차원재료 (7 dig)			계 재고율								
품 목 별 (8 dig)											
⋮ ⋮											
2차원재료 (7 dig)											
품목별											
⋮											
연료및전력 (7 dig)											
무 연 탄											
전 력											
⋮											
중 분 류 (2 dig)											
⋮											
대 분 류 (1 dig)											
⋮											
Total (대분류계)											
〈원재료별 산업별통계〉											
원재료품목별 (8dig)											
소분류별 (3 dig)											
⋮											
계 (소분류계)											
원재료품목별 (8dig)											
소분류별 (3 dig)											
⋮											

6.3.2 원재료별 물량은 사업체 M/F(1.5)로부터 (5.7)에서 계산된 사업체별 합계를 해당월의 물량란에 수록

6.3.3 소비, 재고지수산출(국산, 수입, 계의 지수도 동일)

6.3.3.1 품목별 지수

6.3.3.1－1 국산품 또는 수입품 지수

〈공식〉 $I_t = I_{t-1} \times \frac{Q_t}{Q_{t-1}}$

I_t : t 시점의 지수

I_{t-1} : t-1 시점의 지수

Q_t : t 시점의 물량

Q_{t-1} : t-1 시점의 물량

단, $I_1 = \frac{Q_1}{Q_0} \times 100$

6.3.3.1－2 품목계 지수(국산품 + 수입품)

〈공식〉 $I_T = \frac{W_D\ I_D + W_E\ I_E}{W_D + W_E}$

I_T : 품목합계 지수 W_D : 국산품 가중치

I_D : 국산품 지수 W_E : 수입품 가중치

I_E : 수입품 지수

6.3.3.2 산업별, 특수분류별 지수{원재료(소원재료, 제품원재료), 연료 및 동력}

〈공식〉 $I_t = \frac{\sum W_i \cdot I_{it}}{\sum W_i}$

W_i : 품목가중치

I_{it} : A시점의 i 품목지수

6.3.4 원재료 재고율 지수산출

〈공식〉 재고율지수 $= \frac{\text{재고지수}}{\text{소비지수}} \times 100$

6.3.5 수입의존율 지수산출(품목별, 산업별 합계(국산품+수입품)에만 해당)

〈공식〉 수입의존율지수 = $\frac{\text{수입품 소비지수}}{\text{계(국산+수입)소비지수}} \times 100$

6.3.6 통계표작성(별첨참조)

6.4 고용, 급여, 생산성 통계 M/F 및 결과표 작성

6.4.1 M/F구성요소 및 형식

산업분류	산업분류명	구분	1 (mi)	구분2 (mi)	단위	기준용량	가중치	년도	1월		.....
									물량	지수	
		생산종업원	월말종업원수 이직자수 입직자수	평균조업일수							
		사무 및 기타	man-day	산출량지수							
		자영 및 무급	man-hour	투입량지수 (1)							
		계	월중급여액	〃 (2)							
			평균급여액	〃 (3)							
			임금지수	노동생산성지수							

6.4.1 − 1 산업별 노동투입량(월말종업원수 M/D, M/H) 및 급여액은 M/F(1.1)로부터 (5.7)에 계산된 합계를 해당월의 물량란에 수록

6.4.2 입직·이직률산출

〈공식〉 입직률 $= \dfrac{\text{금월입직자수}}{\text{전월말종업원수}} \times 100 (= 00.0)$

이직률 $= \dfrac{\text{금월이직자수}}{\text{전월말종업원수}} \times 100 (= 00.0)$

6.4.3 평균급여액산출

〈공식〉 평균급여액 $= \dfrac{\text{급여액합계}}{\text{월말종업원수}} (= 0000)$

6.4.4 임금지수산출(임금총액 및 특별급여제외로 각각 산출)

〈공식〉 $I_w = \sum W_w \times I_{wi} \left(= \dfrac{W_1}{W_0} \right)$

I_{wi} : 개별임금지수

W_w : 개별지수의 가중치

W_0 : 기준시점의 평균임금

W_1 : 비교시점의 평균임금

6.4.5 평균근로일수산출

〈공식〉 평균근로일수 $= \dfrac{\text{월중연근무인원수}}{\text{월말종업원수}} (= 00.0)$

6.4.6 산출량지수산출

생산지수와 같음.

6.4.7 투입량지수산출

6.4.7 − 1 개별지수

〈공식〉 $\frac{\text{비교시 투입량}}{\text{기준시 투입량}} \times 100(=00.0)$ (월말종업원수, 월중연근무 인원수, 월중연근무시간수)

6.4.7 − 2 종합지수

$$I_t = \frac{\sum W_i I_i}{\sum W_i} \qquad \left(\begin{array}{l} W_i : \text{개별지수의 가중치} \\ I_i : \text{개별지수} \end{array} \right)$$

6.4.8 노동생산성지수산출

6.4.8.1 개별지수

〈공식〉 $\text{개별지수} = \frac{\text{산출량지수}}{\text{투입량지수}} \times 100(=00.0)$

6.4.8.2 종합지수

〈공식〉 $I_t = \frac{\sum W_i I_{ti}}{\sum W_i} \quad (=00.0)$

$$\left(\begin{array}{l} W_i : \text{개별지수의 가중치} \\ I_{ti} : \text{개별생산성지수} \end{array} \right)$$

6.4.9 통계표 작성

(별첨참조)

6.5 생산능력 및 가동율 지수산출

6.5.1 생산능력지수

6.5.1.1 품목별

$$C_{ijk} = \sum_{i=1}^{a} \sum_{j=1}^{b} \sum_{k=1}^{r} \frac{C_{1ijk}}{C_{0ijk}} \times 100$$

6.5.1.2 소분류별

$$C_{ijo} = \sum_{i=1}^{a} \sum_{j=1}^{b} \frac{\left(\sum_{k=1}^{r} W_{ijk} \cdot C_{ijk} \right)}{W_{ijo}}$$

6.5.1.3 중분류별

$$C_{ioo} = \sum_{i=1}^{a} \frac{\left(\sum_{j=1}^{b} W_{ijo} \cdot C_{ijo} \right)}{W_{ioo}}$$

6.5.1.4 총합지수

$$SC = \left(\sum_{i=1}^{a} W_{ioo} \cdot C_{ioo} \right) / 10{,}000$$

* (C : 생산능력 S : 총합
O : 기준년도 i : 중분류
I : 비교년도 j : 소분류
W : 생산능력지수 가중치 k : 품목
M : 생산지수, 가동율, 가동율지수 가중치)

6.5.2 생산능력 생산지수

6.5.2.1 품목별

$$PI_{ijk} = \sum_{i=1}^{a} \sum_{j=1}^{b} \sum_{k=1}^{r} \frac{Q_{1ijk}}{Q_{0ijk}} \times 100$$

6.5.2.2 소분류별

$$PI_{ijo} = \sum_{i=1}^{a} \sum_{j=1}^{b} \frac{\left[\sum_{k=1}^{r} (M_{ijk} \cdot PI_{ijk}) \right]}{M_{ijo}}$$

6.5.2.3 중분류별

$$PI_{ioo} = \sum_{i=1}^{a} \frac{\left(\sum_{i=1}^{b} M_{ijo} \cdot PI_{ijo}\right)}{M_{ioo}}$$

6.5.2.4 총합지수

$$SPI = \left(\sum_{i=1}^{a} M_{ioo} \cdot PI_{ioo}\right) / 10,000$$

$$* \left[\begin{array}{l} PI : \text{생산능력, 생산지수} \\ Q : \text{실생산량} \end{array}\right]$$

6.5.3 가동율

6.5.3.1 품목별

$$O_{ijk} = \sum_{i=1}^{a} \sum_{j=1}^{b} \sum_{k=1}^{r} \frac{Q_{ijk}}{C_{1\,ijk}} \times 100$$

6.5.3.2 소분류별

$$O_{ijo} = \sum_{i=1}^{a} \sum_{j=1}^{b} \frac{\left(\sum_{k=1}^{r} M_{ijk} \cdot O_{ijk}\right)}{M_{ijo}}$$

6.5.3.3 중분류별

$$O_{ioo} = \sum_{i=1}^{a} \frac{\left(\sum_{j=1}^{b} M_{ijo} \cdot O_{ijo}\right)}{M_{ioo}}$$

6.5.3.4 총합

$$SO = \left(\sum_{i=1}^{a} M_{ioo} \cdot O_{ioo}\right) / 10.000$$

$$* \left[\begin{array}{l} O : \text{가동율} \\ OI : \text{가동율지수} \end{array}\right]$$

6.5.4 가동율지수

6.5.4.1 품목별

$$OI_{ijk} = \sum_{i=1}^{a} \sum_{j=1}^{b} \sum_{k=1}^{r} \frac{O_{1\,ijk}}{O_{0\,ijk}} \times 100$$

6.5.4.2 소분류별

$$OI_{ijo} = \sum_{i=1}^{a} \sum_{j=1}^{b} \frac{\left(\sum_{k=1}^{r} M_{ijk} \cdot OI_{ijk} \right)}{M_{ijo}}$$

6.5.4.3 중분류별

$$OI_{ioo} = \sum_{i=1}^{a} \frac{\left(\sum_{j=1}^{b} M_{ijo} \cdot OI_{ijo} \right)}{M_{ioo}}$$

6.5.4.4 총합지수

$$SOI = \left(\sum_{i=1}^{a} M_{ioo} \cdot OI_{ioo} \right) / 10.000$$

6.5.5 통계표 작성

(별첨참조)

＃1.　鑛工業動態調査

製品統計結果表(案)

(1982. 6)

각표의 공통사항

1. 각표의 (A)란은 생산(대기업, 중소기업), 출하(내수, 수출), 재고(재고율)별로 구분하여 Print out
2. 각표의 (No.)란은 결과표 고유번호를 Print out
3. 각표의 Time (연월, 연분기)는 좌측 상단에 Print out
4. 지수, 증감율(%)은 소숫점 첫째자리까지 계산(둘째자리에서 사사오입)
5. 기여도는 소숫점 둘째자리까지 계산(셋째자리에서 사사오입)

用途	結果表名	結果表 原指數 (P) 生産	大企業	中小企業	(S) 出荷	內需	輸出	在庫 (I)	在庫率 (I/S)
	月別(M)								
分析用	品目別 指數(增減率 및 寄與度)統計表	P 1011	P 1012	P 1013	S 1011	S 1012	S 1013	I 1011	I/S 1011
	產業別 前月對比 統計表(增減率 및 寄與度)	P 1021	P 1022	P 1023	S 1021	S 1022	S 1023	I 1021	I/S 1021
	產業別 前年同月對比 統計表(增減率 및 寄與度)	P 1031	P 1032	P 1033	S 1031	S 1032	S 1033	I 1031	I/S 1031
	產業別 前年同期對比 統計表(增減率 및 寄與度)	P 1041	P 1042	P 1043	S 1041	S 1042	S 1043	I 1041	I/S 1041
	製造業 用途別 特殊分類指數統計表(增減率 및 寄與度)	P 1051	P 1052	P 1053	S 1051	S 1052	S 1053	I 1051	I/S 1051
	製造業 構造別 特殊分類指數統計表(增減率 및 寄與度)	P 1061	P 1062	P 1063	S 1061	S 1062	S 1063	I 1061	I/S 1061
報告用	品目別 前月對比表(物量 및 指數)	P 1071	P 1072	P 1073	S 1071	S 1072	S 1073	I 1071	I/S 1071
	品目別 前年同月對比表(物量 및 指數)	P 1081	P 1082	P 1083	S 1081	S 1082	S 1083	I 1081	I/S 1081
	品目別 地年同期對比表(物量 및 指數)	P 1091	P 1092	P 1093	S 1091	S 1092	S 1093	I 1091	I/S 1091
	產業別 指數統計表	P 1101	P 1102	P 1103	S 1101	S 1102	S 1103	I 1101	I/S 1101
	原指數 統計表(指數 및 增減率)	P 1111	P 1112	P 1113	S 1111	S 1112	S 1113	I 1111	I/S 1111
	季節變動調整指數統計表(指數 및 增減率)	–	–	–	–	–	–	–	–
	主要製品 生產,出荷,在庫量表	P 1121	–	–	–	–	–	–	–
	業種 및 品目別 前月比增減表	P 1131	P 1132	P 1133	S 1131	S 1132	S 1133	I 1131	I/S 1131
	業種 및 品目別 前年同月比增減表	P 1141	P 1142	P 1143	S 1142	S 1142	S 1143	I 1141	I/S 1141
	業種 및 品目別 前年同期比增減表	P 1151	P 1152	P 1153	S 1151	S 1152	S 1153	I 1151	I/S 1151

一　覽

固有番號								結果表書式	備考
季節變動調整指數									
(P) 生産	大企業	中小企業	(S) 出荷	內需	輸出	在庫 (I)	在庫率 (I/S)		
-									
-	-	-	-	-	-	-	-	# 1	
P 2021	P 2022	P 2023	S 2021	S 2022	S 2023	I 2021	I/S 2021	# 2	
P 2031	P 2032	P 2033	S 2031	S 2032	S 2033	I 2031	I/S 2031	# 3	
P 2041	P 2042	P 2043	S 2041	S 2042	S 2043	I 2041	I/S 2041	# 4	
P 2051	P 2052	P 2053	S 2051	S 2052	S 2053	I 2051	I/S 2051	# 5	
P 2061	P 2062	P 2063	S 2061	S 2062	S 2063	I 2061	I/S 2061	# 6	
-	-	-	-	-	-	-	-	# 7	
-	-	-	-	-	-	-	-	# 8	
-	-	-	-	-	-	-	-	# 9	
P 2101	P 2102	P 2103	S 2101	S 2102	S 2103	I 2101	I/S 2101	# 10	
-	-	-	-	-	-	-	-	# 11	
P 2111	P 2112	P 2113	S 2111	S 2112	S 2113	I 2111	I/S 2111	# 12	
-	-	-	-	-	-	-	-	# 13	出荷, 在庫, 在庫率은 같은 票에 作成
P 2131	P 2132	P 2133	S 2131	S 2132	S 2133	I 2131	I/S 2131	# 14	季節調査指數는 業種別로 作成
P 2141	P 2142	P 2143	S 2141	S 2142	S 2143	I 2141	I/S 2141	# 14	
P 2151	P 2152	P 2153	S 2151	S 2152	S 2153	I 2151	I/S 2151	# 14	

用途	結果表名	結果表							
		原指数							
		(P) 生産	大企業	中小企業	(S) 出荷	内需	輸出	在庫 (I)	在庫率 (I/S)
報告用	品目別 前月比 増減表	P 1161	P 1162	P 1163	S 1161	S 1162	S 1163	I 1161	I/S 1161
	品目別 前年同月比 増減表	P 1171	P 1172	P 1173	S 1171	S 1172	S 1173	I 1171	I/S 1171
	品目別 前年同期比 増減表	P 1181	P 1182	P 1183	S 1181	S 1182	S 1183	I 1181	I/S 1181
臺帳	月別 品目別 物量臺帳	P 1191	–	–	–	–	–	–	–
	月別 品目別 物量累計臺帳	P 1201	–	–	–	–	–	–	–
	月別 品目別 指數臺帳	P 1211	P 1212	P 1213	S 1211	S 1212	S 1213	I 1211	I/S 1211
	月別 産業別 指數臺帳	P 1221	P 1222	P 1223	S 1221	S 1222	S 1223	I 1221	I/S 1221
	製造業 用途別 指數臺帳	P 1231	P 1232	P 1233	S 1231	S 1232	S 1233	I 1231	I/S 1231
	製造業 構造別 指數臺帳	P 1241	P 1242	P 1243	S 1241	S 1242	S 1243	I 1241	I/S 1241
	分期別 (Q)								
分析用	産業別 前分期 對比表	P 3221	P 3222	P 3223	S 3221	S 3222	S 3223	I 3221	I/S 3221
	産業別 前年同分期 對比表	P 3231	P 3232	P 3233	S 3231	S 3232	S 3233	I 3231	I/S 3231
	分期別 製造業 用途別 特殊分類 指數表	P 3241	P 3242	P 3243	S 3241	S 3242	S 3243	I 3241	I/S 3241
	分期別 製造業 構造別 特殊分類 指數表	P 3251	P 3252	P 3253	S 3251	S 3252	S 3253	I 3251	I/S 3251
報告用	品目別 前分期 對比表	P 3261	P 3262	P 3263	S 3261	S 3262	S 3263	I 3261	I/S 3261
	品目別 前年同分期 對比表	P 3271	P 3272	P 3273	S 3271	S 3272	S 3273	I 3271	I/S 3271

固有番號								結果表書式	備考
季節變動調整指數									
(P) 生産	大企業	中小企業	(S) 出荷	內需	輸出	在庫 (I)	在庫率 (I/S)		
-	-	-	-	-	-	-	-	#15	
-	-	-	-	-	-	-	-	#15	
-	-	-	-	-	-	-	-	#15	
-	-	-	-	-	-	-	-	#16	出荷,在庫量은 같은 票에 作成
-	-	-	-	-	-	-	-	#16	〃
-	-	-	-	-	-	-	-	#17	
P 2221	P 2222	P 2223	S 2221	S 2222	S 2223	I 2221	I/S 2221	#17	
P 2231	P 2232	P 2233	S 2231	S 2232	S 2233	I 2231	I/S 2231	#17	
P 2241	P 2242	P 2243	S 2241	S 2242	S 2243	I 2241	I/S 2241	#17	
P 4221	P 4222	P 4223	S 4221	S 4222	S 4223	I 4221	I/S 4221	# 2	
P 4231	P 4232	P 4233	S 4231	S 4232	S 4233	I 4231	I/S 4231	# 3	
P 4241	P 4242	P 4243	S 4241	S 4242	S 4243	I 4241	I/S 4241	# 5	
P 4251	P 4252	P 4253	S 4251	S 4252	S 4253	I 4251	I/S 4251	# 6	
-	-	-	-	-	-	-	-	# 7	
-	-	-	-	-	-	-	-	# 8	

用途	結果表名	結果表							
		原指數							
		(P) 生産	大企業	中小企業	(S) 出荷	内需	輸出	在庫 (I)	在庫率 (I/S)
報告用	分期別 産業別 指數 統計表	P 3281	P 3282	P 3283	S 3281	S 3282	S 3283	I 3281	I/S 3281
	分期別 季節變動調整指數 統計表	–	–	–	–	–	–	–	–
	分期別 原指數 統計表	P 3291	P 3292	P 3293	S 3291	S 3292	S 3293	I 3291	I/S 3291
臺帳	分期別 品目別 物量 臺帳	P 3301	–	–	–	–	–	–	–
	分期別 品目別 指數 臺帳	P 3311	P 3312	P 3313	S 3311	S 3312	S 3313	I 3311	I/S 3311
	分期別 産業別 指數 臺帳	P 3321	P 3322	P 3323	S 3321	S 3322	S 3323	I 3321	I/S 3321
	分期別 製造業 用途別 指數 臺帳	P 3331	P 3332	P 3333	S 3331	S 3332	S 3333	I 3331	I/S 3331
	分期別 製造業 構造別 指數 臺帳	P 3341	P 3342	P 3343	S 3341	S 3342	S 3343	I 3341	I/S 3341
	年別 (Y)								
臺帳	年度別 品目別 指數 統計表	P 5351	P 5352	P 5353	S 5351	S 5352	S 5353	I 5351	I/S 5351
	年度別 産業別 指數 統計表	P 5361	P 5362	P 5363	S 5361	S 5362	S 5363	I 5361	I/S 5361
	年度別 製造業 用途別 指數 統計表	P 5371	P 5372	P 5373	S 5371	S 5372	S 5373	I 5371	I/S 5371
	年度別 製造業 構造別 指數 統計表	P 5381	P 5382	P 5383	S 5381	S 5382	S 5383	I 5381	I/S 5381
	品目別 生産(出荷)量 統計表	P 5391	P 5392	P 5393	S 5391	S 5392	S 5393	–	–
	品目別 在庫量 統計表	–	–	–	–	–	–	I 5401	–

固有番號								結果表書式	備考
季節變動調整指數									
(P) 生産	大企業	中小企業	(S) 出荷	內需	輸出	在庫 (I)	在庫率 (I/S)		
P 4281	P 4282	P 4283	S 4281	S 4282	S 4283	I 4281	I/S 4281	#10	
P 4291	P 4292	P 4293	S 4291	S 4292	S 4293	I 4291	I/S 4291	#12	
–	–	–	–	–	–	–	–	#11	
–	–	–	–	–	–	–	–	#16	出荷, 在庫量은 같은 票에 作成
–	–	–	–	–	–	–	–	#17	
P 4321	P 4322	P 4323	S 4321	S 4322	S 4323	I 4321	I/S 4321	#17	
P 4331	P 4332	P 4333	S 4331	S 4332	S 4333	I 4331	I/S 4331	#17	
P 4341	P 4342	P 4343	S 4341	S 4342	S 4343	I 4341	I/S 4341	#17	
–	–	–	–	–	–	–	–	#18	
P 6361	P 6362	P 6363	S 6361	S 6362	S 6363	I 6361	I/S 6361	#18	
P 6371	P 6372	P 6373	S 6371	S 6372	S 6373	I 6371	I/S 6371	#18	
P 6381	P 6382	P 6383	S 6381	S 6382	S 6383	I 6381	I/S 6381	#18	
–	–	–	–	–	–	–	–	#19	
–	–	–	–	–	–	–	–	#20	

1

TABLE OF INDEX BY COMMODITIES BY MONTHLY

(A) (NO)

月別 品目別 指數 統計表

MONTH

CODE	SPE	PM-1	PM	INC - DEC (%)			DAE-SO GI-YU-DO			DAE-JUNG GI-YU-DO			DAE-DAE GI-YU-DO			DAE-CHONG GI-YU-DO		
				VJW	VJNDW	VJNDG	VJW	VJNDW	VJNDG	VJW	VJNDW	VJNDG	VJW	VJNDW	VJNDG	VJW	VJNDW	VJNDG
品目番號	特殊分類附號	前月指數	今月指數	增減率 (%)			對小寄與度			對中寄與度			對大寄與度			對總寄與度		
				前月比	前年同月比	前年同期比	前月比	前年同月比	前年同期比	前月比	前年同月比	前年同期比	前月比	前年同月比	前年同期比	前月比	前年同月比	前年同期比

4 TABLE OF CHANGE FROM SAME PERIOD OF PREVIOUS YEAR BY INSUSTRIES

(A) (No.)

월별 산업별 전년동기 대비 통계표

CODE	PY-2 (A)	PY-1 (B)	P (C)	INC-DEC (%)		DAE-JUNG-GI-YU-DO		DAE-DAE GI-YU-DO		DAE-CHONG GI-YU-DO	
				B/A	C/B	B/A	C/B	B/A	C/B	B/A	C/B
산업분류	전전년 동기지수	전년동기 지수	금년동기 지수	증감율 (%)		대중기여도		대대기여도		대총기여도	
				전전년 동기비	전년 동기비	전전년 동기비	전년 동기비	전전년 동기비	전년 동기비	전전년 동기비	전년 동기비
산업총계											
2											
21											
210											
⋮											
410											

5 TABLE OF INDEX BY SPECIAL GROUPS(USER) BY MONTHLY

(A) (No.)

월별 제조업 용도별 특수분류 지수 통계표

SCODE	AVPY-2	AVPY-1	PM-2	PM-1	PM	INC-DEC (%)			DAE-JUNG GI-YU-DO			DAE-DAE GI-YU-DO		
						VJW	VJNDW	VIJNDG	VJW	VIJNDW	VIJNDG	VJW	VIJNDW	VIJNDG
특수 분류번호	전전년 평균지수	전년 평균지수	전전월 지수	전월지수	금월지수	증감율 (%)			대중기여도			대대기여도		
						전월비	전년 동월비	전년 동기비	전월비	전년 동월비	전년 동기비	전월비	전년 동월비	전년 동기비
제조업 총계														
8														
81														
82														
88														
89														
9														
91														
92														
93														
⋮														

「註」 1) 8 =생산재, 81 =내구생산재, 82 =비내구생산재, 88 =원재료, 89 =건설자재, 9 =소비재
91 =단용재, 92 =반내구재, 93 =내구재

6 TABLE OF INDEX BY SPECIAL GROUP(STRUCTURE) BY MONTHLY

(A) (No.)

월별 제조업 특수분류(구조별) 지수

CODE	AVPY-2	AVPY-1	PM-2	PM-1	PM	INC-DEC (%)			DAE-JUNG GI-YU-DO			DAE-DAE GI-YU-DO		
						VJW	VJNDW	VJNDG	VJW	VJNDW	VJNDG	VJW	VJNDW	VJNDG
산업분류	전전년 평균지수	전년 평균지수	전전월 지수	전월지수	금월지수	증감율 (%)			대중기여도			대대기여도		
						전월비	전년 동월비	전년 동기비	전월비	전년 동월비	전년 동기비	전월비	전년 동월비	전년 동기비
제조업 총계														
중공업														
341														
351														
352														
353														
354														
361														
362														
369														
371														
372														
381														
382														
383														
384														
385														
경공업														
311														
313														
314														
321														
322														
323														
324														
331														
332														
342														
355														
356														
390														

TABLE OF CHANGE FROM SAME MONTH OF PREVIOUS YEAR OF INDEX QUANTITY BY COMMONTIES

8

품 목 별 전 년 동 월 대 비 통 계 표

CODE	DIVIS-ION	UNIT	WEIG-HT	QUANTITY			INC-DEC(%)		INDEX			INC-DEC(%)		NOTE
				PY-2PM (A)	PY-1PM (B)	PYPM (C)	B/A	C/B	PY-2PM (a)	PY-1PM (b)	PYPM (c)	b/a	c/b	
분류번호	구 분	단위	가중치	물 량			증감율(%)		지 수			증감율 (%)		비 고
				전전년동월	전년동월	금년동월	B/A	C/B	전전년동월	전년동월	금년동월	b/a	c/b	
총 계	(1)생 산 (2)대기업 (3)중 소 기 업 (4)출 하 (5)내 수 (6)수 출 (7)재 고 (8)재고율													
2														
21														
210														
001														
⋮														
4														

9 TABLE OF CHANGE FROM SAME PERIOD OF PREVIOUS YEAR OF INDE & QUANTITY BY COMMONTIES

품 목 별 전 년 동 기 대 비 표

CODE	DIVI-SION	UNIT	WEI-GHT	QUANTITY			INC-DEC(%)		INDEX			INC-DEC(%)		비 고
				PY-2(1~PM)(A)	PY-1(1~PM)(B)	PY(1~PM)(C)	B/A	C/B	PY-2(1~PM)(a)	PY-1(1~PM)(b)	PY(1~PM)(c)	b/a	c/b	
분 류 번 호	구 분	단위	가중치	물 량			증감율(%)		지 수			증감율(%)		
				PY-2(1~PM)(A)	PY-1(1~PM)(B)	PY(1~PM)(C)	B/A	C/B	PY-2(1~PM)(a)	PY-1(1~PM)(b)	PY(1~PM)(c)	b/a	c/b	
총 계	(1)생 산 (2)대기업 (3)중소기업 (4)출 하 (5)내 수 (6)수 출 (7)재 고 (8)재고율													
2														
21														
210														
001														
⋮														
4														

#11 TABLE OF INDEX BY(B) BY MONTHL (A)

월별원지수통계표(지수 및 증감율)

CODE	WEIGHT	AVPY-2	AVPY-1	PM-2	PM-1	P M	INC-DEC (%)		
							VJW	VJNDW	VJNDG
산 업 분 류	가중치	전전년 평균지수	전 년 평균지수	전전월 지 수	전월지수	금월지수	증 감 율 (%)		
							전 월 비	전년동월비	전년동기비
산 업 총 계									
광 업									
21									
22									
⋮									
전 기 업									

「주」(A) 나에는 Production great sanaee Shipment Shipment Domestic Inventory I/S Ratio

(B) 나에는 Industries, Special groups (user), Special groups (Stru cture)

12 TABLE OF INDEX S·A INDEX BY(B) BY MONTHLY(A)

월 별 계 절 변 동 조 정 지 수

C O D E	WEIGHT	PM - 2	PM - 1	P M	INC - DEC (%)		
					V J W	VJNDW	VJNDG
산 업 분 류	가 중 치	전전월지수	전월지수	금월지수	증 감 율 (%)		
					전 월 비	전년동월비	전년동기비

13 PRODUCTION (SHIPMENT, INVENTORY) OF PRINOPAL PRODUCTS

주요제품생산출하재고실적 (계속)

COMMODITES	UNIT	DIVISION	PM-1	P M	INC - DEC (%)		
					V J W	V J N D W	V J N D G
품 목 명	단 위	구 분	전 월	금 월	증 감 율 (%)		
					전 월 비	전년동월비	전년동기비
무 연 탄	M/T	(1) 생 산 량 (2) 대 기 업 량 (3) 중소기업량 (4) 출 하 량 (5) 내 수 량 (6) 수 출 량 (7) 재 고 량					

14

TABLE OF INC-DEC BY INDUSTRIES BY COMMODITIES
업 종 및 품 목 별 증 감 표

DIVISON	CODE	GI-YU INC-DO DEC		CODE	GI-YU INC-DO DEC		WEIGHT	UNIT	PM-1	PM	NOTE
구 분	산 업 분 류	대 총 기여도	증감율	품 목 명	대 총 기여도	증감율	가중치	단위	물 량 전 월	물 량 금 월	비 고
1) 증가	321	0.3	20.5	화학섬유연사	0.2	11.8					
				면 사	0.1	18.3					
				모 사	0.1	19.2					
				⋮							
	311~312	0.2	30.8	소 세 지	0.2	19.9					
				⋮							
2) 감소											

「주」 1) (A)내서는 생산, 대기업, 중소기업, 출하, 내수, 수출, 재고별로 각각 구분
2) (B)내서는 전월비, 전년동월비, 전년동기비 별로 각각 구분
3) 증가업종과 감소업종을 구분하여 대총기여도가 큰 순서대로 작성하되 기여도가 같을 경우는 산입분류순으로 작성
4) 품목선정은 해당산업, 분류내에서 기여도가 큰순서로 작성하되 기여도가 같을 경우는 품목번호순으로작성
5) 기여도가 ±0.1 이상만 Print

15

TABLE OF INC - DEC BY COMMODITIES
품 목 별 증 감 표

DIVISON	CODE	GI-YU-DO	INC-DEC	WEIGHT	UNIT	PM-1	PM	NOTE
						물	량	
구 분	품 목 명	대총기여도	증 감 율	가 중 치	단위	전 월	금 월	비 고
1) 증 가	화학섬유연사	0.4	17.4					
	면 사	0.2	23.8					
	모 사	0.1	9.4					
	⋮							
2) 감 소	맥 주	-0.3	-13.7					
	방 카 C 유	-0.2	-21.5					
	칼 라 T V	-0.1	- 3.2					
	⋮							

「주」 1) (A)내에서는 생산, 출하, 내수, 수출, 재고별로 각각 구분
2) (B)내에서는 전월비, 전년동월비, 전년동기비 별로 각각 구분
3) 증가품목과 감소품목을 구분하여 대총기여도가 큰 순서로 작성하되 대총기여도가 같을 경우 대증기여도순위도 작성
4) 기여도가 ±0.1이상만 Print

16

QUANTITY LIST BY COMMODITIES BY MONTHLY
월별 품목별 물량 대장

CODE	DIVISON	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	TOTAL	AVE
품목번호	구 분	1	2	3	4	5	6	7	8	9	10	11	12	계	평균
001	(1)생 산 (2)대 기 업 (3)중소기업 (4)출 하 (5)내 수 (6)수 출 (7)재 고														

17

LIST OF (B) BY MONTHLY

월 별 지 수 대 장

CODE	DIVISION	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	AVE
일련번호	구 분	1	2	3	4	5	6	7	8	9	10	11	12	평 균
001	(1)생 산													
⋮	(2)대 기 업													
	(3)중소기업													
	(4)출 하													
	(5)내 수													
	(6)수 출													
	(7)재 고													
	(8)재 고 율													

A) by commodities, by industries, by special groups(user) by special groups(structure)

18 INDEX LIST WITHOUT SEASONAL ADJUSTMENT (A)(B)

년도별 지수 통계표 (업종별, 특수분류별)

YEAR	AVR	1	2	3	4	5	6	7	8	9	10	11	12	YEAR
1960														1960
1961														1961
1963														1962
1964														1963
1965														1964
⋮														1965
														⋮

(A)내에는 Production great production Small production Shipment, Domestic export, Inventory. (= Special Groups)
(B)내에는 업종별 특수번호를 기입

19 QUANTITIES TABLE OF PRODUCTION(or A) BY COMMODITIES (B)

품목별 생산량(또는 대기업, 중소기업, 출하, 내수, 수출)통계표

YEAR	TOTAL	1	2	3	4	5	6
	7	8	9	10	11	12	YEAR

A) 내에는 Great production Small Production, Shipment, Domestic, export 로 구분

B) 는 품목번호

20

QUANTITIES TABLE OF INVENTOKIES BY COMMODITIES(A)

품목별 재고량 통계표

YEAR	1	2	3	4	5	6
7	8	9	10	11	12	YEAR

작성주기	결과표명	결과표							
		고용(L)·급여(S)통계				산출량지수(PI)			투
		물량	전월비	전년동월비	전년동기비	전월비	전년동월비	전년동기비	전월비
월별	〈대 장 용〉								
	소분류별사업체별고용급여대장	-	MLS1	-	-	-	-	-	-
	산업소분류별고용급여대장	MLS2	-	-	-	-	-	-	-
	〈분 석 용〉								
	고용급여산출표	MLS3	-	-	-	-	-	-	-
	명목 및 실질임금대비표	-	-	-	-	-	-	-	-
	명목임금 및 실질임금계절조정지수	-	-	-	-	-	-	-	-
	산업별노동생산성지수대비표	-	-	-	-	MPI10	MPI11	MPI12	MLI 13
	〈보 고 용〉								
	고용 및 급여통계	-	MLS19	-	-	-	-	-	-
	고용동향	-	ML20	-	-	-	-	-	-
	명목임금및실질임금지수(총급여)	-	-	-	-	-	-	-	-
	〃 (특별급여제외)	-	-	-	-	-	-	-	-
	업종별노동생산성지수	-	-	-	-	MPI27	MPI28	MPI 29	MLI 30
주별									
	〈분 석 용〉								
	명목 및 실질임금대비표	-	-	-	-	-	-	-	-
	명목임금및 실질임금계절조정지수	-	-	-	-	-	-	-	-
	산업별노동생산성지수대비표	-	-	-	-	QPI 7	QPI 8	QPI 9	QLI10
	〈보 고 용〉								
	고용동향	-	GL16	-	-	-	-	-	-
	명목임금및실질임금지수(총급여)	-	-	-	-	-	-	-	-
	〃 (특별급여제외)	-	-	-	-	-	-	-	-
	업종별노동생산성지수	-	-	-	-	QPI 23	QPI 24	QPI 25	QLI26

지수 전산처리결과일람표

고유번호								결과표	비고
입량지수(LI)		노동생산성지수(LPI)			임금지수(WI)				
전년동월비	전년동기비	전월비	전년동월비	전년동기비	전월비	전년동월비	전년동기비		
-	-	-	-	-	-	-	-	#1	
-	-	-	-	-	-	-	-	#1	
-	-	-	-	-	-	-	-	#2	잠정, 확정
-	-	-	-	-	MWI4	MWI5	MWI6	#12	〃 (총급여 특별급여제외급여액)
-	-	-	-	-	MWI7	MWI8	MWI9	#2	〃
MLI 14	MLI15	MLPI 16	MLPI 17	MLPI 18	-	-	-	#6	〃
-	-	-	-	-	-	-	-	#2	잠정, 확정
-	-	-	-	-	-	-	-	#2	〃
-	-	-	-	-	MWI21	MWI22	MWI23	#1	잠 정
-	-	-	-	-	MWI24	MWI25	MWI26	#1	〃
MLI31	MLI32	MLPI33	MLPI34	MLPI35	-	-	-	#1	〃
-	-	-	-	-	QWI1	QWI2	QWI3	#8	잠정, 확정 (총급여 특별급여제외급여액)
-	-	-	-	-	QWI4	QWI5	QWI6	#1	잠정, 확정
QLI11	QLI12	QLPI13	QLPI14	QLPI15	-	-	-	#4	잠정, 확정
-	-	-	-	-	-	-	-	#2	잠정, 확정
-	-	-	-	-	QWI17	QWI18	QWI19	#1	잠 정
-	-	-	-	-	QWI20	QWI21	QWI22	#1	〃
QLI27	QLI28	QLPI29	QLPI30	QLPI31	-	-	-	#1	〃

소 분 류 별 사 업 체 별

SAN	JOSAGU	JOB	JAN EMP.	G.ACC	G.SEP	GUM emp.	MAN day	
							PM-1	PM
산업분류	조사구 번호	직 종	전 월 종업원수	금 월 입직자수	금 월 이직자수	금 월 종업원수	월중연근무인원수	
							전 월	금 월
		SAENG						
		SANG						
		IMSI						
		SAMUJ						
		MUGUB						
		TOTAL						
	JANTOT	〃						
	YOBONXX	〃						
	PYOBON	〃						
	PYOCH	〃						
	GTOTAL	〃						

고 용 급 여 대 장

1

MAN Hour		G. Total		G. Special		INC-DEC (%)	
						M. D	G.Total
PM-1	PM	PM-1	PM	PM-1	PM	PM-1	PM
월중연근무시간수		총 급 여		총 특 별 급 여		증 감 율 (%)	
						월중연근무 인 원 수	총 급 여
전 월	금 월	전 월	금 월	전 월	금 월	전 월	금 월

산 업 소 분 류 별 고 용

SAN	Ja. Emp.	G. Acc.	G. Sep.	Gum emp.	Man•Day	
					PM-1	PM
산업 분류	전 월 종업원수	금 월 입직자수	금 월 이직자수	금 월 종업원수	월중 연근무 인원수	
					전 월	금 월

* 각 산업별에 대한 각 종목별로 작성

급 여 대 조 표

#2 -1

Man-Hour		G.Total		G.Special	
PM-1	PM	PM-1	PM	PM-1	PM
월중 연근무 시간수		총 급 여		총 특 별 급 여	
전 월	금 월	전 월	금 월	전 월	금 월

고 용 급 여

SAN	Blue.W.	Mugib	Total Blue.W.	Man·Day	D.W.	Man·Hour
	Total W.	Mugib	Total			
산업	생산직종업원수	무급종사자	생산직종업원수 (무급종사자포함)	월중연근무	조업일수	월중연근무
분류	총 종 업 원 수	무급종사자	총 종 업 원 수 (무급종사자포함)	인 원 수		시 간 수

＊ 잠정, 확정 각 1부작성

산 출 표

#3

H.W.	G.Total	Average Total	G.Special	Average Special	G.Contrac-tual	Average Contra-ctual
1 일 근무시간	총 급 여	1 인 평 균 총급여	총특별 급여	1 인 평 균 특별급여	총특별급여 제외총급여	특별급여 제외 1인 평균급여

고 용 및

SAN	Employees		D.W.		Average Total		Average Contractual	
	PM1-11	PM1	PM-1	PM*	PM-1	PM	PM÷1	PM
산업 분류	종 업 원 수		조 업 일 수		1인 평균 총급여		총특별급여 제외 1인 평균 급여	
	전 월	금 월	전 월	금 월	전 월	금 월	전 월	금 월

* 잠정, 확정 각각 작성

급 여 통 계

#4

Average Special		H.W.		INC·DEC (%)				
PM-1	PM	PM-1	PM	Emp	D.W.	Aver. Total	Aver. Cont.	H.W.
1인 평균 특별급여		1일 근무시간		증 감 율 (%)				
전 월	금 월	전 월	금 월	종업원수	조업일수	1인평균 총 급 여	특별 급여 제외평균급여	근무시간

SAN	JAN. Dong. Emp.	J.Acc.	J.Sep.	Jan.Emp.	G.Acc.	G.Sep.	Gum.Emp.
산업 분류	전년동월 종업원수	전 월 입직자수	전 월 이직자수	전 월 종업원수	금 월 입직자수	금 월 이직자수	금 월 종업원수

＊잠정, 확정 각각 작성 및 분기별 작성(전분기, 전년동분기)

동　　향

5

INC·DEC		Acc·INC·DEC(%)			Sep·INC·DEC(%)			Emp·INC·DEC(%)		
PM-1	PM-2	PM-1	PM-2	PM-3	PM-1	PM-2	PM-3	PM-1	PM-2	PM-3
증감인원		입직증감율			이직증감율			종업원 증감율		
전 월	전년동월	전 월	전년동월	전년동기	전 월	전년동월	전년동기	전 월	전년동월	전년동기

명 목 및 실 질

SAN	Weight	Nomiral Wage				Real	
		PM-1		PM		PM-1	
		Wage	Index	Wage	Index	Wage	Index
산업 분류	가 중 치	명 목 임 금				실 질	
		전 월		금 월		전 월	
		금 액	지 수	금 액	지 수	금 액	지 수

* 1. 전년동월, 전년동기비도 같은 방법으로 작성
2. 확정, 잠정 각각 작성
3. 총급여, 특별급여 제외급여로 각각 작성(실질임금도 동일)
4. 분기별로도 작성(전분기, 전년동분기)

임 금 대 비 표 (전 월 / 전년동월 / 전년동기)

6

Wage		Nomiral Wage			Real Wage		
PM		INC·DEC	Gi-Yu Yuel		INC·DEC	Gi-Yu Yuel	
Wage	Index	(%)	Dae Dae	Dae Chang	(%)	Dae Dae	Dae Chang
임 금		명 목 임 금			실 질 임 금		
금 월		증감율 (%)	기 여 율 (%)		증감율(%)	기 여 도	
금 액	지 수		대 대	대 총		대 대	대 총

SAN	Weight	Noniral Wage						
		PM-1		PM		Inc. Dec (%)		
		Wage	Index	Wage	Index	PM-1	PM-2	PM-3
산업분류	가 중 치	명 목 임 금						
		전 월		금 월		증 감 율 (%)		
		금 액	지 수	금 액	지 수	전월비	전년동월비	전년동기비

* 월별, 분기별(잠정치만 작성)

* 총급여액, 특별급여제외 급여로 각각 작성

임 금 지 수

#7

Real Wage							Note
PM-1		PM		Inc.Dec (%)			
Wage	Index	Wage	Index	PM-1	PM-2	PM-3	
실 질 임 금							비 고
전 월		금 월		증 감 율 (%)			
금 액	지 수	금 액	지 수	전월비	전년동월비	전년동기비	

SAN	Weight	Nomiral wage Index					
		PY	PM-1	PM	INC · DEC (%)		
					PM - 1	PM - 2	PM - 3
산업분류	가 중 치	명 목 임 금 지 수					
		전 년	전 월	금 월	증 감 율 (%)		
					전월비	전년동월비	전년동기비

※ 1. 월별, 분기별 작성 (잠정, 확정)
2. 총임금에 대해서만 작성

임 금 계 정 조 정 지 수

8

	Real wage Index						
Gi-Yu YUEL / Dae·Chong	PY	PM - 1	PM	INC · DEC (%)			Gi-Yu YUEL / Dae·Chong
				PM - 1	PM - 2	PM - 3	
	실 질 임 금 지 수						
기여도 / 대 총	전 년	전 월	금 월	증 감 율 (%)			기여도 / 대 총
				전월비	전년동월비	전년동기비	

산 업 별 (세세분류별 / 소 분 류 별)

CODE	OUT - PUT INDEX					IN - PUT IN		
	WEIGHT	INDEX		INC-DEC (%)	GI-YU YUEL	WEIGHT	INDEX	
		PM - 1	PM				PM - 1	PM
산업분류	산 출 량 지 수					투 입 량		
	가중치	지 수		증감율 (%)	기여율	가중치	지 수	
		전 월	금 월				전 월	금 월

※ 1. 각 지수별 기여율은 대총기여율로 작성
2. 잠정·확정별로 각각 작성
3. 분기별로도 작성 (전분기, 전년동분기)

지 수 (전 월 / 전년동월 / 전년동기) 대 비 표

9

DEX		LABOR PRODUTIVE INDEX					NOTE
INC-DEC (%)	GI-YU YUEL	WEIGHT	INDEX		INC-DEC (%)	GI-YU YUEL	
			PM-1	PM			
지 수		생 산 성 지 수					비 고
증감율 (%)	기여율	가중치	지 수		증감율 (%)	기여율	
			전 월	금 월			

업 종 별 광 공 업

SAN	LABOR PRODUTIVE INDEX							OUT -	
	WEIGHT	PY	PM - 1	PM	INC - DEC (%)			PM - 1	PM
					PM-1	PM-2	PM-3		
산업분류	노 동 생 산 성 지 수							산 출	
	가중치	전 년	전 월	금 월	증 감 율 (%)			전 년	금 월
					전월비	전 년 동월비	전 년 동기비		

※ 월별, 분기별로 작성

노 동 생 산 성 지 수

10

PUT INDEX			IN - PUT INDEX					
INC - DEC (%)			WEIGHT	PM - 1	PM	INC - DEC (%)		
PM-1	PM-2	PM-3				PM-1	PM-2	PM-3
량 지 수			투 입 량 지 수					
증 감 율 (%)			가중치	전 월	금 월	증 감 율 (%)		
전월비	전 년 동월비	전 년 동기비				전월비	전 년 동월비	전 년 동기비

작성주기	결 과 표 명	결 과 표				
		원				
		생 산 능 력 (C)			가 동 율	
		전 월	전년동월	전년동기	전 월	전년동월
월별	〈 분 석 용 〉					
(M)	월별품목별 물량전월비 증감율표	C1	C2	C3	O1	O2
	〃 지수통계표 (증감율 및 기여도)	C11	C12	C13	O11	O12
	월별산업별 지수통계표(〃)	C21	C22	C23	O21	O22
	월별산업별 전년동월지수통계표 (〃)	C31	C32	C33	O31	O32
	월별산업별 전년동기지수통계표 (〃)	C41	C42	C43	O41	O42
	〈 보 고 용 〉					
	월별 품목별 지수표	C51	C52	C53	O51	O52
	품목별 대비표 (지수, 증감율 기여도)	C61	C62	C63	O61	O62
	〈 대 장 〉					
	품목별 월별물량대장	C81	C82	C83	O81	O82

통 계 결 과 일 람 표

고 유 번 호							결과표 양식	비 고
지 수								
(O)	가 동 율 지 수 (OI)			생 산 량 (PQ)				
전년동기	전 월	전년동월	전년동기	전 월	전년동월	전년동기		
O3	OI1	OI2	OI3	PQ1	PQ2	PQ3	#1	
O13	OI11	OI12	OI13	PQ11	PQ12	PQ13	#2	
O23	OI21	OI22	OI23	PQ21	PQ22	PQ23	#3	
O33	OI31	OI32	OI33	PQ31	PQ32	PQ33	#4	
O43	OI41	OI42	OI43	PQ41	PQ42	PQ43	#5	
O53	OI51	OI52	OI53	PQ51	PQ52	PQ53	#6	
O63	OI61	OI62	OI63	PQ61	PQ62	PQ63	#7	
O83	OI81	OI82	OI83	PQ81	PQ82	PQ83	#9	

작성주기	결 과 표 명	결 과 표				
		원				
		생 산 능 력 (C)			가 동 율	
		전 월	전년동월	전년동기	전 월	전년동월
월별	〈 분 석 용 〉					
(M)	월별품목별 물량전월비 증감율표	C 501	C502	C503	O501	O 502
	월별품목별 지수통계표 (증감율 및 기여도)	C 511	C512	C513	O511	O 512
	월별산업별 지수통계표 (〃)	C 521	C522	C523	O521	O 522
	월별산업별 전년동월지수 (〃)	C 531	C532	C533	O531	O 532
	월별산업별 전년동기지수 (〃)	C 541	C542	C543	O541	O 542
	〈 보 고 용 〉					
	월별품목별 지수표	C 551	C552	C553	O551	O 552
	품목별 대비표 (지수,증감율,기여도)	C 561	C562	C563	O561	O 562
	월별계절변동조정지수 (증감율)	C 571	C572	C573	O571	O 572

통 계 결 과 일 람 표

고 유 번 호							결과표 양식	비 고
지 수								
(O)	가 동 율 지 수 (OI)			생 산 량 (PQ)				
전년동기	전 월	전년동월	전년동기	전 월	전년동월	전년동기		
O503	OI501	OI502	OI503	PQ501	PQ502	PQ503	# 1	
O513	OI511	OI512	OI513	PQ511	PQ512	PQ513	# 2	
O523	OI521	OI522	OI523	PQ521	PQ522	PQ523	# 3	
O533	OI531	OI532	OI533	PQ531	PQ532	PQ533	# 4	
O543	OI541	OI542	OI543	PQ541	PQ542	PQ543	# 5	
O553	OI551	OI552	OI553	PQ551	PQ552	PQ553	# 6	
O563	OI561	OI562	OI563	PQ561	PQ562	PQ563	# 7	
O573	OI571	OI572	OI573	PQ571	PQ572	PQ573	# 8	

1 TABLE OF QUANTITY BY
월 별 품 목

CODE	JO - SA	UNIT	CAP - QUN		INC-DEC (%)	PRO -
			PM-1	PM		PM-1
품목번호	조사구 및 사업체 No	단 위	능 력 물 량		증 감 율 (%)	생 산
			전 월	금 월		전 월
1122100						

COMMODITIE BY MONTHLY

별 물 량

QUN	INC-DEC (%)	OPERATION		INC-DEC (%)	NOTE
PM		PM-1	PM		
물 량	증 감 율 (%)	가 동 율 (%)		증 감 율 (%)	비 고
금 월		전 월	금 월		

#2 TABLE OF INDEX BY COMMODITIES

(NO) 월 별 품 목 별 지 수

CODE	DIVISION	WEIGHT	INDEX		INC
			PM-1	PM	VJU
품목번호	구 분	가 중 치	지 수		증
			전 월	금 월	전월비
3	(1) 생산능력지수				
31	(2) 생 산 지 수				
311	(3) 가동률지수				
31122100					
⋮					
312					
31213111					
⋮					
313					
31311000					
⋮					
32					
321					
32111111					
⋮					
⋮					
8442111					

BY MONTHLY

(생산능력, 생산가동률)

-DEC (%)		GI-YU-DO			NOTE
VJNDW	VJNDG	VJW	VJNDW	VJNDG	
감률 (%)		기여도			비고
전년동월비	전년동기비	전월비	전년동월비	전년동기비	

$$* \quad \text{기여도} = \frac{(\text{금월지수} - \text{전월지수}) \times W}{\text{전월 확정지수} \times 100}$$

(제조업통계)

#3 IABLE OF INDEX BY

(NO) 월별 산업별 지수

CODE	DIVISION	WEIGHT	INDEX			INC
			PM-2	PM-1	PM	VJW
산업분류	구 분	가 중 치	지		수	증
			전 전 월	전 월	금 월	전 월 비
3	(1) 생산능력					
31	(2) 생 산					
311	(3) 가 동 률					
313	(4) 가 동 률 지 수					
32						
321						
323						
⋮						
38						
382						
383						
384						

INDUSTRIES BY MONTHLY

(생산능력, 생산, 가동률, 가동률지수)

- D E C (%)			GI - YU - DO			
VJNDW	VJJNDW	VJNDG	V J W	VJNDW	VJJNDW	VJNDG
감 률 (%)			기 여 도			
전년동월비	전전년동월비	전년동기비	전 월 비	전년동기비	전전년동월비	전년동기비

#4 TABLE OF INDEX CHANGE FROM SAME MONTH

(NO) 월별 산업별 전년

CODE	DIVISION	WEIGHT	INDEX		
			PMY - 2	PMY - 1	PM
산업분류	구 분	가 중 치	지 수		
			전전년동월	전년동월	금 월
3					
31					
311					
⋮					
32					
321					
323					
⋮					
38					
382					
383					
384					

OF PREVIOUS YEAR BY INDUSTRIES

동월지수(능력, 생산, 가동률)

INC - DEC (%)		GI - YU - DO		NOTE
VJJNDW	VJNDW	VJJNDW	VJNDW	
증 감 률 (%)		기 여 도		비 고
전전년동월비	전년동월비	전전년동월비	전년동월비	

(NO) 월별 산업별 전년동기비수

CODE	DIVISION	WEIGHT	INDEX	
			JJNDG	JNDG
산업분류	구 분	가 중 치	지 수	
			전전년동기	전년동기
3				
31				
311				
⋮				
32				
321				
⋮				
38				
381				
382				
383				
384				

(능 력 , 생 산 , 가 동 률)

	INC-DEC (%)		GI - YU - DO	
	VJJNDG	VJNDG	VJJNDG	VJNDG
	증 감 률 (%)		기 여 도	
금 년 동 기	전전년동기비	전년동기비	전전년동기비	전년동기비

#6 TABLE OF INDEX BY

(NO) 월별 품목별 지수표

품목번호	구 분	1	2	3	4
3	(1) 생산능력				
31	(2) 생 산				
311	(3) 가 동 률				
3122100	(4) 품목별 월별 증감률표				
⋮	(5) 〃 〃 동월비표				
	(6) 〃 〃 동기비표				
	(7) 소분류별 월별 지 수 표				
	(8) 〃 월별 증 감 률표				
	(9) 〃 월별 동 월 비 표				
	(10) 〃 월별 동 기 비 표				
	(11) 〃 월차 평균지수표				

COMMODITIES BY MONTHLY

(능력, 생산, 가동률)

5	6	7	8	9	10	11	12	평 균

#7 TABLE OF CHANGE

(NO) 품 목 별 대 비 표

CODE	DIVISION	CODE	UNIT	WEIGHT	QUANTITY	
					PM - 1	PM
품목번호	구 분	품 목 명	단위	가 중 치	물 량	
					전 월	금 월
31122100						
⋮						
38442111						

BY COMMODITIES

(능 력 , 생 산)

I N D E X		I N C - D E C (%)			GI - YU - DO		
PM - 1	PM	VJW	VDW	VDG	PM - 1	DW	DG
지 수		증 감 률 (%)			기 여 도		
전 월	금 월	전월비	동월비	동기비	전 월	동 월	동 기

#8 TABLE OF INDEX S·A INDEX

(NO) 월별 계절변동

CODE	WEIGHT	PM - 2	PM - 1
산업분류	가 중 치	전전월지수	전 월 지 수
3			
311			
313			
⋮			
32			
321			
⋮			
38			
381			
⋮			

BY(B) BY MONTHLY(A)

조 정 지 수

PM	INC - DEC (%)		
	VJW	VJNDW	VJNDG
금월지수	증감률 (%)		
	전월비	전년동월비	전년동기비

9 TABLE OF QUANTITY BY
品 目 別 月 別 物 量

품목번호	1	2	3	4	5	6
31122100						
38442111						

MONTHLY BY COMMODITIES
臺　帳　（生産能力，生産）

7	8	9	10	11	12	計

Ⅵ. 생 산 지 수 보 정

산업생산지수보정에 관한 건

회의일시 : 1983. 9.14　　15:00

장　　소 : 당국　소회의실

참가범위 : 기준과, 조사과, 유통과, 분석과, 처리과　실무자

	취 급 자	처 리 관	전산담당	행정담당	과　　장

1. 문제점

1) 지수보정에 따른 기준설정 미확실

2) 신규품목의 가중치를 적용할 경우 해당 산업의 Weight 의 결정문제

3) 82 년 동계의 부가가치를 이용하게 됨으로 사실상 개편임.

(기준년 80 년 → 82 년이 됨)

* 기개편 작업소요일 약 6 개월, 보정작업 약 5 개월 예상

2. 대 안

1) 개편한 시기가 얼마되지 않으므로 신규품목 또는 고급화제품이 있을지라도 년평균 작업에만 반영함이 타당함.

(별지 제 2호서식)

협 조 문			응신기일 198 . . .
분류기호 및 문서번호	통산 394-212	제목	산업생산지수보정에 관한 회의
수 신	자료처리과장	발신일자 : 1983. 9.13.	

당과에서는 신개발품목 및 급성장품목을 지수에 반영, 보다 현실적인 산업활동동향을 파악하기 위한 지수보정작업을 진행하고 있는 바, 보정방법 및 적용 타당성을 토의하고자 다음과 같이 실무자 회의를 개최하오니 참석하여 주시기 바랍니다.

다 음

1. 회 의 일 시 : 1983. 9.14 15:00
2. 장 소 : 소회의실
3. 참석대상자

과 별	참 석 자
기 준 과	한성찬 사무관
조사관리과	이정대 사무관, 계훈방 사무관
유통통계과	이기섭 사무관
통계분석과	권오술 사무관
자료처리과	김응구 처리관

첨부 : 회의자료 1부 끝.

	취 급 자	처 리 관	전산담당	행정담당	과 장

1. 산업생산지수의 의의

생산지수란 광공업생산의 시간적 변화를 표현해 주고 있는 지표이며 생산의 개념을 광공업에서 이루어진 작업량이라고 규정하고 그 수량적 표현으로서 생산의 결과인 산출량을 대치시키고 있는 것이다.

위에서 이루어진 작업량이란 측정치를 생산지수의 경우에는 생산활동에 의하여 첨가된 가치액으로 나타나며 이를 부가가치액이라 칭한다.

즉, 제품의 생산량을 기초계열로 하여 개별지수를 산출하고 이들 개별지수를 하나로 묶어서 평균화한 것이 산업생산지수이다.

2. 지수보정의 필요성

가. 라스파이레스 산식은 산업구조와 가격 및 품목의 급격한 변화없이 안정적일 경우에는 문제가 없지만 이러한 조건이 급격히 변할 경우에는 지수가 현실과 다르게 나타나게 되므로 기준년도 개편작업을 하여야 하나 지수개편작업은 큰 작업이므로 간단하게 지수자체를 부분적으로 보정하는 방법을 이용하는 것이다.

나. 즉, '80년 기준년도 개편시에 채택되지 않은 품목중 급성장한 품목과 개편후 새로 개발된 품목중 급성장한 품목들을 현재 편제하고 있는 생산지수에 반영되지 않고 있으므로, 이러한 품목들을 현행 지수에 부분적으로 반영하여 현실적인 산업활동을 나타내는 지수를 편제하기 위함이다.

다. 지수보정을 하면 증감율은 기존지수와 큰 차이는 없으나, 지수수준 자체는 높아져 절대수준 즉, 경제규모의 확대를 나타낸다.

라. 또한 지수보정을 하면 증감율 있어 총지수에는 큰 영향을 주지 않지만 특정한 업종에서는 큰 영향을 주게 되므로 산업구조의 변화가 반영된다.

3. 품목에 대한 자료수집

(생산액 : 백만원)

품목명	단위	'80년 연간	'81년 연간	'82년 연간	'82. 1~4월	'83. 1~4월	증감율	현대상품목중 비교되는 품목. ( )는 가중치	비 고
전자레인지	대	-	16,993	43,751	7,218	30,524	322.9	핸드백 (15.7), 살충제 (17.2), 등유 (19.3), 아연도강판 (15.2), 공장용수공구 (16.5)	전파를 이용 유전가열방식으로 조리하는 기기로서 내수 및 수출이 더욱 증가될 전망임.
소형컴퓨터	대	6,315	24,949	34,671	2,911	13,513	364.2	해산물통조림 (13.5), 신문용지 (14.7), 앰프 (13.6), 전화기 (13.2), 지퍼 (13.4)	전자계산기산업을 포함한 컴퓨터를 의미한다.
전자계측기	대	5,717	8,237	16,810	3,494	7,236	107.1	알미늄괴 (6.1), 커피포트 (6.0), 전축 (6.1), 철도	전기의 흐름 (전류, 전압)을 측정하는

품목명	단위	'80년 연간	'81년 연간	'82년 연간	'82. 1~4월	'83. 1~4월	증감율	현대상품목중 비교되는 품목. ( )는 가중치	비 고
								차량륜 (6.0)	기기이다
전 자 복사기	대	3,343	3,949	8,853	459	7,144	1,456.4	포르말린 (3.5), 고무호스 (3.6), 자동판매기 (3.1), 전기보온밥통 (4.2), 철강유조선 (3.8)	사무기기로서 전기가 열로서 복사하는 기기
V.T.R	대	1,140	1,504	25,628	1,436	24,852	1,630.6	전기세탁기(11.1), 황산 (10.1), 기어 (9.8)	녹화 및 재싱기로서 재영할 수 있는 기기
특 수 운동화	천켤레	3,343	7,149	20,000	3,334	4,105	23.0	방카유 (169.5), 시멘트 (177.8)	운동화에 가죽 및 트랙을 부착시켜 고급화시킨 제품

4. 지수보정방법

가. 신규 고급제품의 수치를 유사제품에 환산 합산 통합하는 방법 특수운동화 (월드컵, 나이키 등)를 추가조사, 기존 운동화 물량에 합산하여 지수를 보정하는 방법

나. 전혀 생산되지 않던 품목이 새로 개발되어 급격히 증가하는 품목

은 부가가치를 산정하여 지수에 보정하는 방법

(V.T.R, 전자계측기, 전자복사기, 소형컴퓨터, 전자레인지)

5. 지수보정방법(실례)

가. 전혀 생산되지 않던 품목이 새로 개발되어 급격히 증가하는 품목을 지수에 보정하는 방법

(V.T.R, 마이크로컴퓨터, 전자복사기, 전자계측기, 전자레인지)

예) 1) '80년도 기준 전기기기 제조업의 부가가치액은 795,568백만원이고

2) '82년 전기기기 제조업의 연평균 지수는 128.8이므로

3) '82년 전기기기 제조업의 부가가치액은 795,568 × 122.7하면 976,162백만원이 된다.

4) 그런데, 지수보정품목의 '82년도 부가가치액은 48,513백만원이므로

(단위 : 백만원)

	V.T.R	전자레인지	마이크로컴퓨터	전자복사기	전자계측기	계
'82년 년간 부가가치액 ('80년 불변)	9,585	16,363	12,967	3,311	6,287	48,513

주) : 전자진흥회 및 당과 직원의 자료수집에 의한 잠정 생산액과 '82 광공업 통계조사의 전기기기 제조업의 잠정 부가가치율을 이용.

5) 전기기기 제조업의 '82년 연간 조정 부가가치액은 976,162 + 48,513 = 1,024,675백만원이 된다.

6) 그러므로 조정한 '82년 전기기기제조업 평균지수는 1,024,675 ÷ 795,568 = 128.8이 된다.

7) 여기서, 전기기기 제조업의 가중치 환산은

	가중치	지 수	적산(積算)
전기기기제조업	759.5	122.7	759.5×122.7=93,191
보 정 품 목	-	100.0	- -
합 계	759.5	128.8	759.5×128.8= 97,824

가 된다.

그러므로 보정품목의 가중치는 97,824 − 93,191 = 4,633이므로 4,633÷100=46.3이 된다.

8) 46.3은 5개 보정품목 가중치의 합계이므로 품목별로 계산하면 다음과 같다.

품 목	V.T.R	전자 레인지	마이크로 컴퓨터	전자 복사기	전자 계측기	계
가 중 치	9.1	15.6	12.4	3.2	6.0	46.3

9) 전기기기 제조업의 기존 품목의 가중치는

46.3 + 759.5 = 805.8

759.5÷805.8=0.94254로 하여 전기기기 제조업의 품목별 가중치('80년 기준)에 0.94254를 각각 곱하여 가중치를 조정한다.

10) 보정품목의 '82년 평균지수가 100.0이므로, '82년 월평균 물량이 기준물량이 되며, 기준물량은 대상사업체(모집단 별첨)에서 직접 수집하여 산정하고, 월별로 물량을 조사하여 비교물량으로 한다.

나. 신규 고급제품의 수치를 유사제품에 환산 합산 통합하는 방법

특수운동화(월드컵, 나이키 등)

예)

(단위 : 천켤레)

품 목 명	'80년 물량	'82년 물량	단 가	가 중 치	'82년 개별지수
운 동 화	157,700	182,100	1,200	148.3	114.3
특수운동화	-	18,000	4,000		
운 동 화 (지수보정후)	157,700	241,500		148.3	153.1

1) 단가의 비율을 4,000 : -1,200 = 3.3 하여 구한다.

2) 이 비율을 적용하여 특수운동화의 물량을 운동화 물량으로 환산한 후 합산한다.

18,000 × 3.3 = 59,400

182,100 + 59,400 = 241,500

3) 특수운동화 합산물량을 포함한 수치로 지수를 계산한다.

241,500 ÷ 157,700 = 153.1 가 된다.

다. 특수운동화와 신규개발품목(5개품목)을 반영한 '82년 평균지수를 계산하면 다음과 같다.

	기존지수	보정후지수	차 이
총 지 수	118.1	119.2	1.1
제 조 업	119.2	120.4	1.2
화학, 석유, 고무제조업 (35)	105.9	108.7	2.8
고무제품제조업 (355)	97.7	118.6	20.9
조립금속제조업 (38)	136.3	138.6	2.3
전기기기제조업 (383)	122.7	128.8	6.1

6. 월별 지수보정방법

가. 현 황

(생산액 : 백만원)

	'82.1～4	'83.1～4	전년동기비(%)
V.T.R	1,436	24,852	1,630.6
마이크로컴퓨터	2,911	13,513	364.2
전자계측기	3,494	7,236	107.1
전자복사기	459	7,144	1,456.4
전자레인지	7,218	30,524	322.9
계	15,518	83,269	436.9

위표에서 보는바와 같이 107.1%－1,630.6%를 증가한 것으로서 보정품목들은 '82년말부터 급성장하였음을 알 수 있다.

'82년(기준년)의 생산액을 기초로 부가가치율을 적용, 부가가치액을 추정하여 가중치를 산정하면 46.3(1/10,000)이 되어 가중치 자체는 크지 않지만 총지수는 가중치×개별지수로 하여 적산되는 것이므로 지수가 급격히 증가하면 총지수에 영향을 주게 된다. 즉, 보정품목이 '83년에 급격히 증가한 것은 개별지수의 증가를 의미하며, 이는 곧 총지수에 커다란 영향을 주게 된다.

나. '83년 3월 지수보정(예)

'83년 3월에 지수보정품목(VTR 등)이 평균금액(1～4월)만큼 생산하였다고 가정하고 전기기기 제조업 지수를 보정하면 아래와 같다.

1) 지수보정품목(VTR 등)의 '82년간 생산액이 129,713백만원이므로 월평균 생산액은 10,809백만원이 된다.

2) '83년 3월 보정품목의 생산액이 20,817백만원이므로 지수보정품목의 '83년 3월지수는 192.6이 된다(20,817÷10,809)

3) 기존 '83년 3월 전기기기 제조업지수는 140.5이므로 이 지수를 보정하면

품 목	지 수	가 중 치	적 산
기 존 품 목	140.5	(759.5-46.3) 713.2	100,205
보 정 품 목	192.6	46.3	8,197
계	143.7	759.5	109,122

4) 특수운동화의 물량을 환산 합산하여 지수보정하면

	기준물량	83.3 물량	단 가	가 중 치	83.3지수
운 동 화	13,141	17,560	1,200	148.3	133.6
특 수 운 동 화		2,771	4,000		
운동화 (보정후)		26,704		148.3	203.2

5) '83.3월 보정지수는 다음과 같다.

	기존지수	보정후지수	차 이
총 지 수	130.8	133.4	2.6
제 조 업	132.7	135.6	2.9
화학, 석유, 고무제조업 (35)	118.6	123.6	4.5
고 무 제 품 제 조 업 (355)	114.2	152.0	37.8
조립금속기계제조업 (38)	154.2	155.4	1.2
전 기 기 기 제 조 업 (383)	140.5	143.7	3.2

7. 자료처리과 협조사항

가. '82년 연간 지수시산

품목별 생산, 내수, 수출, 재고물량의 전산처리, 가중치 조정 및 '83년 월별, 품목별, 업종별 지수를 시산하여 수록.

8. 조사관리과 협조사항

가. 대상 사업체 선정 월별조사실시

나. 조사지침 사무소에 시달

9. 지수보정방법결정을 위한 자료수집(추가분)

가. 부가가치액 산정 (품목별)

'82년 연초, 연말 재고액, 출하액, 생산비를 조사하여 품목 부가가치율을 유도하여 부가가치액을 산정한다.

나. 기준물량산정

모집단 '82년 월평균 물량을 산정하기 위하여 사업체에서 생산하는 품목의 월별 물량조사

10. 기성고 조사품목(공정진척율에 의한 조사품목)

가. 필요성

현행 지수품목중 금속탱크, 전동기 등 일부 장기 (공정) 생산품목이 "완성고" 기준으로 조사하면 지수의 증감폭이 심하므로, 이를 "기성고" 기준으로 조사하여 지수의 정도를 제고하고자 함.

나. 자료의 비교

품 목 명	구 분	생 산			출 하			재 고		
		1	2	3	1	2	3	1	2	3
전동기 (HP)	완성고	29	65	65	244	46	28	151	170	207
	기성고	37	40	40	30	34	30	7	13	23
금속탱크및 용기 (천kℓ)	완성고	36	0	0	36	0	6	0	0	0
	기성고	10	11	0	10	11	0	0	0	0

주) : 전동기는 효성중공업, 금속탱크 및 용기는 삼성중공업(주) 창원.

다. 조사대상품목 : 전동기, 금속탱크 및 용기, 변압기, 열교환기

라. 조사방법 :

2개월 이상의 장기 공정을 요하는 제품에 대하여 공정 진척율을 기준으로 생산 및 출하량을 산정조사한다.

*예 : 금월생산(출하)량

$$= \text{생산예정량(수주량)} \times \frac{(\text{금월까지의 공정율} - \text{전월까지의 공정율})}{100}$$

마. 조사관리과 협조사항

해당 품목에 대한 조사지침서 보완 및 변경조사지침을 지방사무소에 시달하고 조사 확인

11. 금액병행조사

가. 필요성

품질규격에 따라 가격의 차이가 심한 품목을 물량과 금액으로 조사하여 물량조사의 단점을 보완하고자 함.

나. 자료의 비교

		'83.1		2		3		4	
			증감율(%)		증감율(%)		증감율(%)		증감율(%)
선반	물량(대)	32	-	32	0.0	43	34.4	20	-53.5
	금액(백만원)	200	-	221	10.5	209	5.4	139	-33.5
밸브	물량(천개)	88	-	46	-47.7	86	87.0	94	9.3
	금액(백만원)	395	-	519	31.4	634	22.2	654	3.2

주) : 선반-대구중공업, 밸브-범한금속공업(주)

다. 대상품목

선반, 밸브, 압출기, 밀링기

라. 조사방법

지정품목에 대하여 물량과 금액으로 조사

마. 한국은행의 도매물가 및 수출물가지수를 이용, '80년 불변가격으로 환산하고, 물량과 불변금액을 월별로 추세를 비교 파악하며, 해당 품목의 연간 출하액(광공업 통계조사)과 연간물량(광공업동태)으로 단가를 산정하여 조사된 물량을 조정할 계획임('80년, '81년, '82년, '83년 단가를 비교 검토하여 기준설정)

바. 조사관리과 협조사항

품목에 대한 조사지침서를 보완하여 지방사무소에 통보

사. 자료처리과 협조사항

해당품목에 대한 금액을 매월 집계하여 제표

12. 추가품목

가. 선정시기 : '82년 광공업 통계조사결과를 이용하여 선정
('83년 12월경)

나. 품목번호를 부여한 후 조사하여 주요품목동향, 차기 지수개편시 및 지수보정에 활용함.

13. 일 정 표

가. 추가자료수집

1) 사업체별, 품목별로 부가가치 산출자료 수집 9.5~9.12

2) 기준물량 ('82년) 및 '83년 월별물량자료수립 9.5～9.17

나. 품목별 부가가치 산정 9.19～10. 4

다. 기준물량산정 9.19～10. 4

라. '83년 월별 지수시산 10.10～10.29

마. 보정지수 작성 (잠정) : 9월분 지수부터 '83.12월분까지

바. 보정지수로 공표 : '84.1월분부터 ('84.2.20일경)

사. 기성고 중심조사실시 : 9월분부터 조사실시

아. 금액병행조사실시 : 9월분부터 조사실시

(별지 제 2 호서식)

협 조 문	응신기일 198 . . .
분류기호 및 문서번호 통산 394-11	제목 : 지수보정에 관한 협조
수 신 자료처리과장	발신일자 1984. 1. 11.

지수보정작업의 일환으로 지수시산에 관한 전산처리를 다음과 같이 의뢰하오니 적극 협조하여 주시기 바랍니다.

다 음

1. 신개발품목 : 소형컴퓨터, VTR, 전자레인지, 전자복사기, 전자계측기 (5 개품목)

가. 기준년도 : 1982 년으로 고정

나. 기준물량 : 품목별생산, 출하, 재고물량을 화일에 편제 (별첨 5 - 1)

다. 가 중 치 : 전기기기제조업 (383) 분류내의 품목별 가중치 (별첨 4 - 1)

라. 산 식 : 라스파이레스산식 이용

마. 지수시산기간 : '82. 1 ~ '83. 12

바. 지수보정방법 (생산지수)

1) 보정품목

가) '82 년의 월평균 물량을 기준으로 품목별 지수작성 (별첨 5 - 1 참조)

나) 품목별 지수로 월간 부가가치를 유도 (별첨 2 - 1 참조)

월간 부가가치 $= \frac{\text{품목지수}}{100} \times$ '82년 월평균 VA 운영실심사천공

2) 부가가치 유도

가) 5개 보정품목의 월간 부가가치를 합한다.

나) 383분류의 해당 월 부가가치와 가) 항 부가가치를 합한다.

3) 보정지수시산 : $\frac{\text{2) 의 나) 항의 부가가치} \times 100}{\text{383분류 '80년 월평균 부가가치}}$

사. 지수보정 실례

(금액 : 백만원)

품 목 명	82년 월평균VA ①	'82 기준물량 ②	'83.7월			'83.8월		
			물량 ③	지수 ④	VA ⑤	물량	지수	VA
V T R	1,027	4,132	13,860	335.4	3,445	11,957	289.4	2,972
전자레인지	2,107	32	98	306.3	6,454	104	325.0	6,848
컴퓨터및주변기기	1,263	3,238	4,379	135.2	1,708	21,136	652.8	8,245
전자계측기	645	1,657	2,166	130.7	843	2,808	169.5	1,093
전자복사기	333	593	605	102.0	340	603	101.9	339
소 계	5,375				⑥ 12,790			19,497

1) ①항, ②항, ③항은 수집된 자료

2) 지수산정 : $\frac{\text{③항 (7월 물량)}}{\text{②항 (82년 기준물량)}} \times 100$

3) 부가가치 : 지수×①항 ('82년 월평균 VA) ÷ 100

4) 그러므로 전체보정품목의 7월 부가가치는 ⑥항 (12,790백만원)

7 월 보 정	8 월 보 정
가) 80년 383소 분류의 부가가치 8,443억원이므로 월평균 VA는 70,358백만원	가) 좌 동
나) 383기존 소분류의 지수는 170.7	나) 383기존 소분류의 지수는 175.2
∴ 기존 7월의 383 VA = 170.7 × 70,358 ÷ 100 = 120,101 백만원	∴ 기존 8월의 383 VA = 175.2 × 70,358 ÷ 100 = 123,267 백만원
다) 전체보정품목의 7월 VA = 12,790 백만원	다) 전체보정품목의 8월 VA = 19,497 백만원
라) 보정품목의 VA를 합한 '83.7월의 383소분류 VA = 12,790 + 120,101 = 132,891 백만원 그러므로 '83.7월 지수는 132,891 ÷ 70,358 × 100 = 188.9	라) 보정품목의 VA를 합한 '83.8월의 383소분류 VA = 19,497 + 123,267 = 142,764 백만원 그러므로 '83.8월의 지수는 142,764 ÷ 70,358 × 100 = 202.9

상술한 방법으로 생산지수를 '82년 1월부터 '83년 12월까지 월별 조정 (별첨 2 - 1, 2 - 2)

아. 지수보정사항

1) 출하, 내수, 수출, 재고 (별첨 2 - 1, 2 - 2)를 '82.1 ~ '83.12

월까지 보정

2) 제조업 특수분류별 생산, 출하, 재고(별첨 3 - 1)를 '82.1~'83.12까지 보정

자. 기여도 산출

1) 383소 분류의 대총기여도 산출

$$\text{대총기여도} = \frac{\text{금월지수} - \text{전월지수}}{\text{총 지 수}} \times \frac{\text{가중치}}{10,000}$$

2) 품목별 대소기여도 산출

$$\text{대소기여도} = \frac{\text{금월품목지수} - \text{금월품목지수}}{\text{전월 383 지수}} \times$$

3) 품목별 대총기여도

$$\text{품목별 대총기여도} \times \frac{\text{소분류가중치}}{10,000 \times \text{전월총지수}}$$

2. 급성장 고급화품목 : 특수운동화(나이키, 월드컵 등)

가. '80년 이후 생산된 물량을 단가로 환산하여 기존운동화품목에 반영(별첨 6 - 1)

나. 기존운동화와 세분하여 화일 편제

다. 보정방법

(천켤레)

품목번호	품 목 명	단 가	조정전생산물량	조정후생산물량
55081	운 동 화	1,300	100	100
55082	특수운동화	3,900	30	90
55080	운 동 화		130	190

그러므로 조정후 생산물량 190 (천켤레) 물량을 운동화품목의 지수로 작성하여 지수보정

라. 지수시산기간 : 80 년 1 월 이후

3. 연간 물량 보정 (이충학)

사업체누락, 중복 및 응답자, 조작착오 등으로 기조사된 품목별 물량을 수정해야 하는 필요성이 제기되었는 바, 이에 대한 수정물량표를 별첨과 같이 송부합니다 (별첨 7 - 1).

첨 부 : 별첨 1 ~ 별첨 7 1 부. 끝.

산업통계과장 엄 준 용

산 업 통 계 과 장

지 정 품 목 번 호 표 (별첨 1 - 1)

품 목 번 호	품 목 명	단 위
83390	VTR	대
83400	전자레인지	대
83410 (83418)	컴퓨터 및 그 주변기기	금액 (대)
83420 (83428)	전자계측기	금액 (대)
83430 (83438)	전자복사기	대 (금액병행)
55082	특수운동화	천 켤 레

'82년 품목별 생산, 출하, 재고 월평균 금액 (별첨 2 - 1)

(금액 : 백만원)

품 목 명	82년월별평균부가가치 (생산)	출하액 E	내수금액 F	수 출 G	재 고 H
V T R	1,027	2,356	2,356	0	774
전 자 레 인 지	2,107	2,542	718	4,824	485
컴퓨터및주변기기	1,263	3,083	500	2,583	2,503
전 자 계 측 기	645	1,684	245	1,439	1,268
전 자 복 사 기	333	862	794	68	500
계	5,375	10,527	4,613	8,914	5,530

'80 년 383 소분류의 생산, 출하, 재고 월평균 금액 (별첨 2-2)

(금액 : 백만원)

	VA (A)	출 하			재 고
			내 수	수 출	
383 소분류 금 액	70,358	212,746	102,952	109,794	292,730

특수분류별 생산, 출하, 재고지수 추계시 분류번호 및
'80 년 월평균 금액 (별첨 3 - 1)

〈 내구소비재 〉

품 목 명	분 류 번 호
V T R	81
전 자 레 인 지	81
컴퓨터 및 주변기기	81
전 자 계 측 기	32
전 자 복 사 기	32

〈 내구생산재 〉

	V A	출 하	재 고
내 구 생 산 재	69,737	239,797	335,639
내 구 소 비 재	54,517	170,324	239,242

가 중 치 조 정 (별첨 4 - 1)

품목번호	품 목 명	가 중 치							
		생 산	생산액	대기업	중소기업	출 하	내 수	수 출	재 고
83010	직류전동기	6.6	5.1	3.2	3.4	5.7	5.3	0.4	10.5
83020	교류전동기	12.6	9.8	3.1	9.5	10.4	10.0	0.4	36.8
83030	변 압 기	24.9	20.3	13.0	11.9	21.0	16.5	3.5	33.6
83040	배 전 압	13.6	11.6	4.7	8.9	12.8	10.2	2.6	5.8
83050	전력회로 차단기	11.5	9.8	6.5	5.0	11.3	11.2	0.1	10.6
83060	흑 백	53.3	59.3	53.3	0	63.1	15.3	47.8	37.7
83070	칼 라	41.3	45.8	41.3	0	53.1	22.5	30.6	21.8
83080	컴비네이션	2.8	3.2	2.0	0.8	2.7	0.1	2.6	7.7
83090	라 디 오	17.5	19.6	10.4	7.1	19.7	1.5	18.2	30.8
83100	전 축	5.6	6.2	5.4	0.2	7.7	4.2	3.5	35.1
83110	녹 음 기	49.9	55.9	43.6	6.3	58.4	14.0	44.4	49.2
83120	튜 너	13.4	14.9	13.2	0.2	15.7	2.6	13.1	11.3
83130	확 성 기	36.1	40.3	12.2	23.9	13.4	12.8	0.6	15.1
83140	앰 프	12.5	14.0	12.4	0.1	14.7	3.0	11.7	14.1
83150	전 화 기	12.2	8.3	9.9	2.3	8.0	7.2	1.2	38.3
83160	자동식전화 교 환 기	44.8	30.3	43.8	1.0	40.7	38.8	1.9	-
83170	브라운관	28.0	23.6	19.9	8.1	18.9	3.0	15.9	7.2
83180	마이오드	4.4	3.8	3.2	1.2	3.4	0.2	3.2	11.3
83190	집적회로	47.1	39.5	46.2	0.9	39.4	1.8	37.6	24.0
83200	트랜지스터	23.2	19.6	23.2	0	20.5	2.6	17.9	8.2
83210	가변저항기	4.0	3.5	2.9	1.1	3.5	3.1	0.4	8.9

품목번호	품 목 명	가 중 치							
		생 산	생산액	대기업	중소기업	출 하	내 수	수 출	재 고
83220	고정저항기	6.9	5.9	4.0	2.9	6.9	1.3	5.6	8.2
83230	가변콘덴서	6.1	5.2	4.7	1.4	5.5	1.1	4.4	8.2
83240	고정콘덴서	21.8	18.4	15.5	6.3	19.9	6.9	13.0	32.2
83250	전자기기용 유 도 차	28.6	24.1	17.4	11.2	25.7	4.6	21.1	24.8
83260	녹음테이프	14.9	13.3	8.0	6.9	14.5	5.5	9.0	12.1
83270	전기냉장고	44.9	30.2	44.9	0	31.7	27.0	4.7	100.1
83280	선 풍 기	2.0	6.4	1.7	0.3	9.8	8.9	0.9	58.9
83290	전기세탁기	10.2	7.4	10.2	0	7.9	7.3	0.6	11.2
83300	전 기 밥 솥	3.8	4.7	1.3	2.5	5.5	3.7	1.8	6.1
83310	전 기 보온밥통	3.7	4.6	0.6	3.1	4.5	4.5	-	7.0
83320	통신선 및 케 블 선	38.6	37.1	33.5	5.1	40.5	38.6	1.9	23.8
83330	전력선 및 케 블 선	25.9	25.1	16.6	9.03	27.3	22.5	4.8	30.6
83340	형 광 전 구	4.0	3.0	2.6	1.4	3.2	3.0	0.2	4.2
83350	장식용전구	4.6	3.5	1.3	3.3	3.6	0.1	3.5	3.0
83360	축 전 지	10.8	10.6	10.1	0.7	11.1	6.3	4.8	12.5
83370	건 전 지	6.5	6.2	6.5	0	7.0	6.1	0.9	18.1
83380	스 위 치	3.7	3.0	1.8	1.9	3.3	1.6	1.7	3.8
83390	V T R	10.9	1.6	10.9	0	5.4	5.4	-	0.8
83400	전자레인지	22.5	3.2	22.5	0	12.8	1.7	11.1	0.9
83410	컴퓨터 및 주변기기	13.5	2.0	10.9	2.6	7.1	1.2	5.9	6.7
83420	전자계측기	6.8	1.0	4.9	1.9	3.8	0.6	3.2	1.5
83430	전자복사기	3.5	0.5	3.5	0	2.0	1.8	0.2	1.4

기준물량표 (별첨 5－1)

품 목 번 호	83390	품 목 명	VTR	

구 분	생 산	출 하	내 수	수 출	재 고
가 중 치	10.9	5.4	5.4	0	0.8
'82. 1	657		1,006		12
2	1,594		585		1,021
3	1,875		2,491		405
4	2,352		2,056		701
5	3,042		3,253		490
6	3,111		3,029		572
7	3,640		3,733		479
8	4,611		4,616		474
9	6,218		5,691		1,001
10	6,034		5,166		1,869
11	5,611		4,500		2,980
12	10,837		13,324		493
'83. 1	8,885		9,108		270
2	9,300		8,899		671
3	9,947		9,294		1,324
4	13,342		13,294		1,372
5	15,175		12,850		3,697
6	14,457		11,011		7,143
7	13,860		10,101		10,902
8	11,957		10,788		12,072

품목번호	83400	품목명	전자레인지	

구 분	생 산	출 하	내 수	수 출	재 고
가 중 치	22.6	12.8	1.7	11.1	0.9
'82. 1	14	16	1	15	1
2	29	29	8	21	2
3	27	26	2	24	2
4	35	32	2	30	5
5	32	33	1	32	4
6	33	32	2	30	5
7	35	34	6	28	6
8	37	39	2	37	4
9	38	37	4	33	5
10	39	39	6	33	5
11	34	35	3	32	4
12	33	33	9	24	4
'83. 1	50	43	8	35	11
2	41	40	4	36	12
3	46	46	9	37	12
4	69	70	11	59	11
5	77	75	14	61	13
6	99	107	32	75	5
7	98	90	12	78	13
8	104	97	4	93	20

품목번호	83410	품목명	컴퓨터및주변기기	

구 분	생 산	출 하	내 수	수 출	재 고
가 중 치	13.5	7.1	1.2	5.9	6.7
'82. 1	1,762	2,460	8	2,452	1,121
2	3,238	3,911	11	3,900	448
3	2,971	1,910	92	1,818	1,509
4	2,247	2,101	75	2,026	1,655
5	3,471	2,964	554	2,410	2,162
6	3,666	2,840	390	2,450	2,988
7	2,049	2,093	567	1,526	2,944
8	1,634	1,522	538	984	3,056
9	3,359	2,774	509	2,265	3,641
10	3,123	3,274	1,029	2,245	3,490
11	4,649	4,797	671	4,126	3,342
12	6,688	6,351	1,555	4,796	3,679
'83. 1	4,219	3,304	2,150	1,154	4,594
2	4,597	4,712	2,846	1,866	4,479
3	4,517	4,631	2,586	2,045	4,365
4	4,988	4,498	2,585	1,913	4,855
5	4,741	4,526	2,690	1,836	5,070
6	6,284	5,165	3,213	1,952	5,189
7	4,379	4,855	2,886	1,969	4,713
8	21,136	18,804	7,374	11,430	7,045

품목번호	83420	품목명	전자계측기	

구 분	생 산	출 하	내 수	수 출	재 고
가 중 치	6.8	ϕ 3.8	0.6	3.2	1.5
'82. 1	988	795	202	593	1,092
2	1,342	1,057	98	959	1,377
3	1,662	1,395	145	1,250	1,644
4	1,419	1,505	146	1,359	1,558
5	1,462	1,540	131	1,409	1,480
6	1,942	1,856	358	1,498	1,566
7	1,770	1,691	370	1,321	1,645
8	1,610	1,993	190	1,803	1,262
9	2,331	2,520	262	2,258	1,073
10	1,522	1,797	306	1,491	798
11	1,906	1,795	405	1,390	909
12	1,930	2,023	334	1,689	816
'83. 1	1,596	1,336	222	1,114	1,076
2	2,074	2,173	322	1,851	977
3	2,296	2,180	254	1,926	1,093
4	2,337	2,149	425	1,724	1,281
5	1,778	1,936	230	1,706	1,123
6	2,436	2,184	362	1,822	1,375
7	2,166	2,176	418	1,758	1,365
8	2,808	3,122	540	2,582	1,051

품목번호	83430	품목명	전자복사기	

구 분	생 산	출 하	내 수	수 출	재 고
가 중 치	3.5	2.0	1.8	0.2	1.4
'82. 1	473	510	370	140	535
2	482	481	390	91	536
3	509	564	507	57	481
4	644	637	637	-	488
5	781	575	535	40	694
6	560	638	548	90	616
7	518	525	475	50	609
8	586	608	523	85	587
9	708	679	634	45	616
10	662	617	592	25	661
11	540	727	687	40	474
12	651	626	616	10	499
'83. 1	742	579	565	14	662
2	698	744	744	-	616
3	858	916	831	85	558
4	909	843	738	105	624
5	836	670	600	70	790
6	809	905	825	80	694
7	605	696	636	60	603
8	603	754	709	45	452

(별첨 6-1)

기 준 물 량 표

품목번호	55080	품목명	특수운동화	

구 분	생 산	출 하	내 수	수 출	재 고
가 중 치					
'82. 1	19,403	18,753	1,997	16,756	17,153
2	19,382	20,343	2,397	17,946	16,138
3	22,097	22,095	2,812	19,283	16,098
4	23,653	23,573	3,024	20,549	16,183
5	23,889	24,327	3,131	21,196	15,709
6	23,351	22,566	2,599	19,967	16,492
7	20,768	21,740	2,671	19,069	15,494
8	19,765	20,655	2,837	17,818	14,588
9	23,186	22,535	4,081	18,454	15,223
10	22,048	22,028	2,330	19,698	15,399
11	24,145	23,674	2,152	21,522	15,896
12	26,876	25,225	2,180	23,045	17,663
'83. 1	23,995	23,475	2,619	20,856	20,328
2	20,393	20,871	2,839	18,032	19,946
3	24,216	24,100	3,054	21,046	20,032
4	23,862	24,179	3,615	20,564	19,677
5	23,463	23,243	3,170	20,073	19,855
6	22,002	22,665	2,928	19,737	19,130
7	22,781	23,176	3,133	20,043	18,673
8	22,614	20,340	3,465	16,875	19,136
기 준 물 량					

지 수 보 정 작 업

1. 보정소급기간 : '82.1 ~ 83.12

2. 보정품목 : 소분류 "383" 38 개

특수품목 5 개

※ 품질고급화 품목인 경우 보통가격으로 환산하여 물량확대

3. 자료입력 : 83.1 ~ 9 월까지 완료

82.1 ~ 12 월과 83.9 ~ 12 월은 소급조사후 산업과에서 가중치 산정관계로 계류중임.

가) 지수보정방법 (생산지수)

1) 보정품목

(가) 82 년의 월평균 물량을 기준으로 품목별 지수작성 (별첨 5 - 1 참조)

(나) 품목별 지수로 월간 부가가치를 유도 (별첨 2-1 참조)

$$\text{월간부가가치} = \frac{\text{품목지수}}{100} \times 82\text{년 월평균 VA 부가가치}$$

나) ④ 항지수 = 물량 ÷ 기준물량

13,860 ÷ 4,132 = 335.4

⑤ 항 VA = 82 년 월평균 VA × 지수 ÷ 100

7 월보정 (별첨 2-2 참조)

다) 지수보정사항

라) 기여도 산출

1) 383 소분류의 대총기여도 산출

$$\text{대총기여도} = \frac{\text{금월지수} - \text{전월지수}}{\text{품목별지수} \times \text{총지수}} \times \frac{\text{가중치}}{10,000}$$

별첨 2-1

VTR 82년 월별평균 부가가치(생산)

1,027 × 3,011

별첨 2-2

VA(부가가치)

383 소분류금액 70,358

별첨 2-1 5개품목 합한것

재고 292,730 → 보정된 자료

별첨 3-1

8 → 내구생산재 3 → 내구소비재

55080(운동화)

구 분	생 산 (A)	생산액 (B)	대기업 (C)	중소기업 (D)	출 하 (E)	내 수 (F)	수 출 (G)	재 고 (H)
물 량	19,255	19,255	19,107	489	17,565	3,367	14,198	12,920

1. 보정 file만 고칠 것(SM, SS는 대상 외)
2. 80년 기준물량만을 수정하고
3. 지수산출은 82년 1월이후부터 현재까지 고쳐진 기준물량을 이용한다.
4. 운동화 등 품목부터 총지수까지 제표

특수운동화 (보정후물량)

구분 / 월별	(A) 생 산	(C) 대기업	(D) 중소기업	(E) 출 하	(F) 내 수	(G) 수 출	(H) 재 고
'83. 9	25,588	23,068	2,520	21,481	4,827	16,654	20,982
10	32,885	29,647	3,238	31,462	5,165	26,297	22,405
11	39,859	23,950	2,616	37,753	5,322	32,431	24,511
12	39,180	35,952	3,228	38,552	3,353	35,199	26,430
'84. 1	39,669	35,763	3,906	43,938	3,783	40,155	28,359
2	40,173	36,218	3,955	33,190	4,606	28,584	35,342
3	41,656	38,462	3,194	36,970	3,645	33,325	40,028
4	42,154	36,367	5,787	40,286	4,811	35,475	41,717

특 수 운 동 화 보 정 물 량

구분 / 월별	(A) 생 산	(C) 대기업	(D) 중소기업	(E) 출 하	(F) 내 수	(G) 수 출	(H) 재 고
'83. 1	30,678	29,405	1,273	30,090	2,968	27,122	19,251
2	27,764	26,611	1,153	29,347	2,871	26,476	17,668
3	34,033	32,521	1,512	33,034	3,054	29,980	18,667
4	35,679	34,261	1,418	31,578	3,615	27,963	22,768
5	35,863	34,478	1,385	35,444	4,641	30,803	23,187
6	32,867	31,503	1,364	32,279	4,291	27,988	23,775
7	34,845	33,399	1,446	35,151	4,592	30,559	23,469
8	34,282	32,859	1,423	35,432	5,254	30,178	23,136
9	39,287	37,757	1,530	30,535	7,320	23,215	31,888
10	38,443	36,847	1,596	34,186	7,929	26,257	36,145
11	27,383	26,658	725	37,730	8,279	29,451	25,798
12	37,363	35,812	1,551	37,069	5,046	32,023	26,092

지정품목번호표

5.10 접수

품목번호	품 목 명	단 위
82440(82448)	컴퓨터 및 그 주변기기	금액(백만원), (대)
83390	V·T·R	대
83400	전자레인지	대
85120(85128)	〃 계측기	금액(백만원), (대)
85130(85138)	〃 복사기	대, (금액)
55082	특수운동화	천켤레

'82 품목별 생산, 출하, 재고월평균금액

품목번호	VA (A)	생산액 (B)	대기업 (C)	중소기업 (D)	출하 (E)	내수 (F)	수출 (G)	재고 (H)
82440	1,263	3,238	758	505	3,083	512	2,571	2,503
83390	1,027	2,635	1,027	-	2,356	2,356	-	774
83400	2,107	5,396	2,107	-	5,542	718	4,824	485
85120	645	1,657	454	191	1,664	98	1,566	1,268
85130	333	851	333	-	862	794	68	500

보 정 특 수 분 류 번 호

품목번호	품 목 명	구 분	CODE	
82440	컴 퓨 터	3	810	내 구 생 산 재
83390	V T R	8	910	내 구 소 비 재
83400	전자레인지	8	910	〃
85120	전자계측기	3	810	내 구 생 산 재
85130	전자복사기	3	810	〃

80 년 월 평 균 금 액

	V A	출 하	재 고	내 수	수 출
내구생산재	69,737	239,797	335,639	167,510	72,287
내구소비재	54,517	170,324	239,242	82,756	87,568

품목번호	82440			품 목 명	컴퓨터 및 그 주변기기		
구 분	생 산	대기업	중소기업	출 하	내 수	수 출	재 고
'82. 1	1,762	894	868	2,460	8	2,452	1,121
2	3,238	1,964	1,274	3,911	11	3,900	448
3	2,971	1,436	1,535	1,910	92	1,818	1,509
4	2,247	1,140	1,107	2,101	75	2,026	1,655
5	3,471	1,931	1,540	2,964	554	2,410	2,162
6	3,666	2,064	1,602	2,840	390	2,450	2,988
7	2,049	1,240	809	2,093	567	1,526	2,944
8	1,634	1,104	530	2,522	538	1,984	2,056
9	3,359	1,725	1,634	2,774	509	2,265	2,641
10	3,123	1,445	1,678	3,274	1,029	2,245	2,490
11	3,649	1,910	1,739	3,797	671	3,126	2,342
12	5,688	3,250	2,438	5,351	1,555	3,796	2,679
'83. 1	4,219	2,223	1,996	4,304	2,150	2,154	2,594
2	4,597	2,256	2,341	4,452	2,586	1,866	2,739
3	4,517	2,501	2,016	4,361	2,846	1,515	2,895
4	5,188	2,974	2,214	5,398	2,585	2,813	2,685

품목번호	82440			품 목 명	컴퓨터 및 그 주변기기		
구 분	생 산	대기업	중소기업	출 하	내 수	수 출	재 고
'83. 5	5,241	3,023	2,218	5,226	2,690	2,536	2,700
6	6,284	3,490	2,794	6,065	3,213	2,852	2,919
7	5,379	3,037	2,342	6,555	3,886	2,669	2,743
8	6,594	4,257	2,337	7,008	3,736	3,272	2,329
9	7,193	4,562	2,631	7,185	3,545	3,640	2,337
10	8,426	5,212	3,214	8,799	3,884	4,915	1,964
11	10,606	6,816	3,790	8,559	2,964	6,876	2,731
12	9,806	6,637	3,171	9,525	4,838	4,687	3,014
'84. 1	7,177	4,764	2,413	6,587	2,457	4,130	3,735
2	9,901	6,211	3,690	9,508	2,908	6,606	5,100
3	45,575	41,342	4,233	13,575	8,059	5,516	37,328
4	1,608	1,608		1,479	801	678	637
5	2,101	2,101		2,053	689	1,364	685

품목번호	83390			품 목 명		V T R	
구 분	생 산	대기업	중소기업	출 하	내 수	수 출	재 고
'82. 1	657	657		1,006	1,006		12
2	1,594	1,594		585	585		1,021
3	1,875	1,875		2,491	2,491		405
4	2,352	2,352		2,056	2,056		701
5	3,042	3,042		3,253	3,253		490
6	3,111	3,111		3,029	3,029		572
7	3,640	3,640		3,733	3,733		479
8	4,611	4,611		4,616	4,616		474
9	6,218	6,218		5,691	5,691		1,001
10	6,034	6,034		5,166	5,166		1,869
11	5,611	5,611		4,500	4,500		2,980
12	10,837	10,837		13,324	13,324		493
'83. 1	8,885	8,885		9,108	9,108		270
2	9,300	9,300		8,899	8,899		671
3	9,947	9,947		9,294	9,294		1,324
4	13,342	13,342		13,294	13,294		1,372
5	15,175	15,175		12,850	12,850		3,697
6	14,457	14,457		11,011	11,011		7,143
7	13,860	13,860		10,101	10,101		10,902
8	11,957	11,957		10,788	10,788		12,072
9	9,799	9,799		11,358	11,358		11,382
10	10,874	10,874		9,962	9,962		12,294
11	13,760	13,760		12,556	12,556		13,498
12	19,865	19,865		21,585	21,585		11,778
'84. 1	11,687	11,687		17,855	17,855		5,610
2	21,034	21,034		17,279	17,279		9,360
3	27,404	27,404		23,893	23,893		12,871

보정 file 및 orrginal file

가 중 치 변 동 내 역				
	대 기 업	중소기업	내 수	수 출
000	6,744.0	3,256.0	6,914.6	3,085.4
300	5,978.3	3,152.4	6,328.4	3,074.2
380	1,394.9	614.1	1,111.7	742.6
382	165.1	148.7	213.1	48.2
383	603.6	155.9	336.6	366.5
385	71.7	42.2	35.2	48.1

'80 지수보정분류의 생산, 출하, 재고월평균금액

분 류	VA	생산액	대기업	중소기업	출 하	내 수	수 출	재 고
382	29,073	81,240	15,352	13,721	79,184	66,638	12,546	84,061
383	70,358	152,485	55,629	14,729	212,746	102,952	109,794	292,730
385	10,565	26,581	6,651	3,914	24,891	10,727	14,164	17,984

품 목 번 호		83400		품 목 명		전자레인지	
구 분	생 산	대기업	중소기업	출 하	내 수	수 출	재 고
82. 1	14,000	14,000		16,000	1,000	15,000	1,000
2	29,000	29,000		29,000	8,000	21,000	2,000
3	27,000	27,000		26,000	2,000	24,000	2,000
4	35,000	35,000		32,000	2,000	30,000	5,000
5	32,000	32,000		33,000	1,000	32,000	4,000
6	33,000	33,000		32,000	2,000	30,000	5,000
7	35,000	35,000		34,000	6,000	28,000	6,000
8	37,000	37,000		39,000	2,000	37,000	4,000
9	38,000	38,000		37,000	4,000	33,000	5,000
10	39,000	39,000		39,000	6,000	33,000	5,000
11	34,000	34,000		35,000	3,000	32,000	4,000
12	33,000	33,000		33,000	9,000	24,000	4,000
83. 1	50,000	50,000		43,000	8,000	35,000	11,000
2	41,000	41,000		40,000	4,000	36,000	12,000
3	46,000	46,000		46,000	9,000	37,000	12,000
4	69,000	69,000		70,000	11,000	59,000	11,000
5	77,000	77,000		75,000	14,000	61,000	13,000
6	99,000	99,000		107,000	32,000	75,000	5,000
7	98,000	98,000		90,000	12,000	78,000	13,000
8	104,000	104,000		97,000	4,000	93,000	20,000
9	92,598	92,598		90,551	9,314	81,237	25,349
10	85,892	85,892		87,457	3,981	83,476	23,784
11	97,199	97,199		97,625	9,969	87,656	23,358
12	101,808	101,808		97,519	13,019	84,500	24,062
84. 1	118,795	118,795		122,837	6,667	116,170	20,020
2	119,693	119,693		119,692	5,577	114,115	20,087
3	163,541	163,541		136,592	8,180	128,412	46,970

품 목 번 호		85120		품 목 명		전자계측기	
구 분	생 산	대기업	중소기업	출 하	내 수	수 출	재 고
82. 1	655	483	172	526	54	472	589
2	889	707	182	789	26	763	743
3	1,101	936	165	1,033	39	994	887
4	940	693	247	1,120	39	1,081	841
5	969	733	236	1,155	35	1,120	798
6	1,287	949	338	1,286	95	1,191	845
7	1,173	1,001	172	1,149	99	1,050	887
8	1,067	848	219	1,485	51	1,434	681
9	1,545	1,318	227	1,865	70	1,795	579
10	1,009	802	207	1,268	82	1,186	431
11	1,263	1,004	259	1,213	108	1,105	490
12	1,279	1,017	262	1,432	89	1,343	440
83. 1	1,058	868	190	945	59	886	580
2	1,375	1,173	202	1,558	86	1,472	527
3	1,522	1,324	198	1,599	68	1,531	590
4	1,549	1,321	228	1,484	113	1,371	691
5	1,178	1,011	167	1,417	61	1,356	606
6	1,614	1,398	216	1,546	97	1,449	742
7	1,436	1,217	219	1,509	111	1,398	736
8	1,861	1,623	238	2,197	144	2,053	567
9	1,676	1,315	361	1,895	191	1,704	405
10	1,673	1,379	294	1,500	157	1,343	578
11	1,502	1,226	276	1,561	192	1,369	519
12	1,586	1,350	236	1,545	197	1,348	560
84. 1	1,218	897	321	1,328	186	1,142	474
2	1,524	1,299	225	1,184	152	1,032	682
3	1,741	1,384	357	1,568	219	1,349	178

품 목 번 호		85130		품 목 명		전자복사기	
구 분	생 산	대기업	중소기업	출 하	내 수	수 출	재 고
82. 1	473	473		510	370	140	535
2	482	482		481	390	91	536
3	509	509		564	507	57	481
4	644	644		637	637	-	488
5	781	781		575	535	40	694
6	560	560		638	548	90	616
7	518	518		525	475	50	609
8	586	586		608	523	85	587
9	708	708		679	634	45	616
10	662	662		617	592	25	661
11	540	540		727	687	40	474
12	651	651		626	616	10	499
83. 1	742	742		579	565	14	662
2	698	698		744	744	-	616
3	858	858		916	831	85	558
4	909	909		843	738	105	624
5	836	836		670	600	70	790
6	809	809		905	825	80	694
7	605	605		696	636	60	603
8	614	614		704	659	45	439
9	486	486		588	542	46	337
10	734	734		572	512	60	499
11	724	724		837	761	76	457
12	890	890		1,153	972	181	193
84. 1	888	888		660	660	-	421
2	829	829		876	876	-	374
3	951	951		861	861	-	464

품목번호	품 목 명	가 중 치							
		생 산	생산액	대기업	중 소 기 업	출 하	내 수	수 출	재 고
82010		2.4	2.2	1.8	0.6	2.1	2.1	-	19.5
20		2.5	1.9	-	2.5	2.1	2.1	-	14.2
30		34.3	26.1	27.3	7.0	29.4	28.3	1.1	35.7
40		4.3	3.2	1.3	3.0	3.5	3.5	-	13.5
50		1.5	1.1	1.4	0.1	1.3	1.1	0.2	-
60		16.8	10.0	12.3	4.5	11.6	5.7	5.9	96.6
70		3.1	1.9	1.1	2.0	2.1	1.5	0.6	8.8
80		7.1	4.8	0.1	7.0	4.9	4.8	0.1	18.1
90		3.5	2.5	0.1	3.4	2.4	2.4	-	3.7
100		2.2	1.5	-	2.2	1.5	1.4	0.1	-
110		18.2	10.1	2.6	15.6	10.6	10.4	0.2	-
120		2.0	3.9	1.3	0.7	4.9	4.9	-	45.1
130		6.4	13.3	1.3	5.1	14.9	13.6	1.3	19.4
140		2.7	5.4	2.6	0.1	6.9	5.9	1.0	48.7
150		4.0	2.6	-	4.0	3.0	1.5	1.5	-
160		12.5	7.9	4.9	7.6	8.8	7.9	0.9	12.6
170		4.7	2.9	-	4.7	3.5	2.5	1.0	-
180		4.0	2.6	-	4.0	2.7	2.6	0.1	-
190		6.2	3.9	-	6.2	4.3	3.9	0.4	-
200		13.0	7.0	10.8	2.2	7.6	7.6	-	-
210		2.3	8.7	1.6	0.7	9.2	4.9	4.3	22.2

품목번호	품 목 명	가 중 치							
		생 산	생산액	대기업	중소기업	출 하	내 수	수 출	재 고
82220		0.7	1.5	0.1	0.6	2.0	1.5	0.5	3.5
230		3.5	1.2	-	3.5	1.9	1.9	-	8.4
240		22.0	15.2	14.2	7.8	15.9	15.2	0.7	8.7
250		5.5	3.8	2.1	3.4	3.7	3.5	0.2	3.3
260		5.2	3.0	-	5.2	3.4	3.4	-	-
270		1.7	1.2	-	1.7	1.1	1.1	-	3.8
280		6.9	4.8	5.0	1.9	5.6	4.2	1.4	7.6
290		15.4	10.8	7.0	8.4	10.9	10.7	0.2	-
300		7.1	5.1	1.7	5.4	5.3	5.3	-	5.3
310		8.7	6.8	8.7	-	6.1	5.3	0.8	20.7
320		5.7	4.8	3.6	2.1	8.1	1.7	6.4	1.2
330		6.2	5.2	2.4	3.8	7.2	4.3	2.9	19.0
340		3.0	2.3	2.9	0.1	2.8	2.6	0.2	5.8
350		4.5	2.7	1.1	3.4	2.8	2.6	0.2	3.3
360		9.4	5.8	6.0	3.4	5.5	5.5	-	7.8
370		16.2	10.2	12.4	3.8	12.9	8.0	4.9	28.5
380		5.6	3.8	4.7	0.9	2.7	2.7	-	41.2
390		6.0	4.2	5.1	0.9	4.5	4.3	0.2	9.4
400		6.0	4.1	5.9	0.1	4.4	4.4	-	7.6
410		2.9	2.1	1.0	1.9	2.9	1.6	1.3	5.4
420		3.0	2.2	3.0	-	3.1	1.6	1.5	14.7

품목번호	품 목 명	가 중 치							
		생 산	생산액	대기업	중소기업	출 하	내 수	수 출	재 고
82430		1.8	1.2	-	1.8	1.7	1.7	-	-
440		13.1	9.0	7.7	5.4	9.8	1.6	8.2	17.1
83010		6.9	5.1	3.2	3.7	5.6	5.2	0.4	10.7
20		13.1	9.5	6.1	7.0	10.5	10.0	0.5	36.9
30		25.7	19.9	13.3	12.4	21.2	16.6	4.6	33.7
40		14.0	11.3	4.8	9.2	13.0	10.2	2.8	5.9
50		11.9	9.5	6.6	5.3	11.3	11.2	0.1	10.8
60		55.1	58.2	54.8	0.3	63.6	15.4	48.2	37.8
70		42.6	44.9	42.4	0.2	53.6	22.7	30.9	21.8
80		3.0	3.1	2.0	1.0	2.8	0.1	2.7	7.8
90		18.1	19.1	10.6	7.5	19.9	1.5	18.4	30.9
100		5.8	6.0	5.5	0.3	7.7	4.1	3.6	35.2
110		51.5	54.4	44.7	6.8	58.9	14.2	44.7	49.4
120		13.9	14.6	13.6	0.3	15.9	2.6	13.3	11.4
130		37.4	39.4	12.4	25.0	13.5	4.2	9.3	15.1
140		13.0	13.7	12.7	0.3	14.9	3.0	11.9	14.1
150		12.6	8.1	10.2	2.4	8.1	7.2	0.9	38.5
160		46.3	29.6	44.9	1.4	41.0	39.0	2.0	-
170		28.9	23.1	20.4	9.5	19.0	3.0	16.0	7.3
180		4.6	3.7	3.3	1.3	3.5	0.2	3.3	-
190		48.7	38.7	47.3	1.4	39.7	1.8	37.9	11.4

품목번호	품 목 명	가 중 치							
		생 산	생산액	대기업	중소기업	출 하	내 수	수 출	재 고
83200		24.0	19.1	23.9	0.1	20.7	2.6	18.1	24.0
210		4.2	3.4	2.9	1.3	3.6	3.1	0.5	8.3
220		7.1	5.9	4.1	3.0	6.9	1.3	5.6	9.0
230		6.4	5.2	4.8	1.6	5.5	1.2	4.3	8.3
240		22.6	17.9	15.9	6.7	20.1	6.9	13.2	32.4
250		29.6	23.5	17.9	11.7	25.9	4.6	21.3	24.9
260		15.4	13.0	8.2	7.2	14.6	5.5	9.1	12.2
270		46.4	29.5	46.1	0.3	32.0	27.2	4.8	100.6
280		2.1	6.2	1.8	0.3	9.8	8.9	0.9	59.1
290		10.6	7.2	10.5	0.1	8.0	7.3	0.7	11.3
300		4.1	4.7	1.3	2.8	5.1	3.8	1.3	6.2
310		3.9	4.6	0.7	3.2	4.6	4.6	-	7.1
320		39.8	36.4	34.4	5.4	40.8	38.8	2.0	23.8
330		26.8	24.5	17.0	9.8	27.5	22.6	4.9	30.7
340		4.2	3.0	2.6	1.6	3.2	3.0	0.2	4.3
350		4.8	3.4	1.4	3.4	3.7	0.1	3.6	3.0
360		11.2	10.3	10.3	0.9	11.1	6.3	4.8	12.5
370		6.8	6.1	6.7	0.1	7.0	6.1	0.9	18.2
380		4.0	3.0	1.9	2.1	3.3	1.6	1.7	3.9
390		10.6	7.4	10.6	-	7.7	7.7	-	4.8
400		21.8	15.2	21.8	-	18.3	1.2	17.1	2.6

품목번호	품 목 명	가 중 치							
		생 산	생산액	대기업	중소기업	출 하	내 수	수 출	재 고
85010		19.7	11.1	17.9	1.8	11.7	6.0	5.7	-
20		8.2	4.5	0.5	7.7	4.1	3.8	0.3	22.0
30		6.0	3.8	2.5	3.5	4.0	0.4	3.6	1.7
40		4.8	3.3	3.2	1.6	3.6	0.6	3.0	1.6
50		2.1	2.4	1.1	1.0	2.9	1.8	1.1	1.5
60		2.4	3.0	0.5	1.9	3.0	0.3	2.7	3.6
70		2.5	1.2	2.4	0.1	1.3	0.3	1.0	-
80		38.1	29.5	31.9	6.2	32.3	12.4	19.9	39.0
90		9.6	7.4	1.8	7.8	8.1	4.3	3.8	13.1
100		1.5	1.1	0.1	1.4	1.3	1.1	0.2	2.1
110		9.3	4.6	2.0	7.3	3.3	1.6	1.7	8.3
120		6.4	4.5	4.5	1.9	5.1	0.3	4.8	6.6
130		3.3	2.3	3.3	-	2.6	2.3	0.3	2.6

지수산출방법 (예)

지수	보정후	흑백 TV 150,170	80 년대비 83.8 의 증감
134.9, 146.4	383(20.0)	100	추이
	보정지수	50	
	170,200.0	전기다리미 170,200	부가가치
		(5.0)	
		25	
		VTR	82 년대비 증감추이
		(5.0)	
		25	

주서 : VTR, 전자렌지 → 82 년대비 지수

보정후 383. 소분류의 대총 기여도 = 4.4

흑백 VT의 대소기여도 = 5.9

전기 〃 = 3.8

VTR = 1.8

383 경우 해당되는 품목 → 전자레인지와 VTR

1) 보정품목 2개 기여도의 합이 나온다.

가) 합을 2개로 나누어 주는 방법

VTR의 경우 $\dfrac{\{(\text{VTR의 자율지수증감}) \times \text{VTR의 가중치}\} \times 2\text{개}}{\{(\text{VTR의 자율지수증감}) \times \text{VTR의 가중치}\} + \text{금월} \sim \text{자율지수}}$

$\dfrac{\text{품목의 기여}}{\{(\text{전자레인지 자율지수증감}) \times \text{전자레인지 가중치})\}}$

예) 2개의 기여도가 0.5일 경우

VTR의

품 목	가 중 치	지 수		기 여 도
		전 월	금 월	
V T R	10.6	1,630.0	1,320.0	△ 1.2
전자레인지	21.8	650.0	850.0	1.7
				0.5

$$\text{전자레인지} = \frac{4,360}{1,286} = 3.4 \times 0.5 = 1.7$$

$$\text{VTR} = \frac{(-3,074 \quad \times}{(-3,074) + 4,360} = -2.4 \times 0.5$$

$$= \triangle 1.2$$

〈기여도 산출〉

기여도 : 개개의 품목 혹은 산업이 총지수 혹은 상위의 분류에 대한 증감의 기여수치

예) 총출하가 전월대비 5.0% 증가했을 경우

분 류	가 중 치	전월지수	금월지수	전 월 비	기 여 도	증 감
총 지 수	10,000.0	135.0	141.8	5.0		
383	700.0	126.0	150.0	19.0	1.2	
집 적 회 로	50.0	129.0	170.0	31.8	0.2	41.0
칼 라 T V					0.5	
흑 백 T V					0.4	
냉 장 고					0.1	

집적회로의 대총기여도 산출 예)

$$\text{품목의 대총기여도 (전월비)} = \frac{(\text{금월지수} - \text{전월지수}) \times \text{가중치}}{\text{전월 총지수} \times 100}$$

실제 지수보정에서 기여를 산출하는 경우 (생산)

분 류	가중치	지수		증감율	기여도
		전 월	금 월		
총 지 수		150.0	154.0	2.7	
2	10,000.0				
3					
31					
32					
38					
382	235.0	135.0	150.0		
383	750.0	130.0	150.0		
385	130.0	150.0	190.0		
384					
39					
4					

	A	B	C	D	E	F	G	H	
382	313.8	234.5	165.1	148.7	261.3	213.1	48.2	590.4	보정후
	313.8	234.5	165.7	148.1	261.3	219.9	41.4	590.4	보정전
383	759.5	661.4	603.6	155.9	703.1	336.6	366.5	785.9	보정후
	759.5	661.4	600.5	159.0	703.1	340.7	362.4	785.9	보정전
385	113.9	78.7	71.7	42.2	83.3	35.2	48.1	102.1	보정후
	113.9	78.7	71.7	42.2	83.3	35.9	47.4	102.1	보정전
381									
	415.6	288.7	194.3	221.3	315.0	226.8	88.2	337.9	보정전
384									
	406.2	452.5	360.2	46.0	491.6	300.0	191.6	293.5	보정전
380			1,394.9	614.1		1,111.7	742.6		
	2,009.0	1,715.8	1,392.4	616.6	1,854.3	1,123.3	731.0	2,109.8	보정전
300			5,978.3	3,152.4		6,328.4	3,074.2		
	9,130.7	9,437.0	5,975.8	3,154.9	9,402.6	6,340.0	3,062.6	9,914.8	보정전
000			6,744.0	3,256.0		6,914.6	3,085.4		
	10,000	10,000	6,741.5	3,258.5	10,000	6,926.2	3,073.8	10,000	보정전

총 지 수	대 분 류	중 분 류	소 분 류	중·경공업
		210		
		230		
	200	290		
			311	경
			313	〃
		310	314	〃
			321	〃
			322	〃
			323	〃
		320	324	〃
			331	〃
		330	332	〃
			341	중
		340	342	경
			351	중
			352	〃
			353	〃
			354	〃
			355	경
		350	356	〃
			361	중
			362	〃

총 지 수	대 분 류	중 분 류	소 분 류	중·경공업
		360	369	중
			371	〃
		370	372	〃
			381	〃
			382	〃
			383	〃
			384	〃
		380	385	〃
	300	390		경
000	400			

(광 동 지 수)

1. 84.7월에 디젤기관차의 기준물량을 SM file에서 다음과 같이 고침. (84년 5월 확정치부터 고쳐짐)

품목 84060 A 81 ←(13) B 81 C 81 D 0 E 81

F 81 G 0 H 0 I 0

SM file에서는 84060 (기관차)의 A에서의 기준물량은 5월부터 81이니 4월 이전은 B으로 지수가 산출되었음. BJ file에서는 84.9.28 기준물량을 SM file과 마찬가지로 고쳤음 (8월 잠정 보정지수작업에서) 지수시산은 84년 1월부터 (Do Lee 30 To 37)시산이 되었음.

2. 특수운동화 (55080)의 기준물량을 9월 17일 고치고 지수시산은 82년 1월부터 고쳐진 기준물량에 의해 개정되었음.

(Do Lee = 8 To 37)

※ 광공업 동태지수 (보정업무) 작업순서

1. Line - up

조사표에서 추출된 자료를 File화 한다.

2. insert 1차수정

내수지수와 수출지수는 출하지수에서 물량 삽입.

대기업과 중소기업은 생산지수와 생산액 가중치로 만든다.

◎ 출하지수 (E) → 내수지수(F) + 수출지수(G)

◎ 생산지수 (A), 생산액가중치 (B) → 대기업지수 (C) + 중소기업지수 (D)}

물량삽입

◎ 재고지수 (H) → 재고지수 (H)

◎ 재고율지수 (i) → 가중치까지 내용만 삽입

◎ 철도선박제외 지수

철도선박제외 생산지수 (P) ← 생산지수 (A)

〃 출하지수 (Q) ← 출하지수 (E)

〃 재고시수 (R) ← 재고지수 (H)

각 유별지수 산출에서 각 소분류별 품목에 물량을 보정하여 지수수정

3. List PROGRAM

총지수, 대분류, 중분류, 소분류 및 각 품목별 대기업, 중소기업, 내수, 수출, 재고의 지수를 확인

4. BACK - UP

Original File에서 2 CARD (생산)를 BACK-UP

5. insert 2차수정

Original File에서 보정 File로 37개월(전월, 금월) 지수물량을 만든다.

6. SUMMARY

Original File에서 보정 6개품목은 BYPASS시키고 보정 File에서 보정 5개 품목은 READ하고 특수운동화는 상수를 곱한다.

7. 지수시산 : 금월과 전월지수를 시험 산출

8. EXCHANGE

보정품목의 소분류 "382" "383" "385"의 각 품목을 바뀌어진 가중치로 생산액 가중치를 바꿔준다.

9. List

지수 산적되어 있는 File을 확인

총지수	대분류	중분류	소분류	품목명
000	300	380	382(82440)	컴퓨터및주변기기
			383(83390)	V T R
			383(83400)	전자레인지
			385(85120)	전자계측기
			385(85130)	전자복사기

광동지수 보정 DOCUMENTATION

가) 383품목중에서 5개품목(82년을 기준)과 특수운동화가 추가되어 지수산정에 필요로 하여 보정을 하게 됨.

5개품목은 컴퓨터(83390), 전자레인지(83400), 컴퓨터 및 주변기기(83410), 전자계측기(83420), 전자복사기(83430)이며 특수운동화는 기존 DATA에서(80년 지수개편후) 약 3배를 합산하여 보정한다.

나) 지수보정실례

품목명	82년 ①	82년 ②	'83. 7월			'83. 8월		
	월평균UA	기준물량	물량③	지수④	UA ⑤	물량	지수	UA
VTR	1,027	4,132	13,860	335.4	3,445	11,957	289.4	2,972
전자레인지	2,107	32	98	306.3	6,454	104	325.0	6,848
컴퓨터	1,263	3,238	4,379	135.2	1,708	21,136	652.8	8,245
전자계측기	645	1,657	2,166	130.7	843	2,808	169.5	1,093
전자복사기	333	593	605	102.0	340	603	101.9	339
소계	5,375				12,790			19,497

1) ① ② ③항은 수집된 자료

2) 지수산정 : $\frac{\text{7월물량 ③}}{\text{'82 기준물량 ②}} \times 100$

3) 부가가치 : 지수 × ①항(82년 월평균 부가가치) ÷ 100

① insert program

대기업 생산지수(C), 내수지수(F), 재고지수(H), 철도선박제의 재고지수(R), 중소기업 생산지수(D), 생산지수(A), 생산액 가중치 (B), 철도선박제의 생산지수(P), 수출지수(G), 출하지수(E), 철도선박제의 출하지수(Q) 등 각 유별지수 산출에서 각 소분류별 품목에 물량을 보정하여 지수를 산출

② List program

총지수, 대분류, 중분류, 소분류 및 각 품목별 대기업, 중소기업, 내수, 수출, 재고의 지수를 확인하기 위한 PROGRAM

③ BACK - UP

Original File에서 2CARD (생산)를 BACK - UP

④ EXCHAGE PROGRAM

① ②번 작업후 보정품목의 소분류 "382" "383" "385" 의 각 품목을 바뀌어진 가중치도 생산액가중치를 바꿔준다.

⑤ LIST PROGRAM

지수산적이 되어 있는 File을 Listing

총지수	대분류	중분류	소분류	품 목
000	300	350	355	특수운동화
		380	382	컴퓨터및주변기기
			383	VTR
			383	전자레인지
			385	전자계측기
			385	전자복사기

⑥ 지수시산

각 STEP별 작업완료후 금월과 전월지수를 시험산출

⑦ insert PROGRAM (2번째)

시산후 Original File에서 보정 File로 37개월 (전월, 금월) 지수, 물량을 만든다.

Ⅶ. Program Source List

SMICMS12

MASTER CREATION

(SOURCE PROGRAM)

```
//SMICMS12  JOB  TYPRUN=HOLD                                          SMI00010
//SS EXEC PLIFCL                                                      SMI00080
//PLI.SYSIN DD *                                                      SMI00090
* PROCESS GS,NEST,OPT(2);
  AA: PROC OPTIONS(MAIN);                                             SMI00100
   DCL D     CHAR(80), AB CHAR(5) DEF D ;                             SMI00120
   DCL INF FILE RECORD INPUT;                                         SMI00120
   DCL OTT FILE RECORD KEYED ENV(VSAM);
   DCL ONCODE  BUILTIN;
   ON KEY(OTT) BEGIN;  CNT=CNT-1;
     IF ONCODE=52 THEN PUT SKIP EDIT(D)(A(80));
   END;
   OPEN FILE(INF);  CNT=0;                                            SMI00150
   OPEN FILE(OTT) SEQUENTIAL OUTPUT;
   ON ENDFILE(INF) GOTO LT;                                           SMI00190
  RC: READ FILE(INF) INTO(D);                                         SMI00200
   IF SUBSTR(D,40,1)=' ' | SUBSTR(D,40,1)='6' |
    SUBSTR(D,40,1)='9' THEN;   ELSE GOTO RC;
   WRITE FILE(OTT) FROM(D) KEYFROM(AB); CNT=CNT+1;
   GOTO RC;                                                           SMI00250
  LT: CLOSE FILE(OTT), FILE(INF);                                     SMI00260
    PUT SKIP EDIT('TOTAL RECORDS= ',CNT)(A,F(5));
      END AA;                                                         SMI00260
/*                                                                    SMI00270
//LKED.SYSLMOD DD DSN=USER.LOADLIB(SMIMS12),DISP=SHR
//
```

SMICOU12

OK UPDATE

(SOURCE PROGRAM)

```
//SMICOU12 JOB TYPRUN=HOLD                                          SMI00010
//  EXEC PLIFCL                                                     SMI00020
//PLI.SYSIN DD *                                                    SMI00030
* PROCESS GS,NEST,A,XREF,OPT(2);                                    SMI00040
 UPDATE: PROC OPTIONS(MAIN);                                        SMI00050
 DCL CMONTH          PIC'99';                                       SMI00070
 DCL RETURN_CODE     FIXED BIN(15);
 DCL ONCODE          BUILTIN;                                       SMI00110
 DCL TAIN FILE RECORD INPUT;                                        SMI00120
 DCL EDIT3 FILE RECORD KEYED ENV(VSAM);                             SMI00130
 DCL (RTYPE,NOER,NOINV)  BIT(1) INIT('1'B);                         SMI00150
 DCL TREC                CHAR(80);                                  SMI00170
 DCL SREC                CHAR(80);                                  SMI00180
 DCL 1 T1 DEF SREC,                                                 SMI00190
      2 SAUP             CHAR(5),                                   SMI00200
      2 CD               CHAR(1),                                   SMI00210
      2 NUM              CHAR(2),                                   SMI00220
      2 DATA             CHAR(67),                                  SMI00230
      2 TAB             CHAR(1),                                    SMI00240
      2 TMON            CHAR(2),                                    SMI00250
      2 F1              CHAR(1),                                    SMI00260
      2 CHECK           CHAR(1);                                    SMI00270
 DCL PIC7                PIC'------9',                              SMI00280
     PIC8                PIC'(8)9',                                 SMI00290
     PIC5                PIC'99999';                                SMI00300
 DCL 1 OREC,                                                        SMI00310
      2 EKEY             CHAR(8),                                   SMI00320
      2 TDATA            CHAR(40),                                  SMI00330
      2 OCHECK           CHAR(1),                                   SMI00340
      2 OFFF             CHAR(1);                                   SMI00340
 DCL 1 OD1 DEF SREC POS(9),                                         SMI00350
      2 OINDUST           CHAR(5),                                  SMI00360
      2 OSIZE             CHAR(1),                                  SMI00370
      2 OYUGO1            CHAR(1);                                  SMI00380
 DCL 1 CD1 DEF TDATA,                                               SMI00420
      2 INDUST           CHAR(5),                                   SMI00430
      2 SIZE             CHAR(1),                                   SMI00440
      2 YUGO1            CHAR(1),                                   SMI00450
      2 TBL              CHAR(1),                                   SMI00490
      2 MON              PIC'99';                                   SMI00500
 DCL 1 CD2 DEF TDATA,                                               SMI00510
      2 PUM              CHAR(5),                                   SMI00520
      2 YUGO2            CHAR(1),                                   SMI00530
      2 DAT2(7)          FIXED BIN(31),                             SMI00540
      2 WHY(2)           CHAR(1);                                   SMI00550
 DCL 1 CD3 DEF TDATA,                                               SMI00560
      2 WON              CHAR(4),                                   SMI00570
      2 YUGO3            CHAR(1),                                   SMI00580
      2 GU               CHAR(1),                                   SMI00590
      2 DAT3(2)          FIXED BIN(31);                             SMI00600
 DCL DAT4(10)            FIXED BIN(31)  DEF TDATA;                  SMI00610
 DCL DAT5(7)             FIXED BIN(31)  DEF TDATA;                  SMI00620
 DCL 1 CD6 DEF TDATA,                                               SMI00630
      2 CPUM             CHAR(5),                                   SMI00640
      2 YUGO6            CHAR(1),                                   SMI00650
```

```
     2 DAT6(2)          FIXED BIN(31);                                  SMI00660
 DCL SCND      BIT(1) INIT('1'B);
 ON KEY(EDIT3) BEGIN;                                                  SMI00670
   IF ONCODE=51 THEN NOINV='0'B;                                       SMI00680
 END;                                                                  SMI00690
  GET SKIP EDIT(CMONTH)(P'99');                                        SMI00750
  OPEN FILE(TAIN); OPEN FILE(EDIT3) SEQUENTIAL UPDATE;                 SMI01380
 RPO:  READ FILE(TAIN) INTO(TREC);                                     SMI01410
  CALL PLISRTD(' SORT FIELDS=(77,2,CH,A,80,1,CH,D,1,8,CH,A) ',
   'RECORD TYPE=F,LENGTH=(80) ', 45000, RETURN_CODE, E15X, E35X);
  CALL PLIRETC(RETURN_CODE);
 E15X: PROC RETURNS(CHAR(80));
  ON ENDFILE(TAIN) RTYPE='0'B;
  DO WHILE(RTYPE);
    SREC=TREC;
    CALL KEYCHEK;
  RRT:  READ FILE(TAIN) INTO(TREC);      /*********************/
    IF NOER THEN DO;
      CALL PLIRETC(12);
      RETURN(SREC);
    END;
    NOER='1'B;
  END;
  CALL PLIRETC(8);
 END E15X;
 E35X: PROC(SRTREC);
  DCL SRTREC   CHAR(80);
  SREC=SRTREC;
  IF CHECK='1' THEN CALL DELTRTN;                                      SMI01420
  ELSE CALL UPDTRTN;                                                   SMI01430
  CALL PLIRETC(4);
 END E35X;
 DELTRTN: PROC;                                                        SMI01440
  ON ENDFILE(EDIT3) GOTO DEND;
  IF DATA¬=(63)' ' THEN DO;
    PUT SKIP EDIT(SREC,'FUNCTION ERROR')(A(85),A); RETURN;
  END;
  IF CD=' ' THEN DO;
  BB:  IF SAUP>SUBSTR(EKEY,1,5) THEN DO;
      READ FILE(EDIT3) INTO(OREC);  GOTO BB;
    END;
    IF SAUP<SUBSTR(EKEY,1,5) THEN DO;
      PUT SKIP EDIT(SREC,'DELETE NOT FOUND')(A(85),A); RETURN;
    END;  /***
    IF OCHECK='1' THEN DO;
      PUT SKIP EDIT(SREC,'DELETE CHECK ERROR')(A(85),A); RETURN;
    END;  **********************/
    DO WHILE(SAUP=SUBSTR(EKEY,1,5));                                   SMI01480
      DELETE FILE(EDIT3) KEY(EKEY);                                    SMI01490
      PUT SKIP EDIT(EKEY,'DELETE')(A(85),A);
      READ FILE(EDIT3) INTO(OREC);
    END;                                                               SMI01500
  END;
  ELSE IF NUM=' ' THEN DO;
  BBC: IF SAUP || CD>SUBSTR(EKEY,1,6) THEN DO;
```

```
      READ FILE(EDIT3) INTO(OREC);  GOTO BBC;
    END;
    IF SAUP || CD<SUBSTR(EKEY,1,6) THEN DO;
      PUT SKIP EDIT(SREC,'DELETE NOT FOUND')(A(85),A); RETURN;
    END;  /************
    IF OCHECK='1' THEN DO;
      PUT SKIP EDIT(SREC,'DELETE CHECK ERROR')(A(85),A); RETURN;
    END;   ****************/
    DO WHILE(SAUP || CD=SUBSTR(EKEY,1,6));                          SMI01480
      DELETE FILE(EDIT3) KEY(EKEY);                                 SMI01490
      PUT SKIP EDIT(EKEY,'DELETE')(A(85),A);
      READ FILE(EDIT3) INTO(OREC);
    END;                                                            SMI01500
  END;
  ELSE DO;
  BBD: IF SAUP || CD || NUM>EKEY THEN DO;
      READ FILE(EDIT3) INTO(OREC); GOTO BBD;
    END;
    IF SAUP || CD || NUM<EKEY THEN DO;
      PUT SKIP EDIT(SREC,'DELETE NOT FOUND')(A(85),A); RETURN;
    END;     /*************
    IF OCHECK='1' THEN DO;
      PUT SKIP EDIT(SREC,'DELETE CHECK ERROR')(A(85),A); RETURN;
    END;   *************/
    DELETE FILE(EDIT3) KEY(EKEY);
    PUT SKIP EDIT(EKEY,'DELETE')(A(85),A);
  END;
  DEND:
 END DELTRTN;                                                       SMI01550
 UPDTRTN: PROC;                                                     SMI01560
  IF SCND THEN DO;                                                  SMI01570
    SCND='0'B; CLOSE FILE(EDIT3); OPEN FILE(EDIT3) DIRECT UPDATE;   SMI01580
  END;
  EKEY=SUBSTR(SREC,1,8);
  READ FILE(EDIT3) INTO(OREC) KEY(EKEY);                            SMI01590
  IF NOINV THEN DO;    /*********                                  SMI01600
    IF OCHECK='1' THEN DO;
      PUT SKIP EDIT(SREC,'CHECK ERROR')(A(85),A); RETURN;
    END;    ***********/
    CALL CDCHECK;                                                   SMI01620
    REWRITE FILE(EDIT3) FROM(OREC) KEY(EKEY);                       SMI01630
  END;                                                              SMI01640
  ELSE DO;  /****
    SUBSTR(EKEY,6,3)='100'; NOINV='1'B;
    READ FILE(EDIT3) INTO(OREC) KEY(EKEY);
    IF NOINV & OCHECK='1' THEN DO;
      PUT SKIP EDIT(SREC,'CHECK2 ERROR')(A(85),A); RETURN;
    END;   ******/
    NOINV='0'B; TDATA=' '; OCHECK=' ';
    EKEY=SUBSTR(SREC,1,8);
    CALL CDCHECK;                                                   SMI01690
    WRITE FILE(EDIT3) FROM(OREC) KEYFROM(EKEY);                     SMI01700
    NOINV='1'B;
  END;                                                              SMI01750
 END UPDTRTN;                                                       SMI01760
```

```
 KEYCHEK: PROC;                                                          SMI02140
 IF CHECK='1' THEN                                                       SMI02150
    SUBSTR(SREC,1,5)=TRANSLATE(SUBSTR(SREC,1,5),'0',' ');                SMI02150
  ELSE DO;                                                               SMI02150
    SUBSTR(SREC,1,8)=TRANSLATE(SUBSTR(SREC,1,8),'0', ' ');
    DO I=1 TO 8;
      IF SUBSTR(SREC,I,1)<'0' | SUBSTR(SREC,I,1)>'9' THEN DO;
        NOER='0'B; RETURN; END;
    END;
  END;
  IF CD='1' | CD='4' | CD='5' THEN NUM='00';
  IF CD='4' THEN SREC=SUBSTR(SREC,1,13) || (5)' ' || SUBSTR(SREC,14,28)
     || (8)' ' || SUBSTR(SREC,42,21) || SUBSTR(SREC,76,5);
  IF CD='5' THEN SREC=SUBSTR(SREC,1,16) || (8)' ' || SUBSTR(SREC,17,51)
     || SUBSTR(SREC,76,5);
  IF TMON='  ' THEN NOER='0'B;
  ELSE DO;
    IF SUBSTR(TMON,2,1)=' ' THEN MON=SUBSTR(TMON,1,1);                   SMI02160
    ELSE MON=TMON;                                                       SMI02170
    TMON=MON;
    IF MON¬=CMONTH THEN NOER='0'B;                                       SMI02180
    IF CHECK¬='1' & (CD<'1' | CD>'6') THEN NOER='0'B;                    SMI02190
  END;
  IF NOER='0'B THEN PUT SKIP EDIT(SREC,'KEY ERROR')(A(85),A);
 END KEYCHEK;                                                            SMI02200
 CDCHECK: PROC;                                                          SMI02210
  IF CD='1' THEN CALL CD1RTN;                                            SMI02220
  ELSE IF CD='2' THEN CALL CD2RTN;                                       SMI02230
  ELSE IF CD='3' THEN CALL CD3RTN;                                       SMI02240
  ELSE IF CD='4' THEN CALL CD4RTN;                                       SMI02250
  ELSE IF CD='5' THEN CALL CD5RTN;                                       SMI02260
  ELSE CALL CD6RTN;                                                      SMI02270
 END CDCHECK;                                                            SMI02280
 CD1RTN: PROC;                                                           SMI02290
  IF NOINV THEN PUT SKIP EDIT(EKEY,INDUST,SIZE,YUGO1,TBL,MON,'ORIGINAL')SMI02300
    (A(8),A(5),2 A(1),X(60),A(1),A(2),X(7),A);                           SMI02310
  IF OINDUST='-' THEN INDUST=' ';                                        SMI02320
  ELSE IF OINDUST¬=' ' THEN INDUST=OINDUST;                              SMI02320
  IF OSIZE='-' THEN SIZE=' ';                                            SMI02330
  ELSE IF OSIZE¬=' ' THEN SIZE=OSIZE;                                    SMI02330
  IF OYUGO1='-' THEN YUGO1=' ';                                          SMI02340
  ELSE IF OYUGO1¬=' ' THEN YUGO1=OYUGO1;                                 SMI02340
  IF TAB='-' THEN TBL=' ';                                               SMI02410
  ELSE IF TAB¬=' ' THEN TBL=TAB;                                         SMI02410
  MON=TMON;                                                              SMI02420
  PUT SKIP EDIT(EKEY,INDUST,SIZE,YUGO1,TBL,MON,'UPDATE')                 SMI02430
    (A(8),A(5),2 A(1),X(60),A(1),A(2),X(7),A);                           SMI02440
  PUT SKIP;
 END CD1RTN;                                                             SMI02450
 CD2RTN: PROC;                                                           SMI02460
  IF NOINV THEN PUT SKIP EDIT(EKEY,PUM,YUGO2,(DAT2(I) DO I=1 TO 7),      SMI02470
  WHY(1),WHY(2),'ORIGINAL')(A(8),A(5),A(1),7 F(7),X(1),2 A(1),X(19),A); SMI02480
  IF SUBSTR(DATA,1,5)¬=' ' THEN PUM=SUBSTR(DATA,1,5);                    SMI02490
  IF SUBSTR(DATA,6,1)='-' THEN YUGO2=' ';                                SMI02500
  ELSE IF SUBSTR(DATA,6,1)¬=' ' THEN YUGO2=SUBSTR(DATA,6,1);             SMI02500
```

```
  IF SUBSTR(DATA,56,1)='-' THEN WHY(1)=' ';                          SMI02510
  ELSE IF SUBSTR(DATA,56,1)¬=' ' THEN WHY(1)=SUBSTR(DATA,56,1);      SMI02510
  IF SUBSTR(DATA,57,1)='-' THEN WHY(2)=' ';                          SMI02520
  ELSE IF SUBSTR(DATA,57,1)¬=' ' THEN WHY(2)=SUBSTR(DATA,57,1);      SMI02520
  K=0;                                                               SMI02530
  DO I=7,14,21,28,35,42,49;                                          SMI02540
    K=K+1;                                                           SMI02550
    IF SUBSTR(DATA,I,1)¬=' ' THEN CALL RJ7RTN;                       SMI02560
    ELSE IF NOINV='0'B THEN DAT2(K)=0;
  END;                                                               SMI02570
  PUT SKIP EDIT(EKEY,PUM,YUGO2,(DAT2(I) DO I=1 TO 7),WHY(1),         SMI02580
  WHY(2),'UPDATE')(A(8),A(5),A(1),7 F(7),X(1),2 A(1),X(19),A);       SMI02590
  PUT SKIP;
 END CD2RTN;                                                         SMI02600
 CD3RTN: PROC;                                                       SMI02610
  IF NOINV THEN PUT SKIP EDIT(EKEY,WON,YUGO3,GU,DAT3(1),DAT3(2),     SMI02620
    'ORIGINAL')(A(8),A(4),2 A(1),2 F(7),X(57),A);                    SMI02630
  IF SUBSTR(DATA,1,4)¬=' ' THEN WON=SUBSTR(DATA,1,4);                SMI02640
  IF SUBSTR(DATA,5,1)='-' THEN YUGO3=' ';                            SMI02650
  ELSE IF SUBSTR(DATA,5,1)¬=' ' THEN YUGO3=SUBSTR(DATA,5,1);         SMI02650
  IF SUBSTR(DATA,6,1)¬=' ' THEN GU=SUBSTR(DATA,6,1);                 SMI02660
  K=0;                                                               SMI02670
  DO I=7,14;                                                         SMI02680
    K=K+1;                                                           SMI02690
    IF SUBSTR(DATA,I,1)¬=' ' THEN CALL RJ7RTN;                       SMI02700
    ELSE IF NOINV='0'B THEN DAT2(K)=0;
  END;                                                               SMI02710
  PUT SKIP EDIT(EKEY,WON,YUGO3,GU,DAT3(1),DAT3(2),'UPDATE')          SMI02720
    (A(8),A(4),2 A(1),2 F(7),X(57),A);                               SMI02730
  PUT SKIP;
 END CD3RTN;                                                         SMI02740
 CD4RTN: PROC;                                                       SMI02750
 IF NOINV THEN PUT SKIP EDIT(EKEY,(DAT4(I) DO I=1,3 TO 7,9,10),      SMI02760
  'ORIGINAL')                                                        SMI02760
   (A(8),5 F(5),3 F(8),X(15),A);                                     SMI02770
  K=0;                                                               SMI02780
  DO I=1,6,11,16,21,26;                                              SMI02790
    K=K+1;                                                           SMI02800
    IF SUBSTR(DATA,I,1)¬=' ' THEN DO;                                SMI02810
      J=INDEX(SUBSTR(DATA,I,5),' ');                                 SMI02820
      IF J¬=0 THEN DO;                                               SMI02830
        CALL SUBNUM;
        PIC5=SUBSTR(DATA,I,J-1);                                     SMI02840
        DAT4(K)=PIC5;                                                SMI02850
      END;                                                           SMI02860
      ELSE DO; J=6; CALL SUBNUM;
        DAT4(K)=SUBSTR(DATA,I,5); END;
    END;                                                             SMI02870
    ELSE IF NOINV='0'B THEN DAT4(K)=0;
  END;                                                               SMI02880
  DO I=31,39,47,55;                                                  SMI02890
    K=K+1;                                                           SMI02900
    IF SUBSTR(DATA,I,1)¬=' ' THEN DO;                                SMI02910
      J=INDEX(SUBSTR(DATA,I,8),' ');                                 SMI02920
      IF J¬=0 THEN DO;                                               SMI02930
```

```
        CALL SUBNUM;
        PIC8=SUBSTR(DATA,I,J-1);                                    SMI02940
        DAT4(K)=PIC8;                                               SMI02950
      END;                                                          SMI02960
      ELSE DO; J=9; CALL SUBNUM;
         DAT4(K)=SUBSTR(DATA,I,8); END;
    END;                                                            SMI02970
    ELSE IF NOINV='0'B THEN DAT4(K)=0;
  END;                                                              SMI02980
  PUT SKIP EDIT(EKEY,(DAT4(I) DO I=1,3 TO 7,9,10),'UPDATE')         SMI02990
   (A(8),5 F(5),3 F(8),X(15),A);                                    SMI03000
  PUT SKIP;
 END CD4RTN;                                                        SMI03010
 CD5RTN: PROC;                                                      SMI03020
 IF NOINV THEN PUT SKIP EDIT(EKEY,(DAT4(I) DO I=1,3 TO 7),'ORIGINAL')  SMI03030
   (A(8),6 F(8),X(21),A);                                           SMI03040
  K=0;                                                              SMI03050
  DO I=1,9,17,25,33,41,49;                                          SMI03060
    K=K+1;                                                          SMI03070
    IF SUBSTR(DATA,I,1)¬=' ' THEN DO;                               SMI03080
      J=INDEX(SUBSTR(DATA,I,8),' ');                                SMI03090
      IF J¬=0 THEN DO;                                              SMI03100
        CALL SUBNUM;
        PIC8=SUBSTR(DATA,I,J-1);                                    SMI03110
        DAT5(K)=PIC8;                                               SMI03120
      END;                                                          SMI03130
      ELSE DO; J=9; CALL SUBNUM;
        DAT5(K)=SUBSTR(DATA,1,8); END;
    END;                                                            SMI03140
    ELSE IF NOINV='0'B THEN DAT5(K)=0;
  END;                                                              SMI03150
  PUT SKIP EDIT(EKEY,(DAT4(I) DO I=1,3 TO 7),'UPDATE')              SMI03160
   (A(8),6 F(8),X(21),A);                                           SMI03170
  PUT SKIP;
 END CD5RTN;                                                        SMI03180
 CD6RTN: PROC;                                                      SMI03190
  IF NOINV THEN PUT SKIP EDIT(EKEY,CPUM,YUGO6,DAT6(1),DAT6(2),      SMI03200
    'ORIGINAL')(A(8),A(5),A(1),2 F(7),X(57),A);                     SMI03210
  IF SUBSTR(DATA,1,5)¬=' ' THEN CPUM=SUBSTR(DATA,1,5);              SMI03220
  IF SUBSTR(DATA,6,1)='-' THEN YUGO6=' ';                           SMI03230
  ELSE IF SUBSTR(DATA,6,1)¬=' ' THEN YUGO6=SUBSTR(DATA,6,1);        SMI03230
  K=0;                                                              SMI03240
  DO I=7,14;                                                        SMI03250
    K=K+1;                                                          SMI03260
    IF SUBSTR(DATA,I,1)¬=' ' THEN CALL RJ7RTN;                      SMI03270
    ELSE IF NOINV='0'B THEN DAT2(K)=0;
  END;                                                              SMI03280
  PUT SKIP EDIT(EKEY,CPUM,YUGO6,DAT6(1),DAT6(2),'UPDATE')           SMI03290
    (A(8),A(5),A(1),2 F(7),X(57),A);                                SMI03300
  PUT SKIP;
 END CD6RTN;                                                        SMI03310
 RJ7RTN: PROC ;                                                     SMI03320
  J=INDEX(SUBSTR(DATA,I,7),' ');                                    SMI03330
  IF J¬=0 THEN DO;                                                  SMI03340
    CALL SUBNUM;
```

```
    PIC7=SUBSTR(DATA,I,J-1);                                          SMI03350
    DAT2(K)=PIC7;                                                     SMI03360
  END;                                                                SMI03370
  ELSE DO; J=8; CALL SUBNUM;
    DAT2(K)=SUBSTR(DATA,I,7); END;
 END RJ7RTN;                                                          SMI03380
 SUBNUM: PROC;
  IF SUBSTR(DATA,I,1)='-' THEN DO;
    IF SUBSTR(DATA,I+1,1)=' ' THEN SUBSTR(DATA,I,1)='0';
  END;
  ELSE IF (SUBSTR(DATA,I,1)<'0' | SUBSTR(DATA,I,1)>
   '9') THEN SUBSTR(DATA,I,1)='0';
  DO Q=1 TO J-1;
    IF SUBSTR(DATA,I+Q,1)<'0' | SUBSTR(DATA,I+Q,1)>'9' THEN
     SUBSTR(DATA,I+Q,1)='0';
  END;
 END SUBNUM;
 END UPDATE;                                                          SMI03390
/*                                                                    SMI03400
//LKED.SYSLMOD DD DSN=USER.LOADLIB(SMIOU12),DISP=SHR
//
```

SMICMC12

FILE CHANGE

```
//SMICMC12 JOB TYPRUN=HOLD                                          SMI00010
//ST3 EXEC PLIFCL                                                   SMI00030
//PLI.SYSIN DD *                                                    SMI00040
   CREAT:  PROC OPTIONS(MAIN);                                      SMI00050
     DCL DIREC FILE RECORD KEYED ENV(VSAM);                         SMI00070
     DCL TA    FILE RECORD INPUT;                                   SMI00080
     DCL ONCODE BUILTIN;                                            SMI00090
     DCL TREC   CHAR(50);                                           SMI00100
     DCL DK   CHAR(8) DEF TREC;                                     SMI00110
     ON ENDFILE(TA) GOTO LT;   OPEN FILE(TA);                       SMI00150
     OPEN  FILE(DIREC) SEQUENTIAL OUTPUT;   CNT=0;                  SMI00160
     ON KEY(DIREC) BEGIN;
       IF ONCODE=52 THEN PUT EDIT(DK)(X(2),A(8));
     END;
   RT: READ FILE(TA) INTO(TREC);  CNT=CNT+1;                        SMI00170
        WRITE FILE(DIREC) FROM(TREC) KEYFROM(DK);                   SMI00180
         GOTO RT;                                                   SMI00190
   LT: PUT SKIP EDIT('CNT=',CNT)(A,F(6));                           SMI00200
     CLOSE  FILE(DIREC), FILE(TA);                                  SMI00210
    END CREAT;                                                      SMI00220
/*                                                                  SMI00230
//LKED.SYSLMOD DD DSN=USER.LOADLIB(SMIMC12),DISP=SHR                SMI00240
//                                                                  SMI00270
```

SMICMV12

SUMFILE MOVE
(SOURCE PROGRAM)

```
//SMICMV12 JOB CLASS=1,TYPRUN=HOLD                                      SMI00010
//  EXEC PLIFCL                                                         SMI00020
//PLI.SYSIN DD *                                                        SMI00030
* PROCESS GS,NEST,A,XREF OPT(2);                                        SMI00040
 SUMFILE:PROC OPTIONS(MAIN);                                            SMI00050
  DCL SUMF FILE RECORD KEYED ENV(VSAM);                                 SMI00060
 DCL 1 SREC,                                                            SMI00070
      2 OKEY        CHAR(10),                                           SMI00080
      2 SPC         CHAR(1),                                            SMI00090
      2 OYEAR       PIC'99',                                            SMI00100
      2 OMONTH      PIC'99',                                            SMI00110
      2 WEIGHT      FLOAT,                                              SMI00120
      2 BASEM  FIXED BIN(31),                                           SMI00130
      2 DATAO(37),                                                      SMI00140
        3 ODATA FIXED BIN(31),                                          SMI00150
        3 JDATA FLOAT,                                                  SMI00160
      2 SIGN        CHAR(1);                                            SMI00170
 DCL ONCODE      BUILTIN;                                               SMI00180
 ON KEY(SUMF) BEGIN;                                                    SMI00190
   IF ONCODE=52 THEN PUT EDIT(OKEY)(A(12));                             SMI00200
 END;  TOT=0;                                                           SMI00210
 ON ENDFILE(SUMF) GOTO LT;                                              SMI00220
 OPEN FILE(SUMF) SEQUENTIAL UPDATE;                                     SMI00230
 RT: READ FILE(SUMF) INTO(SREC);                                        SMI00240
   IF TOT=0 THEN PUT SKIP EDIT(OKEY,OYEAR,OMONTH,(JDATA(I) DO I=1 TO    SMI00250
     12))(A(12),2 P'99',12 F(8,1));                                     SMI00260
   IF OMONTH=12 THEN DO;                                                SMI00270
     OYEAR=OYEAR+1;  OMONTH=1; END;                                     SMI00280
   ELSE OMONTH=OMONTH+1;                                                SMI00290
   DO I=1 TO 36;                                                        SMI00300
     ODATA(I)=ODATA(I+1); JDATA(I)=JDATA(I+1);                          SMI00310
   END;                                                                 SMI00320
   ODATA(37)=0;  JDATA(37)=0;                                           SMI00330
   IF TOT=0 THEN PUT SKIP EDIT(OKEY,OYEAR,OMONTH,(JDATA(I) DO I=1 TO    SMI00340
     12))(A(12),2 P'99',12 F(8,1));                                     SMI00350
   REWRITE FILE(SUMF) FROM(SREC);                                       SMI00360
   TOT=TOT+1;                                                           SMI00370
   GOTO RT;                                                             SMI00380
  LT: CLOSE FILE(SUMF);                                                 SMI00390
   PUT SKIP EDIT('TOT RECORD= ',TOT)(A,F(10));                          SMI00400
 END SUMFILE;                                                           SMI00410
/*                                                                      SMI00420
//LKED.SYSLMOD DD DSN=USER.LOADLIB(SMIMV12),DISP=SHR                    SMI00430
//                                                                      SMI00440
```

SMICCV12

CONVERT
(SOURCE PROGRAM)

```
//SMICCV12 JOB TYPRUN=HOLD                                              SMI00010
//  EXEC PLIFCL                                                         SMI00020
//PLI.SYSIN DD *                                                        SMI00030
* PROCESS GS,NEST,A,XREF,OPT(2);                                        SMI00040
 UPDATE: PROC OPTIONS(MAIN);                                            SMI00050
 DCL RETURN_CODE    FIXED BIN(15);                                      SMI00070
 DCL ONCODE         BUILTIN;                                            SMI00110
 DCL EDIT2 FILE RECORD KEYED ENV(VSAM);                                 SMI00130
 DCL EDIT3 FILE RECORD KEYED ENV(VSAM);                                 SMI00130
 DCL RTYPE            BIT(1) INIT('1'B);                                SMI00150
 DCL 1 OREC,                                                            SMI00310
      2 EKEY             CHAR(8),                                       SMI00320
      2 TDATA            CHAR(40),                                      SMI00330
      2 FA               CHAR(2);                                       SMI00340
 DCL 1 EKK DEF EKEY,
      2 SAUP            CHAR(5),
      2 CD              CHAR(1),
      2 NUM             CHAR(2);
 DCL 1 CD2 DEF TDATA,                                                   SMI00510
      2 PUM              CHAR(5),                                       SMI00520
      2 DAT2             CHAR(35);                                      SMI00530
 DCL 1 CD3 DEF TDATA,                                                   SMI00560
      2 WON              CHAR(4),                                       SMI00570
      2 YUGO3            CHAR(1),                                       SMI00580
      2 GU               CHAR(1),                                       SMI00590
      2 DAT3             CHAR(34);                                      SMI00600
 DCL 1 CD6 DEF TDATA,                                                   SMI00630
      2 CPUM             CHAR(5),                                       SMI00640
      2 DAT6             CHAR(35);                                      SMI00660
 DCL PREC          CHAR(53);
 DCL 1 PREC1 DEF PREC,
      2 PKEY       CHAR(11),
      2 PDATA      CHAR(40),
      2 PFA        CHAR(2);
 ON KEY(EDIT2) BEGIN;
   IF ONCODE=52 THEN PUT SKIP EDIT('DATA DUPLICATE : ',PKEY)(A,A(11));
 END;
 OPEN FILE(EDIT3) SEQUENTIAL INPUT;                                     SMI01040
 OPEN FILE(EDIT2) SEQUENTIAL OUTPUT;                                    SMI01040
 ON ENDFILE(EDIT3) RTYPE='0'B;                                          SMI01060
 READ FILE(EDIT3) INTO(OREC);                                           SMI01070
 CALL PLISRTD(' SORT FIELDS=(1,11,CH,A) ',' RECORD TYPE=F,LENGTH=(53) ',
  45000, RETURN_CODE, E15X, E35X);
 CALL PLIRETC(RETURN_CODE);
 E15X: PROC RETURNS(CHAR(53));
 DO WHILE(RTYPE);
  IF CD='1' THEN CALL CD1RTN;
  ELSE IF CD='2' THEN CALL CD2RTN;
  ELSE IF CD='3' THEN CALL CD3RTN;
  ELSE IF CD='4' THEN CALL CD1RTN;
  ELSE IF CD='5' THEN CALL CD1RTN;
  ELSE CALL CD6RTN;
  PFA=FA;
  READ FILE(EDIT3) INTO(OREC);
  CALL PLIRETC(12);
```

```
  RETURN(PREC);
 END;
 CALL PLIRETC(8);
 END E15X;
 E35X: PROC(KREC);
  DCL KREC    CHAR(53);
  PREC=KREC;
  WRITE FILE(EDIT2) FROM(PREC) KEYFROM(PKEY);
  CALL PLIRETC(4);
 END E35X;
 CD1RTN: PROC;
  PKEY=SAUP || CD ;  PDATA=TDATA;
 END CD1RTN;
 CD2RTN: PROC;
  PKEY=SAUP || CD || PUM; PDATA=NUM || DAT2;
 END CD2RTN;
 CD3RTN: PROC;
  PKEY=SAUP || CD || WON || GU;
  PDATA=NUM || YUGO3 || DAT3;
 END CD3RTN;
 CD6RTN: PROC;
  PKEY=SAUP || CD || CPUM; PDATA=NUM || .DAT6;
 END CD6RTN;
 END UPDATE;                                                          SMI03390
/*                                                                    SMI03400
//LKED.SYSLMOD DD DSN=USER.LOADLIB(SMICV12),DISP=SHR
//
```

SMICCR12

CREATION
(SOURCE CREATION)

```
//SMICCR12 JOB TYPRUN=HOLD                                          SMI00010
//JOBCAT DD DSN=SMICT12,DISP=SHR                                    SMI00020
//  EXEC PLIFCL                                                     SMI00030
//PLI.SYSIN DD *                                                    SMI00040
* PROCESS GS,NEST,A,XREF,OPT(2);                                    SMI00050
 UPDATE: PROC OPTIONS(MAIN);                                        SMI00060
 DCL CMONTH          PIC'99';                                       SMI00070
 DCL ONCODE          BUILTIN;                                       SMI00080
 DCL TAIN FILE RECORD INPUT;                                        SMI00090
 DCL EDIT3 FILE RECORD KEYED SEQUENTIAL OUTPUT ENV(VSAM);           SMI00100
 DCL (RTYPE,NOER)       BIT(1) INIT('1'B);                          SMI00110
 DCL TREC                 CHAR(80);                                 SMI00120
 DCL SREC                 CHAR(80);                                 SMI00130
 DCL 1 T1 DEF SREC,                                                 SMI00140
      2 SAUP              CHAR(5),                                  SMI00150
      2 CD                CHAR(1),                                  SMI00160
      2 NUM               CHAR(2),                                  SMI00170
      2 DATA              CHAR(67),                                 SMI00180
      2 TAB              CHAR(1),                                   SMI00190
      2 TMON             CHAR(2),                                   SMI00200
      2 F1               CHAR(1),                                   SMI00210
      2 CHECK            CHAR(1);                                   SMI00220
 DCL PIC7                 PIC'------9',                             SMI00230
     PIC8                 PIC'(8)9',                                SMI00240
     PIC5                 PIC'99999';                               SMI00250
 DCL 1 OREC,                                                        SMI00260
      2 EKEY              CHAR(8),                                  SMI00270
      2 TDATA             CHAR(40),                                 SMI00280
      2 FA                CHAR(2);                                  SMI00290
 DCL 1 OD1 DEF SREC POS(9),                                         SMI00300
      2 OINDUST            CHAR(5),                                 SMI00310
      2 OSIZE              CHAR(1),                                 SMI00320
      2 OYUGO1             CHAR(1);                                 SMI00330
 DCL 1 CD1 DEF TDATA,                                               SMI00340
      2 INDUST            CHAR(5),                                  SMI00350
      2 SIZE              CHAR(1),                                  SMI00360
      2 YUGO1             CHAR(1),                                  SMI00370
      2 TBL               CHAR(1),                                  SMI00380
      2 MON               PIC'99';                                  SMI00390
 DCL 1 CD2 DEF TDATA,                                               SMI00400
      2 PUM               CHAR(5),                                  SMI00410
      2 YUGO2             CHAR(1),                                  SMI00420
      2 DAT2(7)           FIXED BIN(31),                            SMI00430
      2 WHY(2)            CHAR(1);                                  SMI00440
 DCL 1 CD3 DEF TDATA,                                               SMI00450
      2 WON               CHAR(4),                                  SMI00460
      2 YUGO3             CHAR(1),                                  SMI00470
      2 GU                CHAR(1),                                  SMI00480
      2 DAT3(2)           FIXED BIN(31);                            SMI00490
 DCL DAT4(10)             FIXED BIN(31)  DEF TDATA;                 SMI00500
 DCL DAT5(7)              FIXED BIN(31)  DEF TDATA;                 SMI00510
 DCL 1 CD6 DEF TDATA,                                               SMI00520
      2 CPUM              CHAR(5),                                  SMI00530
      2 YUGO6             CHAR(1),                                  SMI00540
      2 DAT6(2)           FIXED BIN(31);                            SMI00550
```

```
 DCL RETURN_CODE FIXED BIN(15);                                          SMI00560
 DCL SAVSA       CHAR(5);                                                SMI00570
 GET SKIP EDIT(CMONTH)(P'99');                                           SMI00580
 ON KEY(EDIT3) BEGIN;                                                    SMI00590
   IF ONCODE=52 THEN PUT SKIP EDIT('DATA DUPLICATE : ',EKEY)(A,A(8));    SMI00600
 END;                                                                    SMI00610
 OPEN FILE(TAIN);                                                        SMI00620
 OPEN FILE(EDIT3);                                                       SMI00630
 ON ENDFILE(TAIN) RTYPE='0'B;                                            SMI00640
  READ FILE(TAIN) INTO(TREC);                                            SMI00650
  CALL PLISRTD(' SORT FIELDS=(1,8,CH,A) ',                               SMI00660
   ' RECORD TYPE=F,LENGTH=(80) ', 45000, RETURN_CODE, E15X, E35X);       SMI00670
  CALL PLIRETC(RETURN_CODE);                                             SMI00680
 E15X: PROC RETURNS(CHAR(80));                                           SMI00690
  DO WHILE(RTYPE);                                                       SMI00700
    SREC=TREC;                                                           SMI00710
    CALL KEYCHEK;                                                        SMI00720
    READ FILE(TAIN) INTO(TREC);                                          SMI00730
    IF NOER THEN DO ;                                                    SMI00740
      CALL PLIRETC(12);                                                  SMI00750
      RETURN(SREC);                                                      SMI00760
    END;                                                                 SMI00770
    NOER='1'B;                                                           SMI00780
  END;                                                                   SMI00790
  CALL PLIRETC(8);                                                       SMI00800
 END E15X;                                                               SMI00810
 E35X: PROC(SRTREC) ;                                                    SMI00820
 DCL SRTREC        CHAR(80) ;                                            SMI00830
  SREC=SRTREC;                                                           SMI00840
  TDATA=' '; FA=' ';                                                     SMI00850
  CALL CDCHECK;                                                          SMI00860
  EKEY=SUBSTR(SREC,1,8);                                                 SMI00870
  IF CD='1' | CD='4' | CD='5' THEN SUBSTR(EKEY,7,2)='00';                SMI00880
  WRITE FILE(EDIT3) FROM(OREC) KEYFROM(EKEY);                            SMI00890
  TDATA=' ';                                                             SMI00900
  CALL PLIRETC(4);                                                       SMI00910
 END E35X;                                                               SMI00920
 KEYCHEK: PROC;                                                          SMI00930
  SUBSTR(SREC,1,8)=TRANSLATE(SUBSTR(SREC,1,8),'0',' ');                  SMI00940
  IF CD='1' | CD='4' | CD='5' THEN NUM='00';                             SMI00950
  IF CD='4' THEN SREC=SUBSTR(SREC,1,13) || (5)' ' || SUBSTR(SREC,14,28) SMI00960
    || (8)' ' || SUBSTR(SREC,42,21) || SUBSTR(SREC,76,5);                SMI00970
  IF CD='5' THEN SREC=SUBSTR(SREC,1,16) || (8)' ' || SUBSTR(SREC,17,51) SMI00980
    || SUBSTR(SREC,76,5);                                                SMI00990
  DO I=1 TO 8;                                                           SMI01000
    IF SUBSTR(SREC,I,1)<'0' | SUBSTR(SREC,I,1)>'9' THEN DO;              SMI01010
      NOER='0'B; RETURN;  END;                                           SMI01020
  END;                                                                   SMI01030
  IF TMON='  ' THEN TMON='00';                                           SMI01040
  IF SUBSTR(TMON,2,1)=' ' THEN MON=SUBSTR(TMON,1,1);                     SMI01050
  ELSE MON=TMON;                                                         SMI01060
  TMON=MON;                                                              SMI01070
  IF MON¬=CMONTH THEN NOER='0'B;                                         SMI01080
  IF CD<'1' | CD>'6' THEN NOER='0'B;                                     SMI01090
  IF NOER='0'B THEN PUT SKIP EDIT(SREC,'KEY ERROR')(A(85),A);            SMI01100
```

```
 END KEYCHEK;                                                          SMI01110
 CDCHECK: PROC;                                                        SMI01120
  IF CD='1' THEN CALL CD1RTN;                                          SMI01130
  ELSE IF CD='2' THEN CALL CD2RTN;                                     SMI01140
  ELSE IF CD='3' THEN CALL CD3RTN;                                     SMI01150
  ELSE IF CD='4' THEN CALL CD4RTN;                                     SMI01160
  ELSE IF CD='5' THEN CALL CD5RTN;                                     SMI01170
  ELSE CALL CD6RTN;                                                    SMI01180
 END CDCHECK;                                                          SMI01190
 CD1RTN: PROC;                                                         SMI01200
  INDUST=OINDUST;                                                      SMI01210
  SIZE=OSIZE;                                                          SMI01220
  YUGO1=OYUGO1;                                                        SMI01230
  TBL=TAB;                                                             SMI01240
  MON=TMON;                                                            SMI01250
 END CD1RTN;                                                           SMI01260
 CD2RTN: PROC;                                                         SMI01270
  PUM=SUBSTR(DATA,1,5);                                                SMI01280
  YUGO2=SUBSTR(DATA,6,1);                                              SMI01290
  WHY(1)=SUBSTR(DATA,56,1);                                            SMI01300
  WHY(2)=SUBSTR(DATA,57,1);                                            SMI01310
  K=0;                                                                 SMI01320
  DO I=7,14,21,28,35,42,49;                                            SMI01330
    K=K+1;                                                             SMI01340
    IF SUBSTR(DATA,I,1)¬=' ' THEN CALL RJ7RTN;                         SMI01350
    ELSE DAT2(K)=0;                                                    SMI01360
  END;                                                                 SMI01370
 END CD2RTN;                                                           SMI01380
 CD3RTN: PROC;                                                         SMI01390
  WON=SUBSTR(DATA,1,4);                                                SMI01400
  YUGO3=SUBSTR(DATA,5,1);                                              SMI01410
  GU=SUBSTR(DATA,6,1);                                                 SMI01420
  K=0;                                                                 SMI01430
  DO I=7,14;                                                           SMI01440
    K=K+1;                                                             SMI01450
    IF SUBSTR(DATA,I,1)¬=' ' THEN CALL RJ7RTN;                         SMI01460
    ELSE DAT2(K)=0;                                                    SMI01470
  END;                                                                 SMI01480
 END CD3RTN;                                                           SMI01490
 CD4RTN: PROC;                                                         SMI01500
  K=0;                                                                 SMI01510
  DO I=1,6,11,16,21,26;                                                SMI01520
    K=K+1;                                                             SMI01530
    IF SUBSTR(DATA,I,1)¬=' ' THEN DO;                                  SMI01540
      J=INDEX(SUBSTR(DATA,I,5),' ');                                   SMI01550
      IF J¬=0 THEN DO;                                                 SMI01560
        CALL SUBNUM;                                                   SMI01570
        PIC5=SUBSTR(DATA,I,J-1);                                       SMI01580
        DAT4(K)=PIC5;                                                  SMI01590
      END;                                                             SMI01600
      ELSE DO;  J=6; CALL SUBNUM;                                      SMI01610
        DAT4(K)=SUBSTR(DATA,I,5); END;                                 SMI01620
    END;                                                               SMI01630
    ELSE DAT4(K)=0;                                                    SMI01640
  END;                                                                 SMI01650
```

```
  DO I=31,39,47,55;                                                    SMI01660
   K=K+1;                                                              SMI01670
   IF SUBSTR(DATA,I,1)¬=' ' THEN DO;                                   SMI01680
     J=INDEX(SUBSTR(DATA,I,8),' ');                                    SMI01690
     IF J¬=0 THEN DO;                                                  SMI01700
       CALL SUBNUM;                                                    SMI01710
       PIC8=SUBSTR(DATA,I,J-1);                                        SMI01720
       DAT4(K)=PIC8;                                                   SMI01730
     END;                                                              SMI01740
     ELSE DO; J=9; CALL SUBNUM;                                        SMI01750
       DAT4(K)=SUBSTR(DATA,I,8); END;                                  SMI01760
   END;                                                                SMI01770
   ELSE DAT4(K)=0;                                                     SMI01780
  END;                                                                 SMI01790
 END CD4RTN;                                                           SMI01800
 CD5RTN: PROC;                                                         SMI01810
  K=0;                                                                 SMI01820
  DO I=1,9,17,25,33,41,49;                                             SMI01830
   K=K+1;                                                              SMI01840
   IF SUBSTR(DATA,I,1)¬=' ' THEN DO;                                   SMI01850
     J=INDEX(SUBSTR(DATA,I,8),' ');                                    SMI01860
     IF J¬=0 THEN DO;                                                  SMI01870
       CALL SUBNUM;                                                    SMI01880
       PIC8=SUBSTR(DATA,I,J-1);                                        SMI01890
       DAT5(K)=PIC8;                                                   SMI01900
     END;                                                              SMI01910
     ELSE DO; J=9; CALL SUBNUM;                                        SMI01920
       DAT5(K)=SUBSTR(DATA,I,8); END;                                  SMI01930
   END;                                                                SMI01940
   ELSE DAT5(K)=0;                                                     SMI01950
  END;                                                                 SMI01960
 END CD5RTN;                                                           SMI01970
 CD6RTN: PROC;                                                         SMI01980
  CPUM=SUBSTR(DATA,1,5);                                               SMI01990
  YUGO6=SUBSTR(DATA,6,1);                                              SMI02000
  K=0;                                                                 SMI02010
  DO I=7,14;                                                           SMI02020
   K=K+1;                                                              SMI02030
   IF SUBSTR(DATA,I,1)¬=' ' THEN CALL RJ7RTN;                          SMI02040
   ELSE DAT2(K)=0;                                                     SMI02050
  END;                                                                 SMI02060
 END CD6RTN;                                                           SMI02070
 RJ7RTN: PROC ;                                                        SMI02080
  J=INDEX(SUBSTR(DATA,I,7),' ');                                       SMI02090
  IF J¬=0 THEN DO;                                                     SMI02100
   CALL SUBNUM;                                                        SMI02110
   PIC7=SUBSTR(DATA,I,J-1);                                            SMI02120
   DAT2(K)=PIC7;                                                       SMI02130
  END;                                                                 SMI02140
  ELSE DO; J=8; CALL SUBNUM;                                           SMI02150
   DAT2(K)=SUBSTR(DATA,I,7); END;                                      SMI02160
 END RJ7RTN;                                                           SMI02170
 SUBNUM: PROC;                                                         SMI02180
  IF SUBSTR(DATA,I,1)='-' THEN DO;                                     SMI02190
   IF SUBSTR(DATA,I+1,1)=' ' THEN SUBSTR(DATA,I,1)='0';                SMI02200
```

```
  END;                                                                  SMI02210
  ELSE IF (SUBSTR(DATA,I,1)<'0' | SUBSTR(DATA,I,1))                     SMI02220
   '9') THEN SUBSTR(DATA,I,1)='0';                                      SMI02230
  DO Q=1 TO J-1;                                                        SMI02240
    IF SUBSTR(DATA,I+Q,1)<'0' | SUBSTR(DATA,I+Q,1)>'9' THEN             SMI02250
     SUBSTR(DATA,I+Q,1)='0';                                            SMI02260
  END;                                                                  SMI02270
 END SUBNUM;                                                            SMI02280
 END UPDATE;                                                            SMI02290
/*                                                                      SMI02300
//LKED.SYSLMOD DD DSN=USER.LOADLIB(SMICR12),DISP=SHR                    SMI02310
//                                                                      SMI02320
```

SMICED66

EDITING
(CATAL PROGRAM)

```
//SMICED66 JOB CLASS=1,TYPRUN=HOLD                                    SMI00010
//   EXEC PLIFCL                                                      SMI00020
//PLI.SYSIN DD *                                                      SMI00030
* PROCESS GS,NEST,A,XREF,OPT(2);                                      SMI00040
 EDITING:PROC OPTIONS(MAIN);                                          SMI00050
 DCL ONCODE          BUILTIN;                                         SMI00060
 DCL DATE            BUILTIN;                                         SMI00070
 DCL JOSA   PIC'999';                                                 SMI00080
 DCL NUM   PIC'99';                                                   SMI00090
 DCL TOTAL   FIXED(7) INIT(0);                                        SMI00100
 DCL PG   FIXED  INIT(0);                                             SMI00110
 DCL OKREC       CHAR(10);                                            SMI00120
 DCL EDIT3 FILE RECORD INPUT;                                         SMI00130
 DCL EDIT2 FILE RECORD KEYED ENV(VSAM);                               SMI00140
 DCL PUMF FILE RECORD KEYED ENV(VSAM);                                SMI00150
 DCL TOUT FILE RECORD OUTPUT;                                         SMI00160
 DCL PREC CHAR(80);                                                   SMI00170
 DCL WWL  CHAR(2);                                                    SMI00180
 DCL PKEY CHAR(6);                                                    SMI00190
 DCL 1 TREC,                                                          SMI00200
      2 SAUP         CHAR(5),                                         SMI00210
      2 CD           PIC'9',                                          SMI00220
      2 NUMB         CHAR(2),                                         SMI00230
      2 DATA         CHAR(40),                                        SMI00240
      2 CHECK        CHAR(1),                                         SMI00250
      2 F1           CHAR(1),                                         SMI00260
      2 F333         CHAR(3);                                         SMI00270
 DCL OSAUP           CHAR(5);                                         SMI00280
 DCL BSAN CHAR(3) INIT('000');                                        SMI00290
 DCL RTYPE                BIT(1) INIT('1'B);                          SMI00300
 DCL ER              BIT(1) INIT('0'B);                               SMI00310
 DCL INV             CHAR(1) INIT('0');                               SMI00320
 DCL DAY(12) PIC'99' INIT(31,29,31,30,31,30,31,31,30,31,30,31);       SMI00330
 DCL 1 OD1 DEF DATA,                                                  SMI00340
      2 OINDUST       CHAR(5),                                        SMI00350
      2 OSIZE         CHAR(1),                                        SMI00360
      2 OYUGO1        CHAR(1),                                        SMI00370
      2 OTABLE        CHAR(1),                                        SMI00380
      2 OMONTH        PIC'99';                                        SMI00390
 DCL 1 OD2 DEF DATA,                                                  SMI00400
      2 OPUMNO        CHAR(5),                                        SMI00410
      2 OYUGO2        CHAR(1),                                        SMI00420
      2 ODATA2(7)     FIXED BIN(31),                                  SMI00430
      2 OWHY          CHAR(2);                                        SMI00440
 DCL 1 OD3 DEF DATA,                                                  SMI00450
      2 OWONNO        CHAR(4),                                        SMI00460
      2 OYUGO3        CHAR(1),                                        SMI00470
      2 OGUBUN        CHAR(1),                                        SMI00480
      2 ODATA3(2)     FIXED BIN(31);                                  SMI00490
 DCL ODATA4(10)  FIXED BIN(31) DEF DATA,                              SMI00500
     ODATA5(7)   FIXED BIN(31) DEF DATA;                              SMI00510
 DCL 1 OD6 DEF DATA,                                                  SMI00520
      2 OCPUMNO       CHAR(5),                                        SMI00530
      2 OYUGO6        CHAR(1),                                        SMI00540
      2 ODATA6(2)     FIXED BIN(31);                                  SMI00550
```

```
DCL 1 D1,                                                          SMI00560
      2 SNUM          CHAR(5),                                     SMI00570
      2 INDUST        CHAR(5),                                     SMI00580
      2 SIZE          CHAR(1),                                     SMI00590
      2 YUGO1         CHAR(1),                                     SMI00600
      2 TABLE         CHAR(1),                                     SMI00610
      2 MONTH         PIC'99',                                     SMI00620
      2 F222          CHAR(3);                                     SMI00630
DCL 1 D2(30),                                                      SMI00640
      2 PIL           PIC'99',                                     SMI00650
      2 PUMNO         CHAR(5),                                     SMI00660
      2 YUGO2         CHAR(1),                                     SMI00670
      2 DATA2(7)      FLOAT,                                       SMI00680
      2 WHY           CHAR(2);                                     SMI00690
DCL 1 D3(13),                                                      SMI00700
      2 WIL           PIC'99',                                     SMI00710
      2 WONNO         CHAR(4),                                     SMI00720
      2 YUGO3         CHAR(1),                                     SMI00730
      2 GUBUN         CHAR(1),                                     SMI00740
      2 DATA3(2)      FLOAT;                                       SMI00750
DCL DATA4(10)         FLOAT DEC(15),                               SMI00760
    DATA5(7)          FLOAT DEC(15);                               SMI00770
DCL 1 D6(17),                                                      SMI00780
      2 CIL           PIC'99',                                     SMI00790
      2 CPUMNO        CHAR(5),                                     SMI00800
      2 YUGO6         CHAR(1),                                     SMI00810
      2 DATA6(2)      FLOAT;                                       SMI00820
DCL CD2EB             FIXED,                                       SMI00830
    CD3EB             FIXED,                                       SMI00840
    CD4EB             FIXED,                                       SMI00850
    CD5EB             FIXED,                                       SMI00860
    CD6EB             FIXED;                                       SMI00870
DCL 1 ED1,                                                         SMI00880
      2 ESNUM         CHAR(5),                                     SMI00890
      2 EINDUST       CHAR(5),                                     SMI00900
      2 ESIZE         CHAR(4),                                     SMI00910
      2 EYUGO1        CHAR(4),                                     SMI00920
      2 ETABLE        CHAR(4),                                     SMI00930
      2 EMONTH        CHAR(4);                                     SMI00940
DCL 1 ED2(30),                                                     SMI00950
      2 EPIL          CHAR(4),                                     SMI00960
      2 EPUMNO        CHAR(5),                                     SMI00970
      2 EYUGO2        CHAR(3),                                     SMI00980
      2 EDATA2(7)     CHAR(5),                                     SMI00990
      2 EWHY          CHAR(4);                                     SMI01000
DCL 1 ED3(13),                                                     SMI01010
      2 EWIL          CHAR(4),                                     SMI01020
      2 EWONNO        CHAR(4),                                     SMI01030
      2 EYUGO3        CHAR(3),                                     SMI01040
      2 EGUBUN        CHAR(3),                                     SMI01050
      2 EDATA3(2)     CHAR(5);                                     SMI01060
DCL EDATA4(10)        CHAR(5),                                     SMI01070
    EDATA5(7)        CHAR(5);                                      SMI01080
DCL 1 ED6(17),                                                     SMI01090
      2 ECIL          CHAR(4),                                     SMI01100
```

```
     2 ECPUMNO      CHAR(5),                                          SMI01110
     2 EYUGO6       CHAR(3),                                          SMI01120
     2 EDATA6(2)    CHAR(5);                                          SMI01130
 DCL 1 KREC,                                                          SMI01140
     2 KSAUP         CHAR(5),                                         SMI01150
     2 KCD           PIC'9',                                          SMI01160
     2 KNUMB         CHAR(5),                                         SMI01170
     2 KDATA         CHAR(42);                                        SMI01180
 DCL 1 KD1 DEF KDATA,                                                 SMI01190
     2 KINDUST       CHAR(5),                                         SMI01200
     2 KSIZE         CHAR(1),                                         SMI01210
     2 KYUGO1        CHAR(1),                                         SMI01220
     2 KTABLE        CHAR(1),                                         SMI01230
     2 KMONTH        PIC'99';                                         SMI01240
 DCL 1 KD2 DEF KDATA,                                                 SMI01250
     2 KPUMNO        CHAR(2),                                         SMI01260
     2 KYUGO2        CHAR(1),                                         SMI01270
     2 KDATA2(7)     FIXED BIN(31),                                   SMI01280
     2 KWHY          CHAR(2);                                         SMI01290
 DCL 1 KD3 DEF KDATA,                                                 SMI01300
     2 KWONNO        CHAR(2),                                         SMI01310
     2 KYUGO3        CHAR(1),                                         SMI01320
     2 KDATA3(2)     FIXED BIN(31);                                   SMI01330
 DCL KDATA4(10)  FIXED BIN(31) DEF KDATA,                             SMI01340
    KDATA5(7)   FIXED BIN(31) DEF KDATA;                              SMI01350
 DCL 1 KD6 DEF KDATA,                                                 SMI01360
     2 KCPUMNO       CHAR(2),                                         SMI01370
     2 KYUGO6        CHAR(1),                                         SMI01380
     2 KDATA6(2)     FIXED BIN(31);                                   SMI01390
  DCL HAP FIXED(9) INIT(0);                                           SMI01400
  DCL OKCT         FIXED  INIT(0);                                    SMI01410
  DCL (ATABLE,ASIZE)  CHAR(1);                                        SMI01420
  DCL HDATA(7)     FLOAT;                                             SMI01430
  DCL ADATA(2)     FLOAT;                                             SMI01440
  DCL (ERDAT(50),QDATA4(10),RDATA4(10))  FIXED BIN(31);               SMI01450
  DCL (ERNO,LA,LB)  FLOAT;                                            SMI01460
  ON ENDFILE(EDIT3) GOTO REND;                                        SMI01470
  ON KEY(PUMF) BEGIN;                                                 SMI01480
    IF ONCODE=51 THEN INV='1'B;                                       SMI01490
  END;                                                                SMI01500
  ON KEY(EDIT2) BEGIN;                                                SMI01510
    IF ONCODE=51 THEN INV='1'B;                                       SMI01520
  END;                                                                SMI01530
  OPEN FILE(TOUT);                                                    SMI01540
  OPEN FILE(PUMF) DIRECT INPUT;                                       SMI01550
  OPEN FILE(EDIT3);                                                   SMI01560
  OPEN FILE(EDIT2) DIRECT INPUT;                                      SMI01570
  OPEN FILE(SYSPRINT) LINESIZE(132);                                  SMI01580
  CALL CLEAR;                                                         SMI01590
 R1: READ FILE(EDIT3) INTO(TREC);                                     SMI01600
  IF CHECK='1' THEN GOTO R1;                                          SMI01610
  IF BSAN ¬= F333 THEN PUT PAGE;                                      SMI01620
  BSAN=F333;                                                          SMI01630
  SNUM=SAUP;                                                          SMI01640
  F222=F333;                                                          SMI01650
```

```
RERE:  IF SNUM¬=SAUP THEN DO;                                          SMI01660
    CALL ERCHECK;                                                      SMI01670
    IF ER THEN DO;                                                     SMI01680
       CALL MSGPUT;                                                    SMI01690
       IF ERNO>0 THEN PUT SKIP(2) EDIT('ERROR DATA: ',(ERDAT(I)        SMI01700
        DO I=1 TO ERNO))(A(12),(ERNO) F(12));                          SMI01710
    END;                                                               SMI01720
    ELSE DO;                                                           SMI01730
      OKREC=SNUM || '1';                                               SMI01740
      WRITE FILE(TOUT) FROM(OKREC); OKCT=OKCT+1;                       SMI01750
    END;                                                               SMI01760
    CALL CLEAR;                                                        SMI01770
END;                                                                   SMI01780
IF CD=1 THEN DO;                                                       SMI01790
  SNUM=SAUP;                                                           SMI01800
  F222=F333;                                                           SMI01810
  INDUST=OINDUST;                                                      SMI01820
  SIZE=OSIZE;                                                          SMI01830
  YUGO1=OYUGO1;                                                        SMI01840
  TABLE=OTABLE;                                                        SMI01850
  MONTH=OMONTH;                                                        SMI01860
END;                                                                   SMI01870
ELSE IF CD=2 THEN DO;                                                  SMI01880
  IF CD2EB>29 THEN GOTO R2;                                            SMI01890
  CD2EB=CD2EB+1;                                                       SMI01900
  PIL(CD2EB)=NUMB; SNUM=SAUP;                                          SMI01910
  F222=F333;                                                           SMI01920
  PUMNO(CD2EB)=OPUMNO;                                                 SMI01930
  YUGO2(CD2EB)=OYUGO2;                                                 SMI01940
  DO I=1 TO 7;                                                         SMI01950
    DATA2(CD2EB,I)=ODATA2(I);                                          SMI01960
  END;                                                                 SMI01970
  WHY(CD2EB)=OWHY;                                                     SMI01980
END;                                                                   SMI01990
ELSE IF CD=3 THEN DO;                                                  SMI02000
  IF CD3EB>12 THEN GOTO R2;                                            SMI02010
  CD3EB=CD3EB+1;                                                       SMI02020
  WIL(CD3EB)=NUMB;  SNUM=SAUP;                                         SMI02030
  F222=F333;                                                           SMI02040
  WONNO(CD3EB)=OWONNO;                                                 SMI02050
  YUGO3(CD3EB)=OYUGO3;                                                 SMI02060
  GUBUN(CD3EB)=OGUBUN;                                                 SMI02070
  DO I=1,2;                                                            SMI02080
    DATA3(CD3EB,I)=ODATA3(I);                                          SMI02090
  END;                                                                 SMI02100
END;                                                                   SMI02110
ELSE IF CD=4 THEN DO;                                                  SMI02120
  CD4EB=1;                                                             SMI02130
  SNUM=SAUP;                                                           SMI02140
  F222=F333;                                                           SMI02150
  DO I=1 TO 10;                                                        SMI02160
    DATA4(I)=ODATA4(I);                                                SMI02170
  END;                                                                 SMI02180
END;                                                                   SMI02190
ELSE IF CD=5 THEN DO;                                                  SMI02200
```

```
    CD5EB=1;                                                          SMI02210
    SNUM=SAUP;                                                        SMI02220
    F222=F333;                                                        SMI02230
    DO I=1 TO 7;                                                      SMI02240
      DATA5(I)=ODATA5(I);                                             SMI02250
    END;                                                              SMI02260
  END;                                                                SMI02270
  ELSE IF CD=6 THEN DO;                                               SMI02280
    IF CD6EB>16 THEN GOTO R2;                                         SMI02290
    CD6EB=CD6EB+1;                                                    SMI02300
    CIL(CD6EB)=NUMB;  SNUM=SAUP;                                      SMI02310
    F222=F333;                                                        SMI02320
    CPUMNO(CD6EB)=OCPUMNO;                                            SMI02330
    YUGO6(CD6EB)=OYUGO6;                                              SMI02340
    DO I=1,2;                                                         SMI02350
      DATA6(CD6EB,I)=ODATA6(I);                                       SMI02360
    END;                                                              SMI02370
  END;                                                                SMI02380
 R2: READ FILE(EDIT3) INTO(TREC);                                     SMI02390
  IF CHECK='1' THEN GOTO R2;                                          SMI02400
  ELSE GOTO RERE;                                                     SMI02410
  REND:CALL ERCHECK;                                                  SMI02420
  IF ER THEN CALL MSGPUT;                                             SMI02430
  ELSE DO;                                                            SMI02440
     OKREC=SNUM || '1';                                               SMI02450
     WRITE FILE(TOUT) FROM(OKREC); OKCT=OKCT+1;                       SMI02460
  END;                                                                SMI02470
  PUT SKIP(3) EDIT('TOTAL TABLE =',TOTAL,'ERROR TABLE =',PG,          SMI02480
   'OK    TABLE =',OKCT)(3 (X(5),A,F(7)));                            SMI02490
 CLEAR: PROC;                                                         SMI02500
   PIL=0; WIL=0; CIL=0; DATA2=0; DATA3=0; DATA6=0;                    SMI02510
   PUMNO=' '; CPUMNO=' '; WONNO=' '; YUGO2=' '; YUGO3=' ';            SMI02520
   YUGO6=' '; WHY=' '; SNUM,INDUST,SIZE,YUGO1,TABLE=' ';              SMI02530
   STRING(ED1)=' '; STRING(ED2)=' '; STRING(ED3)=' '; STRING(ED6)=' ';SMI02540
   DATA4=0;  DATA5=0;  MONTH=0;                                       SMI02550
   CD2EB=0; CD3EB=0; CD6EB=0; CD4EB=0; CD5EB=0;                       SMI02560
   EDATA4=' ';  EDATA5=' ';  ERNO=0;                                  SMI02570
   ER='0'B;  HDATA=0;                                                 SMI02580
 END CLEAR;                                                           SMI02590
 MSGPUT: PROC;                                                        SMI02600
  PG=PG+1;                                                            SMI02610
  PUT SKIP(4) EDIT((10)'=','THE CURRENT INDUSTRIAL PRODUCTION SURVEY',SMI02620
   (10)'=','JOB EDIT DATE :',DATE,(10)'=','PAGE :',PG,(10)'=')        SMI02630
   (A(10),X(2),A,X(2),A(10),X(2),A,X(1),P'XXXXXX',X(2),A(10),         SMI02640
   X(2),A,X(1),F(4),X(2),A(10));                                      SMI02650
  PUT SKIP EDIT('I',MONTH,'I',EINDUST,'I',ESIZE,'I',EYUGO1,ETABLE)    SMI02660
   (A(14),P'99',X(8),A(15),A(20),4 A(3),X(20),A(4));                  SMI02670
  PUT SKIP EDIT('I  KEY CODE :',SNUM,'I  1 CARD',INDUST,'I',SIZE,'I', SMI02680
   YUGO1,'    SANUP= ',F222,TABLE)(A,X(1),A(10),A(10),                SMI02690
   X(5),A(20),4 A(3),A(11),A(3),X(6),A(4));                           SMI02700
  IF CD2EB+CD3EB+CD4EB+CD5EB+CD6EB=0 THEN RETURN;                     SMI02710
  IF CD2EB=0 THEN GOTO PSKIP;                                         SMI02720
  PUT SKIP EDIT((127)'-')(A(127));                                    SMI02730
  PUT SKIP EDIT(EPIL(1),EPUMNO(1),EYUGO2(1),(EDATA2(1,J) DO J=1 TO 7  SMI02740
   ),EWHY(1))                                                         SMI02750
```

```
  (X(10),A(8),A(7),A(3),7 (X(8),A(5)),X(5),A(2));                    SMI02760
 IF TABLE='1' THEN                                                   SMI02770
 PUT SKIP EDIT('  2 CARD',PIL(1),':',PUMNO(1),YUGO2(1),(DATA2(1,J) DO SMI02780
  J=1 TO 7),WHY(1))                                                  SMI02790
  (A(10),P'99',X(2),A(4),A(7),A(3),7 F(13),X(5),A(2));               SMI02800
 ELSE PUT SKIP EDIT('  2 CARD',PIL(1),':',PUMNO(1),YUGO2(1),         SMI02810
  (DATA2(1,J) DO J=1 TO 6),WHY(1))                                   SMI02820
  (A(10),P'99',X(2),A(4),A(7),A(3),6 F(13),X(18),A(2));              SMI02830
 DO I=2 TO CD2EB;                                                    SMI02840
   PUT SKIP EDIT(EPIL(I),EPUMNO(I),EYUGO2(I),(EDATA2(I,J) DO J=1 TO  SMI02850
    7),EWHY(I))                                                      SMI02860
    (X(10),A(8),A(7),A(3),7 (X(8),A(5)),X(5),A(2));                  SMI02870
   IF TABLE='1' THEN                                                 SMI02880
   PUT SKIP EDIT(PIL(I),':',PUMNO(I),YUGO2(I),(DATA2(I,J) DO J=1 TO  SMI02890
    7),WHY(I))(X(10),P'99',X(2),A(4),A(7),A(3),7 F(13),X(5),A(2));   SMI02900
   ELSE PUT SKIP EDIT(PIL(I),':',PUMNO(I),YUGO2(I),(DATA2(I,J) DO J=1 SMI02910
    TO 6),WHY(I))(X(10),P'99',X(2),A(4),A(7),A(3),6 F(13),X(18),A(2)); SMI02920
 END;                                                                SMI02930
 PSKIP: IF CD3EB=0 & CD4EB=0 & CD5EB=0 & CD6EB=0 THEN DO;            SMI02940
   ER='1'B; RETURN;  END;                                            SMI02950
        PUT SKIP EDIT((127)'-')(A(127));                             SMI02960
 PUT SKIP EDIT('3 CARD','I','4 CARD','I','5 CARD','I','6 CARD')      SMI02970
  (X(13),A(27),A(10),A(17),A(3),A(9),A(13),A(33));                   SMI02980
 DO I=1 TO 4;                                                        SMI02990
   IF CD3EB<I THEN PUT SKIP EDIT('I')(X(40),A(1));                   SMI03000
   ELSE PUT SKIP EDIT(EWIL(I),EWONNO(I),EYUGO3(I),EGUBUN(I),         SMI03010
    (EDATA3(I,J) DO J=1,2),'I')                                      SMI03020
    (X(2),A(7),A(6),2 A(3),2 (X(4),A(5)),X(1),A(1));                 SMI03030
   IF I=3 THEN PUT EDIT(EDATA4(4),'I','I')                           SMI03040
    (X(10),A(16),A(12),A(1));                                        SMI03050
   ELSE IF I=4 THEN PUT EDIT(EDATA4(5),EDATA4(10),'I',EDATA5(4),'I') SMI03060
    (X(10),A(10),A(6),A(6),A(6),A(1));                               SMI03070
   ELSE IF I=2 THEN PUT EDIT(EDATA4(3),EDATA4(9),'I',EDATA5(3),'I')  SMI03080
    (X(10),A(10),A(6),A(6),A(6),A(1));                               SMI03090
   ELSE PUT EDIT(EDATA4(I),EDATA4(I+6),'I',EDATA5(I),'I')            SMI03100
    (X(10),A(10),A(6),A(6),A(6),A(1));                               SMI03110
   IF CD6EB>=I THEN PUT EDIT(ECIL(I),ECPUMNO(I),EYUGO6(I),(EDATA6(I,J) SMI03120
    DO J=1,2))                                                       SMI03130
    (X(3),A(7),A(7),A(3),2 (X(4),A(5)));                             SMI03140
   IF CD3EB<I THEN PUT SKIP EDIT('I')(X(40),A(1));                   SMI03150
   ELSE PUT SKIP EDIT(WIL(I),':',WONNO(I),YUGO3(I),GUBUN(I),         SMI03160
    (DATA3(I,J) DO J=1,2),'I')                                       SMI03170
    (X(2),A(4),A(3),A(6),2 A(3),2 F(9),X(1),A(1));                   SMI03180
   IF I=3 THEN PUT EDIT ('(',I,')',DATA4(4),'I','I')                 SMI03190
    (X(3),A(1),F(1),A(3),F(7),X(11),A(12),A(1));                     SMI03200
   ELSE IF I=4 THEN PUT EDIT('(',I,')',DATA4(5),DATA4(10),'I',DATA5(4),SMI03210
    'I')(X(3),A(1),F(1),A(3),F(7),F(10),X(1),A(2),F(9),X(1),A(1));   SMI03220
   ELSE IF I=2 THEN                                                  SMI03230
   PUT EDIT('(',I,')',DATA4(3),DATA4(9),'I',DATA5(3),'I')            SMI03240
    (X(3),A(1),F(1),A(3),F(7),F(10),X(1),A(2),F(9),X(1),A(1));       SMI03250
   ELSE PUT EDIT('(',I,')',DATA4(I),DATA4(I+6),'I',DATA5(I),'I')     SMI03260
    (X(3),A(1),F(1),A(3),F(7),F(10),X(1),A(2),F(9),X(1),A(1));       SMI03270
   IF CD6EB>=I THEN PUT EDIT(CIL(I),':',CPUMNO(I),YUGO6(I),(DATA6(I,J) SMI03280
    DO J=1,2))                                                       SMI03290
    (X(3),A(4),A(3),A(7),A(3),2 F(9));                               SMI03300
```

```
  END;                                                                  SMI03310
  IF CD3EB<5 THEN PUT SKIP EDIT('I')(X(40),A(1));                       SMI03320
  ELSE PUT SKIP EDIT(EWIL(5),EWONNO(5),EYUGO3(5),EGUBUN(5),             SMI03330
   (EDATA3(5,J) DO J=1,2),'I')                                          SMI03340
   (X(2),A(7),A(6),2 A(3),2 (X(4),A(5)),X(1),A(1));                     SMI03350
  PUT EDIT(EDATA4(6),'I',EDATA5(5),'I')                                 SMI03360
   (X(10),A(16),A(6),A(6),A(1));                                        SMI03370
  IF CD6EB>=5 THEN PUT EDIT(ECIL(5),ECPUMNO(5),EYUGO6(5),(EDATA6(5,J)   SMI03380
   DO J=1,2))                                                           SMI03390
   (X(3),A(7),A(7),A(3),2 A(9));                                        SMI03400
  IF CD3EB<5 THEN PUT SKIP EDIT('I')(X(40),A(1));                       SMI03410
  ELSE PUT SKIP EDIT(WIL(5),':',WONNO(5),YUGO3(5),GUBUN(5),             SMI03420
   (DATA3(5,J) DO J=1,2),'I')                                           SMI03430
   (X(2),A(4),A(3),A(6),2 A(3),2 F(9),X(1),A(1));                       SMI03440
  PUT EDIT(DATA4(6),(10)'-',DATA5(5),'I')                               SMI03450
   (X(7),F(8),X(1),A(10),X(3),F(8),X(1),A(1));                          SMI03460
  IF CD6EB>=5 THEN PUT EDIT(CIL(5),':',CPUMNO(5),YUGO6(5),(DATA6(5,J)   SMI03470
   DO J=1,2))                                                           SMI03480
   (X(3),A(4),A(3),A(7),A(3),2 F(9));                                   SMI03490
  IF CD3EB<6 THEN PUT SKIP EDIT('I')(X(40),A(1));                       SMI03500
  ELSE PUT SKIP EDIT(EWIL(6),EWONNO(6),EYUGO3(6),EGUBUN(6),             SMI03510
   (EDATA3(6,J) DO J=1,2),'I')                                          SMI03520
   (X(2),A(7),A(6),2 A(3),2 A(9),X(1),A(1));                            SMI03530
  PUT EDIT('I',(EDATA5(J) DO J=6,7),'I')                                SMI03540
    (X(16),A(3),2 (X(4),A(5)),X(1),A(1));                               SMI03550
  IF CD6EB>=6 THEN PUT EDIT(ECIL(6),ECPUMNO(6),EYUGO6(6),(EDATA6(6,J)   SMI03560
   DO J=1,2))                                                           SMI03570
   (X(3),2 A(7),A(3),2 (X(4),A(5)));                                    SMI03580
  IF CD3EB<6 THEN PUT SKIP EDIT('I')(X(40),A(1));                       SMI03590
  ELSE PUT SKIP EDIT(WIL(6),':',WONNO(6),YUGO3(6),GUBUN(6),             SMI03600
   (DATA3(6,J) DO J=1,2),'I')                                           SMI03610
   (X(2),A(4),A(3),A(6),2 A(3),2 F(9),X(1),A(1));                       SMI03620
  PUT EDIT((17)'-',(DATA5(J) DO J=6,7),'I')                             SMI03630
   (A(17),X(2),2 F(9),X(1),A(1));                                       SMI03640
  IF CD6EB>=6 THEN PUT EDIT(CIL(6),':',CPUMNO(6),YUGO6(6),(DATA6(6,J)   SMI03650
   DO J=1,2))                                                           SMI03660
   (X(3),A(4),A(3),A(7),A(3),2 F(9));                                   SMI03670
  IF CD6EB>CD3EB THEN K=CD6EB; ELSE K=CD3EB;                            SMI03680
  DO I=7 TO K;                                                          SMI03690
    IF CD3EB<I THEN PUT SKIP EDIT('I','I')(X(40),A(39),A(1));           SMI03700
    ELSE PUT SKIP EDIT(EWIL(I),EWONNO(I),EYUGO3(I),EGUBUN(I),           SMI03710
     (EDATA3(I,J) DO J=1,2),'I','I')                                    SMI03720
     (X(2),A(7),A(6),2 A(3),2 A(9),X(1),A(39),A(1));                    SMI03730
    IF CD6EB>=I THEN PUT EDIT(ECIL(I),ECPUMNO(I),EYUGO6(I),(EDATA6(I,J) SMI03740
     DO J=1,2))                                                         SMI03750
     (X(3),2 A(7),A(3),2 A(9));                                         SMI03760
    IF CD3EB<I THEN PUT SKIP EDIT('I','I')(X(40),A(39),A(1));           SMI03770
    ELSE PUT SKIP EDIT(WIL(I),':',WONNO(I),YUGO3(I),GUBUN(I),           SMI03780
     (DATA3(I,J) DO J=1,2),'I','I')                                     SMI03790
     (X(2),A(4),A(3),A(6),2 A(3),2 F(9),X(1),A(39),A(1));               SMI03800
    IF CD6EB>=I THEN PUT EDIT(CIL(I),':',CPUMNO(I),YUGO6(I),(DATA6(I,J) SMI03810
     DO J=1,2))                                                         SMI03820
     (X(3),A(4),A(3),A(7),A(3),2 F(9));                                 SMI03830
  END;                                                                  SMI03840
 END MSGPUT;                                                            SMI03850
```

```
  (NOZERODIVIDE):                                                       SMI03860
 ERCHECK: PROC;                                                         SMI03870
  TOTAL=TOTAL+1;                                                        SMI03880
  IF YUGO1¬=' ' THEN ER='1'B;                                           SMI03890
  IF CD2EB=0 THEN DO;                                                   SMI03900
    IF CD6EB=0 THEN DO; EYUGO1='**'; ER='1'B; RETURN; END;              SMI03910
    ELSE ER='1'B;                                                       SMI03920
  END;                                                                  SMI03930
  ELSE IF CD4EB=0 | CD5EB=0 THEN DO; EYUGO1='**'; ER='1'B; END;         SMI03940
  IF YUGO1='1' | YUGO1='4' THEN DO;                                     SMI03950
  IF SIZE='1' THEN DO;                                                  SMI03960
    IF DATA4(5)<200 THEN DO; ESIZE='**'; ER='1'B; END;                  SMI03970
  END ;                                                                 SMI03980
  ELSE IF SIZE='2' THEN DO;                                             SMI03990
    IF DATA4(5)>400 THEN DO; ESIZE='**'; ER='1'B; END;                  SMI04000
  END;                                                                  SMI04010
  ELSE DO;  ESIZE='**'; ER='1'B; END;                                   SMI04020
  IF TABLE='1' THEN DO;                                                 SMI04030
    IF SUBSTR(INDUST,1,3)='322' | SUBSTR(INDUST,1,4)='3520' THEN DO;    SMI04040
      ETABLE='**'; ER='1'B; END;                                        SMI04050
  END;                                                                  SMI04060
  ELSE IF TABLE='2' THEN DO;                                            SMI04070
    IF SUBSTR(INDUST,1,3)¬='322' & SUBSTR(INDUST,1,4)¬='3520' THEN DO;  SMI04080
      ETABLE='**';  ER='1'B;  END;                                      SMI04090
  END;                                                                  SMI04100
  ELSE DO; ETABLE='**';  ER='1'B;  END;                                 SMI04110
  ATABLE=TABLE; ASIZE=SIZE;                                             SMI04120
  END;                                                                  SMI04130
 ELSE DO;                                                               SMI04140
    INV='0'B;                                                           SMI04150
    READ FILE(EDIT2) INTO(KREC) KEY(SNUM || '1');                       SMI04160
    IF INV THEN DO;                                                     SMI04170
      EYUGO1='$$'; ER='1'B; RETURN;                                     SMI04180
    END;                                                                SMI04190
    ELSE DO;                                                            SMI04200
      IF SIZE¬=KSIZE THEN DO;                                           SMI04210
        ESIZE='$$'; ER='1'B;                                            SMI04220
      END;                                                              SMI04230
      IF TABLE¬=KTABLE THEN DO;                                         SMI04240
        ETABLE='$$'; ER='1'B;                                           SMI04250
      END;                                                              SMI04260
    END ;  ATABLE=KTABLE;  ASIZE=KSIZE;                                 SMI04270
 END;                                                                   SMI04280
  J=0;                                                                  SMI04290
  DO I=1 TO CD2EB;                                                      SMI04300
      J=J+1;                                                            SMI04310
      IF PIL(I)=99 THEN;                                                SMI04320
      ELSE IF PIL(I)>30 THEN DO; EPIL(I)='**'; ER='1'B; END;            SMI04330
      ELSE IF PIL(I)¬=J THEN DO;                                        SMI04340
        EPIL(I)='**';  J=PIL(I);   ER='1'B;                             SMI04350
      END;                                                              SMI04360
    DO K=1 TO I-1;                                                      SMI04370
      IF PUMNO(K)=PUMNO(I) THEN DO; EPUMNO(I) ='*****';  ER='1'B; END;  SMI04380
    END;                                                                SMI04390
    IF (YUGO2(I)='2' | YUGO2(I)='3' | YUGO2(I)='5') & DATA2(I,2)        SMI04400
```

```
 +DATA2(I,3)+DATA2(I,4)+DATA2(I,5)¬=0  THEN DO;                          SMI04410
  EYUGO2(I)='**';  ER='1'B; GOTO SK2;                                    SMI04420
END;                                                                     SMI04430
IF ATABLE='1' & (DATA2(I,1)+DATA2(I,2))¬=(DATA2(I,3)+DATA2(I,4)+         SMI04440
 DATA2(I,5)-DATA2(I,6)+DATA2(I,7)) THEN DO;                              SMI04450
  EDATA2(I,7)=(5)'*';   ER='1'B; END;                                    SMI04460
IF ATABLE='2' & (DATA2(I,1)+DATA2(I,2))¬=(DATA2(I,3)+DATA2(I,4)-         SMI04470
 DATA2(I,5)+DATA2(I,6)) THEN DO;                                         SMI04480
  EDATA2(I,6)=(5)'*';  ER='1'B; END;                                     SMI04490
IF PIL(I)=99 THEN DO K=1 TO 7;                                           SMI04500
  IF HDATA(K)¬=DATA2(I,K) THEN DO;                                       SMI04510
    EDATA2(I,K)=(5)'*'; ER='1'B;  END;                                   SMI04520
END;                                                                     SMI04530
IF PIL(CD2EB)=99 THEN HDATA(*)=HDATA(*)+DATA2(I,*);                      SMI04540
IF YUGO2(I)='1' | YUGO2(I)='4' THEN GOTO SK2;                            SMI04550
IF SUBSTR(EPUMNO(I),1,1)='*' THEN GOTO SK2;                              SMI04560
INV='0'B;                                                                SMI04570
READ FILE(EDIT2) INTO(KREC) KEY(SNUM || '2' || PUMNO(I));                SMI04580
IF INV THEN DO;                                                          SMI04590
  EYUGO2(I)='$$'; ER='1'B;                                               SMI04600
END;                                                                     SMI04610
ELSE DO;                                                                 SMI04620
  IF YUGO2(I)¬=' ' THEN GOTO SK2;                                        SMI04630
  IF ASIZE='1' THEN DO; LA=0.5; LB=5; END;                               SMI04640
  ELSE DO; LA=0.2; LB=10; END;                                           SMI04650
  IF ATABLE='2' THEN K=3; ELSE K=4;                                      SMI04660
  IF DATA2(I,1)¬=KDATA2(K+3) THEN DO;                                    SMI04670
  - EDATA2(I,1)=(5)'$'; ER='1'B;                                         SMI04680
    ERNO=ERNO+1; ERDAT(ERNO)=KDATA2(K+3); END;                           SMI04690
  IF KDATA2(2)=0 THEN DO;                                                SMI04700
    IF DATA2(I,2)=0 THEN GOTO K2;                                        SMI04710
    ADATA(2)=1;   END;                                                   SMI04720
  ELSE ADATA(2)=KDATA2(2);                                               SMI04730
  ADATA(1)=DATA2(I,2);   F1=SUBSTR(WHY(I),1,1);                          SMI04740
  IF F1¬=' ' THEN DO;                                                    SMI04750
    IF ADATA(1)/ADATA(2)>1 & (F1='2' | F1='4' | F1='6' | F1='8')         SMI04760
    THEN DO; ER='1'B; SUBSTR(EWHY(I),1,1)='$'; END;                      SMI04770
    ELSE IF ADATA(1)/ADATA(2)<1 & (F1='1' | F1='3' | F1='5' |            SMI04780
    F1='7' | F1='9') THEN DO; ER='1'B; SUBSTR(EWHY(I),1,1)='$'; END;SMI04790
  END;  F1=SUBSTR(WHY(I),2,1);                                           SMI04800
  IF ADATA(1)/ADATA(2)<LA | ADATA(1)/ADATA(2)>LB                         SMI04810
   THEN DO; EDATA2(I,2)=(5)'$'; ER='1'B;                                 SMI04820
    ERNO=ERNO+1; ERDAT(ERNO)=KDATA2(2); END;                             SMI04830
K2: IF KDATA2(K)+KDATA2(K+1)=0 THEN DO;                                  SMI04840
    IF DATA2(I,K)+DATA2(I,K+1)=0 THEN GOTO SK2;                          SMI04850
    ADATA(2)=1;   END;                                                   SMI04860
  ELSE ADATA(2)=KDATA2(K)+KDATA2(K+1);                                   SMI04870
  ADATA(1)=DATA2(I,K)+DATA2(I,K+1);                                      SMI04880
  IF F1¬=' ' THEN DO;                                                    SMI04890
    IF ADATA(1)/ADATA(2)>1 & (F1='2' | F1='4' | F1='6' | F1='8')         SMI04900
    THEN DO; ER='1'B; SUBSTR(EWHY(I),2,1)='$'; END;                      SMI04910
    ELSE IF ADATA(1)/ADATA(2)<1 & (F1='1' | F1='3' | F1='5' |            SMI04920
    F1='7' | F1='9') THEN DO; ER='1'B; SUBSTR(EWHY(I),2,1)='$'; END;SMI04930
  END;                                                                   SMI04940
  IF ADATA(1)/ADATA(2)<LA | ADATA(1)/ADATA(2)>LB THEN                    SMI04950
```

```
      DO; EDATA2(I,K)=(5)'$'; ER='1'B;                                  SMI04960
          ERNO=ERNO+1; ERDAT(ERNO)=KDATA2(K)+KDATA2(K+1); END;          SMI04970
    END;                                                                SMI04980
  SK2: END;                                                             SMI04990
  J=0;                                                                  SMI05000
  DO I=1 TO CD3EB;                                                      SMI05010
      J=J+1;                                                            SMI05020
      IF WIL(I)<51 THEN  DO;                                            SMI05030
        IF WIL(I)>13 THEN DO; EWIL(I)='**'; ER='1'B; END;               SMI05040
        ELSE IF WIL(I)¬=J THEN DO;                                      SMI05050
          EWIL(I)='**'; J=WIL(I);  ER='1'B;                             SMI05060
        END;                                                            SMI05070
      END;                                                              SMI05080
      ELSE IF WIL(I)>57 THEN DO; EWIL(I)='**';  ER='1'B; END;           SMI05090
      ELSE DO;                                                          SMI05100
        WWL=WIL(I);                                                     SMI05110
        IF SUBSTR(WONNO(I),2,1)¬='0' | SUBSTR(WONNO(I),1,1)¬='9' |      SMI05120
         SUBSTR(WWL,2,1)¬=SUBSTR(WONNO(I),3,1)THEN DO;                  SMI05130
          ER='1'B; EWIL(I)=(2)'*';  END;                                SMI05140
      END;                                                              SMI05150
    PKEY='3' || WONNO(I);  INV='0'B;                                    SMI05160
    READ FILE(PUMF) INTO(PREC) KEY(PKEY);                               SMI05170
    IF INV THEN DO; ER='1'B; INV='0'B; EWONNO(I)='****'; END;           SMI05180
    IF GUBUN(I)¬='1' & GUBUN(I)¬='2' THEN DO;                           SMI05190
      EGUBUN(I)='**'; ER='1'B; END;                                     SMI05200
    ELSE IF WIL(I)=57 & GUBUN(I)='2' THEN DO;                           SMI05210
      EGUBUN(I)='**'; ER='1'B; END;                                     SMI05220
    ELSE DO K=1 TO I-1;                                                 SMI05230
      IF WONNO(K) || GUBUN(K)=WONNO(I) || GUBUN(I) THEN DO;             SMI05240
        EWONNO(I)='****';  ER='1'B; END;                                SMI05250
    END;                                                                SMI05260
    IF YUGO3(I)='2' | YUGO3(I)='3' | YUGO3(I)='5' THEN DO;              SMI05270
      IF DATA3(I,1)¬=0 THEN DO; EYUGO3(I)='**'; ER='1'B; END;           SMI05280
      ELSE IF DATA3(I,2)¬=0 THEN DO; EYUGO3(I)='**'; ER='1'B; END;      SMI05290
    END;    /*************                                             SMI05300
    IF YUGO3(I)='1' | YUGO3(I)='4' THEN GOTO SK3;                       SMI05310
    INV='0'B;                                                           SMI05320
    READ FILE(EDIT2) INTO(KREC) KEY(SNUM || '3' || WONNO(I) ||GUBUN(I));SMI05330
    IF INV THEN DO;                                                     SMI05340
      EYUGO3(I)='$$'; ER='1'B;                                          SMI05350
    END;                                                                SMI05360
    ELSE DO;                                                            SMI05370
      IF YUGO3(I)>'0' & YUGO3(I)<'6' THEN GOTO SK3;                     SMI05380
      IF DATA4(5)>199 THEN DO; LA=0.5; LB=2; END;                       SMI05390
      ELSE IF DATA4(5)>99 THEN DO; LA=0.3; LB=2.5; END;                 SMI05400
      ELSE DO; LA=0.2; LB=3; END;                                       SMI05410
      ADATA(1)=DATA3(I,1); ADATA(2)=KDATA3(1);                          SMI05420
      IF ADATA(1)/ADATA(2)<LA | ADATA(1)/ADATA(2)>LB                    SMI05430
       THEN DO; EDATA3(I,1)=(5)'$'; ER='1'B;                            SMI05440
        ERNO=ERNO+1; ERDAT(ERNO)=KDATA3(1); END;                        SMI05450
      ADATA(1)=DATA3(I,2); ADATA(2)=KDATA3(2);                          SMI05460
      IF ADATA(1)/ADATA(2)<LA | ADATA(1)/ADATA(2)>LB                    SMI05470
       THEN DO; EDATA3(I,2)=(5)'$'; ER='1'B;                            SMI05480
        ERNO=ERNO+1; ERDAT(ERNO)=KDATA3(2); END;                        SMI05490
    END;          *************/                                        SMI05500
```

```
 SK3: END;                                                              SMI05510
 IF  DATA4(5)=0 THEN DO;                                                SMI05520
   EDATA4(5)=(5)'*'; ER='1'B;  CALL ER6RTN; RETURN;  END;               SMI05530
 IF DATA4(5)¬=(DATA4(1)+DATA4(2)+DATA4(3)+DATA4(4)) THEN DO;            SMI05540
   EDATA4(5)=(5)'*'; ER='1'B; END;                                      SMI05550
 IF DATA4(10)¬=(DATA4(7)+DATA4(8)+DATA4(9)) THEN DO;                    SMI05560
   EDATA4(10)=(5)'*'; ER='1'B; END;                                     SMI05570
 IF DATA5(4)¬=(DATA5(1)+DATA5(2)+DATA5(3)) THEN DO;                     SMI05580
   EDATA5(4)=(5)'*'; ER='1'B; END;                                      SMI05590
 IF DATA4(1)+DATA4(2)¬=0 THEN DO;                                       SMI05600
   IF (DATA5(6)/(DATA4(1)+DATA4(2))))>DAY(MONTH) THEN DO;               SMI05610
     EDATA5(6)=(5)'*'; ER='1'B; END;                                    SMI05620
   ELSE IF (YUGO1¬='1' & YUGO1¬='2' & YUGO1¬='3') & (DATA5(6)/(DATA4(1)SMI05630
     +DATA4(2))) <15 THEN DO;                                           SMI05640
     EDATA5(6)=(5)'*'; ER='1'B; END;                                    SMI05650
 END;                                                                   SMI05660
 IF (DATA5(7)>DATA5(6)*13) | (DATA5(7)<DATA5(6)*4) THEN  DO;            SMI05670
  EDATA5(7)=(5)'*'; ER='1'B; END;                                       SMI05680
 DO I=1,2,3;                                                            SMI05690
   IF DATA4(I+6)¬=0 & DATA4(I+6)<=DATA5(I) THEN DO;                     SMI05700
     EDATA5(I)=(5)'*'; ER='1'B; END;                                    SMI05710
   IF DATA4(I)¬=0 THEN DO;                                              SMI05720
 IF I¬=2 & ((DATA4(I+6)-DATA5(I))/DATA4(I)>1000 | (DATA4(I+6)-DATA5(I))SMI05730
     /DATA4(I)<50) THEN DO; EDATA4(I+6)=(5)'*'; ER='1'B; END;           SMI05740
   END;                                                                 SMI05750
 END;                                                                   SMI05760
 IF YUGO1¬='3' &                                                        SMI05770
    DATA4(4)=0 & DATA4(3)=0 THEN DO; EDATA4(3)='*****'; ER='1'B; END;   SMI05780
 IF YUGO1>'0' & YUGO1<'6' THEN GOTO SK4;                                SMI05790
 INV='0'B;                                                              SMI05800
 READ FILE(EDIT2) INTO(KREC) KEY(SNUM || '4');                          SMI05810
 IF INV THEN DO;                                                        SMI05820
   EYUGO1='$$'; ER='1'B;                                                SMI05830
 END;                                                                   SMI05840
 ELSE DO;                                                               SMI05850
   IF KDATA4(5)+DATA4(6)¬=DATA4(5)+DATA5(5) THEN DO;                    SMI05860
     EDATA4(6)=(5)'*'; ER='1'B; END;                                    SMI05870
   IF KDATA4(5)=0 THEN GOTO SK4;                                        SMI05880
   IF KDATA4(5)>199 THEN DO; LA=0.8; LB=1.2; END;                       SMI05890
   ELSE IF KDATA4(5)>99 THEN DO; LA=0.5; LB=1.5; END;                   SMI05900
   ELSE IF KDATA4(5)>49 THEN DO; LA=0.3; LB=2; END;                     SMI05910
   ELSE DO; LA=0.1; LB=3; END;                                          SMI05920
   IF KDATA4(1)=0 THEN GOTO K3;                                         SMI05930
   HDATA(1)=DATA4(1); HDATA(2)=KDATA4(1);                               SMI05940
   IF HDATA(1)/HDATA(2)<LA | HDATA(1)/HDATA(2)>LB THEN DO;              SMI05950
     EDATA4(1)=(5)'$'; ER='1'B;                                         SMI05960
     ERNO=ERNO+1; ERDAT(ERNO)=KDATA4(1); END;                           SMI05970
  K3: IF KDATA4(3)=0 THEN GOTO K4;                                      SMI05980
   HDATA(1)=DATA4(3); HDATA(2)=KDATA4(3);                               SMI05990
   IF HDATA(1)/HDATA(2)<LA | HDATA(1)/HDATA(2)>LB THEN DO;              SMI06000
     EDATA4(3)=(5)'$'; ER='1'B;                                         SMI06010
     ERNO=ERNO+1; ERDAT(ERNO)=KDATA4(3); END;                           SMI06020
  K4: QDATA4=DATA4;   RDATA4=KDATA4;                                    SMI06030
 END;                                                                   SMI06040
 INV='0'B;                                                              SMI06050
```

```
  READ FILE(EDIT2) INTO(KREC) KEY(SNUM || '5');                          SMI06060
  IF INV THEN DO;                                                        SMI06070
    EYUGO1='$$'; ER='1'B; END;                                           SMI06080
  ELSE DO;                                                               SMI06090
    IF KDATA5(6)=0 THEN GOTO K5;                                         SMI06100
    HDATA(1)=DATA5(6);  HDATA(2)=KDATA5(6);                              SMI06110
    IF HDATA(1)/HDATA(2)<LA | HDATA(1)/HDATA(2)>LB THEN DO;              SMI06120
      EDATA5(6)=(5)'$'; ER='1'B;                                         SMI06130
      ERNO=ERNO+1; ERDAT(ERNO)=KDATA5(6); END;                           SMI06140
   K5: HDATA(1)=QDATA4(7)+QDATA4(8)-DATA5(1)-DATA5(2);                   SMI06150
    HDATA(2)=RDATA4(7)+RDATA4(8)-KDATA5(1)-KDATA4(2);                    SMI06160
    IF HDATA(2)=0 THEN GOTO SK4;                                         SMI06170
    LA=HDATA(1)/HDATA(2);                                                SMI06180
    HDATA(1)=QDATA4(9)-DATA5(3);                                         SMI06190
    HDATA(2)=RDATA4(9)-KDATA5(3);                                        SMI06200
    LB=HDATA(1)/HDATA(2);                                                SMI06210
    IF LA<0.5 | LA>2 THEN DO; EDATA4(7)=(5)'$$'; ER='1'B;                SMI06220
      ERNO=ERNO+1; ERDAT(ERNO)=RDATA4(7); END;                           SMI06230
    IF LB<0.5 | LB>2 THEN DO; EDATA4(9)=(5)'$$'; ER='1'B;                SMI06240
      ERNO=ERNO+1; ERDAT(ERNO)=RDATA4(9); END;                           SMI06250
  END;                                                                   SMI06260
  SK4: CALL ER6RTN;                                                      SMI06270
 ER6RTN: PROC;                                                           SMI06280
  J=0;                                                                   SMI06290
  DO I=1 TO CD6EB;                                                       SMI06300
      J=J+1;                                                             SMI06310
      IF CIL(I)<51 THEN DO;                                              SMI06320
        IF CIL(I)>17 THEN DO; ECIL(I)='**'; ER='1'B; END;                SMI06330
        ELSE IF CIL(I)¬=J THEN DO;                                       SMI06340
          ECIL(I)='**'; J=CIL(I); ER='1'B;                               SMI06350
        END;                                                             SMI06360
      END;                                                               SMI06370
      ELSE IF CIL(I)>57 THEN DO; ECIL(I)='**'; ER='1'B; END;             SMI06380
    DO K=1 TO I-1;                                                       SMI06390
      IF CPUMNO(K)=CPUMNO(I) THEN DO; ECPUMNO(I)='*****'; ER='1'B; END;  SMI06400
    END;                                                                 SMI06410
    IF YUGO6(I)='2' | YUGO6(I)='3' | YUGO6(I)='5' THEN DO;               SMI06420
      IF DATA6(I,1)¬=0 THEN DO; EYUGO6(I)='**'; ER='1'B; END;            SMI06430
      ELSE IF DATA6(I,2)¬=0 THEN DO; EYUGO6(I)='**'; ER='1'B; END;       SMI06440
    END;                                                                 SMI06450
    IF DATA6(I,2)/DATA6(I,1)>2 THEN DO; EDATA6(I,1)=(5)'*'; ER='1'B;     SMI06460
    END;                                                                 SMI06470
    IF YUGO6(I)='1' | YUGO6(I)='4' THEN GOTO SK6;                        SMI06480
    INV='0'B;                                                            SMI06490
    READ FILE(EDIT2) INTO(KREC) KEY(SNUM || '6' || CPUMNO(I));           SMI06500
    IF INV THEN DO;                                                      SMI06510
      EYUGO6(I)='$$'; ER='1'B;                                           SMI06520
    END;                                                                 SMI06530
    ELSE DO;                                                             SMI06540
      IF YUGO6(I)>'0' & YUGO6(I)<'6' THEN GOTO SK6;                      SMI06550
      HDATA(1)=DATA6(I,1); HDATA(2)=KDATA6(1);                           SMI06560
      HDATA(3)=DATA6(I,2); HDATA(4)=KDATA6(2);                           SMI06570
      IF HDATA(2)=0 THEN GOTO K6;                                        SMI06580
      IF HDATA(1)/HDATA(2)<0.8 | HDATA(1)/HDATA(2)>1.2                   SMI06590
       THEN DO; EDATA6(I,1)=(5)'$'; ER='1'B;                             SMI06600
```

```
        ERNO=ERNO+1; ERDAT(ERNO)=KDATA6(1); END;                        SMI06610
    K6:  IF HDATA(4)=0 THEN GOTO SK6;                                   SMI06620
      IF HDATA(3)/HDATA(4)<0.5 | HDATA(3)/HDATA(4)>2                    SMI06630
       THEN DO; EDATA6(I,2)=(5)'$'; ER='1'B;                            SMI06640
        ERNO=ERNO+1; ERDAT(ERNO)=KDATA6(2); END;                        SMI06650
    END;                                                                SMI06660
  SK6: END;                                                             SMI06670
 END ER6RTN;                                                            SMI06680
 END ERCHECK;                                                           SMI06690
  LLLT:                                                                 SMI06700
 CLOSE FILE(PUMF), FILE(EDIT3), FILE(TOUT), FILE(EDIT2);                SMI06710
 END EDITING;                                                           SMI06720
/*                                                                      SMI06730
//LKED.SYSLMOD DD DSN=USER.LOADLIB(SMIED65),DISP=SHR                    SMI06740
//                                                                      SMI06750
```

SMICOS12

OK SELECT
(CATAL PROGRAM)

```
//SMICOS12 JOB TYPRUN=HOLD                                          SMI00010
//   EXEC PLIFCL                                                    SMI00020
//PLI.SYSIN DD *                                                    SMI00030
* PROCESS GS,NEST,A,XREF,OPT(2);                                    SMI00040
 CLEAR:PROC OPTIONS(MAIN);                                          SMI00050
  DCL EDIT3 FILE RECORD KEYED ENV(VSAM);                            SMI00060
  DCL TAIN FILE RECORD INPUT;                                       SMI00070
  DCL 1 EREC,                                                       SMI00080
       2 SAUP   CHAR(5),                                            SMI00090
       2 CD     CHAR(1),                                            SMI00100
       2 NUM    CHAR(2),                                            SMI00110
       2 DATA   CHAR(40),                                           SMI00120
       2 CHECK  CHAR(2);                                            SMI00130
  DCL 1 TREC,                                                       SMI00140
       2 TSA    CHAR(5),                                            SMI00150
       2 SGN    CHAR(1),                                            SMI00160
       2 FIL    CHAR(4);                                            SMI00170
  OPEN FILE(EDIT3) SEQUENTIAL UPDATE;                               SMI00180
  OPEN FILE(TAIN);                                                  SMI00190
  ON ENDFILE(EDIT3) GOTO LT;  TOT=0; TAT=0;                         SMI00200
  ON ENDFILE(TAIN) GOTO LT;                                         SMI00210
  RT: READ FILE(TAIN) INTO(TREC);                                   SMI00220
      TOT=TOT+1;                                                    SMI00230
      PUT  EDIT(TSA,SGN,'TT')(X(2),A(5),A(1),A(2));                 SMI00240
      IF SAUP>=TSA THEN GOTO BK;                                    SMI00250
  RE: READ FILE(EDIT3) INTO(EREC);                                  SMI00260
  BK:IF SAUP=TSA THEN DO;                                           SMI00270
        IF SGN='1' THEN DO;                                         SMI00280
           CHECK=SGN;  TAT=TAT+1;                                   SMI00290
           REWRITE FILE(EDIT3) FROM(EREC);                          SMI00300
           GOTO RE;                                                 SMI00310
        END;                                                        SMI00320
     END;                                                           SMI00330
     ELSE IF SAUP > TSA THEN GOTO RT;                               SMI00340
     ELSE GOTO RE;                                                  SMI00350
 LT: PUT SKIP EDIT('TOTAL RECORDS = ',TOT,TAT)(A,F(10),F(10));      SMI00360
     CLOSE FILE(EDIT3),FILE(TAIN);                                  SMI00370
 END CLEAR;                                                         SMI00380
/*                                                                  SMI00390
//LKED.SYSLMOD DD DSN=USER.LOADLIB(SMIOS12),DISP=SHR                SMI00400
//                                                                  SMI00410
```

SMICUP12

UPDATE
(CATAL PROGRAM)

```
//SMICUP12 JOB TYPRUN=HOLD                                          SMI00010
//  EXEC PLIFCL                                                     SMI00030
//PLI.SYSIN DD *                                                    SMI00040
* PROCESS GS,NEST,A,XREF,OPT(2);                                    SMI00050
 UPDATE: PROC OPTIONS(MAIN);                                        SMI00060
 DCL CMONTH          PIC'99';                                       SMI00080
 DCL RETURN_CODE     FIXED BIN(15);                                 SMI00090
 DCL ONCODE          BUILTIN;                                       SMI00100
 DCL TAIN FILE RECORD INPUT;                                        SMI00110
 DCL EDIT3 FILE RECORD KEYED ENV(VSAM);                             SMI00120
 DCL (RTYPE,NOER,NOINV)  BIT(1) INIT('1'B);                         SMI00130
 DCL TREC                CHAR(80);                                  SMI00140
 DCL SREC                CHAR(80);                                  SMI00150
 DCL 1 T1 DEF SREC,                                                 SMI00160
      2 SAUP             CHAR(5),                                   SMI00170
      2 CD               CHAR(1),                                   SMI00180
      2 NUM              CHAR(2),                                   SMI00190
      2 DATA             CHAR(67),                                  SMI00200
      2 TAB             CHAR(1),                                    SMI00210
      2 TMON            CHAR(2),                                    SMI00220
      2 F1              CHAR(1),                                    SMI00230
      2 CHECK           CHAR(1);                                    SMI00240
 DCL PIC7                PIC'------9',                              SMI00250
     PIC8                PIC'(8)9',                                 SMI00260
     PIC5                PIC'99999';                                SMI00270
     1 OREC,                                                        SMI00280
      2 EKEY             CHAR(8),                                   SMI00290
      2 TDATA            CHAR(40),                                  SMI00300
      2 OCHECK           CHAR(1),                                   SMI00310
      2 OFFF             CHAR(1);                                   SMI00320
 DCL 1 OD1 DEF SREC POS(9),                                         SMI00330
      2 OINDUST           CHAR(5),                                  SMI00340
      2 OSIZE             CHAR(1),                                  SMI00350
      2 OYUGO1            CHAR(1);                                  SMI00360
 DCL 1 CD1 DEF TDATA,                                               SMI00370
      2 INDUST           CHAR(5),                                   SMI00380
      2 SIZE             CHAR(1),                                   SMI00390
      2 YUGO1            CHAR(1),                                   SMI00400
      2 TBL              CHAR(1),                                   SMI00410
      2 MON              PIC'99';                                   SMI00420
 DCL 1 CD2 DEF TDATA,                                               SMI00430
      2 PUM              CHAR(5),                                   SMI00440
      2 YUGO2            CHAR(1),                                   SMI00450
      2 DAT2(7)          FIXED BIN(31),                             SMI00460
      2 WHY(2)           CHAR(1);                                   SMI00470
 DCL 1 CD3 DEF TDATA,                                               SMI00480
      2 WON              CHAR(4),                                   SMI00490
      2 YUGO3            CHAR(1),                                   SMI00500
      2 GU               CHAR(1),                                   SMI00510
      2 DAT3(2)          FIXED BIN(31);                             SMI00520
 DCL DAT4(10)            FIXED BIN(31)  DEF TDATA;                  SMI00530
 DCL DAT5(7)             FIXED BIN(31)  DEF TDATA;                  SMI00540
 DCL 1 CD6 DEF TDATA,                                               SMI00550
      2 CPUM             CHAR(5),                                   SMI00560
      2 YUGO6            CHAR(1),                                   SMI00570
```

```
     2 DAT6(2)          FIXED BIN(31);                                  SMI00580
 DCL SCND       BIT(1) INIT('1'B);                                      SMI00590
 ON KEY(EDIT3) BEGIN;                                                   SMI00600
   IF ONCODE=51 THEN NOINV='0'B;                                        SMI00610
 END;                                                                   SMI00620
  GET SKIP EDIT(CMONTH)(P'99');                                         SMI00630
  OPEN FILE(TAIN); OPEN FILE(EDIT3) SEQUENTIAL UPDATE;                  SMI00640
 RPO:  READ FILE(TAIN) INTO(TREC);                                      SMI00650
  CALL PLISRTD(' SORT FIELDS=(77,2,CH,A,80,1,CH,D,1,8,CH,A) ',          SMI00660
   'RECORD TYPE=F,LENGTH=(80) ', 45000, RETURN_CODE, E15X, E35X);       SMI00670
  CALL PLIRETC(RETURN_CODE);                                            SMI00680
 E15X: PROC RETURNS(CHAR(80));                                          SMI00690
  ON ENDFILE(TAIN) RTYPE='0'B;                                          SMI00700
  DO WHILE(RTYPE);                                                      SMI00710
    SREC=TREC;                                                          SMI00720
    CALL KEYCHEK;                                                       SMI00730
  RRT: READ FILE(TAIN) INTO(TREC);       /*********************/         SMI00740
    IF NOER THEN DO;                                                    SMI00750
      CALL PLIRETC(12);                                                 SMI00760
      RETURN(SREC);                                                     SMI00770
    END;                                                                SMI00780
    NOER='1'B;                                                          SMI00790
  END;                                                                  SMI00800
  CALL PLIRETC(8);                                                      SMI00810
 END E15X;                                                              SMI00820
 E35X: PROC(SRTREC);                                                    SMI00830
  DCL SRTREC   CHAR(80);                                                SMI00840
  SREC=SRTREC;                                                          SMI00850
  IF CHECK='1' THEN CALL DELTRTN;                                       SMI00860
  ELSE CALL UPDTRTN;                                                    SMI00870
  CALL PLIRETC(4);                                                      SMI00880
 END E35X;                                                              SMI00890
 DELTRTN: PROC;                                                         SMI00900
  ON ENDFILE(EDIT3) GOTO DEND;                                          SMI00910
  IF DATA¬=(63)' ' THEN DO;                                             SMI00920
    PUT SKIP EDIT(SREC,'FUNCTION ERROR')(A(85),A); RETURN;              SMI00930
  END;                                                                  SMI00940
  IF CD=' ' THEN DO;                                                    SMI00950
  BB:  IF SAUP>SUBSTR(EKEY,1,5) THEN DO;                                SMI00960
      READ FILE(EDIT3) INTO(OREC);  GOTO BB;                            SMI00970
    END;                                                                SMI00980
    IF SAUP<SUBSTR(EKEY,1,5) THEN DO;                                   SMI00990
      PUT SKIP EDIT(SREC,'DELETE NOT FOUND')(A(85),A); RETURN;          SMI01000
    END;                                                                SMI01010
    IF OCHECK='1' THEN DO;                                              SMI01020
      PUT SKIP EDIT(SREC,'DELETE CHECK ERROR')(A(85),A); RETURN;        SMI01030
    END;                                                                SMI01040
    DO WHILE(SAUP=SUBSTR(EKEY,1,5));                                    SMI01050
      DELETE FILE(EDIT3) KEY(EKEY);                                     SMI01060
      PUT SKIP EDIT(EKEY,'DELETE')(A(85),A);                            SMI01070
      READ FILE(EDIT3) INTO(OREC);                                      SMI01080
    END;                                                                SMI01090
  END;                                                                  SMI01100
  ELSE IF NUM=' ' THEN DO;                                              SMI01110
  BBC: IF SAUP || CD>SUBSTR(EKEY,1,6) THEN DO;                          SMI01120
```

```
      READ FILE(EDIT3) INTO(OREC);  GOTO BBC;                          SMI01130
    END;                                                               SMI01140
    IF SAUP || CD<SUBSTR(EKEY,1,6) THEN DO;                            SMI01150
      PUT SKIP EDIT(SREC,'DELETE NOT FOUND')(A(85),A); RETURN;         SMI01160
    END;                                                               SMI01170
    IF OCHECK='1' THEN DO;                                             SMI01180
      PUT SKIP EDIT(SREC,'DELETE CHECK ERROR')(A(85),A); RETURN;       SMI01190
    END;                                                               SMI01200
    DO WHILE(SAUP || CD=SUBSTR(EKEY,1,6));                             SMI01210
      DELETE FILE(EDIT3) KEY(EKEY);                                    SMI01220
      PUT SKIP EDIT(EKEY,'DELETE')(A(85),A);                           SMI01230
      READ FILE(EDIT3) INTO(OREC);                                     SMI01240
    END;                                                               SMI01250
  END;                                                                 SMI01260
  ELSE DO;                                                             SMI01270
  BBD: IF SAUP || CD || NUM>EKEY THEN DO;                              SMI01280
      READ FILE(EDIT3) INTO(OREC);  GOTO BBD;                          SMI01290
    END;                                                               SMI01300
    IF SAUP || CD || NUM<EKEY THEN DO;                                 SMI01310
      PUT SKIP EDIT(SREC,'DELETE NOT FOUND')(A(85),A); RETURN;         SMI01320
    END;                                                               SMI01330
    IF OCHECK='1' THEN DO;                                             SMI01340
      PUT SKIP EDIT(SREC,'DELETE CHECK ERROR')(A(85),A); RETURN;       SMI01350
    END;                                                               SMI01360
    DELETE FILE(EDIT3) KEY(EKEY);                                      SMI01370
    PUT SKIP EDIT(EKEY,'DELETE')(A(85),A);                             SMI01380
  END;                                                                 SMI01390
  DEND:                                                                SMI01400
 END DELTRTN;                                                          SMI01410
 UPDTRTN: PROC;                                                        SMI01420
  IF SCND THEN DO;                                                     SMI01430
    SCND='0'B; CLOSE FILE(EDIT3); OPEN FILE(EDIT3) DIRECT UPDATE;      SMI01440
  END;                                                                 SMI01450
  EKEY=SUBSTR(SREC,1,8);                                               SMI01460
  READ FILE(EDIT3) INTO(OREC) KEY(EKEY);                               SMI01470
  IF NOINV THEN DO;                                                    SMI01480
    IF OCHECK='1' THEN DO;                                             SMI01490
      PUT SKIP EDIT(SREC,'CHECK ERROR')(A(85),A); RETURN;              SMI01500
    END;                                                               SMI01510
    CALL CDCHECK;                                                      SMI01520
    REWRITE FILE(EDIT3) FROM(OREC) KEY(EKEY);                          SMI01530
  END;                                                                 SMI01540
  ELSE DO;                                                             SMI01550
    SUBSTR(EKEY,6,3)='100'; NOINV='1'B;                                SMI01560
    READ FILE(EDIT3) INTO(OREC) KEY(EKEY);                             SMI01570
    IF NOINV & OCHECK='1' THEN DO;                                     SMI01580
      PUT SKIP EDIT(SREC,'CHECK2 ERROR')(A(85),A); RETURN;             SMI01590
    END;                                                               SMI01600
    NOINV='0'B; TDATA=' '; OCHECK=' ';                                 SMI01610
    EKEY=SUBSTR(SREC,1,8);                                             SMI01620
    CALL CDCHECK;                                                      SMI01630
    WRITE FILE(EDIT3) FROM(OREC) KEYFROM(EKEY);                        SMI01640
    NOINV='1'B;                                                        SMI01650
  END;                                                                 SMI01660
 END UPDTRTN;                                                          SMI01670
```

```
KEYCHEK: PROC;                                                          SMI01680
IF CHECK='1' THEN                                                       SMI01690
   SUBSTR(SREC,1,5)=TRANSLATE(SUBSTR(SREC,1,5),'0',' ');                SMI01700
 ELSE DO;                                                               SMI01710
   SUBSTR(SREC,1,8)=TRANSLATE(SUBSTR(SREC,1,8),'0',' ');                SMI01720
   DO I=1 TO 8;                                                         SMI01730
     IF SUBSTR(SREC,I,1)<'0' | SUBSTR(SREC,I,1)>'9' THEN DO;            SMI01740
       NOER='0'B; RETURN; END;                                          SMI01750
   END;                                                                 SMI01760
 END;                                                                   SMI01770
 IF CD='1' | CD='4' | CD='5' THEN NUM='00';                             SMI01780
 IF CD='4' THEN SREC=SUBSTR(SREC,1,13) || (5)' ' || SUBSTR(SREC,14,28) SMI01790
    || (8)' ' || SUBSTR(SREC,42,21) || SUBSTR(SREC,76,5);               SMI01800
 IF CD='5' THEN SREC=SUBSTR(SREC,1,16) || (8)' ' || SUBSTR(SREC,17,51) SMI01810
    || SUBSTR(SREC,76,5);                                               SMI01820
 IF TMON='  ' THEN NOER='0'B;                                           SMI01830
 ELSE DO;                                                               SMI01840
   IF SUBSTR(TMON,2,1)=' ' THEN MON=SUBSTR(TMON,1,1);                   SMI01850
   ELSE MON=TMON;                                                       SMI01860
   TMON=MON;                                                            SMI01870
   IF MON¬=CMONTH THEN NOER='0'B;                                       SMI01880
   IF CHECK¬='1' & (CD('1' | CD)'6') THEN NOER='0'B;                    SMI01890
 END;                                                                   SMI01900
 IF NOER='0'B THEN PUT SKIP EDIT(SREC,'KEY ERROR')(A(85),A);
END KEYCHEK;                                                            SMI01910
CDCHECK: PROC;                                                          SMI01920
 IF CD='1' THEN CALL CD1RTN;                                            SMI01930
 ELSE IF CD='2' THEN CALL CD2RTN;                                       SMI01940
 ELSE IF CD='3' THEN CALL CD3RTN;                                       SMI01950
 ELSE IF CD='4' THEN CALL CD4RTN;                                       SMI01960
 ELSE IF CD='5' THEN CALL CD5RTN;                                       SMI01970
 ELSE CALL CD6RTN;                                                      SMI01980
END CDCHECK;                                                            SMI01990
CD1RTN: PROC;                                                           SMI02000
 IF NOINV THEN PUT SKIP EDIT(EKEY,INDUST,SIZE,YUGO1,TBL,MON,'ORIGINAL')SMI02010
   (A(8),A(5),2 A(1),X(60),A(1),A(2),X(7),A);                           SMI02020
 IF OINDUST='-' THEN INDUST=' ';                                        SMI02030
 ELSE IF OINDUST¬=' ' THEN INDUST=OINDUST;                              SMI02040
 IF OSIZE='-' THEN SIZE=' ';                                            SMI02050
 ELSE IF OSIZE¬=' ' THEN SIZE=OSIZE;                                    SMI02060
 IF OYUGO1='-' THEN YUGO1=' ';                                          SMI02070
 ELSE IF OYUGO1¬=' ' THEN YUGO1=OYUGO1;                                 SMI02080
 IF TAB='-' THEN TBL=' ';                                               SMI02090
 ELSE IF TAB¬=' ' THEN TBL=TAB;                                         SMI02100
 MON=TMON;                                                              SMI02110
 PUT SKIP EDIT(EKEY,INDUST,SIZE,YUGO1,TBL,MON,'UPDATE')                 SMI02120
   (A(8),A(5),2 A(1),X(60),A(1),A(2),X(7),A);                           SMI02130
END CD1RTN;                                                             SMI02140
CD2RTN: PROC;                                                           SMI02150
 IF NOINV THEN PUT SKIP EDIT(EKEY,PUM,YUGO2,(DAT2(I) DO I=1 TO 7),      SMI02160
 WHY(1),WHY(2),'ORIGINAL')(A(8),A(5),A(1),7 F(7),X(1),2 A(1),X(19),A); SMI02170
 IF SUBSTR(DATA,1,5)¬=' ' THEN PUM=SUBSTR(DATA,1,5);                    SMI02180
 IF SUBSTR(DATA,6,1)='-' THEN YUGO2=' ';                                SMI02190
 ELSE IF SUBSTR(DATA,6,1)¬=' ' THEN YUGO2=SUBSTR(DATA,6,1);             SMI02200
 IF SUBSTR(DATA,56,1)='-' THEN WHY(1)=' ';                              SMI02210
```

```
  ELSE IF SUBSTR(DATA,56,1)¬=' ' THEN WHY(1)=SUBSTR(DATA,56,1);          SMI02220
  IF SUBSTR(DATA,57,1)='-' THEN WHY(2)=' ';                              SMI02230
  ELSE IF SUBSTR(DATA,57,1)¬=' ' THEN WHY(2)=SUBSTR(DATA,57,1);          SMI02240
  K=0;                                                                   SMI02250
  DO I=7,14,21,28,35,42,49;                                              SMI02260
    K=K+1;                                                               SMI02270
    IF SUBSTR(DATA,I,1)¬=' ' THEN CALL RJ7RTN;                           SMI02280
    ELSE IF NOINV='0'B THEN DAT2(K)=0;                                   SMI02290
  END;                                                                   SMI02300
  PUT SKIP EDIT(EKEY,PUM,YUGO2,(DAT2(I) DO I=1 TO 7),WHY(1),             SMI02310
  WHY(2),'UPDATE')(A(8),A(5),A(1),7 F(7),X(1),2 A(1),X(19),A);           SMI02320
 END CD2RTN;                                                             SMI02330
 CD3RTN: PROC;                                                           SMI02340
  IF NOINV THEN PUT SKIP EDIT(EKEY,WON,YUGO3,GU,DAT3(1),DAT3(2),         SMI02350
    'ORIGINAL')(A(8),A(4),2 A(1),2 F(7),X(57),A);                        SMI02360
  IF SUBSTR(DATA,1,4)¬=' ' THEN WON=SUBSTR(DATA,1,4);                    SMI02370
  IF SUBSTR(DATA,5,1)='-' THEN YUGO3=' ';                                SMI02380
  ELSE IF SUBSTR(DATA,5,1)¬=' ' THEN YUGO3=SUBSTR(DATA,5,1);             SMI02390
  IF SUBSTR(DATA,6,1)¬=' ' THEN GU=SUBSTR(DATA,6,1);                     SMI02400
  K=0;                                                                   SMI02410
  DO I=7,14;                                                             SMI02420
    K=K+1;                                                               SMI02430
    IF SUBSTR(DATA,I,1)¬=' ' THEN CALL RJ7RTN;                           SMI02440
    ELSE IF NOINV='0'B THEN DAT2(K)=0;                                   SMI02450
  END;                                                                   SMI02460
  PUT SKIP EDIT(EKEY,WON,YUGO3,GU,DAT3(1),DAT3(2),'UPDATE')              SMI02470
    (A(8),A(4),2 A(1),2 F(7),X(57),A);                                   SMI02480
 END CD3RTN;                                                             SMI02490
 CD4RTN: PROC;                                                           SMI02500
 IF NOINV THEN PUT SKIP EDIT(EKEY,(DAT4(I) DO I=1 TO 10),'ORIGINAL')     SMI02510
   (A(8),6 F(5),4 F(8),X(15),A);                                         SMI02520
  K=0;                                                                   SMI02530
  DO I=1,6,11,16,21,26;                                                  SMI02540
    K=K+1;                                                               SMI02550
    IF SUBSTR(DATA,I,1)¬=' ' THEN DO;                                    SMI02560
      J=INDEX(SUBSTR(DATA,I,5),' ');                                     SMI02570
      IF J¬=0 THEN DO;                                                   SMI02580
        CALL SUBNUM;                                                     SMI02590
        PIC5=SUBSTR(DATA,I,J-1);                                         SMI02600
        DAT4(K)=PIC5;                                                    SMI02610
      END;                                                               SMI02620
      ELSE DO; J=6; CALL SUBNUM;                                         SMI02630
        DAT4(K)=SUBSTR(DATA,I,5); END;                                   SMI02640
    END;                                                                 SMI02650
    ELSE IF NOINV='0'B THEN DAT4(K)=0;                                   SMI02660
  END;                                                                   SMI02670
  DO I=31,39,47,55;                                                      SMI02680
    K=K+1;                                                               SMI02690
    IF SUBSTR(DATA,I,1)¬=' ' THEN DO;                                    SMI02700
      J=INDEX(SUBSTR(DATA,I,8),' ');                                     SMI02710
      IF J¬=0 THEN DO;                                                   SMI02720
        CALL SUBNUM;                                                     SMI02730
        PIC8=SUBSTR(DATA,I,J-1);                                         SMI02740
        DAT4(K)=PIC8;                                                    SMI02750
      END;                                                               SMI02760
```

```
      ELSE DO; J=9; CALL SUBNUM;                                        SMI02770
         DAT4(K)=SUBSTR(DATA,I,8); END;                                 SMI02780
    END;                                                                SMI02790
    ELSE IF NOINV='0'B THEN DAT4(K)=0;                                  SMI02800
  END;                                                                  SMI02810
  PUT SKIP EDIT(EKEY,(DAT4(I) DO I=1 TO 10),'UPDATE')                   SMI02820
   (A(8),6 F(5),4 F(8),X(15),A);                                        SMI02830
 END CD4RTN;                                                            SMI02840
 CD5RTN: PROC;                                                          SMI02850
 IF NOINV THEN PUT SKIP EDIT(EKEY,(DAT4(I) DO I=1 TO 7),'ORIGINAL')     SMI02860
   (A(8),7 F(8),X(21),A);                                               SMI02870
  K=0;                                                                  SMI02880
  DO I=1,9,17,25,33,41,49;                                              SMI02890
    K=K+1;                                                              SMI02900
    IF SUBSTR(DATA,I,1)¬=' ' THEN DO;                                   SMI02910
      J=INDEX(SUBSTR(DATA,I,8),' ');                                    SMI02920
      IF J¬=0 THEN DO;                                                  SMI02930
        CALL SUBNUM;                                                    SMI02940
        PIC8=SUBSTR(DATA,I,J-1);                                        SMI02950
        DAT5(K)=PIC8;                                                   SMI02960
      END;                                                              SMI02970
      ELSE DO; J=9; CALL SUBNUM;                                        SMI02980
        DAT5(K)=SUBSTR(DATA,I,8); END;                                  SMI02990
    END;                                                                SMI03000
    ELSE IF NOINV='0'B THEN DAT5(K)=0;                                  SMI03010
  END;                                                                  SMI03020
  PUT SKIP EDIT(EKEY,(DAT4(I) DO I=1 TO 7),'UPDATE')                    SMI03030
   (A(8),7 F(8),X(21),A);                                               SMI03040
 END CD5RTN;                                                            SMI03050
 CD6RTN: PROC;                                                          SMI03060
  IF NOINV THEN PUT SKIP EDIT(EKEY,CPUM,YUGO6,DAT6(1),DAT6(2),          SMI03070
    'ORIGINAL')(A(8),A(5),A(1),2 F(7),X(57),A);                         SMI03080
  IF SUBSTR(DATA,1,5)¬=' ' THEN CPUM=SUBSTR(DATA,1,5);                  SMI03090
  IF SUBSTR(DATA,6,1)='-' THEN YUGO6=' ';                               SMI03100
  ELSE IF SUBSTR(DATA,6,1)¬=' ' THEN YUGO6=SUBSTR(DATA,6,1);            SMI03110
  K=0;                                                                  SMI03120
  DO I=7,14;                                                            SMI03130
    K=K+1;                                                              SMI03140
    IF SUBSTR(DATA,I,1)¬=' ' THEN CALL RJ7RTN;                          SMI03150
    ELSE IF NOINV='0'B THEN DAT2(K)=0;                                  SMI03160
  END;                                                                  SMI03170
  PUT SKIP EDIT(EKEY,CPUM,YUGO6,DAT6(1),DAT6(2),'UPDATE')               SMI03180
    (A(8),A(5),A(1),2 F(7),X(57),A);                                    SMI03190
 END CD6RTN;                                                            SMI03200
 RJ7RTN: PROC ;                                                         SMI03210
  J=INDEX(SUBSTR(DATA,I,7),' ');                                        SMI03220
  IF J¬=0 THEN DO;                                                      SMI03230
    CALL SUBNUM;                                                        SMI03240
    PIC7=SUBSTR(DATA,I,J-1);                                            SMI03250
    DAT2(K)=PIC7;                                                       SMI03260
  END;                                                                  SMI03270
  ELSE DO; J=8; CALL SUBNUM;                                            SMI03280
    DAT2(K)=SUBSTR(DATA,I,7); END;                                      SMI03290
 END RJ7RTN;                                                            SMI03300
 SUBNUM: PROC;                                                          SMI03310
```

```
  IF SUBSTR(DATA,I,1)='-' THEN DO;                                    SMI03320
    IF SUBSTR(DATA,I+1,1)=' ' THEN SUBSTR(DATA,I,1)='0';              SMI03330
  END;                                                                SMI03340
  ELSE IF (SUBSTR(DATA,I,1)<'0' | SUBSTR(DATA,I,1)>                   SMI03350
   '9') THEN SUBSTR(DATA,I,1)='0';                                    SMI03360
  DO Q=1 TO J-1;                                                      SMI03370
    IF SUBSTR(DATA,I+Q,1)<'0' | SUBSTR(DATA,I+Q,1)>'9' THEN           SMI03380
     SUBSTR(DATA,I+Q,1)='0';                                          SMI03390
  END;                                                                SMI03400
 END SUBNUM;                                                          SMI03410
 END UPDATE;                                                          SMI03420
/*                                                                    SMI03430
//LKED.SYSLMOD DD DSN=USER.LOADLIB(SMIUP12),DISP=SHR                  SMI03440
//                                                                    SMI0
```

SMICOE 12

OK. ERROR 구분
(CATAL PROGRAM)

```
//SMICOE12 JOB TYPRUN=HOLD                                          SMI00010
//   EXEC PLIFCL                                                    SMI00020
//PLI.SYSIN DD *                                                    SMI00030
* PROCESS GS,NEST,A,XREF,OPT(2);                                    SMI00040
 CLEAR:PROC OPTIONS(MAIN);                                          SMI00050
  DCL EDIT3 FILE RECORD KEYED ENV(VSAM);                            SMI00060
  DCL 1 EREC,                                                       SMI00070
       2 SAUP   CHAR(5),                                            SMI00080
       2 CD     CHAR(1),                                            SMI00090
       2 NUM    CHAR(2),                                            SMI00100
       2 DATA   CHAR(40),                                           SMI00110
       2 CHECK  CHAR(2);                                            SMI00120
  DCL  TSA    CHAR(5);                                              SMI00130
  OPEN FILE(EDIT3) SEQUENTIAL UPDATE;                               SMI00140
  ON ENDFILE(EDIT3) GOTO LT;  TOT=0; TAT=0;                         SMI00150
  ON ENDFILE(SYSIN) GOTO LT;                                        SMI00160
  RQ: READ FILE(EDIT3) INTO(EREC);                                  SMI00170
   CHECK='1'; REWRITE FILE(EDIT3) FROM(EREC); GOTO RQ;              SMI00180
  RT: SAUP='  ';                                                    SMI00190
  RX: GET SKIP EDIT(TSA)(A(5));                                     SMI00200
      TOT=TOT+1;                                                    SMI00210
      PUT  EDIT(TSA,SGN,'TT')(X(2),A(5),A(1),A(2));                 SMI00220
      IF SAUP>=TSA THEN GOTO BK;                                    SMI00230
  RE: READ FILE(EDIT3) INTO(EREC);                                  SMI00240
  BK:IF SAUP=TSA THEN DO;                                           SMI00250
          CHECK=' ';  TAT=TAT+1;                                    SMI00260
          REWRITE FILE(EDIT3) FROM(EREC);                           SMI00270
          GOTO RE;                                                  SMI00280
     END;                                                           SMI00290
     ELSE IF SAUP > TSA THEN GOTO RX;                               SMI00300
     ELSE GOTO RE;                                                  SMI00310
 LT: CLOSE FILE(EDIT3);                                             SMI00320
     IF TOT=0 THEN DO;  OPEN FILE(EDIT3) SEQUENTIAL UPDATE;         SMI00330
       GOTO RT;  END;                                               SMI00340
     PUT SKIP EDIT('TOTAL RECORDS = ',TOT,TAT)(A,F(10),F(10));      SMI00350
     CLOSE  FILE(TAIN);                                             SMI00360
 END CLEAR;                                                         SMI00370
/*                                                                  SMI00380
//LKED.SYSLMOD DD DSN=USER.LOADLIB(SMIOE12),DISP=SHR                SMI00390
//                                                                  SMI00400
```

SMICFP12

FILE MERGE
(CATAL PROGRAM)

```
//SMICFP12 JOB TYPRUN=HOLD                                          SMI00010
//ST3 EXEC PLIFCL                                                   SMI00030
//PLI.SYSIN DD *                                                    SMI00040
* PROCESS GS,NEST,A,XREF,OPT(2);
   CREAT:  PROC OPTIONS(MAIN);                                      SMI00050
     DCL DIREC FILE RECORD KEYED ENV(VSAM);                         SMI00070
     DCL TA    FILE RECORD KEYED ENV(VSAM);                         SMI00070
     DCL ONCODE BUILTIN;                                            SMI00090
     DCL TREC   CHAR(50);                                           SMI00100
     DCL DKEY   CHAR(8) DEF TREC;                                   SMI00110
   ON KEY(TA) BEGIN;
    IF ONCODE=52 THEN PUT EDIT(DKEY)(A(10));
   END;
     ON ENDFILE(DIREC) GOTO LT;                                     SMI00150
     OPEN  FILE(DIREC) SEQUENTIAL INPUT;   CNT=0;                   SMI00160
     OPEN  FILE(TA) DIRECT UPDATE;                                  SMI00160
   RT: READ FILE(DIREC) INTO(TREC);                                 SMI00170
        WRITE FILE(TA) FROM(TREC) KEYFROM(DKEY);                    SMI00180
        PUT EDIT(SUBSTR(TREC,1,13))(A(15));
         GOTO RT;                                                   SMI00190
   LT:                                                              SMI00200
     CLOSE  FILE(DIREC), FILE(TA);                                  SMI00210
    END CREAT;                                                      SMI00220
/*                                                                  SMI00230
//LKED.SYSLMOD DD DSN=USER.LOADLIB(SMIFP12),DISP=SHR                SMI00240
//                                                                  SMI00270
```

SMICCH12

CARD CHECK

(CATAL PROGRM)

```
//SMICCH12 JOB TYPRUN=HOLD                                          SMI00010
//   EXEC PLIFCL                                                    SMI00030
//PLI.SYSIN DD *                                                    SMI00040
* PROCESS S,GS,NEST,A,XREF,OPT(2);                                  SMI00050
  SEQUENC : PROCEDURE  OPTIONS(MAIN) ;                              SMI00060
  DECLARE  ONCODE          BUILTIN;                                 SMI00070
  DECLARE  OLD    FILE  RECORD KEYED ENV(VSAM);                     SMI00080
  DECLARE  NEW    FILE  RECORD KEYED ENV(VSAM);                     SMI00090
  DECLARE 1  NEW_RECORD,                                            SMI00100
             2 NSAUP       CHARACTER(5),                            SMI00110
             2 NCARD       PICTURE '9',                             SMI00120
             2 NUMB        CHARACTER(2),                            SMI00130
             2 NPUM        CHARACTER(5),                            SMI00140
             2 NF1         CHARACTER(37);                           SMI00150
  DECLARE    W_OLD         CHARACTER(5);                            SMI00160
  DECLARE    W_NEW         CHARACTER(5);                            SMI00170
  DECLARE    W_SAUP        CHARACTER(5);                            SMI00180
  DECLARE    W_NSAUP       CHARACTER(5);                            SMI00190
  DECLARE    YUGOO         CHARACTER(1);                            SMI00200
  DECLARE    YUGON         CHARACTER(1);                            SMI00210
  DECLARE    O_HD(2)       CHARACTER(5);                            SMI00220
  DECLARE    N_HD(2)       CHARACTER(5);                            SMI00230
  DECLARE  1  OLD2(50),                                             SMI00240
              2 PUMNO       CHARACTER(5),                           SMI00250
              2 PIL         CHARACTER(2),                           SMI00250
              2 YUGO2       CHARACTER(1);                           SMI00260
  DECLARE  1  OLD3(30),                                             SMI00270
              2 WONNO       CHARACTER(4),                           SMI00280
              2 WIL         CHARACTER(2),                           SMI00280
              2 YUGO3       CHARACTER(1),                           SMI00290
              2 GUBUN       CHARACTER(1);                           SMI00300
  DECLARE  1  OLD6(30),                                             SMI00310
              2 CPUMNO      CHARACTER(5),                           SMI00320
              2 CIL         CHARACTER(2),                           SMI00320
              2 YUGO6       CHARACTER(1);                           SMI00330
  DECLARE    (N2,N3,N6,O6,O2,O3)  PICTURE '99' ;                    SMI00340
             N2,N3,N6,O6,O2,O3  = 0    ;                           SMI00350
 DECLARE     1  OLD_RECORD,                                         SMI00360
               2  SAUP     CHARACTER(5),                            SMI00370
               2  CARD     PICTURE '9',                             SMI00380
               2  PUM      CHARACTER(5),                            SMI00390
               2  F1       CHARACTER(42);                           SMI00400
  DECLARE  1  NEW2(50),                                             SMI00410
              2  NPUMNO      CHARACTER(5),                          SMI00420
              2  NPIL        CHARACTER(2),                          SMI00420
              2  NYUGO2      CHARACTER(1);                          SMI00430
  DECLARE  1  NEW3(30),                                             SMI00440
              2  NWONNO       CHARACTER(4),                         SMI00450
              2  NWIL        CHARACTER(2),                          SMI00450
              2  NYUGO3      CHARACTER(1),                          SMI00460
              2  NGUBUN       CHARACTER(1);                         SMI00470
  DECLARE  1  NEW6(30),                                             SMI00480
              2  NCPUMNO     CHARACTER(5),                          SMI00490
              2  NCIL        CHARACTER(2),                          SMI00490
              2  NYUGO6      CHARACTER(1);                          SMI00500
```

```
ON  ENDFILE(OLD)  BEGIN ; EOF1 = 0 ; W_OLD   = '99999' ; END ;        SMI00510
ON  ENDFILE(NEW)  BEGIN ; EOF2 = 0 ; W_NEW   = '99999' ; END ;        SMI00520
                     EOF1, EOF2 = 1 ;                                 SMI00530
OPEN  FILE(OLD)  SEQUENTIAL INPUT;                                   SMI00540
OPEN  FILE(NEW)  SEQUENTIAL INPUT;                                   SMI00550
OPEN  FILE(SYSPRINT)  LINESIZE(133)  PAGESIZE(60) ;                  SMI00560
OLD2 = ' ' ; OLD3 = ' ' ; OLD6 = ' ' ;                               SMI00570
NEW2 = ' ' ; NEW3 = ' ' ; NEW6 = ' ' ;                               SMI00580
READ  FILE(OLD)  INTO (OLD_RECORD) ;                                 SMI00590
READ  FILE(NEW)  INTO (NEW_RECORD) ;                                 SMI00600
      W_NEW = NSAUP ;                                                SMI00610
      W_OLD =   SAUP ;                                               SMI00620
      YUGOO  = SUBSTR (F1,7,1) ;                                     SMI00630
      YUGON  = SUBSTR (NF1,2,1) ;                                    SMI00640
DO   WHILE ( EOF1 = 1 | EOF2 = 1 ) ;                                 SMI00650
   IF W_OLD = '99999' THEN K1 = 2 ; ELSE K1 = 1 ;                    SMI00660
   IF W_NEW = '99999' THEN K2 = 2 ; ELSE K2 = 1 ;                    SMI00670
      O_HD(1) = W_OLD ;                                              SMI00680
      O_HD(2) = W_SAUP;                                              SMI00690
      N_HD(1) = W_NEW ;                                              SMI00700
      N_HD(2) = W_NSAUP ;                                            SMI00710
     CALL  PROCESS  ;                                                SMI00720
 END ;                                                               SMI00730
      CALL  PROCESS ;                                                SMI00740
      CLOSE FILE(OLD), FILE(NEW), FILE(SYSPRINT) ;                   SMI00750
                                                                     SMI00760
 PROCESS  :  PROCEDURE  ;                                            SMI00770
   IF W_OLD  =  SAUP  THEN  DO ;                                     SMI00780
   CALL  OLD_R ;                                                     SMI00790
   READ  FILE(OLD)  INTO (OLD_RECORD) ;                              SMI00800
   END ;                                                             SMI00810
 ELSE  DO ;                                                          SMI00820
   IF W_NEW  = NSAUP  THEN  DO ;                                     SMI00830
      CALL  NEW_R ;                                                  SMI00840
   READ  FILE(NEW)  INTO (NEW_RECORD) ;                              SMI00850
   END ;                                                             SMI00860
  ELSE  DO ;                                                         SMI00870
    IF  W_NEW = W_OLD THEN  DO ;                                     SMI00880
        CALL  CHECK_R ;                                              SMI00890
        CALL  OLD_R ;                                                SMI00900
        CALL  NEW_R ;                                                SMI00910
        IF  W_NEW ¬= '99999' THEN                                    SMI00920
        READ FILE(NEW) INTO (NEW_RECORD) ;                           SMI00930
        IF  W_OLD ¬= '99999' THEN                                    SMI00940
        READ FILE(OLD) INTO (OLD_RECORD) ;                           SMI00950
        END ;                                                        SMI00960
    ELSE  DO ;                                                       SMI00970
      IF W_NEW > W_OLD THEN DO ;                                     SMI00980
         CALL OLD_PUT ;                                              SMI00990
         CALL  OLD_R ;                                               SMI01000
         IF  W_OLD ¬= '99999' THEN                                   SMI01010
         READ FILE(OLD) INTO (OLD_RECORD) ;                          SMI01020
         END ;                                                       SMI01030
      ELSE DO ;                                                      SMI01040
         CALL NEW_PUT ;                                              SMI01050
```

```
        CALL  NEW_R ;                                                   SMI01060
        IF  W_NEW ¬= '99999' THEN                                       SMI01070
        READ FILE(NEW) INTO (NEW_RECORD) ;                              SMI01080
        END ;                                                           SMI01090
      END ;  END  ;  END ;                                              SMI01100
      END PROCESS ;                                                     SMI01110
                                                                        SMI01120
    OLD_R  :  PROCEDURE ;                                               SMI01130
      SELECT (CARD) ;                                                   SMI01140
        WHEN (1)  DO ;                                                  SMI01150
             W_SAUP = SAUP ;                                            SMI01160
                  END ;                                                 SMI01170
        WHEN (2)  DO ;                                                  SMI01180
             O2          = O2 + 1 ;                                     SMI01190
             PUMNO (O2) = PUM ;  PIL(O2)=SUBSTR(F1,1,2);                SMI01200
             YUGO2 (O2) = SUBSTR (F1,3,1) ;                             SMI01210
             END ;                                                      SMI01220
        WHEN (3)  DO ;                                                  SMI01230
             O3          = O3 + 1 ;  WIL(O3)=SUBSTR(F1,1,2);            SMI01240
             WONNO (O3) = SUBSTR (PUM,1,4) ;                            SMI01250
             GUBUN (O3) = SUBSTR (PUM,5,1) ;                            SMI01260
             YUGO3 (O3) = SUBSTR (F1,3,1) ;                             SMI01270
             END ;                                                      SMI01280
        WHEN (6)  DO ;                                                  SMI01290
             O6          = O6 + 1 ;                                     SMI01300
             CPUMNO (O6) = PUM ;  CIL(O6)=SUBSTR(F1,1,2);               SMI01310
             YUGO6  (O6) = SUBSTR (F1,3,1) ;                            SMI01320
             END ;                                                      SMI01330
        OTHERWISE ;                                                     SMI01340
        END ;                                                           SMI01350
      END  OLD_R ;                                                      SMI01360
                                                                        SMI01370
    NEW_R  :  PROCEDURE ;                                               SMI01380
      SELECT (NCARD) ;                                                  SMI01390
        WHEN (1)  DO ;                                                  SMI01400
             W_NSAUP = NSAUP ;                                          SMI01410
                  END ;                                                 SMI01420
        WHEN (2)  DO ;                                                  SMI01430
             N2          = N2 + 1 ;                                     SMI01440
             NPUMNO (N2) = NPUM ; NPIL(N2)=NUMB;                        SMI01450
             NYUGO2 (N2) = SUBSTR (NF1,1,1) ;                           SMI01460
             END ;                                                      SMI01470
        WHEN (3)  DO ;                                                  SMI01480
             N3          = N3 + 1 ;  NWIL(N3)=NUMB;                     SMI01490
             NWONNO (N3) = SUBSTR (NPUM,1,4) ;                          SMI01500
             NGUBUN (N3) = SUBSTR (NF1,1,1) ;                           SMI01510
             NYUGO3 (N3) = SUBSTR (NPUM,5,1) ;                          SMI01520
             END ;                                                      SMI01530
        WHEN (6)  DO ;                                                  SMI01540
             N6          = N6 + 1 ;                                     SMI01550
             NCPUMNO (N6) = NPUM ; NCIL(N6)=NUMB;                       SMI01560
             NYUGO6  (N6) = SUBSTR (NF1,1,1) ;                          SMI01570
             END ;                                                      SMI01580
        OTHERWISE ;                                                     SMI01590
        END ;                                                           SMI01600
```

```
      END  NEW_R ;                                                      SMI01610
                                                                        SMI01620
   CHECK_R  :  PROCEDURE ;                                              SMI01630
      DO  I  =  1  TO  O2 ;                                             SMI01640
          DO  J  =  1  TO  N2 ;                                         SMI01650
              IF  PUMNO(I)   =  NPUMNO(J)  THEN                         SMI01660
                  PUMNO(I), NPUMNO(J) = ' ' ;                           SMI01670
          END  ;                                                        SMI01680
      END  ;                                                            SMI01690
                                                                        SMI01700
      DO  I  =  1  TO  O3 ;                                             SMI01710
          DO  J  =  1  TO  N3 ;                                         SMI01720
            IF WONNO(I) || GUBUN(I) = NWONNO(J) || NGUBUN(J) THEN DO ;SMI01730
                WONNO(I)  = ' ' ;  GUBUN(I)  = ' ' ;                    SMI01740
                NWONNO(J) = ' ' ;  NGUBUN(J) = ' ' ;                    SMI01750
          END  ;    END ;                                               SMI01760
      END  ;                                                            SMI01770
                                                                        SMI01780
      DO  I  =  1  TO  O6 ;                                             SMI01790
          DO  J  =  1  TO  N6 ;                                         SMI01800
            IF  CPUMNO(I)  =  NCPUMNO(J)  THEN                          SMI01810
                CPUMNO(I), NCPUMNO(J) = ' ' ;                           SMI01820
          END  ;                                                        SMI01830
      END  ;                                                            SMI01840
                                                                        SMI01850
  IF  N2  <=  O2   THEN  N2  =  O2 ;                                    SMI01860
  IF  N3  <=  O3   THEN  N3  =  O3 ;                                    SMI01870
  IF  N6  <=  O6   THEN  N6  =  O6 ;                                    SMI01880
 DO  I  =  1  TO  N2  ;                                                 SMI01890
    IF PUMNO(I) ¬= '      '  THEN                                       SMI01900
       PUT  SKIP(1) EDIT (O_HD(K1),'2',PIL(I),PUMNO(I),YUGO2(I))        SMI01910
                        (X(5),A,X(1),A,X(1),A,X(2),A,X(3),A) ;          SMI01920
    IF NPUMNO(I) ¬= '      '  THEN                                      SMI01930
       PUT  SKIP(1) EDIT (N_HD(K2),'2',NPIL(I),NPUMNO(I),NYUGO2(I))     SMI01940
                        (X(60),A,X(1),A,X(1),A,X(2),A,X(3),A) ;         SMI01950
 END ;                                                                  SMI01960
  /**************                                                       SMI01970
 DO  I  =  1  TO  N3  ;                                                 SMI01980
    IF WONNO(I) || GUBUN(I) ¬= '      '  THEN                           SMI01990
 PUT SKIP(1) EDIT (O_HD(K1),'3',WIL(I),WONNO(I)||GUBUN(I),YUGO3(I))     SMI02000
                        (X(5),A,X(1),A,X(1),A,X(2),A,X(3),A) ;          SMI02010
    IF NWONNO(I) || NGUBUN(I) ¬= '      '  THEN                         SMI02020
 PUT SKIP(1) EDIT (N_HD(K2),'3',NWIL(I),NWONNO(I)||NGUBUN(I),           SMI02030
     NYUGO3(I)    )        (X(60),A,X(1),A,X(1),A,X(2),A,X(3),A) ;      SMI02040
 END ;                                                                  SMI02050
  ***************/                                                      SMI02060
 DO  I  =  1  TO  N6  ;                                                 SMI02070
    IF CPUMNO(I) ¬= '      '  THEN                                      SMI02080
    PUT  SKIP(1) EDIT (O_HD(K1),'6',CIL(I),CPUMNO(I),YUGO6(I))          SMI02090
                        (X(5),A,X(1),A,X(1),A,X(2),A,X(3),A) ;          SMI02100
    IF NCPUMNO(I) ¬= '      '  THEN                                     SMI02110
    PUT  SKIP(1) EDIT (N_HD(K2),'6',NCIL(I),NCPUMNO(I),NYUGO6(I))       SMI02120
                        (X(60),A,X(1),A,X(1),A,X(2),A,X(3),A) ;         SMI02130
 END ;                                                                  SMI02140
 IF  W_OLD ¬= '99999'  THEN  W_OLD = SAUP ;                             SMI02150
```

```
   IF  W_NEW ¬= '99999'  THEN  W_NEW = NSAUP ;                          SMI02160
   N2,N3,N6,O2,O3,O6 = 0 ;                                              SMI02170
  OLD2 = ' ' ; OLD3 = ' ' ; OLD6 = ' ' ;                                SMI02180
  NEW2 = ' ' ; NEW3 = ' ' ; NEW6 = ' ' ;                                SMI02190
  IF CARD = 1 THEN                                                      SMI02200
  YUGOO  = SUBSTR (F1,9,1) ;                                            SMI02210
  IF NCARD = 1 THEN                                                     SMI02220
  YUGON  = SUBSTR (NF1,2,1) ;                                           SMI02230
 END  CHECK_R ;                                                         SMI02240
                                                                        SMI02250
  OLD_PUT  :  PROCEDURE ;                                               SMI02260
     IF W_SAUP ¬= '      '  THEN                                        SMI02270
      PUT  SKIP(2) EDIT (O_HD(K1),YUGOO,'SAUP NURAK')                   SMI02280
                          (X(5),A,X(15),A,X(5),A) ;                     SMI02290
   DO  I  =  1  TO  50  ;                                               SMI02300
     IF PUMNO(I) ¬= '      '  THEN                                     SMI02310
     PUT  SKIP(1) EDIT (O_HD(K1),'2',PIL(I),PUMNO(I),YUGO2(I))          SMI02320
                          (X(5),A,X(1),A,X(1),A,X(2),A,X(3),A) ;        SMI02330
    END ;       /***************                                        SMI02340
   DO  I  =  1  TO  30  ;                                               SMI02350
     IF WONNO(I) || GUBUN(I) ¬= '      '  THEN                         SMI02360
   PUT SKIP(1) EDIT (O_HD(K1),'3',WIL(I),WONNO(I)||GUBUN(I),YUGO3(I))   SMI02370
                          (X(5),A,X(1),A,X(1),A,X(2),A,X(3),A) ;        SMI02380
   END ;      ********************/                                     SMI02390
   DO  I  =  1  TO  30  ;                                               SMI02400
     IF CPUMNO(I) ¬= '      '  THEN                                    SMI02410
    PUT  SKIP(1) EDIT (O_HD(K1),'6',CIL(I),CPUMNO(I),YUGO6(I))          SMI02420
                          (X(5),A,X(1),A,X(1),A,X(2),A,X(3),A) ;        SMI02430
   END ;                                                                SMI02440
   IF  W_OLD ¬= '99999'  THEN  W_OLD = SAUP ;                           SMI02450
   O2,O3,O6          = 0 ;                                              SMI02460
   OLD2 = ' ' ; OLD3 = ' ' ; OLD6 = ' ' ; YUGOO  = ' ' ;                SMI02470
   IF CARD = 1 THEN                                                     SMI02480
   YUGOO  = SUBSTR (F1,9,1) ;                                           SMI02490
   END  OLD_PUT ;                                                       SMI02500
                                                                        SMI02510
   NEW_PUT  : PROCEDURE ;                                               SMI02520
     IF W_NSAUP ¬= '      '  THEN                                      SMI02530
        PUT  SKIP(2) EDIT (N_HD(K2),YUGON,'SAUP SINKYU')                SMI02540
                          (X(60),A,X(15),A,X(5),A) ;                    SMI02550
   DO  I  =  1  TO  50  ;                                               SMI02560
     IF NPUMNO(I) ¬= '      '  THEN                                    SMI02570
    PUT  SKIP(1) EDIT (N_HD(K2),'2',NPIL(I),NPUMNO(I),NYUGO2(I))        SMI02580
                          (X(60),A,X(1),A,X(1),A,X(2),A,X(3),A) ;       SMI02590
   END ;     /**************                                            SMI02600
   DO  I  =  1  TO  30  ;                                               SMI02610
     IF NWONNO(I) || NGUBUN(I) ¬= '      '  THEN                       SMI02620
   PUT SKIP(1) EDIT (N_HD(K2),'3',NWIL(I),NWONNO(I)||NGUBUN(I),         SMI02630
    NYUGO3(I)     )        (X(60),A,X(1),A,X(1),A,X(2),A,X(3),A) ;     SMI02640
   END ;      **************/                                           SMI02650
   DO  I  =  1  TO  30  ;                                               SMI02660
     IF NCPUMNO(I) ¬= '      '  THEN                                   SMI02670
     PUT  SKIP(1) EDIT (N_HD(K2),'6',NCIL(I),NCPUMNO(I),NYUGO6(I))      SMI02680
                          (X(60),A,X(1),A,X(1),A,X(2),A,X(3),A) ;       SMI02690
   END ;                                                                SMI02700
```

```
  IF  W_NEW ¬= '99999'  THEN  W_NEW = NSAUP ;                          SMI02710
  N2,N3,N6          = 0 ;                                              SMI02720
  NEW2 = ' ' ; NEW3 = ' ' ; NEW6 = ' ' ; YUGON = ' ' ;                 SMI02730
  IF NCARD = 1 THEN                                                    SMI02740
  YUGON  = SUBSTR (NF1,2,1) ;                                          SMI02750
  END  NEW_PUT ;                                                       SMI02760
 END SEQUENC ;                                                         SMI02770
/*                                                                     SMI02780
//LKED.SYSLMOD DD DSN=USER.LOADLIB(SMICH12),DISP=SHR                   SMI02790
//                                                                     SMI02810
```

SMICJS12

승 수 산 업 삽 입

(CATAL PROGRAM)

```
//SMICJS12 JOB CLASS=3,TYPRUN=HOLD                                      SMI00010
//  EXEC PLIFCL                                                         SMI00020
//PLI.SYSIN DD *                                                        SMI00030
* PROCESS GS,NEST,A,XREF,OPT(2);                                        SMI00040
 SUMMARY: PROC OPTIONS(MAIN);                                           SMI00050
 DCL EDIT3 FILE RECORD KEYED ENV(VSAM);                                 SMI00530
 DCL SMAST FILE RECORD KEYED ENV(VSAM);                                 SMI00540
 DCL ONCODE          BUILTIN;
 DCL 1 INREC,                                                           SMI00550
      2 SANUM        CHAR(5),                                           SMI00560
      2 CD           CHAR(1),                                           SMI00570
      2 INUM         CHAR(2),                                           SMI00580
      2 INDATA       CHAR(40),                                          SMI00590
      2 FA           CHAR(2);
 DCL 1 DAT1 DEF INDATA,
      2 INDUST       CHAR(5),
      2 SIZE         CHAR(1),
      2 YUGO1        CHAR(1),
      2 TAB          CHAR(1),
      2 MONTH        CHAR(2),
      2 RATE         CHAR(2);
 DCL RTYPE           BIT(1) INIT('1'B);                                 SMI00930
 DCL 1 SAUP,                                                            SMI00840
      2 XXFM         CHAR(5),                                           SMI00850
      2 SNUMST       CHAR(5),                                           SMI00860
      2 ONUM         CHAR(5),                                           SMI00870
      2 INDT1       CHAR(5),                                            SMI00880
      2 GOYU        CHAR(7),                                            SMI00880
      2 RATEB        CHAR(2),                                           SMI00890
      2 F5           CHAR(51);                                          SMI00910
 DCL OSAUP        CHAR(5);
 DCL AKEY         CHAR(8);
 DCL SBNUM(50)    CHAR(5),
     BINDT(50)    CHAR(5),
     BRATE(50)    CHAR(2);
 ON KEY(SMAST) BEGIN;
   IF ONCODE=51 THEN DO;  INDT1=INDUST;
     PUT  EDIT(SANUM,' INV')(A(5),A); RATEB='01'; END;
 END;  K=0;
  OPEN FILE(EDIT3) SEQUENTIAL UPDATE;                                   SMI01020
  OPEN FILE(SMAST) DIRECT INPUT;                                        SMI01030
  ON ENDFILE(EDIT3) RTYPE='0'B;                                         SMI01170
  READ FILE(EDIT3) INTO(INREC);  CNT=0;
  DO WHILE(RTYPE);                                                      SMI01180
    IF OSAUP¬=SANUM THEN DO;                                            SMI01190
      READ FILE(SMAST) INTO(SAUP) KEY(SANUM);                           SMI01200
      IF CD='1' THEN DO;                                                SMI01210
        INDUST=INDT1; RATE=RATEB;   SIZE=SUBSTR(F5,3,1);   CNT=CNT+1;
        REWRITE FILE(EDIT3) FROM(INREC);
      END;
      ELSE DO;
        PUT  EDIT(SANUM,CD)(A(7),A(5));
        K=K+1; SBNUM(K)=SANUM; BINDT(K)=INDT1; BRATE(K)=RATEB;
      END;
    END; OSAUP=SANUM;                                                   SMI01220
```

```
    READ FILE(EDIT3) INTO(INREC);
  END;                                                                  SMI01300
  CLOSE FILE(EDIT3);
  OPEN FILE(EDIT3) DIRECT UPDATE;
  CD='1'; INUM='00';    SIZE='2';    YUGO1=' '; TAB='1';
  DO I=1 TO K;
    SANUM=SBNUM(I); INDUST=BINDT(I); RATE=BRATE(I);
    AKEY=SANUM || CD || INUM;
    WRITE FILE(EDIT3) FROM(INREC) KEY(AKEY);
  END;
  PUT SKIP EDIT('TOT SAUP = ',CNT)(A,F(5));
  CLOSE FILE(EDIT3), FILE(SMAST);
 END SUMMARY;                                                           SMI06750
/*
//LKED.SYSLMOD DD DSN=USER.LOADLIB(SMIJS12),DISP=SHR
//
```

SMICSM62

SUMMARY

(CATAL PROGRAM)

```
//SMICSM62 JOB CLASS=C,TYPRUN=HOLD                                    SMI00010
//  EXEC PLIFCL                                                       SMI00020
//PLI.SYSIN DD *                                                      SMI00030
* PROCESS GS,NEST,A,XREF,OPT(2);                                      SMI00040
  (NOZERODIVIDE) :                                                    SMI00050
 SUMMARY: PROC OPTIONS(MAIN);                                         SMI00060
 DCL CARD            CHAR(80);                                        SMI00070
 DCL ID              CHAR(1),                                         SMI00080
     CMONTH          PIC'99',                                         SMI00090
     CCARD           CHAR(1);                                         SMI00100
 DCL PRICE(10)   FLOAT;                                               SMI00110
 DCL SUMF FILE RECORD KEYED ENV(VSAM);                                SMI00120
 DCL 1 SUMR,                                                          SMI00130
      2 OKEY         CHAR(10),                                        SMI00140
      2 FILK         CHAR(13),                                        SMI00150
      2 DATAJ(37),                                                    SMI00160
        3 ODATA      FIXED BIN(31),                                   SMI00170
        3 JDATA      FLOAT,                                           SMI00180
      2 OFIL         CHAR(1);                                         SMI00190
 DCL SYSEOF          BIT(1) INIT('1'B);                               SMI00200
 DCL (CD2CH,CD5V)     BIT(1) INIT('1'B);                              SMI00210
 DCL EDIT2 FILE RECORD KEYED ENV(VSAM);                               SMI00220
 DCL EDIT3 FILE RECORD KEYED ENV(VSAM);                               SMI00230
 DCL ONCODE          BUILTIN;                                         SMI00240
 DCL 1 INREC,                                                         SMI00250
      2 SANUM        CHAR(5),                                         SMI00260
      2 CD           CHAR(1),                                         SMI00270
      2 INUM         CHAR(2),                                         SMI00280
      2 INDATA       CHAR(40),                                        SMI00290
      2 FA           CHAR(2);                                         SMI00300
 DCL YUGO1 CHAR(1);                                                   SMI00310
 DCL 1 DTT2 DEF INDATA,                                               SMI00320
      2 JAEPUM       CHAR(5),                                         SMI00330
      2 YUGO2        CHAR(1),                                         SMI00340
      2 DAT2(7)      FIXED BIN(31),                                   SMI00350
      2 WHY(2)       CHAR(1);                                         SMI00360
 DCL 1 DTT3 DEF INDATA,                                               SMI00370
      2 WONJAE       CHAR(4),                                         SMI00380
      2 YUGO3        CHAR(1),                                         SMI00390
      2 GU           CHAR(1),                                         SMI00400
      2 DAT3(2)      FIXED BIN(31);                                   SMI00410
 DCL DAT4(10)        FIXED BIN(31)  DEF  INDATA;                      SMI00420
 DCL DAT5(7)         FIXED BIN(31)  DEF  INDATA;                      SMI00430
 DCL 1 DTT6 DEF INDATA,                                               SMI00440
      2 CAPACT       CHAR(5),                                         SMI00450
      2 YUGO6        CHAR(1),                                         SMI00460
      2 DAT6(2)      FIXED BIN(31);                                   SMI00470
 DCL SORTREC         CHAR(89);                                        SMI00480
 DCL 1 SRT DEF SORTREC,                                               SMI00490
      2 SKEY         CHAR(15),                                        SMI00500
      2 SDATA(17)    FIXED BIN(31),                                   SMI00510
      2 SRATE        PIC'99',                                         SMI00520
      2 SSIZE        CHAR(1),                                         SMI00530
      2 SGRT(2)      CHAR(1),                                         SMI00540
      2 SYUGO        CHAR(1);                                         SMI00550
```

```
 DCL INDUST          CHAR(3);                                          SMI00560
 DCL OFUM     CHAR(5),  OGU  CHAR(1),  KYUGO   CHAR(1);               SMI00570
 DCL ANU   CHAR(10), KNU  CHAR(10);                                    SMI00580
 DCL TAB            CHAR(1);                                           SMI00590
 DCL SIZE           CHAR(1);                                           SMI00600
 DCL SNUM           CHAR(5);                                           SMI00610
 DCL RATE           PIC'99';                                           SMI00620
 DCL RETURN_CODE    FIXED BIN(15);                                     SMI00630
 DCL FEUP           BIT(1);                                            SMI00640
 DCL INN            CHAR(2) DEF SORTREC POS(70);                       SMI00650
 DCL IL(2)          CHAR(2);                                           SMI00660
 DCL MDAT(3,17) FLOAT DEC(16);                                         SMI00670
 DCL QDATA(3,17)   FLOAT DEC(16);                                      SMI00680
 DCL ZB   BIT(1) INIT('1'B);                                           SMI00690
 DCL PDATA(3,17)    FLOAT DEC(16),                                     SMI00700
     TDATA(3,17)    FLOAT DEC(16),                                     SMI00710
     ADATA(3,17)    FLOAT DEC(16),                                     SMI00720
     SDT(3,17)    FLOAT DEC(16),                                       SMI00730
     FDATA(3,17)    FLOAT DEC(16);                                     SMI00740
 DCL (WON,OWON)     CHAR(7);                                           SMI00750
 DCL MONTH          PIC'99';                                           SMI00760
 DCL SINDUST        CHAR(3);                                           SMI00770
 DCL FST            BIT(1) INIT('0'B);                                 SMI00780
 DCL RTYPE          BIT(1) INIT('1'B);                                 SMI00790
 ON KEY(SUMF) BEGIN;                                                   SMI00800
   IF ONCODE=51 THEN DO;                                               SMI00810
     PUT EDIT(OKEY,'INV')(A(11),A(4));  END;                           SMI00820
 END;                                                                  SMI00830
 ON ENDFILE(SYSIN) GOTO LLT;                                           SMI00840
 RRC:  GET SKIP EDIT(CMONTH,CCARD,ID)(P'99',A(1),A(1));                SMI00850
 IF CCARD='2' & ID='3' THEN DO;                                        SMI00860
    GET EDIT((PRICE(I) DO I=1 TO 10))(10 (X(1),P'999V9'));             SMI00870
    PRICE(1)=PRICE(1)*0.65+PRICE(2)*0.35;                              SMI00880
    PRICE(2)=PRICE(6)*0.65+PRICE(7)*0.35;                              SMI00890
  END;                                                                 SMI00900
  OPEN FILE(EDIT2) SEQUENTIAL INPUT;                                   SMI00910
  OPEN FILE(EDIT3) SEQUENTIAL INPUT;                                   SMI00920
  OPEN FILE(SUMF) DIRECT UPDATE;                                       SMI00930
  OPEN FILE(SYSPRINT) LINESIZE(132);                                   SMI00940
  READ FILE(EDIT3) INTO(INREC);                                        SMI00950
  CALL PLISRTD(' SORT FIELDS=(1,15,CH,A) ',                            SMI00960
   ' RECORD TYPE=F,LENGTH=(89) ', 45000, RETURN_CODE, E15X, E35X);     SMI00970
  SUBSTR(SKEY,1,5)='99999';                                            SMI00980
  CALL E35X(SORTREC);                                                  SMI00990
  CALL PLIRETC(RETURN_CODE);                                           SMI01000
 E15X: PROC RETURNS(CHAR(89));                                         SMI01010
  ON ENDFILE(EDIT2) GOTO EN15;                                         SMI01020
  ON ENDFILE(EDIT3) RTYPE='0'B;                                        SMI01030
  DO WHILE(RTYPE);                                                     SMI01040
    IF CD='1' THEN DO;                                                 SMI01050
  IF SUBSTR(INDATA,11,2)='  ' THEN SUBSTR(INDATA,11,2)='01';           SMI01060
  IF SUBSTR(INDATA,11,1)('0' | SUBSTR(INDATA,11,1)>'9'                 SMI01070
         THEN SUBSTR(INDATA,11,1)='0';                                 SMI01080
  IF SUBSTR(INDATA,12,1)('0' | SUBSTR(INDATA,12,1)>'9'                 SMI01090
         THEN SUBSTR(INDATA,12,1)='0';                                 SMI01100
```

```
      INDUST=SUBSTR(INDATA,1,3);  RATE=SUBSTR(INDATA,11,2);          SMI01110
      IF RATE=0 THEN RATE=1;                                          SMI01120
      SIZE=SUBSTR(INDATA,6,1);  TAB=SUBSTR(INDATA,8,1);               SMI01130
      MONTH=CMONTH;             YUGO1=SUBSTR(INDATA,7,1);             SMI01140
      READ FILE(EDIT3) INTO(INREC);                                   SMI01150
    END;                                                              SMI01160
   ELSE DO;   IF CD='2' THEN CD2CH='0'B;                              SMI01170
    IF CCARD='4' & CD2CH THEN GOTO AR;                                SMI01180
    IF CD=CCARD THEN DO;                                              SMI01190
      CALL SORTRTN;                                                   SMI01200
      IF CD5V THEN READ FILE(EDIT3) INTO(INREC);                      SMI01210
      CD5V='1'B;                                                      SMI01220
      CALL PLIRETC(12);                                               SMI01230
      RETURN(SORTREC);                                                SMI01240
    END;                                                              SMI01250
  AR:  READ FILE(EDIT3) INTO(INREC);                                  SMI01260
   END;                                                               SMI01270
  END;                                                                SMI01280
  RTT: IF CD5V THEN READ FILE(EDIT2) INTO(INREC);                     SMI01290
    CD5V='1'B;                                                        SMI01300
    IF CD='1' THEN DO;                                                SMI01310
  IF SUBSTR(INDATA,11,2)='  ' THEN SUBSTR(INDATA,11,2)='01';          SMI01320
  IF SUBSTR(INDATA,11,1)<'0' | SUBSTR(INDATA,11,1)>'9'                SMI01330
        THEN SUBSTR(INDATA,11,1)='0';                                 SMI01340
  IF SUBSTR(INDATA,12,1)<'0' | SUBSTR(INDATA,12,1)>'9'                SMI01350
        THEN SUBSTR(INDATA,12,1)='0';                                 SMI01360
      INDUST=SUBSTR(INDATA,1,3); RATE=SUBSTR(INDATA,11,2);            SMI01370
      IF RATE=0 THEN RATE=1;                                          SMI01380
      SIZE=SUBSTR(INDATA,6,1);  TAB=SUBSTR(INDATA,8,1);               SMI01390
      MONTH=CMONTH-1;           YUGO1=SUBSTR(INDATA,7,1);             SMI01400
    END;                                                              SMI01410
    ELSE DO;  IF CD='2' THEN CD2CH='0'B;                              SMI01420
     IF CCARD='4' & CD2CH THEN GOTO RTT;                              SMI01430
     IF CD=CCARD THEN DO;                                             SMI01440
       CALL SORTRTN;                                                  SMI01450
       CALL PLIRETC(12);                                              SMI01460
       RETURN(SORTREC);                                               SMI01470
     END;                                                             SMI01480
    END;                                                              SMI01490
    GOTO RTT;                                                         SMI01500
 EN15:   CALL PLIRETC(8);                                             SMI01510
END E15X;                                                             SMI01520
E35X: PROC(SRTREC);                                                   SMI01530
DCL SRTREC          CHAR(89);                                         SMI01540
 SORTREC=SRTREC;                                                      SMI01550
 IF CCARD='2' THEN CALL CD2RTN;                                       SMI01560
 ELSE IF CCARD='3' THEN CALL CD3RTN;                                  SMI01570
 ELSE IF CCARD='4' THEN CALL CD4RTN;                                  SMI01580
 ELSE IF CCARD='6' THEN CALL CD6RTN;                                  SMI01590
 CALL PLIRETC(4);                                                     SMI01600
END E35X;                                                             SMI01610
SORTRTN: PROC;                                                        SMI01620
 SORTREC=' ';                                                         SMI01630
 IF CD='2' THEN DO;                                                   SMI01640
   IF RATE=1 THEN SKEY=JAEPUM || '1' || SANUM || MONTH ;              SMI01650
```

```
            ELSE SKEY=JAEPUM || '2' || SANUM || MONTH ;                SMI01660
   INN=INUM;                                                           SMI01670
   IF SUBSTR(JAEPUM,1,4)='2201' | SUBSTR(JAEPUM,1,4)='5206' THEN DO;   SMI01680
     SDATA(1)=DAT2(2);  SDATA(2)=0;                                    SMI01690
     DO I=3 TO 6;                                                      SMI01700
       SDATA(I)=DAT2(I);                                               SMI01710
     END;                                                              SMI01720
   END;                                                                SMI01730
   ELSE DO;                                                            SMI01740
     DO I=1 TO 6;                                                      SMI01750
       SDATA(I)=DAT2(I+1);                                             SMI01760
     END;                                                              SMI01770
   END;                                                                SMI01780
   SRATE=RATE;                                                         SMI01790
   SSIZE=SIZE;                                                         SMI01800
   SGRT=WHY;                                                           SMI01810
   SYUGO=YUGO2;                                                        SMI01820
 END;                                                                  SMI01830
 ELSE IF CD='3' THEN DO;                                               SMI01840
     IF INDUST='312' THEN INDUST='311';                                SMI01850
   SKEY=INDUST || WONJAE || SANUM || GU || MONTH;                      SMI01860
   SDATA(1)=DAT3(1);                                                   SMI01870
   SDATA(2)=DAT3(2);                                                   SMI01880
   SYUGO=YUGO3; INN=INUM;                                              SMI01890
 END;                                                                  SMI01900
 ELSE IF CD='4' THEN DO;                                               SMI01910
     CD2CH='1'B;                                                       SMI01920
     IF INDUST='312' THEN INDUST='311';                                SMI01930
     IF INDUST='410' THEN INDUST='400';                                SMI01940
   IF RATE=1 THEN SKEY=INDUST || '1' || SANUM || MONTH;                SMI01950
            ELSE SKEY=INDUST || '2' || SANUM || MONTH;                 SMI01960
   DO I=1 TO 10;                                                       SMI01970
     SDATA(I)=DAT4(I);                                                 SMI01980
   END;                                                                SMI01990
   SRATE=RATE;                                                         SMI02000
   SYUGO=YUGO1;                                                        SMI02010
   IF RTYPE THEN READ FILE(EDIT3) INTO(INREC);                         SMI02020
   ELSE READ FILE(EDIT2) INTO(INREC);                                  SMI02030
   IF CD='5' THEN DO;                                                  SMI02040
    CD5V='1'B;                                                         SMI02050
   DO I=1 TO 7;                                                        SMI02060
     SDATA(I+10)=DAT5(I);                                              SMI02070
   END; END;                                                           SMI02080
   ELSE DO;  CD5V='0'B;                                                SMI02090
    DO I=1 TO 7;                                                       SMI02100
      SDATA(I+10)=0;                                                   SMI02110
    END;  END;                                                         SMI02120
 END;                                                                  SMI02130
 ELSE IF CD='6' THEN DO;                                               SMI02140
   SKEY=CAPACT || SANUM || MONTH;                                      SMI02150
   SYUGO=YUGO6;  INN=INUM;                                             SMI02160
   SDATA(1)=DAT6(1);                                                   SMI02170
   SDATA(2)=DAT6(2);                                                   SMI02180
 END;                                                                  SMI02190
END SORTRTN;                                                           SMI02200
```

```
CD2RTN: PROC ;                                                          SMI02210
 OGU=GU;  OPUM=JAEPUM;                                                  SMI02220
 JAEPUM=SUBSTR(SKEY,1,5);                                               SMI02230
 GU=SUBSTR(SKEY,6,1);                                                   SMI02240
 SANUM=SUBSTR(SKEY,7,5);                                                SMI02250
 MONTH=SUBSTR(SKEY,12,2);                                               SMI02260
 IF RTYPE='0'B THEN DO;                                                 SMI02270
   PUT PAGE EDIT('PRODUCTS BY ESTABLISHMENT (',CMONTH,')')              SMI02280
                 (X(50),A,F(2),A);  RATE=SRATE; KYUGO=SYUGO;            SMI02290
   OGU=GU;  OPUM=JAEPUM;  RTYPE='1'B;  PUT SKIP; SNUM=SANUM;            SMI02300
 END;                                                                   SMI02310
 ANU=OPUM || SNUM;  KNU=JAEPUM || SANUM;                                SMI02320
 IF ANU¬=KNU THEN DO;                                                   SMI02330
  IF ID='1' THEN DO;                                                    SMI02340
   DO I=1 TO 8;                                                         SMI02350
    IF PDATA(1,I)=0 THEN PDATA(3,I)=0;                                  SMI02360
    ELSE PDATA(3,I)=PDATA(2,I)/PDATA(1,I)*100-100;                      SMI02370
   END;                                                                 SMI02380
   IF FST='0'B THEN                                                     SMI02390
     PUT SKIP(2) EDIT(OPUM,SNUM,RATE,KYUGO,SIZE,IL(1),                  SMI02400
   (PDATA(1,J) DO J=1,4 TO 8),PDATA(3,1),WHY(1),WHY(2))                 SMI02410
 (2 A(7),P'99',X(2),2 A(3),A(3),6 F(10),F(10,1),X(20),A(2),A(1));       SMI02420
   ELSE PUT SKIP(2) EDIT(SNUM,RATE,KYUGO,SIZE,IL(1),(PDATA(1,J)         SMI02430
   DO J=1,4 TO 8),PDATA(3,1),WHY(1),WHY(2))(X(7),A(7),P'99',X(2),       SMI02440
   2 A(3),A(3),6 F(10),F(10,1),X(20),A(2),A(1));                        SMI02450
PUT SKIP EDIT(IL(2),(PDATA(2,J) DO J=1,4 TO 8),(PDATA(3,K) DO K=5,      SMI02460
    6,8))(X(24),A(3),6 F(10),3 F(10,1));                                SMI02470
   FST='1'B;                                                            SMI02480
  END;                                                                  SMI02490
  IF RATE=1 THEN OGU='1'; ELSE OGU='2';                                 SMI02500
  IF SRATE=1 THEN GU='1'; ELSE GU='2';                                  SMI02510
  IF SNUM='06108' THEN DO; SDT=PDATA; GOTO TAKJU;  END;                 SMI02520
  IF OGU='1' THEN TDATA=TDATA+PDATA;                                    SMI02530
  ELSE DO;                                                              SMI02540
    FDATA=FDATA+PDATA;                                                  SMI02550
    ADATA=ADATA+PDATA*RATE;                                             SMI02560
  END;                                                                  SMI02570
  TAKJU: PDATA=0;  SNUM=SANUM; IL=' ';                                  SMI02580
 IF OPUM¬=JAEPUM THEN DO;                                               SMI02590
  IF ID='1' THEN DO;                                                    SMI02600
    CALL JTPUT('S_TOTAL',FDATA);                                        SMI02610
    CALL JTPUT('R*TOTAL',ADATA);                                        SMI02620
  END;                                                                  SMI02630
  TDATA=TDATA+ADATA;                                                    SMI02640
  CALL JTPUT('A_TOTAL',TDATA);                                          SMI02650
  IF OPUM='13050' THEN DO;                                              SMI02660
    CALL JTPUT('A_TOTAL',SDT); TDATA=SDT;                               SMI02670
  END;                                                                  SMI02680
  IF SUBSTR(OPUM,5,1)='0' & ID='3' THEN CALL IDX2;                      SMI02690
  IF SUBSTR(OPUM,5,1)¬='0' THEN DO;                                     SMI02700
   IF OPUM='83378' | OPUM='82118' | OPUM='62068' | OPUM='62088' |       SMI02710
      OPUM='41088' | OPUM='41138' | OPUM='41148' | OPUM='41158' |       SMI02720
      OPUM='41168' | OPUM='51278' | OPUM='55098' | OPUM='55108' |       SMI02730
      OPUM='69098' | OPUM='69108' | OPUM='69118' | OPUM='81078' |       SMI02740
      OPUM='81108' | OPUM='81138' | OPUM='82068' | OPUM='82078' |       SMI02750
```

```
      OPUM='82098' | OPUM='82108' | OPUM='82138' | OPUM='82208' |        SMI02760
      OPUM='82248' | OPUM='82308' | OPUM='82358' | OPUM='82438' |        SMI02770
      OPUM='82198' | OPUM='82178' | OPUM='83048' | OPUM='82428' |        SMI02780
      OPUM='82278' | OPUM='82168' | OPUM='82258' | OPUM='82268' |        SMI02790
      OPUM='85138' | OPUM='85128' | OPUM='82448' |                       SMI02800
  OPUM='82368' | SUBSTR(OPUM,1,2)='01' | SUBSTR(OPUM,1,4)='2201' THEN;  SMI02810
     ELSE DO;                                                            SMI02820
      QDATA=QDATA+TDATA; ZB='0'B; END; END;                              SMI02830
  IF ZB='0'B & SUBSTR(OPUM,1,4)¬=SUBSTR(JAEPUM,1,4) THEN DO;             SMI02840
   CALL JTPUT('0_TOTAL',QDATA);  ZB='1'B; SUBSTR(OPUM,5,1)='0';          SMI02850
   TDATA=QDATA;                                                          SMI02860
   IF ID='3' THEN CALL IDX2;  QDATA=0;                                   SMI02870
  END;                                                                   SMI02880
   NN: IF SUBSTR(OPUM,1,1)¬=SUBSTR(JAEPUM,1,1) THEN PUT PAGE;            SMI02890
   FDATA=0; ADATA=0; TDATA=0; FST='0'B; OPUM=JAEPUM; OGU=GU;             SMI02900
  END;                                                                   SMI02910
  ELSE IF OGU¬=GU & ID='1' THEN DO;                                      SMI02920
    CALL JTPUT('S_TOTAL',TDATA); OGU=GU; PUT SKIP;                       SMI02930
  END;                                                                   SMI02940
  END;                                                                   SMI02950
  RATE=SRATE; SIZE=SSIZE; WHY=SGRT; KYUGO=SYUGO;                         SMI02960
  OGU=GU;                                                                SMI02970
  IF CMONTH=MONTH THEN I=2;                                              SMI02980
  ELSE I=1;                                                              SMI02990
  IL(I)=INN;                                                             SMI03000
  PDATA(I,1)=SDATA(1);                                                   SMI03010
  DO J=2 TO 6;                                                           SMI03020
    PDATA(I,J+2)=SDATA(J);                                               SMI03030
  END;                                                                   SMI03040
  IF SSIZE='1' THEN PDATA(I,2)=PDATA(I,1);                               SMI03050
  ELSE PDATA(I,3)=PDATA(I,1);                                            SMI03060
 JTPUT: PROC(TITLE,DATA);                                                SMI03070
 DCL TITLE          CHAR(7);                                            SMI03080
 DCL DATA(3,17)     FLOAT DEC(16);                                      SMI03090
 DCL PTIT           CHAR(5) INIT((5)' ');                               SMI03100
  DO I=1 TO 8;                                                           SMI03110
   IF DATA(1,I)=0 THEN DATA(3,I)=0;                                      SMI03120
   ELSE DATA(3,I)=DATA(2,I)/DATA(1,I)*100-100;                           SMI03130
  END;                                                                   SMI03140
  IF SUBSTR(TITLE,1,1)='A' THEN PTIT=OPUM;                               SMI03150
  IF SUBSTR(TITLE,1,1)='S' THEN PUT SKIP;                                SMI03160
  PUT SKIP(2) EDIT(PTIT,TITLE,(DATA(1,I) DO I=1 TO 8),(DATA(3,J) DO J=1 SMI03170
  TO 3))(A(6),A(7),X(1),8 F(10),3 F(10,1));                              SMI03180
  PUT SKIP EDIT((DATA(2,I) DO I=1 TO 8),(DATA(3,J) DO J=5,6,8))          SMI03190
  (X(14),8 F(10),3 F(10,1));                                             SMI03200
 END JTPUT;                                                              SMI03210
 IDX2: PROC;                                                             SMI03220
  IF OPUM='22010' THEN DO;                                               SMI03230
     TDATA(1,*)=TDATA(1,*)/PRICE(1)*100;                                 SMI03240
     TDATA(2,*)=TDATA(2,*)/PRICE(2)*100;  END;                           SMI03250
  IF OPUM='52060' THEN DO;                                               SMI03260
    TDATA(1,*)=TDATA(1,*)/PRICE(3)*100;                                  SMI03270
    TDATA(2,*)=TDATA(2,*)/PRICE(8)*100;  END;                            SMI03280
  IF OPUM='82440' THEN DO;                                               SMI03290
    TDATA(1,*)=TDATA(1,*)/PRICE(4)*100;                                  SMI03300
```

```
    TDATA(2,*)=TDATA(2,*)/PRICE(9)*100;  END;                          SMI03310
  IF OPUM='85120' THEN DO;                                             SMI03320
    TDATA(1,*)=TDATA(1,*)/PRICE(5)*100;                                SMI03330
    TDATA(2,*)=TDATA(2,*)/PRICE(10)*100;  END;                         SMI03340
  IF SUBSTR(OPUM,1,1)='0' THEN INDUST='2' || SUBSTR(OPUM,2,1) || '0';  SMI03350
  ELSE IF SUBSTR(OPUM,1,2)='99' THEN INDUST='400';                     SMI03360
  ELSE INDUST='3' || SUBSTR(OPUM,1,2);                                 SMI03370
  OKEY='2A' || INDUST || OPUM;                                         SMI03380
  READ FILE(SUMF) INTO(SUMR) KEY(OKEY);                                SMI03390
  ODATA(36)=TDATA(1,1); ODATA(37)=TDATA(2,1);                          SMI03400
  REWRITE FILE(SUMF) FROM(SUMR) KEY(OKEY);                             SMI03410
  OKEY='2B' || INDUST || OPUM;                                         SMI03420
  READ FILE(SUMF) INTO(SUMR) KEY(OKEY);                                SMI03430
  ODATA(36)=TDATA(1,1); ODATA(37)=TDATA(2,1);                          SMI03440
  REWRITE FILE(SUMF) FROM(SUMR) KEY(OKEY);                             SMI03450
  OKEY='2C' || INDUST || OPUM;                                         SMI03460
  READ FILE(SUMF) INTO(SUMR) KEY(OKEY);                                SMI03470
  ODATA(36)=TDATA(1,2); ODATA(37)=TDATA(2,2);                          SMI03480
  REWRITE FILE(SUMF) FROM(SUMR) KEY(OKEY);                             SMI03490
  OKEY='2D' || INDUST || OPUM;                                         SMI03500
  READ FILE(SUMF) INTO(SUMR) KEY(OKEY);                                SMI03510
  ODATA(36)=TDATA(1,3); ODATA(37)=TDATA(2,3);                          SMI03520
  REWRITE FILE(SUMF) FROM(SUMR) KEY(OKEY);                             SMI03530
  OKEY='2E' || INDUST || OPUM;                                         SMI03540
  READ FILE(SUMF) INTO(SUMR) KEY(OKEY);                                SMI03550
  ODATA(36)=TDATA(1,5)+TDATA(1,6); ODATA(37)=TDATA(2,5)+TDATA(2,6);    SMI03560
  REWRITE FILE(SUMF) FROM(SUMR) KEY(OKEY);                             SMI03570
  OKEY='2F' || INDUST || OPUM;                                         SMI03580
  READ FILE(SUMF) INTO(SUMR) KEY(OKEY);                                SMI03590
  ODATA(36)=TDATA(1,5); ODATA(37)=TDATA(2,5);                          SMI03600
  REWRITE FILE(SUMF) FROM(SUMR) KEY(OKEY);                             SMI03610
  OKEY='2G' || INDUST || OPUM;                                         SMI03620
  READ FILE(SUMF) INTO(SUMR) KEY(OKEY);                                SMI03630
  ODATA(36)=TDATA(1,6); ODATA(37)=TDATA(2,6);                          SMI03640
  REWRITE FILE(SUMF) FROM(SUMR) KEY(OKEY);                             SMI03650
  OKEY='2H' || INDUST || OPUM;                                         SMI03660
  READ FILE(SUMF) INTO(SUMR) KEY(OKEY);                                SMI03670
  ODATA(36)=TDATA(1,8); ODATA(37)=TDATA(2,8);                          SMI03680
  REWRITE FILE(SUMF) FROM(SUMR) KEY(OKEY);                             SMI03690
 END IDX2;                                                             SMI03700
 END CD2RTN;                                                           SMI03710
 CD3RTN: PROC;                                                         SMI03720
  WON=SUBSTR(SKEY,1,7);                                                SMI03730
  SANUM=SUBSTR(SKEY,8,5);                                              SMI03740
  GU=SUBSTR(SKEY,13,1);                                                SMI03750
  MONTH=SUBSTR(SKEY,14,2);                                             SMI03760
  IF RTYPE='0'B THEN DO;                                               SMI03770
    PUT PAGE EDIT('RAW MATERIAL BY ESTABLISHMENT (',CMONTH,')')        SMI03780
                 (X(28),A,F(2),A);                                     SMI03790
    PUT SKIP;                                                          SMI03800
    OWON=WON; RTYPE='1'B;  YUGO3=SYUGO; SNUM=SANUM; OGU=GU;            SMI03810
  END;                                                                 SMI03820
  IF WON || SANUM || GU¬=OWON || SNUM || OGU THEN DO;                  SMI03830
     IF PDATA(J,1)=0 THEN PDATA(J,3)=0;                                SMI03840
     ELSE PDATA(J,3)=PDATA(J,2)/PDATA(J,1)*100-100;                    SMI03850
```

```
     IF PDATA(J,4)=0 THEN PDATA(J,6)=0;                                  SMI03860
     ELSE PDATA(J,6)=PDATA(J,5)/PDATA(J,4)*100-100;                      SMI03870
    IF ID='3' THEN                                                       SMI03880
    PUT SKIP EDIT(SUBSTR(OWON,1,3),SUBSTR(OWON,4,4),SNUM,OGU,YUGO3,      SMI03890
      (PDATA(J,I) DO I=1 TO 6))(A(5),2 A(7),A(4),A(3),                   SMI03900
       2(2 F(10),F(10,1)));                                              SMI03910
  SNUM=SANUM;  OGU=GU;  PDATA=0;                                         SMI03920
  IF OWON¬=WON THEN DO;                                                  SMI03930
    TDATA(3,*)=TDATA(1,*)+TDATA(2,*);                                    SMI03940
   DO K=1 TO 3;                                                          SMI03950
    IF TDATA(K,1)=0 THEN TDATA(K,3)=0;                                   SMI03960
    ELSE TDATA(K,3)=TDATA(K,2)/TDATA(K,1)*100-100;                       SMI03970
    IF TDATA(K,4)=0 THEN TDATA(K,6)=0;                                   SMI03980
    ELSE TDATA(K,6)=TDATA(K,5)/TDATA(K,4)*100-100;                       SMI03990
   END;                                                                  SMI04000
    PUT SKIP(2) EDIT(SUBSTR(OWON,1,3),SUBSTR(OWON,4,4),                  SMI04010
                     'TOTAL','1',(TDATA(1,I) DO I=1 TO 6))               SMI04020
      (A(5),A(7),A(7),A(4),X(3),2(2 F(10),F(10,1)));                     SMI04030
    PUT SKIP EDIT('2',(TDATA(2,I) DO I=1 TO 6))(X(19),A(4),              SMI04040
              X(3),2(2 F(10),F(10,1)));                                  SMI04050
    PUT SKIP EDIT('3',(TDATA(3,I) DO I=1 TO 6))(X(19),A(4),              SMI04060
             X(3),2(2 F(10),F(10,1)));                                   SMI04070
    IF ID='3' THEN CALL IDX3;                                            SMI04080
    IF SUBSTR(OWON,2,1)¬=SUBSTR(WON,2,1) THEN PUT PAGE;                  SMI04090
    PUT SKIP;                                                            SMI04100
    TDATA=0; OWON=WON;                                                   SMI04110
  END;                                                                   SMI04120
  END;                                                                   SMI04130
  YUGO3=SYUGO;                                                           SMI04140
  IF CMONTH=MONTH THEN I=2;                                              SMI04150
  ELSE I=1;                                                              SMI04160
  IF GU='1' THEN J=1;                                                    SMI04170
  ELSE J=2;                                                              SMI04180
  PDATA(J,I)=SDATA(1);                                                   SMI04190
  PDATA(J,I+3)=SDATA(2);                                                 SMI04200
  TDATA(J,I)=TDATA(J,I)+PDATA(J,I);                                      SMI04210
  TDATA(J,I+3)=TDATA(J,I+3)+PDATA(J,I+3);                                SMI04220
  IDX3: PROC;                                                            SMI04230
  DCL Q   PIC'9';                                                        SMI04240
   DO Q=1 TO 3;                                                          SMI04250
    OKEY='31' || WON || Q;                                               SMI04260
    READ FILE(SUMF) INTO(SUMR) KEY(OKEY);                                SMI04270
    ODATA(36)=TDATA(Q,1); ODATA(37)=TDATA(Q,4);                          SMI04280
    REWRITE FILE(SUMF) FROM(SUMR) KEY(OKEY);                             SMI04290
    OKEY='32' || WON || Q;                                               SMI04300
    READ FILE(SUMF) INTO(SUMR) KEY(OKEY);                                SMI04310
    ODATA(36)=TDATA(Q,2); ODATA(37)=TDATA(Q,5);                          SMI04320
    REWRITE FILE(SUMF) FROM(SUMR) KEY(OKEY);                             SMI04330
   END;                                                                  SMI04340
  END IDX3;                                                              SMI04350
 END CD3RTN;                                                             SMI04360
 CD4RTN: PROC;                                                           SMI04370
  INDUST=SUBSTR(SKEY,1,3);                                               SMI04380
  GU=SUBSTR(SKEY,4,1);                                                   SMI04390
  SANUM=SUBSTR(SKEY,5,5);                                                SMI04400
```

```
MONTH=SUBSTR(SKEY,10,2);                                                SMI04410
IF RTYPE='0'B THEN DO ;                                                 SMI04420
  PUT PAGE EDIT('WORKER & SALARY BY ESTABLISHMENT (',CMONTH,')')        SMI04430
    (X(46),A,F(2),A);  YUGO3=SYUGO;  RATE=SRATE;                        SMI04440
  OGU=GU; SINDUST=INDUST; RTYPE='1'B; PUT SKIP; SNUM=SANUM;             SMI04450
END;                                                                    SMI04460
 INDUST=TRANSLATE(INDUST,'0',' ');                                      SMI04470
IF INDUST || SANUM¬=SINDUST || SNUM THEN DO;                            SMI04480
 IF ID='1' THEN DO;                                                     SMI04490
  DO I=1 TO 4;                                                          SMI04500
   PDATA(3,I)=PDATA(2,I)-PDATA(1,I);                                    SMI04510
   IF PDATA(1,I+10)+PDATA(1,I+6)=0 THEN PDATA(3,I+10)=0;  ELSE          SMI04520
   PDATA(3,I+10)=(PDATA(2,I+10)+PDATA(2,I+6))/(PDATA(1,I+10)+PDATA(1,   SMI04530
   I+6))*100-100;                                                       SMI04540
  END;                                                                  SMI04550
  PDATA(3,5)=PDATA(2,5)-PDATA(1,5);                                     SMI04560
  IF PDATA(1,1)=0 THEN PDATA(3,7)=0;                                    SMI04570
  ELSE PDATA(3,7)=PDATA(2,1)/PDATA(1,1)*100-100;                        SMI04580
  IF PDATA(1,3)=0 THEN PDATA(3,8)=0;                                    SMI04590
  ELSE PDATA(3,8)=PDATA(2,3)/PDATA(1,3)*100-100;                        SMI04600
  IF PDATA(1,4)=0 THEN PDATA(3,9)=0;                                    SMI04610
  ELSE PDATA(3,9)=PDATA(2,4)/PDATA(1,4)*100-100;                        SMI04620
  IF PDATA(1,5)=0 THEN DO; PDATA(3,10)=0;                               SMI04630
    PDATA(3,6)=0;  PDATA(3,15)=0;  END;                                 SMI04640
  ELSE DO; PDATA(3,10)=PDATA(2,5)/PDATA(1,5)*100-100;                   SMI04650
   PDATA(3,6)=PDATA(2,6)/PDATA(1,5)*100;                                SMI04660
   PDATA(3,15)=PDATA(2,15)/PDATA(1,5)*100;                              SMI04670
  END;                                                                  SMI04680
  IF PDATA(1,16)=0 THEN PDATA(3,16)=0;                                  SMI04690
  ELSE PDATA(3,16)=PDATA(2,16)/PDATA(1,16)*100-100;                     SMI04700
  IF PDATA(1,17)=0 THEN PDATA(3,17)=0;                                  SMI04710
  ELSE PDATA(3,17)=PDATA(2,17)/PDATA(1,17)*100-100;                     SMI04720
 PUT SKIP(2) EDIT(SINDUST,SNUM,RATE,YUGO3,((PDATA(J,K) DO K=1,7,11,     SMI04730
   6) DO J=1,2,3))(2 A(7),P'99',X(2),A(3),9 F(9),3 F(9,1));             SMI04740
  PUT SKIP EDIT(((PDATA(J,K) DO K=2,8,12,15) DO J=1,2),(PDATA(3,I)      SMI04750
  DO I=2,12,15))(X(21),8 F(9),F(9),  X(9),2 F(9,1));                    SMI04760
  PUT SKIP EDIT(((PDATA(J,K) DO K=3,9,13,16) DO J=1,2),(PDATA(3,I)      SMI04770
  DO I=3,8,13,16))(X(21),9 F(9),3 (F(9,1)));                            SMI04780
  PUT SKIP EDIT(((PDATA(J,K) DO K=4,17) DO J=1,2),(PDATA(3,I)           SMI04790
 DO I=4,9,17))(X(21),2 (F(9),X(18),F(9)),F(9),F(9,1),X(9),F(9,1));      SMI04800
  PUT SKIP EDIT(((PDATA(J,K) DO K=5,10,14) DO J=1,2,3))                 SMI04810
  (X(21),2(3 F(9),X(9)),F(9),2 F(9,1));                                 SMI04820
 END ;                                                                  SMI04830
 IF OGU='1' THEN TDATA=TDATA+PDATA;                                     SMI04840
 ELSE DO;                                                               SMI04850
   FDATA=FDATA+PDATA;                                                   SMI04860
   ADATA=ADATA+PDATA* RATE;                                             SMI04870
 END;                                                                   SMI04880
 PDATA=0;  SNUM=SANUM;                                                  SMI04890
IF SINDUST¬=INDUST THEN DO;                                             SMI04900
  IF ID='1' THEN DO;                                                    SMI04910
    CALL WTPUT('S_TOTAL',FDATA);                                        SMI04920
    CALL WTPUT('R*TOTAL',ADATA);                                        SMI04930
  END;                                                                  SMI04940
  TDATA=TDATA+ADATA;                                                    SMI04950
```

```
    CALL WTPUT('A_TOTAL',TDATA);                                        SMI04960
    IF ID='3' THEN CALL IDX4;                                           SMI04970
    IF SUBSTR(SINDUST,2,1)¬=SUBSTR(INDUST,2,1) THEN PUT PAGE;           SMI04980
    FDATA=0; ADATA=0; TDATA=0; SINDUST=INDUST;  OGU=GU;                 SMI04990
  END;                                                                  SMI05000
  ELSE IF OGU¬=GU & ID='1' THEN DO; CALL WTPUT('S_TOTAL',TDATA);        SMI05010
    OGU=GU; PUT SKIP; END;                                              SMI05020
  END;                                                                  SMI05030
  RATE=SRATE;  YUGO3=SYUGO; OGU=GU;                                     SMI05040
  IF CMONTH=MONTH THEN I=2;                                             SMI05050
  ELSE I=1;                                                             SMI05060
  PDATA(I,*)=SDATA(*);                                                  SMI05070
 WTPUT : PROC(TITLE,DATA);                                              SMI05080
 DCL TITLE            CHAR(7);                                          SMI05090
 DCL DATA(3,17)       FLOAT DEC(16);                                    SMI05100
 DCL PTIT             CHAR(5) INIT((5)' ');                             SMI05110
    DO I=1 TO 4;                                                        SMI05120
     DATA(3,I)=DATA(2,I)-DATA(1,I);                                     SMI05130
     IF DATA(1,I+10)+DATA(1,I+6)=0 THEN DATA(3,I+10)=0;  ELSE           SMI05140
     DATA(3,I+10)=(DATA(2,I+10)+DATA(2,I+6))/(DATA(1,I+10)+DATA(1,      SMI05150
     I+6))*100-100;                                                     SMI05160
    END;                                                                SMI05170
    DATA(3,5)=DATA(2,5)-DATA(1,5);                                      SMI05180
    IF DATA(1,1)=0 THEN DATA(3,7)=0;                                    SMI05190
    ELSE DATA(3,7)=DATA(2,1)/DATA(1,1)*100-100;                         SMI05200
    IF DATA(1,3)=0 THEN DATA(3,8)=0;                                    SMI05210
    ELSE DATA(3,8)=DATA(2,3)/DATA(1,3)*100-100;                         SMI05220
    IF DATA(1,4)=0 THEN DATA(3,9)=0;                                    SMI05230
    ELSE DATA(3,9)=DATA(2,4)/DATA(1,4)*100-100;                         SMI05240
    IF DATA(1,5)=0 THEN DO; DATA(3,10)=0;                               SMI05250
      DATA(3,6)=0;  DATA(3,15)=0;  END;                                 SMI05260
    ELSE DO; DATA(3,10)=DATA(2,5)/DATA(1,5)*100-100;                    SMI05270
     DATA(3,6)=DATA(2,6)/DATA(1,5)*100;                                 SMI05280
     DATA(3,15)=DATA(2,15)/DATA(1,5)*100;                               SMI05290
    END;                                                                SMI05300
    IF DATA(1,16)=0 THEN DATA(3,16)=0;                                  SMI05310
    ELSE DATA(3,16)=DATA(2,16)/DATA(1,16)*100-100;                      SMI05320
    IF DATA(1,17)=0 THEN DATA(3,17)=0;                                  SMI05330
    ELSE DATA(3,17)=DATA(2,17)/DATA(1,17)*100-100;                      SMI05340
 IF SUBSTR(TITLE,1,1)='A' THEN PTIT=SINDUST;                            SMI05350
  PUT SKIP(2) EDIT(PTIT,TITLE,((DATA(J,K) DO K=1,7,11,6) DO J=1,2,3))   SMI05360
   (A(7),A(7),X(7),9 F(9),3 F(9,1));                                    SMI05370
  PUT SKIP EDIT(((DATA(J,K) DO K=2,8,12,15) DO J=1,2),(DATA(3,I) DO I=  SMI05380
   2,12,15))(X(21),9 F(9),X(9),2 F(9,1));                               SMI05390
  PUT SKIP EDIT(((DATA(J,K) DO K=3,9,13,16) DO J=1,2),(DATA(3,I) DO I=  SMI05400
   3,8,13,16))(X(21),9 F(9),3 F(9,1));                                  SMI05410
  PUT SKIP EDIT(((DATA(J,K) DO K=4,17) DO J=1,2),(DATA(3,I) DO I=4,9,   SMI05420
   17))(X(21),2 (F(9),X(18),F(9)),F(9),F(9,1),X(9),F(9,1));             SMI05430
  PUT SKIP EDIT(((DATA(J,K) DO K=5,10,14) DO J=1,2,3))                  SMI05440
   (X(21),2(3 F(9),X(9)),F(9),2 F(9,1));                                SMI05450
 END WTPUT;                                                             SMI05460
 IDX4:PROC;                                                             SMI05470
 CALL JUPD('1','1',1,3);                                                SMI05480
 CALL JUPD('1','2',5,4);                                                SMI05490
 CALL JAPD('1','3',6);                                                  SMI05500
```

```
 CALL JAPD('1','4',15);                                                  SMI05510
 CALL JAPD('1','5',16);                                                  SMI05520
 CALL JAPD('1','7',17);                                                  SMI05530
 CALL JAPD('2','1',7);                                                   SMI05540
 CALL JAPD('2','2',9);                                                   SMI05550
 CALL JAPD('2','3',10);                                                  SMI05560
 CALL JAPD('3','1',7);                                                   SMI05570
 CALL JAPD('3','2',9);                                                   SMI05580
 CALL JAPD('3','3',10);                                                  SMI05590
 JUPD: PROC(A,B,C,D);                                                    SMI05600
 DCL A     CHAR(1);                                                      SMI05610
 DCL B     CHAR(1);                                                      SMI05620
 DCL C     PIC'99';                                                      SMI05630
 DCL D     PIC'99';                                                      SMI05640
  OKEY='40' || SINDUST || A || B;                                        SMI05650
  READ FILE(SUMF) INTO(SUMR) KEY(OKEY);                                  SMI05660
  ODATA(36)=TDATA(1,C); JDATA(36)=TDATA(1,D);                            SMI05670
  ODATA(37)=TDATA(2,C); JDATA(37)=TDATA(2,D);                            SMI05680
  REWRITE FILE(SUMF) FROM(SUMR) KEY(OKEY);                               SMI05690
 END JUPD;                                                               SMI05700
 JAPD: PROC(A,B,C);                                                      SMI05710
 DCL A     CHAR(1);                                                      SMI05720
 DCL B     CHAR(1);                                                      SMI05730
 DCL C     PIC'99';                                                      SMI05740
  OKEY='40' || SINDUST || A || B;                                        SMI05750
  READ FILE(SUMF) INTO(SUMR) KEY(OKEY);                                  SMI05760
  IF A='2' | A='1' THEN DO;                                              SMI05770
  ODATA(36)=TDATA(1,C); ODATA(37)=TDATA(2,C);                            SMI05780
  JDATA(36)=0;          JDATA(37)=0;          END;                       SMI05790
  ELSE DO;                                                               SMI05800
  ODATA(36)=TDATA(1,C)-TDATA(1,C+4); ODATA(37)=TDATA(2,C)-TDATA(2,C+4);  SMI05810
  JDATA(36)=0;     JDATA(37)=0;                                          SMI05820
  END;                                                                   SMI05830
  REWRITE FILE(SUMF) FROM(SUMR) KEY(OKEY);                               SMI05840
 END JAPD;                                                               SMI05850
 END IDX4;                                                               SMI05860
 END CD4RTN;                                                             SMI05870
 CD6RTN: PROC ;                                                          SMI05880
  CAPACT=SUBSTR(SKEY,1,5);                                               SMI05890
  SANUM=SUBSTR(SKEY,6,5);                                                SMI05900
  MONTH=SUBSTR(SKEY,11,2);                                               SMI05910
  IF RTYPE='0'B THEN DO;                                                 SMI05920
    PUT PAGE EDIT('CAPACITY BY ESTABLISHMENT (',CMONTH,')')              SMI05930
       (X(23),A,F(2),A);                                                 SMI05940
    OPUM=CAPACT; RTYPE='1'B; PUT SKIP; SNUM=SANUM; YUGO3=SYUGO;          SMI05950
  END;                                                                   SMI05960
  CAPACT=TRANSLATE(CAPACT,'0',' ');                                      SMI05970
  IF OPUM || SNUM¬=CAPACT || SANUM THEN DO;                              SMI05980
   IF ID='1' THEN DO;                                                    SMI05990
    IF PDATA(1,1)=0 THEN PDATA(1,3)=0;                                   SMI06000
    ELSE PDATA(1,3)=PDATA(1,2)/PDATA(1,1)*100-100;                       SMI06010
    IF PDATA(1,4)=0 THEN PDATA(1,6)=0;                                   SMI06020
    ELSE PDATA(1,6)=PDATA(1,5)/PDATA(1,4)*100-100;                       SMI06030
    PUT SKIP EDIT(OPUM,SNUM,YUGO3,IL(1),IL(2),(PDATA(1,J) DO             SMI06040
      J=1 TO 6))(2 A(7),A(3),2 A(4),2(2 F(10),F(10,1)));                 SMI06050
```

```
   END ;                                                                SMI06060
    TDATA=TDATA+PDATA;                                                  SMI06070
    PDATA=0;  SNUM=SANUM;  IL=' ';                                      SMI06080
  IF OPUM¬=CAPACT THEN DO;                                              SMI06090
    TDATA(1,3)=TDATA(1,2)/TDATA(1,1)*100-100;                           SMI06100
    TDATA(1,6)=TDATA(1,5)/TDATA(1,4)*100-100;                           SMI06110
    TDATA(1,7)=TDATA(1,4)/TDATA(1,1)*100;                               SMI06120
    TDATA(1,8)=TDATA(1,5)/TDATA(1,2)*100;                               SMI06130
    TDATA(1,9)=TDATA(1,8)-TDATA(1,7);                                   SMI06140
    PUT SKIP(2) EDIT(OPUM,'TOTAL',(TDATA(1,I) DO I=1 TO 9))             SMI06150
      (A(14),A(11),2(2 F(10),F(10,1)),3 F(10,1));                       SMI06160
    IF ID='3' THEN CALL IDX6;                                           SMI06170
    IF SUBSTR(OPUM,1,1)¬=SUBSTR(CAPACT,1,1) THEN PUT PAGE;              SMI06180
    TDATA=0; OPUM=CAPACT;  PUT SKIP;                                    SMI06190
  END;                                                                  SMI06200
  END;                                                                  SMI06210
  YUGO3=SYUGO;                                                          SMI06220
  IF CMONTH=MONTH THEN I=2;                                             SMI06230
  ELSE I=1;                                                             SMI06240
  IL(I)=INN;                                                            SMI06250
  PDATA(1,I)=SDATA(1);                                                  SMI06260
  PDATA(1,I+3)=SDATA(2);                                                SMI06270
  IDX6: PROC;                                                           SMI06280
  IF SUBSTR(OPUM,1,1)='0' THEN INDUST='2' || SUBSTR(OPUM,2,1) || '0';   SMI06290
  ELSE IF SUBSTR(OPUM,1,2)='99' THEN INDUST='4';                        SMI06300
  ELSE INDUST='3' || SUBSTR(OPUM,1,2);                                  SMI06310
  OKEY='61' || INDUST || OPUM;                                          SMI06320
  READ FILE(SUMF) INTO(SUMR) KEY(OKEY);                                 SMI06330
  ODATA(36)=TDATA(1,1); ODATA(37)=TDATA(1,2);                           SMI06340
  REWRITE FILE(SUMF) FROM(SUMR) KEY(OKEY);                              SMI06350
  OKEY='62' || INDUST || OPUM;                                          SMI06360
  READ FILE(SUMF) INTO(SUMR) KEY(OKEY);                                 SMI06370
  ODATA(36)=TDATA(1,4);  ODATA(37)=TDATA(1,5);                          SMI06380
  REWRITE FILE(SUMF) FROM(SUMR) KEY(OKEY);                              SMI06390
 END IDX6;                                                              SMI06400
 END CD6RTN;                                                            SMI06410
 CLOSE FILE(SUMF),FILE(EDIT2),FILE(EDIT3);                              SMI06420
  GOTO RRC;                                                             SMI06430
 LLT:                                                                   SMI06440
 END SUMMARY;                                                           SMI06450
/*                                                                      SMI06460
//LKED.SYSLMOD DD DSN=USER.LOADLIB(SMIWN66),DISP=SHR                    SMI06470
//                                                                      SMI06480
```

SMICIX66

지 수 시 산

(CATAL PROGRAM)

```
//SMICIX66 JOB CLASS=6,TYPRUN=HOLD
//   EXEC PLIFCL
//PLI.SYSIN DD *
* PROCESS GS,NEST,S,A,XREF,OPT(2);                                    AA 00010
  JISURTN: PROC OPTIONS(MAIN);                                        AA 00020
 DCL SUMM FILE RECORD KEYED ENV(VSAM);                                AA 00030
 DCL 1 SUMREC,                        /*********************************/AA 00040
      2 OKEY        CHAR(10),         /***** 37MONTH JISU GYESAN  *****/AA 00050
      2 SPC         CHAR(1),          /*********************************/AA 00060
      2 OYEAR       PIC'99',                                           AA 00070
      2 OMONTH      PIC'99',                                           AA 00080
      2 WEIGHT      FLOAT,                                             AA 00090
      2 BASEM  FIXED BIN(31),                                          AA 00100
      2 DATAO(37),                                                     AA 00110
        3 ODATA FIXED BIN(31),                                         AA 00120
        3 JDATA FLOAT,                                                 AA 00130
      2 SIGN        CHAR(1);                                           AA 00140
 DCL 1 KKEY DEF OKEY,                                                  AA 00160
      2 CD    CHAR(1),                                                 AA 00170
      2 GU    CHAR(1),                                                 AA 00180
      2 INDUST CHAR(3),                                                AA 00190
      2 PUM   CHAR(5);                                                 AA 00200
 DCL YUJISU(9,43,37)      FLOAT DEC(16);                               AA 00210
 DCL IYUL(453,75)         FLOAT DEC(16);                               AA 00220
 DCL SPCJISU(9,5,37)      FLOAT DEC(16);                               AA 00230
                          /************ SPECIAL JACKUP *********/
                          /*   SPCJISU (A,B,C)                  */
                          /*   A MEAN    SPECIAL BUNRYU         */
                          /*   B MEAN  GUBUN  J,K,L,S,T         */
                          /*   C MEAN  LAST MONTH , THIS MONTH */
                          /************ (84.5.10) **************/
 DCL HRJISU(28,3,37)      FLOAT DEC(16);                               AA 00240
 DCL HRDAT(0:2,37)      FLOAT DEC(16);                                 AA 00250
 DCL BASE                 FLOAT DEC(16);                               AA 00260
 DCL DATA(37)             FLOAT DEC(16);                               AA 00270
 DCL JISU                 PIC'99999V9';                                AA 00280
 DCL A10(25) CHAR(5) INIT('62050','81090','81100','81150','82070',
  '82090','82100','82120','82130','82140','82150','82160','82170',
  '82190','82200','82250','82270','82280','82290','82300','82310',
  '82340','82350','84060','84070');
 DCL GUB(20) CHAR(1) INIT('A','B','C','D','E','F','G','H','I','J','K',  AA 00290
     'L','M','N','O','P','Q','R','S','T');                             AA 00300
 DCL HRCD(28)  CHAR(4) INIT('2311','2313','2314','2321','2322','2323',  AA 00310
   '2324','2331','2332','1341','2342','1351','1352','1353','1354',     AA 00320
   '2355','2356','1361','1362','1369','1371','1372','1381','1382',     AA 00330
   '1383','1384','1385','2390');                                       AA 00340
 DCL OGU                  CHAR(1);                                    AA 00350
 DCL OINDUST              CHAR(3);                                    AA 00360
 DCL SOSUM(37)          FLOAT DEC(16),                                AA 00370
     JUSUM(37)          FLOAT DEC(16),                                AA 00380
     DASUM(37)          FLOAT DEC(16),                                AA 00390
     CHSUM(37)          FLOAT DEC(16);                                AA 00400
 DCL Q                    PIC'99';                                    AA 00410
 DCL EXCP(3,37)         FLOAT DEC(16);                                AA 00420
 DCL SUMDAT(9,18,37)      FLOAT DEC(16);                              AA 00430
```

```
 DCL YUCD(43) CHAR(3) INIT('210','230','290','200','311','313','314',   AA 00440
   '310','321','322','323','324','320','331','332','330','341','342',   AA 00450
   '340','351','352','353','354','355','356','350','361','362','369',   AA 00460
   '360','371','372','370','381','382','383','384','385','380','390',   AA 00470
   '300','400','000');                                                 AA 00480
 DCL SPCD(9)  CHAR(3) INIT('000','800','810','820','821','822','900',   AA 00490
   '910','920');                                                       AA 00500
 DCL INT                  BIT(1) INIT('1'B);                           AA 00510
 YUJISU=0; IYUL=0; SPCJISU=0; HRJISU=0; HRDAT=0; BASE=0; DATA=0;
 SOSUM=0; JUSUM=0; DASUM=0; CHSUM=0; SUMDAT=0;  EXCP=0;
  ON KEY(SUMM) BEGIN;
   IF ONCODE=51 THEN PUT EDIT(OKEY,'NOT FOUND')(A(10),A(10));
  END;
  OPEN FILE(SUMM) SEQUENTIAL UPDATE;                                   AA 00580
  READ FILE(SUMM) INTO(SUMREC);                                        AA 00590
  OGU=GU;  J=0; I=0;   BASE=BASEM;                                     AA 00600
   DATA(*)=ODATA(*) ;  CALL MUL10;
  DO UNTIL(GU='I');  I=I+1;                                            AA 00610
    DO WHILE(OGU=GU);                                                  AA 00620
      IF SUBSTR(PUM,5,1)='0' & PUM¬='00000' THEN DO;                   AA 00630
        IF INT THEN DO; OINDUST=INDUST; INT='0'B; END;
        IF BASE=0 THEN DATA=0;                                         AA 00700
        ELSE DATA=DATA/BASE*100;                                       AA 00710
      DO LEE = 1 TO 37 ;
        JISU=DATA(LEE)+0.05; DATA(LEE)=JISU; JDATA(LEE)=JISU;              AA
      END;
        REWRITE FILE(SUMM) FROM(SUMREC);                               AA 00740
        IF I=8 THEN DO;                                                AA 00760
          J=J+1;                                                       AA 00770
        DO LEE = 1 TO 37;
          IYUL(J,LEE+1)=DATA(LEE);                                         AA 0
        END;
          IYUL(J,1)=WEIGHT;                                            AA 00800
        END;                                                           AA 00810
        ELSE IF I=5  THEN DO;                                          AA 00820
          J=J+1;                                                       AA 00830
        DO LEE = 1 TO 37;
          IYUL(J,LEE+38)=DATA(LEE);                                        AA
        END;                                                           AA 00860
        END;                                                           AA 00860
        IF OINDUST¬=INDUST THEN DO;                                    AA 00870
         IF SUBSTR(OINDUST,3,1)='0' THEN CALL CALRTN(JUSUM,DASUM);     AA 00880
         ELSE DO;  CALL CALRTN(SOSUM,JUSUM);                           AA 00880
          IF SUBSTR(OINDUST,1,2)¬=SUBSTR(INDUST,1,2) THEN              AA 00890
            CALL CALRTN(JUSUM,DASUM); END;                             AA 00900
            IF SUBSTR(OINDUST,1,1)¬=SUBSTR(INDUST,1,1) THEN            AA 00910
            CALL CALRTN(DASUM,CHSUM);                                  AA 00920
          OINDUST=INDUST;
        END;                                                           AA 00940
IF SUBSTR(INDUST,1,1)='2' | INDUST='390' THEN JUSUM=JUSUM+DATA*WEIGHT; AA 00950
        ELSE IF SUBSTR(INDUST,1,1)='3' THEN SOSUM=SOSUM+DATA*WEIGHT;   AA 00960
        ELSE DASUM=DASUM+DATA*WEIGHT;                                  AA 00970
        /*************************************************************/
        /********************* SPECIAL JACKUP ************************/
        /*************************************************************/
```

```
        IF I = 1 THEN L = 1 ;                          /*  SAING SAN ******/
        ELSE IF I = 5 THEN L =2 ;                      /*  CHUL HA   ******/
        ELSE IF I = 8 THEN L =3 ;                      /*  JAI GO    ******/
        ELSE IF I = 6 THEN L =4 ;                      /*  NAI SU    ******/
        ELSE IF I = 7 THEN L =5 ;                      /*  SU CHUL   ******/
        IF I=1 | I=5 | I=8 | I=6 | I=7 THEN DO ; /*******************/  AA 00980
        IF SPC¬=' ' THEN DO;                           /*******************/  AA 00990
          Q=SPC;                                       /*******************/  AA 01000
          SPCJISU(Q,L,*)=SPCJISU(Q,L,*)+DATA(*)*WEIGHT;  /***********/  AA 01010
        END; END ;                                     /***********/  AA 01020
        /*************************************************************/
        /*********************** SPECIAL *************************/
        /*************************************************************/
        IF I=1 | I=5 | I=8 THEN DO ;
        IF PUM)'84010' & PUM('84090' THEN EXCP(L,*)=EXCP(L,*)+DATA(*)*  AA 01030
         WEIGHT;                                                        AA 01040
      END;   END;                                                       AA 01070
      READ FILE(SUMM) INTO(SUMREC);                                     AA 01080
      DATA(*)=ODATA(*);                                                 AA 01090
      BASE=BASEM;  CALL MUL10;
    END;                                                                AA 01100
    CALL CALRTN(DASUM,CHSUM);                                           AA 01110
    CALL CALRTN(CHSUM,CHSUM); CHSUM=0;                                  AA 01110
    OGU=GU; J=0;  K=0;  INT='1'B;                                       AA 01120
  END;                                                                  AA 01140
  L=0;  I=9; OINDUST=INDUST;                                            AA 01150
  DO WHILE(GU='I');                                                     AA 01160
    IF PUM¬='00000' THEN DO;                                            AA 01170
      IF INT THEN DO;  INT='0'B; OINDUST=INDUST; END;
      L=L+1;  IF WEIGHT=0 THEN GOTO JYY;                                AA 01180
    DO LEE = 1 TO 37;
      IF IYUL(L,LEE+38)=0 THEN DATA(LEE)=0;
      ELSE DATA(LEE)=IYUL(L,LEE+1)/IYUL(L,LEE+38)*100;
    END;
    DO LEE = 1 TO 37;
      JISU=DATA(LEE)+0.05; JDATA(LEE)=JISU; DATA(LEE)=JISU;                AA
   END;
      REWRITE FILE(SUMM) FROM(SUMREC);                                  AA 01230
    END;                                                                AA 01410
   JYY: READ FILE(SUMM) INTO(SUMREC);                                   AA 01430
  END;                                                                  AA 01440
   CALL CALRTN(DASUM,CHSUM);
   CALL CALRTN(CHSUM,CHSUM);
  CLOSE FILE(SUMM);                                                     AA 01450
  OPEN FILE(SUMM) DIRECT UPDATE; J=0;                                   AA 01460
  DO Q=1 TO 8;  K=0;                                                    AA 01470
    IF Q=1 | Q=5 | Q=8 THEN J=J+1;
    DO I=1 TO 43;                                                       AA 01480
      OKEY='2' || GUB(Q) || YUCD(I);                                    AA 01490
      OKEY=TRANSLATE(OKEY,'0',' ');
      READ FILE(SUMM) INTO(SUMREC) KEY(OKEY);                           AA 01500
      IF WEIGHT=0 THEN DATA=0;
      ELSE DATA(*)=YUJISU(Q,I,*)/WEIGHT;                                AA 01510
    DO LEE = 1 TO 37;
      JISU=DATA(LEE)+0.05; JDATA(LEE)=JISU;                                AA
```

```
    END;
      PUT SKIP EDIT(OKEY,WEIGHT,(JDATA(Z) DO Z=35 TO 37))
         (A(10),4 F(10,1));
      REWRITE FILE(SUMM) FROM(SUMREC) KEY(OKEY);                          AA 01540
      IF Q=1 | Q=5 | Q=8 THEN DO;                                         AA 01550
      IF SUBSTR(INDUST,3,1)¬='0'  THEN DO;                                AA 01560
        K=K+1;                                                            AA 01570
        HRJISU(K,J,*)=YUJISU(Q,I,*);                                      AA 01580
      END;                                                                AA 01590
      ELSE IF INDUST='390'  THEN DO;                                      AA 01560
        K=K+1;                                                            AA 01570
        HRJISU(K,J,*)=YUJISU(Q,I,*);                                      AA 01580
      END;                                                                AA 01590
     IF YUCD(I)='000' | YUCD(I)='300' | YUCD(I)='380' | YUCD(I)='384'     AA 01600
      THEN DO;                                                            AA 01610
      OKEY='2' || GUB(J+15) || YUCD(I);                                   AA 01620
      OKEY=TRANSLATE(OKEY,'0',' ');
      READ FILE(SUMM) INTO(SUMREC) KEY(OKEY);                             AA 01630
      IF WEIGHT=0 THEN DATA=0;
      ELSE DATA(*)=(YUJISU(Q,I,*)-EXCP(J,*))/WEIGHT;                      AA 01640
    DO LEE = 1 TO 37;
      JISU=DATA(LEE)+0.05; JDATA(LEE)=JISU;                                  AA
    END;
      REWRITE FILE(SUMM) FROM(SUMREC) KEY(OKEY);                          AA 01670
     END;                                                                 AA 01680
     END;                                                                 AA 01690
      IF Q=5 THEN YUJISU(9,I,*)=DATA(*);
      IF Q=8 THEN DO;
        YUJISU(8,I,*)=DATA(*);
        IF SUBSTR(INDUST,1,1)='0' THEN DO;
      OKEY='2I' || YUCD(I);                                               AA 01490
      OKEY=TRANSLATE(OKEY,'0',' ');
      READ FILE(SUMM) INTO(SUMREC) KEY(OKEY);                             AA 01500
      IF WEIGHT=0 THEN DATA=0;
      ELSE DATA=CHSUM/WEIGHT;                                             AA 01510
    DO LEE = 1 TO 37;
      JISU=DATA(LEE)+0.05; JDATA(LEE)=JISU;                                  AA
    END;
      REWRITE FILE(SUMM) FROM(SUMREC) KEY(OKEY);                          AA 01540
        END;
      ELSE IF SUBSTR(INDUST,2,1)='0' THEN DO;
      OKEY='2I' || YUCD(I);                                               AA 01490
      OKEY=TRANSLATE(OKEY,'0',' ');
      READ FILE(SUMM) INTO(SUMREC) KEY(OKEY);                             AA 01500
      IF WEIGHT=0 THEN DATA=0;
      ELSE DATA=DASUM/WEIGHT;                                             AA 01510
    DO LEE = 1 TO 37;
      JISU=DATA(LEE)+0.05; JDATA(LEE)=JISU;                                  AA
    END;
      REWRITE FILE(SUMM) FROM(SUMREC) KEY(OKEY);                          AA 01540
      CHSUM=CHSUM+DATA*WEIGHT;  DASUM=0;
        END;
        ELSE DO;
          IF SUBSTR(INDUST,1,1)='2' | SUBSTR(INDUST,1,1)='4' |
             SUBSTR(INDUST,1,2)='39' THEN DO;
```

```
      OKEY='2I' || YUCD(I);                                          AA 01490
      OKEY=TRANSLATE(OKEY,'0',' ');
      READ FILE(SUMM) INTO(SUMREC) KEY(OKEY);                        AA 01500
     DO LEE = 1 TO 37;
      IF YUJISU(9,I,LEE)=0 THEN DATA(LEE)=0;
      ELSE DATA(LEE)=YUJISU(8,I,LEE)/YUJISU(9,I,LEE)*100;            AA 01510
     END;
    DO LEE = 1 TO 37;
      JISU=DATA(LEE)+0.05; JDATA(LEE)=JISU;                              AA
    END;
      REWRITE FILE(SUMM) FROM(SUMREC) KEY(OKEY);                     AA 01540
          DASUM=DASUM+DATA*WEIGHT;
          END;
          ELSE IF SUBSTR(INDUST,3,1)='0' THEN DO;
      OKEY='2I' || YUCD(I);                                          AA 01490
      OKEY=TRANSLATE(OKEY,'0',' ');
      READ FILE(SUMM) INTO(SUMREC) KEY(OKEY);                        AA 01500
      IF WEIGHT=0 THEN DATA=0;
      ELSE DATA(*)=JUSUM/WEIGHT;                                     AA 01510
    DO LEE = 1 TO 37;
      JISU=DATA(LEE)+0.05; JDATA(LEE)=JISU;                              AA
    END;
      REWRITE FILE(SUMM) FROM(SUMREC) KEY(OKEY);                     AA 01540
        DASUM=DASUM+DATA*WEIGHT;  JUSUM=0;
          END;
          ELSE DO;
      OKEY='2I' || YUCD(I);                                          AA 01490
      OKEY=TRANSLATE(OKEY,'0',' ');
      READ FILE(SUMM) INTO(SUMREC) KEY(OKEY);                        AA 01500
     DO LEE = 1 TO 37;
      IF YUJISU(9,I,LEE)=0 THEN DATA(LEE)=0;
      ELSE DATA(LEE)=YUJISU(8,I,LEE)/YUJISU(9,I,LEE)*100;            AA 01510
     END;
    DO LEE = 1 TO 37;
      JISU=DATA(LEE)+0.05; JDATA(LEE)=JISU;                              AA
    END;
      REWRITE FILE(SUMM) FROM(SUMREC) KEY(OKEY);                     AA 01540
       JUSUM=JUSUM+DATA*WEIGHT;
          END;
         END;
       END;
    END;                                                             AA 01700
  END ;    K=0;                                                      AA 01710
  SPCJISU(7,*,*)=SPCJISU(8,*,*)+SPCJISU(9,*,*);                      AA 01720
  SPCJISU(4,*,*)=SPCJISU(5,*,*)+SPCJISU(6,*,*);                      AA 01730
  SPCJISU(2,*,*)=SPCJISU(3,*,*)+SPCJISU(4,*,*);                      AA 01740
  SPCJISU(1,*,*)=SPCJISU(2,*,*)+SPCJISU(7,*,*);                      AA 01750
  DO Q=1 TO 3;                                                       AA 01760
  K=K+1;                                                             AA 01770
    DO I=1 TO 28 ;                                                   AA 01860
       OKEY='2' || GUB(K+12) || HRCD(I);                             AA 01870
       OKEY=TRANSLATE(OKEY,'0',' ');
       READ FILE(SUMM) INTO(SUMREC) KEY(OKEY);                       AA 01880
       IF WEIGHT=0 THEN DATA=0;
       ELSE DATA(*)=HRJISU(I,K,*)/WEIGHT;                            AA 01890
```

```
   DO LEE = 1 TO 37;
     JISU=DATA(LEE)+0.05; JDATA(LEE)=JISU;                                AA
   END;
      REWRITE FILE(SUMM) FROM(SUMREC) KEY(OKEY);                      AA 01920
      IF SUBSTR(HRCD(I),1,1)='1' THEN SOSUM(*)=SOSUM(*)+HRJISU(I,K,*); AA 01930
      ELSE JUSUM(*)=JUSUM(*)+HRJISU(I,K,*);                           AA 01940
   END;                                                               AA 01950
   OKEY='2' || GUB(K+12) || '1';                                      AA 01960
   OKEY=TRANSLATE(OKEY,'0',' ');
   READ FILE(SUMM) INTO(SUMREC) KEY(OKEY);                            AA 01970
   IF WEIGHT=0 THEN DATA=0;
   ELSE DATA=SOSUM/WEIGHT;                                            AA 01980
   DO LEE = 1 TO 37;
     JISU=DATA(LEE)+0.05; JDATA(LEE)=JISU;                                AA
   END;
   DASUM=DASUM+SOSUM;                                                 AA 01990
   REWRITE FILE(SUMM) FROM(SUMREC) KEY(OKEY);                         AA 02010
   OKEY='2' || GUB(K+12) || '2';                                      AA 02020
   OKEY=TRANSLATE(OKEY,'0',' ');
   READ FILE(SUMM) INTO(SUMREC) KEY(OKEY);                            AA 02030
   IF WEIGHT=0 THEN DATA=0;
   ELSE DATA=JUSUM/WEIGHT;                                            AA 02040
   DO LEE = 1 TO 37;
     JISU=DATA(LEE)+0.05; JDATA(LEE)=JISU;                                AA
   END;
   DASUM=DASUM+JUSUM;
   REWRITE FILE(SUMM) FROM(SUMREC) KEY(OKEY);                         AA 02070
   OKEY='2' || GUB(K+12) ;                                            AA 02080
   OKEY=TRANSLATE(OKEY,'0',' ');
   READ FILE(SUMM) INTO(SUMREC) KEY(OKEY);                            AA 02090
   IF WEIGHT=0 THEN DATA=0;
   ELSE DATA=DASUM/WEIGHT;                                            AA 02100
   DO LEE = 1 TO 37;
     JISU=DATA(LEE)+0.05; JDATA(LEE)=JISU;                                AA
   END;
   REWRITE FILE(SUMM) FROM(SUMREC) KEY(OKEY);                         AA 02130
   SOSUM=0;  JUSUM=0;  DASUM=0;
 END;                                                                 AA 02140
   /*****************************************************/
   /*************** SPECIAL JAKUP ************************/
   /*****************************************************/
   DO K = 1 TO 5 ;
      IF K < 4 THEN L = K + 9 ;
               ELSE L = K +15 ;
   DO I=1 TO 9;                                                       AA 01780
     OKEY='2' || GUB(L) || SPCD(I);                                   AA 01790
     OKEY=TRANSLATE(OKEY,'0',' ');
     READ FILE(SUMM) INTO(SUMREC) KEY(OKEY);                          AA 01800
     IF WEIGHT=0 THEN DATA=0;
     ELSE DATA(*)=SPCJISU(I,K,*)/WEIGHT;                              AA 01810
     DO LEE = 1 TO 37 ;
       JISU=DATA(LEE)+0.05; JDATA(LEE)=JISU;
     END ;
     REWRITE FILE(SUMM) FROM(SUMREC) KEY(OKEY);                       AA 01840
   END;                                                               AA 01850
```

```
    END;                                                                AA 01850
    /*************************************************/
    /******************* SPECIAL ************************/
    /*************************************************/
  CLOSE FILE(SUMM);                                                     AA 02150
 CALRTN: PROC(ASUM,BSUM);                                               AA 02170
  DCL ASUM(37)  FLOAT DEC(16);                                          AA 02180
  DCL BSUM(37)  FLOAT DEC(16);                                          AA 02190
  K=K+1;                                                                AA 02200
  IF GU='A' | GU='I' THEN PUT EDIT(YUCD(K),ASUM(2))(A(10),F(10,1));
  YUJISU(I,K,*)=ASUM(*);                                                AA 02210
  BSUM=BSUM+ASUM;                                                       AA 02220
  ASUM=0;                                                               AA 02230
 END CALRTN;                                                            AA 02240
 MUL10: PROC;
  DO M=1 TO 25;
    IF A10(M)=PUM THEN DATA=DATA*10;
  END;
 END MUL10;
 LASS:
 END JISURTN;                                                           AA 02250
/*
//LKED.SYSLMOD DD DSN=USER.LOADLIB(SMISP66),DISP=SHR
//
```

SMIC 3262

TABLE

동태 (월간) CATAL PROGRAM

```
//SMIC3265 JOB CLASS=A                                                  SMI00010
//  EXEC PLIFCL                                                         SMI00020
//PLI.SYSIN DD *                                                        SMI00030
* PROCESS   GS,NEST,A,OPT(2);                                           SMI00040
    TAB:   PROC   OPTIONS(MAIN);                                        SMI00050
         DCL    DIN   FILE RECORD KEYED SEQUENTIAL INPUT ENV(VSAM);     SMI00060
                  DCL    1      IN,                                     SMI00070
                              2  CD        CHAR(1),                     SMI00080
                              2  GU        CHAR(1),                     SMI00090
                              2  SAN       CHAR(3),                     SMI00100
                              2  PUM       CHAR(5),                     SMI00110
                              2  TK        CHAR(1),                     SMI00120
                              2  SI        PIC  '(4)9',                 SMI00130
                              2  GA        FLOAT DEC(6),                SMI00140
                              2  MUL       FIXED BIN(31),               SMI00150
                              2  DATA(37),                              SMI00160
                                 3 DMUL    FIXED BIN(31),               SMI00170
                                 3 DGI     FLOAT DEC(6),                SMI00180
                              2  BB        CHAR(1);                     SMI00190
   DCL AA(11) LABEL INIT(L3,L31,L4,L41,L5,L6,L7,L8,L9,L10,L101);        SMI00200
              DCL  SW    CHAR(1) INIT('*');                             SMI00210
              DCL  SRT(9)     FLOAT DEC(15);                            SMI00220
              DCL  SRTA       FLOAT DEC(15);                            SMI00230
              DCL  OGU        CHAR(1);                                  SMI00240
              DCL  SRTB       FLOAT DEC(15);                            SMI00250
              DCL  E          CHAR(1);                                  SMI00260
              DCL  PYO        PIC  '99' INIT(1);                        SMI00270
              DCL  A          PIC  '99' DEF STOR POS(1);                SMI00280
              DCL  B          PIC  '99' DEF STOR POS(3);                SMI00290
              DCL  C          PIC  '99';                                SMI00300
              DCL  D          PIC  '99';                                SMI00310
              DCL  L          PIC  '99';                                SMI00320
              DCL  STOR       CHAR(4);                                  SMI00330
              DCL  SRTC       FLOAT DEC(15);                            SMI00340
              DCL  SRTD       FLOAT DEC(15);                            SMI00350
              DCL  SRTE       FLOAT DEC(15);                            SMI00360
              DCL  WGTD       FLOAT DEC(15);                            SMI00370
              DCL  WGTG       FLOAT DEC(15);                            SMI00380
              DCL  1     SAVR,                                          SMI00390
                         2 SAR(37),                                     SMI00400
                           3 SMUL    FIXED BIN(31),                     SMI00410
                           3 SGI     FLOAT DEC(6);                      SMI00420
              DCL  1     JSAVR,                                         SMI00430
                         2 JSAR(37),                                    SMI00440
                           3 JMUL    FIXED BIN(31),                     SMI00450
                           3 JGI     FLOAT DEC(6);                      SMI00460
               DCL 1     JJAVR1,                                        SMI00470
                         2 JSSAR1(37),                                  SMI00480
                           3 JSMUL1   FIXED BIN(31),                    SMI00490
                           3 JSGI1    FLOAT DEC(6);                     SMI00500
               DCL 1     JJAVR2,                                        SMI00510
                         2 JSSAR2(37),                                  SMI00520
                           3 JSMUL2   FIXED BIN(31),                    SMI00530
                           3 JSGI2    FLOAT DEC(6);                     SMI00540
               DCL 1     JJAVR3,                                        SMI00550
```

```
                        2 JSSAR3(37),                                   SMI00560
                          3 JSMUL3   FIXED BIN(31),                     SMI00570
                          3 JSGI3    FLOAT DEC(6);                      SMI00580
              DCL BUM(3) CHAR(3) INIT('000','800','900');               SMI00590
              DCL GAM(3) FLOAT DEC(15);                                 SMI00600
    DISPLAY ('*******************************************');            SMI00610
    DISPLAY ('*** PLEASE  KEY IN NUMBER FOUR CHARACTERS ***');          SMI00620
    DISPLAY ('*******************************************')             SMI00630
            REPLY(STOR);                                                SMI00640
  OP:     OPEN FILE(DIN),FILE(SYSPRINT) LINESIZE(132);                  SMI00650
              ON ENDFILE(DIN) GOTO LAST;                                SMI00660
          SW='*';                                                       SMI00670
        IF PYO=3 THEN OGU='J';                                          SMI00680
           ELSE IF PYO=4 THEN OGU='M';                                  SMI00690
           ELSE OGU='A';                                                SMI00700
      IF A=37 THEN E='P';                                               SMI00710
         ELSE E='F';                                                    SMI00720
  RD: READ FILE(DIN) INTO(IN);                                          SMI00730
          IF SUBSTR(SAN,1,1)  ='0' THEN  GOTO RR;                       SMI00740
             GOTO AA(PYO);                                              SMI00750
  L3: IF GU='A' | GU>'D' &  GU<'J' THEN DO;                             SMI00760
      IF  PUM¬='00000' THEN GOTO RD;                                    SMI00770
      IF SW¬= '*' THEN GOTO R1;                                         SMI00780
          CALL PUT1;                                                    SMI00790
 PUT1:    PROC;                                                         SMI00800
          PUT PAGE;                                                     SMI00810
          PUT SKIP(3) EDIT('# 3.  TABLE  OF CHANGE  FROM SAME MONTH  OFSMI00820
 PREVIOUS  YEAR BY   INDUSTRIES')(X(23),A(76),X(33));                   SMI00830
          PUT SKIP    EDIT((76)'-')(X(23),A(76),X(33));                 SMI00840
          PUT SKIP(2) EDIT('MONTH=',B,E,GU)(X(5),A(6),F(2),A,A);        SMI00850
          PUT SKIP(3) EDIT((115)'-')(A(115));                           SMI00860
          PUT SKIP   EDIT('   INC  -   DEC   (%)        DAE - CHONG  GISMI00870
 - YU - DO')(X(61),A(54));                                              SMI00880
         PUT SKIP   EDIT((54)'-')(X(61),A(54));                         SMI00890
         PUT SKIP   EDIT('          CODE              PMY-2          PMY-1SMI00900
           PM           B/A           C/B             B/A            C/B')SMI00910
  (A(115));                                                             SMI00920
          PUT SKIP    EDIT((115)'-')(A(115));                           SMI00930
          SW='$';                                                       SMI00940
          END PUT1;                                                     SMI00950
  R1:     IF OGU¬=GU THEN DO;                                           SMI00960
            CALL PUT1; END;                                             SMI00970
          OGU=GU;                                                       SMI00980
          SRT=0; SRTA=0; SRTB=0; SRTC=0;                                SMI00990
          CALL INA;                                                     SMI01000
          SRTA=0; SRTB=0; SRTC=0;                                       SMI01010
          CALL GYA;                                                     SMI01020
      PUT SKIP(2) EDIT(SAN,DGI(A-24),DGI(A-12),DGI(A),SRT(1),SRT(2),    SMI01030
                SRT(3),SRT(4))(X(10),A(7),5 F(14,1),2 F(14,2));         SMI01040
      GOTO RD; END;                                                     SMI01050
      GOTO RD;                                                          SMI01060
 L31: IF   GU='P' | GU='Q' | GU='R'  THEN DO;                           SMI01070
      IF  PUM¬='00000' THEN GOTO RD;                                    SMI01080
      IF SW¬= '*' THEN GOTO R11;                                        SMI01090
          CALL PUT11;                                                   SMI01100
```

```
PUT11:    PROC;                                                          SMI01110
          PUT PAGE;                                                      SMI01120
          PUT SKIP(3) EDIT('#3-1. TABLE  OF CHANGE  FROM SAME MONTH  OFSMI01130
 PREVIOUS  YEAR BY   INDUSTRIES')(X(23),A(76),X(33));                    SMI01140
          PUT SKIP    EDIT((76)'-')(X(23),A(76),X(33));                  SMI01150
          PUT SKIP(2) EDIT('MONTH=',B,E,GU)(X(5),A(6),F(2),A,A);         SMI01160
          PUT SKIP(3) EDIT((115)'-')(A(115));                            SMI01170
          PUT SKIP   EDIT('   INC  -   DEC   (%)        DAE - CHONG  GISMI01180
 - YU - DO')(X(61),A(54));                                               SMI01190
         PUT SKIP   EDIT((54)'-')(X(61),A(54));                          SMI01200
         PUT SKIP   EDIT('         CODE              PMY-2          PMY-1SMI01210
           PM           B/A           C/B            B/A            C/B')SMI01220
   (A(115));                                                             SMI01230
          PUT SKIP    EDIT((115)'-')(A(115));                            SMI01240
          SW='$';                                                        SMI01250
          END PUT11;                                                     SMI01260
   R11:   IF OGU¬=GU THEN DO;                                            SMI01270
            CALL PUT11; END;                                             SMI01280
          OGU=GU;                                                        SMI01290
          SRT=0; SRTA=0; SRTB=0; SRTC=0;                                 SMI01300
          CALL INA;                                                      SMI01310
          SRTA=0; SRTB=0; SRTC=0;                                        SMI01320
          CALL GYA;                                                      SMI01330
       PUT SKIP(2) EDIT(SAN,DGI(A-24),DGI(A-12),DGI(A),SRT(1),SRT(2),    SMI01340
                   SRT(3),SRT(4))(X(10),A(7),5 F(14,1),2 F(14,2));       SMI01350
       GOTO RD; END;                                                     SMI01360
       GOTO RD;                                                          SMI01370
   L4: IF GU='A' | GU>'D' &  GU<'J' THEN DO;                             SMI01380
       IF PUM¬='00000' THEN GOTO RD;                                     SMI01390
       IF SW¬= '*' THEN GOTO R4;                                         SMI01400
          CALL PUT4;                                                     SMI01410
 PUT4:    PROC;                                                          SMI01420
          PUT PAGE;                                                      SMI01430
          PUT SKIP(3) EDIT('# 4.  TABLE  OF CHANGE  FROM SAME PERIOD OFSMI01440
PREVIOUS  YEAR BY   INDUSTRIES')(X(25),A(74),X(33));                     SMI01450
          PUT SKIP    EDIT((74)'-')(X(25),A(74),X(33));                  SMI01460
          PUT SKIP(2) EDIT('MONTH=',B,E,GU)(X(5),A(6),F(2),A,A);         SMI01470
          PUT SKIP(3) EDIT((115)'-')(A(115));                            SMI01480
          PUT SKIP   EDIT('   INC  -   DEC   (%)        DAE - CHONG  GISMI01490
 - YU - DO')(X(61),A(54));                                               SMI01500
         PUT SKIP   EDIT((54)'-')(X(61),A(54));                          SMI01510
         PUT SKIP   EDIT('         CODE              PY-2           PY-1SMI01520
           P            B/A           C/B            B/A            C/B')SMI01530
   (A(115));                                                             SMI01540
          PUT SKIP    EDIT((115)'-')(A(115));                            SMI01550
          SW='$';                                                        SMI01560
          END PUT4;                                                      SMI01570
    R4:   IF OGU¬=GU THEN DO;                                            SMI01580
            CALL PUT4; END;                                              SMI01590
          OGU=GU;                                                        SMI01600
          SRT=0;                                                         SMI01610
          CALL AVPA;                                                     SMI01620
          SRTA=0; SRTB=0; SRTC=0; SRTD=0;                                SMI01630
          CALL AVPB;                                                     SMI01640
       PUT SKIP(2) EDIT(SAN,(SRT(I) DO I=1 TO 3),SRTA,SRTB,SRTC,SRTD)    SMI01650
```

```
                  (X(10),A(7),3 F(14,1),2 F(14,1),2 F(14,2));          SMI01660
       GOTO RD; END;                                                   SMI01670
       GOTO RD;                                                        SMI01680
  L41: IF GU='P' | GU='Q' |  GU='R' THEN DO;                           SMI01690
       IF PUM¬='00000' THEN GOTO RD;                                   SMI01700
       IF SW¬= '*' THEN GOTO R41;                                      SMI01710
          CALL PUT41;                                                  SMI01720
 PUT41:    PROC;                                                       SMI01730
          PUT PAGE;                                                    SMI01740
          PUT SKIP(3) EDIT('#4-1. TABLE  OF CHANGE  FROM SAME PERIOD OFSMI01750
PREVIOUS  YEAR BY   INDUSTRIES')(X(25),A(74),X(33));                   SMI01760
          PUT SKIP    EDIT((74)'-')(X(25),A(74),X(33));                SMI01770
          PUT SKIP(2) EDIT('MONTH=',B,E,GU)(X(5),A(6),F(2),A,A);       SMI01780
          PUT SKIP(3) EDIT((115)'-')(A(115));                          SMI01790
          PUT SKIP   EDIT('   INC  -   DEC   (%)        DAE - CHONG  GISMI01800
 - YU - DO')(X(61),A(54));                                             SMI01810
         PUT SKIP   EDIT((54)'-')(X(61),A(54));                        SMI01820
         PUT SKIP   EDIT('          CODE              PY-2          PY-1SMI01830
           P           B/A           C/B            B/A           C/B')SMI01840
   (A(115));                                                           SMI01850
          PUT SKIP    EDIT((115)'-')(A(115));                          SMI01860
          SW='$';                                                      SMI01870
          END PUT41;                                                   SMI01880
   R41:    IF OGU¬=GU THEN DO;                                         SMI01890
            CALL PUT41; END;                                           SMI01900
          OGU=GU;                                                      SMI01910
          SRT=0;                                                       SMI01920
          CALL AVPA;                                                   SMI01930
          SRTA=0; SRTB=0; SRTC=0; SRTD=0;                              SMI01940
          CALL AVPB;                                                   SMI01950
       PUT SKIP(2) EDIT(SAN,(SRT(I) DO I=1 TO 3),SRTA,SRTB,SRTC,SRTD)  SMI01960
                  (X(10),A(7),3 F(14,1),2 F(14,1),2 F(14,2));          SMI01970
       GOTO RD; END;                                                   SMI01980
       GOTO RD;                                                        SMI01990
      L5:                                                              SMI02000
      IF GU = 'S' | GU = 'T' THEN GOTO TAB5;                           SMI02010
      IF GU < 'J' | GU > 'L' THEN GOTO RD;                             SMI02020
      IF GU = 'K' THEN DO;                                             SMI02030
                  IF SAN = '000' THEN DO;                              SMI02040
                                 JSSAR1=DATA;                          SMI02050
                                 GAM(1)=GA;                            SMI02060
                                 END;                                  SMI02070
                  IF SAN = '800' THEN DO;                              SMI02080
                                 JSSAR2=DATA;                          SMI02090
                                 GAM(2)=GA;                            SMI02100
                                 END;                                  SMI02110
                  IF SAN = '900' THEN DO;                              SMI02120
                                 JSSAR3=DATA;                          SMI02130
                                 GAM(3)=GA;                            SMI02140
                                 END;                                  SMI02150
         END;                                                          SMI02160
 TAB5:                                                                 SMI02170
      IF SW¬= '*' THEN GOTO R5;                                        SMI02180
      CALL PUT5;                                                       SMI02190
 PUT5: PROC;                                                           SMI02200
```

```
      PUT PAGE;                                                         SMI02210
      PUT SKIP(3) EDIT('# 5.  TABLE   OF  INDEX   BY  SPECIAL GROUPS(USSMI02220
 ER) BY  MONTHLY')(X(35),A(62),X(35));                                  SMI02230
      PUT SKIP EDIT((62)'-')(X(35),A(62),X(35));                        SMI02240
          PUT SKIP(2) EDIT('MONTH=',B,E,GU)(X(5),A(6),F(2),A,A);        SMI02250
          PUT SKIP(3) EDIT((130)'-')(A(130));                           SMI02260
          PUT SKIP    EDIT('        INC  -  DEC  (%)                 GISMI02270
 -YU-DO   DAE-DAE   GI-YU-DO ')(X(57),A);                               SMI02280
          PUT SKIP    EDIT((74)'-')(X(57),A);                           SMI02290
          PUT SKIP    EDIT('SCODE      AVPY-2     AVPY-1       PM-2     SMI02300
 PM-1       PM     VJW   VJNDW   VJNDG     VJW   VJNDW  VJNDG     VJW   SMI02310
 VJNDW   VJNDG')(A(130));                                               SMI02320
          PUT SKIP    EDIT((130)'-')(A(130));                           SMI02330
          SW='$';                                                       SMI02340
          END PUT5;                                                     SMI02350
  R5:     IF OGU¬=GU THEN DO;                                           SMI02360
          CALL PUT5; END;                                               SMI02370
          OGU=GU;                                                       SMI02380
          SRT=0; SRTA=0; SRTB=0; SRTC=0;                                SMI02390
          CALL  INC;                                                    SMI02400
          SRTA=0; SRTB=0; SRTC=0;                                       SMI02410
          IF GU = 'S' | GU = 'T' THEN DO;                               SMI02420
              IF SAN = '000' THEN DO;                                   SMI02430
                              SAR=JSSAR1;                               SMI02440
                              WGTD=GAM(1);                              SMI02450
                              WGTG=GA;                                  SMI02460
                              JSAR=DATA;                                SMI02470
                              END;                                      SMI02480
           GO TO CALL_GYU;                                              SMI02490
           END;                                                         SMI02500
          IF SAN = '800' | SAN = '900' THEN DO;                         SMI02510
                              JSAR=DATA;                                SMI02520
                              WGTG=GA;                                  SMI02530
                              END;                                      SMI02540
      CALL_GYU:                                                         SMI02550
          CALL  GYU;                                                    SMI02560
          SRTA=0; SRTB=0; SRTC=0;                                       SMI02570
          CALL  AVPY;                                                   SMI02580
      PUT SKIP(2) EDIT(GU,SAN,SRTA,SRTB,DGI(A-2),DGI(A-1),DGI(A),(SRT   SMI02590
              (I)  DO 1 = 1 TO 3),(SRT(J) DO J = 4 TO 9))               SMI02600
              (A(1),A(7),5 F(10,1), 3 F(8,1),6 F(8,2));                 SMI02610
          GOTO RD ;                                                     SMI02620
      L6:                                                               SMI02630
      IF GU < 'M' | GU > 'O' THEN GOTO RD;                              SMI02640
      IF SW¬= '*' THEN GOTO R6;                                         SMI02650
      CALL PUT6;                                                        SMI02660
 PUT6: PROC;                                                            SMI02670
      PUT PAGE;                                                         SMI02680
      PUT SKIP(3) EDIT('# 6.  TABLE  OF INDEX  BY SPECIAL GROUP(STRUCTUSMI02690
 RE) BY  MONTHLY')(X(35),A(62),X(35));                                  SMI02700
      PUT SKIP EDIT((62)'-')(X(35),A(62),X(35));                        SMI02710
      PUT SKIP(2) EDIT('MONTH',B,E,GU)(X(5),A(6),F(2),A,A);             SMI02720
          PUT SKIP(3) EDIT((117)'-')(A(117));                           SMI02730
          PUT SKIP    EDIT('        INC  -  DEC  (%)             DAE-CHSMI02740
 ONG   GI-YU-DO')(X(57),A(57));                                         SMI02750
```

```
          PUT SKIP    EDIT((60)'-')(X(57),A(60));                          SMI02760
          PUT SKIP    EDIT('SCODE       AVPY-2     AVPY-1       PM-2       SMI02770
 PM-1       PM        VJW      VJNDW      VJNDG        VJW      VJNDW    VJNSMI02780
DG')(A(117));                                                              SMI02790
          PUT SKIP    EDIT((117)'-')(A(117));                              SMI02800
          SW='$';                                                          SMI02810
          END PUT6;                                                        SMI02820
  R6:    IF OGU¬=GU THEN DO;                                               SMI02830
          CALL PUT6; END;                                                  SMI02840
          OGU=GU;                                                          SMI02850
          SRT=0; SRTA=0; SRTB=0; SRTC=0;                                   SMI02860
          CALL  INC;                                                       SMI02870
          SRTA=0; SRTB=0; SRTC=0;                                          SMI02880
          CALL  GYU;                                                       SMI02890
          SRTA=0; SRTB=0; SRTC=0;                                          SMI02900
          CALL  AVPY;                                                      SMI02910
   PUT SKIP(2) EDIT(GU,SAN,SUBSTR(PUM,1,1),SRTA,SRTB,DGI(A-2),DGI(A-1),SMI02920
  DGI(A),(SRT(I)  DO I = 1 TO 3),(SRT(J) DO J = 7 TO 9))                   SMI02930
  (A,A,A(6),5 F(10,1), 3 F(10,1),3 F(10,2));                               SMI02940
          GOTO RD ;                                                        SMI02950
   L7:                                                                     SMI02960
          IF GU='A' | GU>'B' & GU<'J' THEN DO;                             SMI02970
          IF SW¬='*' THEN GOTO R7;                                         SMI02980
          CALL PUT7;                                                       SMI02990
 PUT7:    PROC;                                                            SMI03000
          PUT PAGE;                                                        SMI03010
          PUT SKIP(3) EDIT('# 7.  TABLE  OF CHANGE  FROM PREVIOUS MONTHSMI03020
 OF INDEX & QUANITITY BY COMMODITIES')(X(30),A);                           SMI03030
          PUT SKIP    EDIT((76)'-')(X(30),A);                              SMI03040
          PUT SKIP(2) EDIT('MONTH=',B,E,GU)(X(5),A(6),F(2),A,A);           SMI03050
          PUT SKIP(3) EDIT((121)'-')(A(121));                              SMI03060
          PUT SKIP   EDIT('              QUANTITY                    INC SMI03070
 - DEC(%)            INDEX                 INC - DEC (%)        ')        SMI03080
            (X(8),A);                                                      SMI03090
        PUT SKIP   EDIT((113)'-')(X(8),A);                                 SMI03100
        PUT SKIP   EDIT('    CODE          PM-2       PM-1         PM      SMI03110
  B/A       C/B    WEIGHT       PM-2       PM-1        PM         B/A     SMI03120
 C/B')(A);                                                                 SMI03130
          PUT SKIP    EDIT((121)'-')(A(121));                              SMI03140
          SW='$';                                                          SMI03150
          END PUT7;                                                        SMI03160
  R7:     IF OGU¬=GU THEN DO;                                              SMI03170
            CALL PUT7; END;                                                SMI03180
          OGU=GU;                                                          SMI03190
          SRT=0; SRTA=0; SRTB=0; SRTC=0;                                   SMI03200
          CALL CCA;                                                        SMI03210
          CALL CGA;                                                        SMI03220
        SRT(5) = DMUL(A-2);                                                SMI03230
        SRT(6) = DMUL(A-1);                                                SMI03240
        SRTA = DMUL(A);                                                    SMI03250
      PUT SKIP(2) EDIT(SAN,PUM,SRT(5),SRT(6),SRTA,SRT(1),SRT(2),GA,        SMI03260
          DGI(A-2),DGI(A-1),DGI(A),SRT(3),SRT(4))(A(3),A(5),X(3),          SMI03270
           3 F(10), 8 F(10,1));                                            SMI03280
      GOTO RD; END;                                                        SMI03290
      GOTO RD;                                                             SMI03300
```

```
     CCA: PROC ;                                                        SMI03310
          SRTB = DMUL(A-1);   SRTC = DMUL(A-2);                         SMI03320
          IF SRTC = 0 THEN SRT(1) = 0;                                  SMI03330
           ELSE SRT(1) =(SRTB / SRTC) * 100 - 100;                      SMI03340
          SRTB = DMUL(A);   SRTC = DMUL(A-1);                           SMI03350
          IF SRTC = 0 THEN SRT(2) = 0;                                  SMI03360
           ELSE SRT(2) =(SRTB / SRTC) * 100 - 100;                      SMI03370
          END CCA ;                                                     SMI03380
     CGA:  PROC ;                                                       SMI03390
          IF  DGI(A-2) = 0 THEN SRT(3) = 0;                             SMI03400
          ELSE SRT(3) =  (DGI(A-1) / DGI(A-2)) * 100 - 100;             SMI03410
          IF  DGI(A-1) = 0 THEN SRT(4) = 0;                             SMI03420
          ELSE SRT(4) =  (DGI(A) / DGI(A-1)) * 100 - 100;               SMI03430
      END CGA;                                                          SMI03440
   L8:                                                                  SMI03450
          IF GU='A' | GU)'B' & GU('J' THEN DO;                          SMI03460
          IF SW¬='*' THEN GOTO R8;                                      SMI03470
          CALL PUT8;                                                    SMI03480
  PUT8:   PROC;                                                         SMI03490
          PUT PAGE;                                                     SMI03500
          PUT SKIP(3) EDIT('# 8.  TABLE  OF CHANGE  FROM SAME MONTH OF SMI03510
  PREVIOUS YEAR OF INDEX & QUANITITY BY COMMODITIES')(X(20),A);         SMI03520
          PUT SKIP    EDIT((93)'-')(X(20),A);                           SMI03530
          PUT SKIP(2) EDIT('MONTH=',B,E,GU)(X(5),A(6),F(2),A,A);        SMI03540
          PUT SKIP(3) EDIT((121)'-')(A(121));                           SMI03550
          PUT SKIP   EDIT('                QUANTITY                INC -SMI03560
  DEC(%)              INDEX              INC - DEC (%)     ')           SMI03570
             (X(21),A);                                                 SMI03580
         PUT SKIP   EDIT((100)'-')(X(21),A);                            SMI03590
         PUT SKIP   EDIT('     CODE        WEIGHT   PY-2PM    PY-1PM    SMI03600
   PYPM       B/A        C/B    PY-2PM    PY-1PM      PYPM       B/A    SMI03610
   C/B')(A);                                                            SMI03620
          PUT SKIP    EDIT((121)'-')(A(121));                           SMI03630
          SW='$';                                                       SMI03640
          END PUT8;                                                     SMI03650
   R8:    IF OGU¬=GU THEN DO;                                           SMI03660
            CALL PUT8; END;                                             SMI03670
          OGU=GU;                                                       SMI03680
          SRT=0; SRTA=0; SRTB=0; SRTC=0;                                SMI03690
          CALL CCB;                                                     SMI03700
          CALL KGB;                                                     SMI03710
        SRT(5) = DMUL(A-24);                                            SMI03720
        SRT(6) = DMUL(A-12);                                            SMI03730
        SRTA = DMUL(A);                                                 SMI03740
       PUT SKIP(2) EDIT(SAN,PUM,GA,SRT(5),SRT(6),SRTA,SRT(1),SRT(2),    SMI03750
          DGI(A-24),DGI(A-12),DGI(A),SRT(3),SRT(4))(A(3),A,X(3),        SMI03760
          F(10,1), 3 F(10), 7 F(10,1));                                 SMI03770
       GOTO RD; END;                                                    SMI03780
       GOTO RD;                                                         SMI03790
     CCB: PROC ;                                                        SMI03800
          SRTB = DMUL(A-12);   SRTC = DMUL(A-24);                       SMI03810
          IF SRTC = 0 THEN SRT(1) = 0;                                  SMI03820
           ELSE SRT(1) =(SRTB / SRTC) * 100 - 100;                      SMI03830
          SRTB = DMUL(A);   SRTC = DMUL(A-12);                          SMI03840
          IF SRTC = 0 THEN SRT(2) = 0;                                  SMI03850
```

```
          ELSE SRT(2) =(SRTB / SRTC) * 100 - 100;                    SMI03860
         END CCB ;                                                   SMI03870
     KGB:  PROC ;                                                    SMI03880
         IF DGI(A-24) = 0 THEN SRT(3) = 0;                           SMI03890
         ELSE  SRT(3) =  (DGI(A-12) / DGI(A-24)) * 100 - 100;        SMI03900
         IF DGI(A-12) = 0 THEN SRT(4) = 0;                           SMI03910
         ELSE  SRT(4) =  (DGI(A) / DGI(A-12)) * 100 - 100;           SMI03920
       END KGB;                                                      SMI03930
   L9:                                                               SMI03940
         IF GU='A' | GU)'B' & GU('J' THEN DO;                        SMI03950
         IF SW¬='*' THEN GOTO R9;                                    SMI03960
         CALL PUT9;                                                  SMI03970
 PUT9:   PROC;                                                       SMI03980
         PUT PAGE;                                                   SMI03990
        PUT SKIP(3) EDIT('# 9.  TABLE  OF CHANGE  FROM SAME PERIOD OF SMI04000
 PREVIOUS YEAR OF INDEX & QUANITITY BY COMMODITIES')(X(20),A);       SMI04010
         PUT SKIP    EDIT((90)'-')(X(20),A);                         SMI04020
         PUT SKIP(2) EDIT('MONTH=',B,E,GU)(X(5),A(6),F(2),A,A);      SMI04030
         PUT SKIP(3) EDIT((121)'-')(A(121));                         SMI04040
         PUT SKIP   EDIT('                 QUANTITY                INC -SMI04050
 DEC(%)              INDEX                INC - DEC (%)     ')       SMI04060
            (X(21),A);                                               SMI04070
        PUT SKIP   EDIT((100)'-')(X(21),A);                          SMI04080
        PUT SKIP   EDIT('     CODE        WEIGHT   PY-2PM    PY-1PM  SMI04090
  PYPM       B/A        C/B    PY-2PM    PY-1PM      PYPM       B/A  SMI04100
  C/B')(A);                                                          SMI04110
         PUT SKIP    EDIT((121)'-')(A(121));                         SMI04120
         SW='$';                                                     SMI04130
         END PUT9;                                                   SMI04140
   R9:   IF OGU¬=GU THEN DO;                                         SMI04150
           CALL PUT9; END;                                           SMI04160
         OGU=GU;                                                     SMI04170
         SRT=0; SRTA=0; SRTB=0; SRTC=0;SRTD = 0;                     SMI04180
         CALL CCC;                                                   SMI04190
         SRTA=0; SRTB=0; SRTC=0;SRTD = 0;                            SMI04200
         CALL CGB;                                                   SMI04210
         SRTD = 0; SRTE=0;                                           SMI04220
       DO I = (A-B+1) TO A;                                          SMI04230
       SRTD = DMUL(I-24); SRT(5) = SRT(5) + SRTD;                    SMI04240
       SRTD = DMUL(I-12); SRT(6) = SRT(6) + SRTD;                    SMI04250
       SRTD = DMUL(I); SRTE = SRTE + SRTD;       END;                SMI04260
       SRTA = SRTA / B;  SRTB = SRTB / B;  SRTC = SRTC / B;          SMI04270
      PUT SKIP(2) EDIT(SAN,PUM,GA,SRT(5),SRT(6),SRTE,SRT(1),SRT(2),  SMI04280
         SRTA,SRTB,SRTC,SRT(3),SRT(4))(A,A,X(3),                     SMI04290
         F(10,1), 3 F(10), 7 F(10,1));                               SMI04300
      GOTO RD; END;                                                  SMI04310
      GOTO RD;                                                       SMI04320
    CCC: PROC ;                                                      SMI04330
         DO I = (A-B+1) TO A;                                        SMI04340
       SRTD = DMUL(I-24); SRTC = SRTC + SRTD;                        SMI04350
       SRTD = DMUL(I-12); SRTB = SRTB + SRTD;                        SMI04360
       SRTD = DMUL(I); SRTA = SRTA + SRTD;       END;                SMI04370
         IF SRTC = 0 THEN SRT(1) = 0;                                SMI04380
          ELSE SRT(1) =(SRTB / SRTC) * 100 - 100;                    SMI04390
         IF SRTB = 0 THEN SRT(2) = 0;                                SMI04400
```

```
          ELSE SRT(2) =(SRTA / SRTB) * 100 - 100;                    SMI04410
         END CCC ;                                                   SMI04420
     CGB:  PROC ;                                                    SMI04430
         SRTA=0;                                                     SMI04440
         DO I = (A-B+1) TO A;                                        SMI04450
         SRTA = DGI(I-24) + SRTA;                                    SMI04460
         SRTB = DGI(I-12) + SRTB;                                    SMI04470
         SRTC = DGI(I) + SRTC;END;                                   SMI04480
         IF SRTA = 0 THEN SRT(3) = 0;                                SMI04490
         ELSE  SRT(3) = (SRTB / SRTA) * 100 - 100;                   SMI04500
         IF SRTB = 0 THEN SRT(4) = 0;                                SMI04510
         ELSE  SRT(4) = (SRTC / SRTB) * 100 - 100;                   SMI04520
       END CGB;                                                      SMI04530
       L10:                                                          SMI04540
         IF GU ='A' | GU )'D' & GU('J' THEN DO;                      SMI04550
      IF SW¬= '*' THEN GOTO R10;                                     SMI04560
      CALL PUT10;                                                    SMI04570
 PUT10: PROC;                                                        SMI04580
      PUT PAGE;                                                      SMI04590
         PUT SKIP(3) EDIT('# 10. TABLE   OF   INDEX  BY  INDUSTRIES  SMI04600
   BY    MONTHLY ')(X(30),A(60));                                    SMI04610
         PUT SKIP    EDIT((60)'-')(X(30),A(60));                     SMI04620
         PUT SKIP(2) EDIT('MONTH=',B,E,GU)(X(5),A(6),F(2),A,A);      SMI04630
         PUT SKIP(3) EDIT((117)'-')(A(117));                         SMI04640
         PUT SKIP    EDIT('        INC  -  DEC  (%)            DAE-CHSMI04650
ONG   GI-YU-DO')(X(57),A(57));                                       SMI04660
         PUT SKIP    EDIT((60)'-')(X(57),A(60));                     SMI04670
         PUT SKIP    EDIT('CODE       AVPY-2    AVPY-1      PM-2     SMI04680
 PM-1       PM       VJW     VJNDW     VJNDG       VJW     VJNDW     VJSMI04690
DG')(A(117));                                                        SMI04700
         PUT SKIP    EDIT((117)'-')(A(117));                         SMI04710
         SW='$' ;                                                    SMI04720
         END PUT10;                                                  SMI04730
  R10:   IF OGU¬=GU THEN DO;                                         SMI04740
         CALL PUT10; END;                                            SMI04750
         OGU=GU;                                                     SMI04760
         IF PUM ¬='00000' THEN GOTO RD;                              SMI04770
         SRT=0; SRTA=0; SRTB=0; SRTC=0;                              SMI04780
         CALL INC;                                                   SMI04790
         SRTA=0; SRTB=0; SRTC=0;                                     SMI04800
         CALL GYU10;                                                 SMI04810
         SRTA=0; SRTB=0; SRTC=0;                                     SMI04820
         CALL AVPY;                                                  SMI04830
         PUT SKIP(2) EDIT(SAN,SRTA,SRTB,DGI(A-2),DGI(A-1),DGI(A),(SRT SMI04840
             (I)  DO I = 1 TO 3),(SRT(J) DO J = 4 TO 6))             SMI04850
             (A(7),5 F(10,1),3 F(10,1),3 F(10,2));                   SMI04860
         GOTO RD ; END;                                              SMI04870
         GOTO RD;                                                    SMI04880
      L101:                                                          SMI04890
         IF GU ='P' | GU ='Q' & GU='R' THEN DO;                      SMI04900
      IF SW¬= '*' THEN GOTO R101;                                    SMI04910
      CALL PUT101;                                                   SMI04920
 PUT101: PROC;                                                       SMI04930
      PUT PAGE;                                                      SMI04940
         PUT SKIP(3) EDIT('#10-1. TABLE   OF   INDEX  BY  INDUSTRIES SMI04950
```

```
   BY    MONTHLY ')(X(30),A(60));                                          SMI04960
         PUT SKIP    EDIT((60)'-')(X(30),A(60));                           SMI04970
         PUT SKIP(2) EDIT('MONTH=',B,E,GU)(X(5),A(6),F(2),A,A);            SMI04980
         PUT SKIP(3) EDIT((117)'-')(A(117));                               SMI04990
         PUT SKIP    EDIT('        INC  -  DEC  (%)              DAE-CHSMI05000
ONG   GI-YU-DO')(X(57),A(57));                                             SMI05010
         PUT SKIP    EDIT((60)'-')(X(57),A(60));                           SMI05020
         PUT SKIP    EDIT('CODE        AVPY-2    AVPY-1       PM-2        SMI05030
 PM-1       PM        VJW     VJNDW     VJNDG       VJW      VJNDW     VJSMI05040
DG')(A(117));                                                              SMI05050
         PUT SKIP    EDIT((117)'-')(A(117));                               SMI05060
         SW='$' ;                                                          SMI05070
         END PUT101;                                                       SMI05080
  R101:   IF OGU¬=GU THEN DO;                                              SMI05090
         CALL PUT101; END;                                                 SMI05100
         OGU=GU;                                                           SMI05110
         IF PUM ¬='00000' THEN GOTO RD;                                    SMI05120
         SRT=0; SRTA=0; SRTB=0; SRTC=0;                                    SMI05130
         CALL INC;                                                         SMI05140
         SRTA=0; SRTB=0; SRTC=0;                                           SMI05150
         CALL GYU10;                                                       SMI05160
         SRTA=0; SRTB=0; SRTC=0;                                           SMI05170
         CALL AVPY;                                                        SMI05180
         PUT SKIP(2) EDIT(SAN,SRTA,SRTB,DGI(A-2),DGI(A-1),DGI(A),(SRT SMI05190
             (I)  DO I = 1 TO 3),(SRT(J) DO J = 4 TO 6))                   SMI05200
             (A(7),5 F(10,1),3 F(10,1),3 F(10,2));                         SMI05210
         GOTO RD ; END;                                                    SMI05220
         GOTO RD;                                                          SMI05230
     INC: PROC ;                                                           SMI05240
         IF DGI(A-1)=0 THEN SRT(1)=0;                                      SMI05250
         ELSE SRT(1) = (DGI(A) / DGI(A-1)) * 100 - 100;                    SMI05260
         IF DGI(A-12)=0 THEN SRT(2)=0;                                     SMI05270
         ELSE SRT(2) = (DGI(A) / DGI(A-12)) * 100 - 100 ;                  SMI05280
         C = (A-B) + 1;                                                    SMI05290
         DO  J = C TO A;                                                   SMI05300
             SRTA = SRTA + DGI(J) ;                                        SMI05310
         END ;                                                             SMI05320
         SRTA = SRTA / B ;                                                 SMI05330
         C = (A-B) - 11;                                                   SMI05340
         D = (C+B) - 1;                                                    SMI05350
         DO  J = C TO D ;                                                  SMI05360
             SRTB = SRTB + DGI(J) ;                                        SMI05370
         END;                                                              SMI05380
         SRTB = SRTB / B ;                                                 SMI05390
         IF SRTB=0 THEN SRT(3)=0;                                          SMI05400
         ELSE  SRT(3) = (SRTA / SRTB) * 100 - 100 ;                        SMI05410
         END INC ;                                                         SMI05420
    GYU10:  PROC ;                                                         SMI05430
         IF SGI(A-1)=0 THEN SRT(4)=0;                                      SMI05440
  ELSE SRT(4) = ((DGI(A) - DGI(A-1)) * GA) / (SGI(A-1) * 100);             SMI05450
         IF SGI(A-12)=0 THEN SRT(5)=0;                                     SMI05460
  ELSE SRT(5) = ((DGI(A) - DGI(A-12)) * GA) / (SGI(A-12) * 100);           SMI05470
         C = (A-B) +1;                                                     SMI05480
         DO  J = C TO A ;                                                  SMI05490
             SRTA = SRTA + DGI(J);                                         SMI05500
```

```
          END;                                                          SMI05510
          SRTA = SRTA / B;                                              SMI05520
          C=(A-B) - 11;                                                 SMI05530
          D=(C+B) - 1;                                                  SMI05540
          DO  J = C TO D ;                                              SMI05550
              SRTB = SRTB + DGI(J);                                     SMI05560
              SRTC = SRTC + SGI(J) ;                                    SMI05570
          END ;                                                         SMI05580
          SRTB = SRTB / B ;  SRTC = SRTC / B ;                          SMI05590
          IF SRTC=0 THEN SRT(6)=0;                                      SMI05600
  ELSE  SRT(6) = ((SRTA - SRTB)* GA) /  (SRTC * 100);                   SMI05610
          END  GYU10 ;                                                  SMI05620
     GYU:  PROC ;                                                       SMI05630
          IF GU = 'S' | GU = 'T' THEN GO TO GYUST;                      SMI05640
          IF SAN = '000' THEN GOTO DAEDAEG;                             SMI05650
 GYUST:                                                                 SMI05660
          IF JGI(A-1)=0 THEN SRT(4)=0;                                  SMI05670
 ELSE SRT(4) = (((DGI(A) - DGI(A-1)) * GA) / (JGI(A-1) * WGTG)) *100;   SMI05680
          IF JGI(A-12)=0 THEN SRT(5)=0;                                 SMI05690
 ELSE SRT(5) = (((DGI(A) - DGI(A-12)) * GA) / (JGI(A-12) * WGTG)) *100;SMI05700
          C = (A-B) +1;                                                 SMI05710
          DO  J = C TO A ;                                              SMI05720
              SRTA = SRTA + DGI(J);                                     SMI05730
          END;                                                          SMI05740
          SRTA = SRTA / B;                                              SMI05750
          C=(A-B) - 11;                                                 SMI05760
          D=(C+B) - 1;                                                  SMI05770
          DO  J = C TO D ;                                              SMI05780
              SRTB = SRTB + DGI(J);                                     SMI05790
              SRTC = SRTC + JGI(J) ;                                    SMI05800
          END ;                                                         SMI05810
          SRTB = SRTB / B ;  SRTC = SRTC / B ;                          SMI05820
          IF SRTC=0 THEN SRT(6)=0;                                      SMI05830
 ELSE  SRT(6) = (((SRTA - SRTB)* GA) /  (SRTC * WGTG)) *100;            SMI05840
     DAEDAEG:                                                           SMI05850
          SRTA=0; SRTB=0; SRTC=0;                                       SMI05860
          IF SGI(A-1)=0 THEN SRT(7)=0;                                  SMI05870
 ELSE SRT(7) = (((DGI(A) - DGI(A-1)) * GA) / (SGI(A-1) * WGTD))*100;    SMI05880
          IF SGI(A-12)=0 THEN SRT(8)=0;                                 SMI05890
 ELSE SRT(8) = (((DGI(A) - DGI(A-12)) * GA) / (SGI(A-12) * WGTD))*100;  SMI05900
          C = (A-B) +1;                                                 SMI05910
          DO  J = C TO A ;                                              SMI05920
              SRTA = SRTA + DGI(J);                                     SMI05930
          END;                                                          SMI05940
          SRTA = SRTA / B;                                              SMI05950
          C=(A-B) - 11;                                                 SMI05960
          D=(C+B) - 1;                                                  SMI05970
          DO  J = C TO D ;                                              SMI05980
              SRTB = SRTB + DGI(J);                                     SMI05990
              SRTC = SRTC + SGI(J) ;                                    SMI06000
          END ;                                                         SMI06010
          SRTB = SRTB / B ;  SRTC = SRTC / B ;                          SMI06020
          IF SRTC=0 THEN SRT(9)=0;                                      SMI06030
  ELSE  SRT(9) = (((SRTA - SRTB)* GA) /  (SRTC * WGTD))*100;            SMI06040
          END  GYU ;                                                    SMI06050
```

```
     INA: PROC ;                                                        SMI06060
          IF DGI(A-24)=0 THEN SRT(1)=0;                                 SMI06070
          ELSE SRT(1) = (DGI(A-12) / DGI(A-24)) * 100 - 100;            SMI06080
          IF DGI(A-12)=0 THEN SRT(2)=0;                                 SMI06090
          ELSE SRT(2) = (DGI(A) / DGI(A-12)) * 100 - 100 ;              SMI06100
          END INA ;                                                     SMI06110
     GYA:  PROC ;                                                       SMI06120
          IF SGI(A-24)=0 THEN SRT(3)=0;                                 SMI06130
      ELSE SRT(3) = ((DGI(A-12) - DGI(A-24)) * GA) / (SGI(A-24) * 100);SMI06140
          IF SGI(A-12)=0 THEN SRT(4)=0;                                 SMI06150
         ELSE SRT(4) = ((DGI(A) - DGI(A-12)) * GA) / (SGI(A-12) * 100);SMI06160
          END  GYA ;                                                    SMI06170
      AVPY:   PROC ;                                                    SMI06180
              L=(A-(B+23));                                             SMI06190
          DO  J = 1 TO 12 ;                                             SMI06200
              SRTA = SRTA + DGI(L);                                     SMI06210
              L=L+1;                                                    SMI06220
          END ;                                                         SMI06230
          SRTA = SRTA / 12;                                             SMI06240
              L=(A-(B+11));                                             SMI06250
          DO  J = 1 TO 12 ;                                             SMI06260
                SRTB = SRTB + DGI(L);                                   SMI06270
                L=L+1;                                                  SMI06280
          END ;                                                         SMI06290
          SRTB = SRTB / 12;                                             SMI06300
          END AVPY;                                                     SMI06310
      AVPA:   PROC ;                                                    SMI06320
              L=(A-(B+23));                                             SMI06330
          DO  J = 1 TO B ;                                              SMI06340
              SRT(1) = SRT(1) + DGI(L);                                 SMI06350
              SRT(4) = SRT(4) + SGI(L);                                 SMI06360
              L=L+1;                                                    SMI06370
          END ;                                                         SMI06380
          SRT(1) = SRT(1) / B;                                          SMI06390
          SRT(4) = SRT(4) / B;                                          SMI06400
              L=(A-(B+11));                                             SMI06410
          DO  J = 1 TO B ;                                              SMI06420
                SRT(2) = SRT(2) + DGI(L);                               SMI06430
                SRT(5) = SRT(5) + SGI(L);                               SMI06440
                L=L+1;                                                  SMI06450
          END ;                                                         SMI06460
          SRT(2) = SRT(2) / B;                                          SMI06470
          SRT(5) = SRT(5) / B;                                          SMI06480
          L=(A-(B-1));                                                  SMI06490
          DO  J = 1 TO B ;                                              SMI06500
                SRT(3) = SRT(3) + DGI(L);                               SMI06510
                L=L+1;                                                  SMI06520
          END ;                                                         SMI06530
          SRT(3) = SRT(3) / B;                                          SMI06540
          END AVPA;                                                     SMI06550
    AVPB: PROC;                                                         SMI06560
          IF SRT(1)=0 THEN SRTA=0;                                      SMI06570
          ELSE SRTA=(SRT(2)/SRT(1)) * 100 - 100;                        SMI06580
          IF SRT(2)=0 THEN SRTB=0;                                      SMI06590
          ELSE SRTB=(SRT(3)/SRT(2)) * 100 - 100;                        SMI06600
```

```
          IF SRT(4)=0 THEN SRTC=0;                                    SMI06610
          ELSE SRTC=((SRT(2)-SRT(1)) * GA) / (SRT(4)*100);            SMI06620
          IF SRT(5)=0 THEN SRTD=0;                                    SMI06630
          ELSE SRTD=((SRT(3)-SRT(2)) * GA) / (SRT(5)*100);            SMI06640
          END AVPB;                                                   SMI06650
       RR:                                                            SMI06660
         SAR=DATA; WGTD=GA; GOTO AA(PYO);                             SMI06670
       LAST:  CLOSE FILE(DIN);                                        SMI06680
              PYO=PYO+1;                                              SMI06690
              IF PYO < 12 THEN GOTO OP;                               SMI06700
       END  TAB ;                                                     SMI06710
/*                                                                    SMI06720
//LKED.SYSLMOD DD DSN=USER.LOADLIB(SMI1232),DISP=SHR                  SMI06730
//                                                                    SMI06740
```

SMICSA35

고용 TABLE

(CATAL PROGRAM)

```
//SMICSA35 JOB CLASS=A,TYPRUN=HOLD                                      SMI00010
//STEP2 EXEC PLIFCL                                                     SMI00430
//PLI.SYSIN DD *                                                        SMI00440
* PROCESS GS,NEST,XREF,A,OPT(2);                                        SMI00450
 SEL2 : PROC OPTIONS(MAIN);                                             SMI00460
  DCL DIN    FILE RECORD SEQUENTIAL INPUT ENV(VSAM) ;                   SMI00470
  DCL DOUT   FILE RECORD OUTPUT;                                        SMI00480
  DCL 1 IN,                                                             SMI00490
      2 DDD        CHAR(11),                                            SMI00500
      2 YYMM       CHAR(4),                                             SMI00510
      2 WEIGHT     FLOAT DEC(6),                                        SMI00520
      2 POINT      FIXED BIN(31),                                       SMI00530
      2 DATA(37),                                                       SMI00540
         3 MUL     FIXED BIN(31),                                       SMI00550
         3 GISU    FLOAT DEC(6),                                        SMI00560
      2 F3         CHAR(1) ;                                            SMI00570
  DCL 1 DD DEF DDD,                                                     SMI00580
      2 CARD       CHAR(1),                                             SMI00590
      2 GUBUN      CHAR(1),                                             SMI00600
      2 SANUP      CHAR(3),                                             SMI00610
      2 PUM        CHAR(5),                                             SMI00620
      2 FIL        CHAR(1);                                             SMI00620
  DCL WCN     PIC'99999' INIT(0) ;
  OPEN FILE(DIN) ;                                                      SMI00650
  ON ENDFILE(DIN) GOTO LAST ;                                           SMI00660
  R1 :     READ FILE(DIN) INTO(IN) ;                                    SMI00670
           IF CARD ¬= '4' THEN GOTO R1 ;                                SMI00680
        WRITE FILE(DOUT) FROM (IN);                                     SMI00700
        PUT DATA(DDD) ;                                                 SMI00710
        WCN = WCN + 1 ;                                                 SMI00720
        GOTO R1 ;                                                       SMI00730
  LAST : PUT SKIP EDIT( 'WRITE COUNT ',WCN ) (A(20)) ;                  SMI00740
         CLOSE FILE(DOUT), FILE(DIN) ;                                  SMI00750
         END SEL2 ;                                                     SMI00760
/*                                                                      SMI00770
//LKED.SYSLMOD DD DSN=USER.LOADLIB(SMISA35),DISP=SHR                    SMI00780
//STEP3 EXEC PLIFCL                                                     SMI00430
//PLI.SYSIN DD *                                                        SMI00440
* PROCESS GS,NEST,A,XREF,OPT(2);                                        SMI00450
 SEL2 : PROC OPTIONS(MAIN);                                             SMI00460
  DCL DIN    FILE RECORD SEQUENTIAL INPUT ENV(VSAM) ;                   SMI00470
  DCL DOUT   FILE RECORD OUTPUT;                                        SMI00480
  DCL 1 IN,                                                             SMI00490
      2 DDD        CHAR(11),                                            SMI00500
      2 YYMM       CHAR(4),                                             SMI00510
      2 WEIGHT     FLOAT DEC(6),                                        SMI00520
      2 POINT      FIXED BIN(31),                                       SMI00530
      2 DATA(37),                                                       SMI00540
         3 MUL     FIXED BIN(31),                                       SMI00550
         3 GISU    FLOAT DEC(6),                                        SMI00560
      2 F3         CHAR(1) ;                                            SMI00570
  DCL 1 DD DEF DDD,                                                     SMI00580
      2 CARD       CHAR(1),                                             SMI00590
      2 GUBUN      CHAR(1),                                             SMI00600
      2 SANUP      CHAR(3),                                             SMI00610
```

```
      2 PUM          CHAR(5),                                          SMI00620
      2 FIL          CHAR(1);                                          SMI00620
  DCL WCN     PIC'99999' INIT(0) ;
  OPEN FILE(DIN) ;                                                     SMI00650
  ON ENDFILE(DIN) GOTO LAST ;                                          SMI00660
  R1 :     READ FILE(DIN) INTO(IN) ;                                   SMI00670
           IF CARD ¬= '2' THEN GOTO R1 ;                               SMI00680
           IF GUBUN ¬= 'A' THEN GOTO R1 ;                              SMI00690
           IF PUM ¬= '00000' THEN GOTO  R1 ;
        WRITE FILE(DOUT) FROM (IN);                                    SMI00700
        PUT DATA(DDD) ;                                                SMI00710
        WCN = WCN + 1 ;                                                SMI00720
        GOTO R1 ;                                                      SMI00730
  LAST : PUT SKIP EDIT( 'WRITE COUNT ',WCN ) (A(20)) ;                 SMI00740
         CLOSE FILE(DOUT), FILE(DIN) ;                                 SMI00750
         END SEL2 ;                                                    SMI00760
/*                                                                     SMI00770
//LKED.SYSLMOD DD DSN=USER.LOADLIB(SMISB35),DISP=SHR                   SMI00780
//STEP6 EXEC PLIFCL
//PLI.SYSIN DD *
* PROCESS GS,NEST,A,XREF,OPT(2);                                       23500010
 DDD1 : PROC OPTIONS(MAIN);                                            23500020
  DCL DIN FILE RECORD INPUT ;                                          23500030
  DCL DOUT FILE RECORD OUTPUT;                                         23500040
  DCL 1 IN,                                                            23500050
      2 CARD        CHAR(1),                                           23500060
      2 F1          PIC'9',                                            23500070
      2 SANUP       CHAR(3),                                           23500080
      2 GUBUN       PIC'9',                                            23500090
      2 BUN         PIC'9',                                            23500100
      2 F2          CHAR(4),                                           23500110
      2 YYMM        CHAR(4),                                           23500120
      2 WHEIT       FLOAT DEC(6),                                      23500130
      2 POINT       FIXED BIN(31),                                     23500140
      2 DATA(37),                                                      23500150
         3 MUL      FIXED BIN(31),                                     23500160
         3 GISU     FLOAT DEC(6),                                      23500170
      2 F3          CHAR(1) ;                                          23500180
  DCL 1 A1(20),                                                        23500190
         3 MUL(2)      FIXED BIN(31) INIT(0),                          23500200
         3 GISU(2)     FLOAT DEC(6) INIT(0) ;                          23500210
  DCL 1 A2(20),                                                        23500220
         3 MUL(2)      FIXED BIN(31) INIT(0),                          23500230
         3 GISU(2)     FLOAT DEC(6) INIT(0) ;                          23500240
  DCL 1 A3(20),                                                        23500250
         3 MUL(2)      FIXED BIN(31) INIT(0),                          23500260
         3 GISU(2)     FLOAT DEC(6) INIT(0) ;                          23500270
  DCL (WCN,ICN)    PIC'(5)9' INIT(0);                                  23500280
  DCL (N,I,K,L,M)    PIC'99' INIT(0);                                  23500290
  DCL SSANUP      CHAR(5) INIT((5)' ') ;                               23500300
  OPEN FILE(DIN), FILE(DOUT) ;                                         23500310
  ON ENDFILE(DIN) GOTO LAST ;                                          23500320
 RR : READ FILE(DIN) INTO(IN);                                         23500330
      ICN = ICN + 1 ;                                                  23500340
       IF SSANUP ¬= SANUP THEN DO ;                                    23500350
```

```
          I = 0 ;                                                          23500360
          SSANUP = SANUP ;     END ;                                       23500370
         IF  SANUP = '000'  THEN DO ;                                      23500380
            N = N + 1;                                                     23500390
            DO K = 1 TO 2 ;                                                23500400
            DATA.MUL(35+K) = A3.MUL(N,K) ;                                 23500410
            DATA.GISU(35+K) = A3.GISU(N,K) ;                               23500420
            END ;                                                          23500430
            WRITE FILE(DOUT) FROM(IN) ;                                    23500440
            WCN = WCN + 1 ;                                                23500450
            GOTO RR ;              END ;                                   23500460
       I = I + 1 ;                                                         23500470
       IF SUBSTR(SANUP,3,1) = '0' THEN DO ;                                23500480
          IF SANUP = '390' | SANUP = '210' | SANUP = '230' |               23500490
                SANUP = '290' THEN DO ;                                    23500500
            DO K = 1 TO 2 ;                                                23500510
            A2.MUL(I,K) = A2.MUL(I,K) + DATA.MUL(K+35) ;                   23500520
            A2.GISU(I,K) = A2.GISU(I,K) + DATA.GISU(K+35) ;                23500530
            END ;                                                          23500540
            GOTO BIE ;                 END ;                               23500550
          IF SUBSTR(SANUP,2,2) = '00' THEN DO ;                            23500560
            IF SANUP = '400' THEN DO ;                                     23500570
            DO K = 1 TO 2 ;                                                23500580
                A3.MUL(I,K) = A3.MUL(I,K) + DATA.MUL(K+35) ;               23500590
                A3.GISU(I,K) = A3.GISU(I,K) + DATA.GISU(K+35) ;            23500600
            END ;                                                          23500610
                GOTO BIE ;        END ;                                    23500620
            DO K = 1 TO 2 ;                                                23500630
                A3.MUL(I,K) = A3.MUL(I,K) + A2.MUL(I,K) ;                  23500640
                A3.GISU(I,K) = A3.GISU(I,K) + A2.GISU(I,K) ;               23500650
              DATA.MUL(K+35) = A2.MUL(I,K) ;                               23500660
              DATA.GISU(K+35) = A2.GISU(I,K) ;                             23500670
              A2.MUL(I,K) = 0 ;  A2.GISU(I,K) = 0 ;                        23500680
            END ;                                                          23500690
              GOTO BIE ;                   END ;                           23500700
            DO K = 1 TO 2 ;                                                23500710
          A2.MUL(I,K) = A2.MUL(I,K) + A1.MUL(I,K) ;                        23500720
          A2.GISU(I,K) = A2.GISU(I,K) + A1.GISU(I,K) ;                     23500730
          DATA.MUL(K+35) = A1.MUL(I,K) ;                                   23500740
          DATA.GISU(K+35) = A1.GISU(I,K) ;                                 23500750
          A1.MUL(I,K) = 0 ;  A1.GISU(I,K) = 0 ;                            23500760
            END ;                                                          23500770
          GOTO BIE ;                   END ;                               23500780
            DO K = 1 TO 2 ;                                                23500790
      A1.MUL(I,K) = A1.MUL(I,K) + DATA.MUL(K+35) ;                         23500800
      A1.GISU(I,K) = A1.GISU(I,K) + DATA.GISU(K+35) ;                      23500810
            END ;                                                          23500820
  BIE : WRITE FILE(DOUT) FROM(IN) ;                                        23500830
        WCN  = WCN + 1;                                                    23500840
      GOTO RR ;                                                            23500850
  LAST : PUT SKIP(2) EDIT('IN_COUNT',ICN,'WRITE_COUNT',WCN)                23500860
                    ((2) (A,X(2),F(5),X(5)));                              23500870
          CLOSE   FILE(DIN),FILE(DOUT);                                    23500880
          END DDD1 ;                                                       23500890
/*                                                                         SMI00990
```

```
//LKED.SYSLMOD DD DSN=USER.LOADLIB(SMISC35),DISP=SHR                    SMI01000
//                                                                      SMI01090
```

SMIWCV34

생산능력 TABLE (월간)

(CATAL PROGRAM)

```
//SMIWCV34 JOB CLASS=A,TYPRUN=HOLD                                      SMI00010
//STEP4 EXEC PLIFCL                                                     SMI00570
//PLI.SYSIN DD *                                                        SMI00580
* PROCESS GS,NEST ;                                                     SMI00590
      POCON:   PROC OPTIONS(MAIN);                                      SMI00600
           DCL DIN FILE RECORD INPUT;                                   SMI00610
           DCL DOG FILE RECORD KEYED DIRECT UPDATE ENV(VSAM);           SMI00620
           DCL 1 IN,                                                    SMI00630
                3 CNO CHAR(1),                                          SMI00640
                3 SEO CHAR(1),                                          SMI00650
                3 SAN CHAR(3),                                          SMI00660
                3 PUM CHAR(5),                                          SMI00670
                3 FL1 CHAR(1),                                          SMI00680
                3 SIL PIC '9999',                                       SMI00690
                3 CHI FLOAT,                                            SMI00700
                3 YANG FIXED BIN(31),                                   SMI00710
                3 DATA(37),                                             SMI00720
                  4 MUL FIXED BIN(31),                                  SMI00730
                  4 JI  FLOAT,                                          SMI00740
                3 FL2 CHAR(1);                                          SMI00750
             DCL AST(37) FLOAT DEC(16) INIT(0);                         SMI00760
             DCL BST(37) FLOAT DEC(16) INIT(0);                         SMI00770
             DCL CST(37) FLOAT DEC(16) INIT(0);                         SMI00780
             DCL SS(37) FLOAT DEC(16) INIT(0);                          SMI00790
             DCL AMT(37) FLOAT DEC(16) INIT(0);                         SMI00800
             DCL BMT(37) FLOAT DEC(16) INIT(0);                         SMI00810
             DCL CMT(37) FLOAT DEC(16) INIT(0);                         SMI00820
             DCL MS(37) FLOAT DEC(16) INIT(0);                          SMI00830
             DCL AGT(37)  FLOAT DEC(16) INIT(0);                        SMI00840
             DCL BGT(37)  FLOAT DEC(16) INIT(0);                        SMI00850
             DCL CGT(37)  FLOAT DEC(16) INIT(0);                        SMI00860
             DCL GS(37)  FLOAT DEC(16) INIT(0);                         SMI00870
           DCL COMP FLOAT DEC(16);                                      SMI00880
           DCL CM(37) FLOAT ;                                           SMI00890
           DCL ACO FLOAT ;                                              SMI00900
           DCL BCO FLOAT ;                                              SMI00900
           DCL DM(37) FLOAT ;                                           SMI00910
           DCL COM1 FLOAT DEC(16);                                      SMI00930
           DCL COM2 FLOAT DEC(16);                                      SMI00940
           DCL CHO FLOAT DEC(16);                                       SMI00940
           DCL BAN  FLOAT DEC(16);                                      SMI00950
           DCL P PIC '99';                                              SMI00960
           DCL AKEY CHAR (10);                                          SMI00970
           ON ENDFILE(DIN) GOTO LAS;                                    SMI00980
      OPEN FILE(DIN),FILE(DOG),FILE(SYSPRINT) LINESIZE(132);            SMI00990
      RD:                                                               SMI01000
           READ FILE(DIN) INTO (IN);                                    SMI01010
           IF PUM ¬> '00000' THEN GO TO SMA;                            SMI01030
           IF SEO = '3' THEN GO TO GDL;                                 SMI01040
           DO P = 36 TO 37;
           ACO = MUL(P);    BCO = YANG;
           JI(P) = ACO / BCO * 100;                                     SMI01050
           END;                                                         SMI01060
           DO P = 36 TO 37;
           COM1 = JI(P) * CHI;                                          SMI01070
```

```
        IF SEO = '1' THEN DO; AST(P) = AST(P) + COM1;                    SMI01090
            CM(P)=MUL(P);
                END;                                                     SMI01130
                    ELSE DO; BST(P) = BST(P) + COM1;                     SMI01140
           DM(P)=MUL(P);                                                 SMI01160
                END;          END;
           GO TO PT;                                                     SMI01190
GDL:    DO P = 36 TO 37;                                                 SMI01200
      IF CM(P)=0 THEN COMP = 0; ELSE COMP = DM(P)/CM(P)*100;
              CHO = COMP;
        MUL(P) = COMP * 10 + 0.5;  BAN = CHO * CHI;                      SMI01210
                        SS(P) = SS(P) + BAN ;  END;                      SMI01220
      DO P = 36 TO 37;
        ACO = MUL(P);   BCO = YANG;
    JI(P) = ACO / BCO * 100;                                             SMI01250
        COM1 = JI(P) * CHI;                                              SMI01260
        CST(P) = CST(P) + COM1;    END;                                  SMI01280
        GO TO PT ;                                                       SMI01310
SMA:                                                                     SMI01320
        IF SUBSTR(SAN,2,2) ¬> '00' THEN GO TO GSM;                       SMI01330
        IF SUBSTR(SAN,3,1) ¬> '0' THEN GO TO MSM;                        SMI01340
        IF SEO = '1' THEN GO TO GA;                                      SMI01350
        IF SEO = '2' THEN GO TO NA;                                      SMI01360
        DO P = 36 TO 37;
        COMP = SS(P) / CHI ;                                             SMI01370
            MUL(P) = COMP *10 + 0.5;                                     SMI01380
           MS(P) = MS(P) + SS(P);     END;     SS = 0;                   SMI01410
        DO P = 36 TO 37;
        COMP = CST(P) / CHI;                                             SMI01420
        JI(P) = COMP;    CMT(P) = CMT(P) + CST(P);                       SMI01430
          END;      CST = 0;                                             SMI01450
        GO TO PT;                                                        SMI01460
GA:                                                                      SMI01470
      DO P = 36 TO 37;
        COMP = AST(P) / CHI;                                             SMI01480
          JI(P) = COMP;      AMT(P) = AMT(P) + AST(P);                   SMI01490
              END;        AST = 0;                                       SMI01510
        GO TO PT;                                                        SMI01520
NA:                                                                      SMI01530
        DO P = 36 TO 37;
        COMP = BST(P) / CHI;                                             SMI01540
        JI(P) = COMP;       BMT(P) = BMT(P) + BST(P);                    SMI01550
        END;       BST = 0;                                              SMI01570
        GO TO PT;                                                        SMI01580
MSM:                                                                     SMI01590
        IF SEO = '1' THEN GO TO GB;                                      SMI01600
        IF SEO = '2' THEN GO TO NB;                                      SMI01610
        DO P = 36 TO 37;                                                 SMI01620
           COMP = MS(P) / CHI ;                                          SMI01620
           MUL(P) = COMP * 10 + 0.5;                                     SMI01630
           GS(P) = GS(P) + MS(P);     END;MS = 0;                        SMI01660
         DO P = 36 TO 37;
        COMP = CMT(P) / CHI;                                             SMI01670
        JI(P) = COMP;   CGT(P) = CGT(P) + CMT(P);                        SMI01680
        END;      CMT = 0;                                               SMI01690
```

```
          GO TO PT;                                                    SMI01710
     GB:  DO P = 36 TO 37;                                             SMI01720
          COMP = AMT(P) / CHI ;                                        SMI01730
          JI(P) = COMP;            AGT(P)=AGT(P)+AMT(P);               SMI01740
          END;     AMT = 0;
          GO TO PT;                                                    SMI01770
     NB:                                                               SMI01780
          DO P = 36 TO 37;
          COMP = BMT(P) / CHI;                                         SMI01790
          JI(P) = COMP;        BGT(P)=BGT(P)+BMT(P);                   SMI01800
          END;        BMT = 0;                                         SMI01820
          GO TO PT;                                                    SMI01830
     GSM:                                                              SMI01840
          IF SEO = '1' THEN GO TO GC;                                  SMI01850
          IF SEO = '2' THEN GO TO NC;                                  SMI01860
          DO P = 36 TO 37;                                             SMI01870
             COMP = GS(P) / CHI ;                                      SMI01870
             MUL(P) = COMP * 10 + 0.5;   END;                          SMI01880
           DO P = 36 TO 37;
          COMP = CGT(P) / CHI;                                         SMI01910
          JI(P) = COMP;                                                SMI01920
           END;                                                        SMI01940
          GO TO PT;                                                    SMI01950
     GC:                                                               SMI01960
          DO P = 36 TO 37;
          COMP = AGT(P) / CHI;                                         SMI01970
            JI(P) = COMP;                                              SMI01980
            END;                                                       SMI02000
          GO TO PT;                                                    SMI02010
     NC:                                                               SMI02020
          DO P = 36 TO 37;
          COMP = BGT(P) / CHI;                                         SMI02030
          JI(P) = COMP;                                                SMI02040
          END;                                                         SMI02060
          GO TO PT;                                                    SMI02070
     PT:                                                               SMI02080
         AKEY = CNO||SEO||SAN||PUM;                                    SMI02090
        REWRITE FILE(DOG) FROM(IN) KEY(AKEY);                          SMI02100
          GO TO RD;                                                    SMI02110
         LAS:                                                          SMI02120
              CLOSE FILE(DIN),FILE(DOG);                               SMI02130
         END POCON;                                                    SMI02140
/*                                                                     SMI02150
//LKED.SYSLMOD DD DSN=USER.LOADLIB(SMICB34),DISP=SHR                   SMI02210
//                                                                     SMI02220
```

SMICTB65

생 산 능 력 (월 간)

(CATAL PROGRAM(2))

```
//SMICTB65 JOB CLASS=V                                                  SMI00010
//STEP3 EXEC PLIFCL                                                     SMI00020
//PLI.SYSIN DD *                                                        SMI00030
* PROCESS GS,NEST;                                                      SMI00040
          POTAB:   PROC OPTIONS(MAIN);                                  SMI00050
          DCL  DIN FILE RECORD INPUT;                                   SMI00060
          DCL  1  IND,                                                  SMI00070
                 2 CNO CHAR(1),                                         SMI00080
                 2 SEE CHAR(1),                                         SMI00090
                 2 SAN CHAR(3),                                         SMI00100
                 2 PUM CHAR(5),                                         SMI00110
                 2 FL  CHAR(1),                                         SMI00120
                 2 FIL PIC '9999',                                      SMI00130
                 2 CHI FLOAT,                                           SMI00140
                 2 YANG FIXED BIN(31),                                  SMI00150
                 2 DATA(37),                                            SMI00160
                   3 MUL FIXED BIN(31),                                 SMI00170
                   3 JISU FLOAT,                                        SMI00180
                 2 FL2  CHAR(1) ;                                       SMI00190
           DCL   1 SDATA(37),                                           SMI00200
                   2 SMU FIXED BIN(31),                                 SMI00210
                   2 SJI FLOAT;                                         SMI00220
          DCL  KI FIXED DEC (2) INIT(36);                               SMI00230
          DCL  K FIXED DEC(2) INIT(0);                                  SMI00240
          DCL  P FIXED DEC(2) INIT(0);                                  SMI00250
          DCL  L FIXED DEC(2) INIT(0);                                  SMI00260
          DCL  SNO CHAR(1) INIT('9');                                   SMI00270
          DCL  SWX CHAR(1) INIT('*');                                   SMI00280
          DCL  KYE (12) FLOAT;                                          SMI00290
          DCL  B1 FLOAT;                                                SMI00300
          DCL  B2 FLOAT;                                                SMI00310
          DCL  B3 FLOAT;                                                SMI00320
          DCL  B4 FLOAT;                                                SMI00330
          DCL  B5 FLOAT;                                                SMI00340
          DCL  B6 FLOAT;                                                SMI00350
          DCL  B7 FLOAT;                                                SMI00360
          DCL  B8 FLOAT;                                                SMI00370
          DCL COM1 FLOAT;                                               SMI00380
          DCL COM2 FLOAT;                                               SMI00390
          DCL CP1 FLOAT;                                                SMI00400
          DCL CP2 FLOAT;                                                SMI00410
          DCL CP3 FLOAT;                                                SMI00420
          DCL CP4 FLOAT;                                                SMI00430
          DCL ACCEPT CHAR(4);                                           SMI00440
          DCL CSC PIC '(4)9';                                           SMI00450
          DCL SIM PIC '(4)9';                                           SMI00460
       DISPLAY('WHAT MONTH OF BASE?')                                   SMI00470
       REPLY(ACCEPT);                                                   SMI00480
       SIM = ACCEPT;                                                    SMI00490
       DISPLAY('WHAT SHALL I DO?')                                      SMI00500
       REPLY(ACCEPT);                                                   SMI00510
       CSC = ACCEPT;                                                    SMI00520
        B7 = SUBSTR(SIM,1,2) * 12 + SUBSTR(SIM,3,2);                    SMI00530
        B8 = SUBSTR(CSC,1,2) * 12 + SUBSTR(CSC,3,2);                    SMI00540
        B6 = 37 - (B7 - B8);                                            SMI00550
```

```
          KI = B6;                                                    SMI00560
       K = 14 - SUBSTR(CSC,3,2);                                      SMI00570
          OPEN FILE(SYSPRINT) LINESIZE(132);                          SMI00580
          CALL TAB2;                                                  SMI00590
          CALL TAB3;                                                  SMI00600
          CALL TAB32;                                                 SMI00610
          CALL TAB4;                                                  SMI00620
          CALL TAB5;                                                  SMI00630
          CALL TAB64;                                                 SMI00640
          CALL TAB65;                                                 SMI00650
          CALL TAB66;                                                 SMI00660
          CALL TAB67;                                                 SMI00670
          CALL TABA;                                                  SMI00680
          SWX = 'A';                                                  SMI00690
          CALL TAB64;                                                 SMI00700
          CALL TAB65;                                                 SMI00710
          CALL TAB66;                                                 SMI00720
          CALL TAB67;                                                 SMI00730
          CALL TABA;                                                  SMI00740
          CALL TAB7;                                                  SMI00750
          CALL TAB8;                                                  SMI00760
          CALL TAB9;                                                  SMI00770
          CALL TAB10;                                                 SMI00780
          GO TO EVE;                                                  SMI00790
  TAB2: PROC;                                                         SMI00800
          ON ENDFILE(DIN) GO TO LA1;                                  SMI00810
          OPEN FILE(DIN),FILE(SYSPRINT) LINESIZE(132);                SMI00820
    R1:                                                               SMI00830
          READ FILE(DIN) INTO(IND);                                   SMI00840
          IF SEE = SNO THEN GO TO MP1;                                SMI00850
          PUT PAGE;      SNO = SEE;                                   SMI00860
          PUT SKIP(3)                                                 SMI00870
           EDIT(CSC,'#2 TABLE OF INDEX BY COMMODITIES BY MONTHLY')    SMI00880
               (X(10),A,X(31),A);                                     SMI00890
    PUT SKIP(2) EDIT('CODE  DIVISION WEIGHT      P M - 1        P   M  SMI00900
  INC-JW    INC-JNDW    I-VJNDG    KI-VJW     VJNDW     VJNDG') (X(2),A);  SMI00910
        IF SAN ¬) '300' THEN SDATA = DATA;                            SMI00920
  MP1:                                                                SMI00930
       IF JISU(KI-1) = 0 THEN  B1 = 0;                                SMI00940
       ELSE                                                           SMI00950
       B1 = JISU(KI) / JISU(KI-1) * 100 - 100;                        SMI00960
       IF SJI(KI-1) = 0 THEN  B5 = 0;                                 SMI00970
       ELSE  B5 = (JISU(KI)-JISU(KI-1)) * CHI /(SJI(KI-1) * 100);     SMI00980
    IF JISU(KI-12) = 0 THEN  B2=0;                                    SMI00990
     ELSE  B2 = JISU(KI) / JISU(KI-12) * 100 - 100;                   SMI01000
    IF SJI(KI-12) = 0 THEN  B6 = 0;                                   SMI01010
  ELSE                                                                SMI01020
       B6 = (JISU(KI)-JISU(KI-12)) * CHI /(SJI(KI-12) * 100);         SMI01030
       K = 14 - SUBSTR(SIM,3,2);                                      SMI01040
           COM1 = 0;CP3 = 0;                                          SMI01050
           COM2 = 0;CP4 = 0;                                          SMI01060
       DO I = K TO 13;                                                SMI01070
           COM1 = COM1 + JISU(I+24);                                  SMI01080
           COM2 = COM2 + JISU(I+12);                                  SMI01090
           CP4  = CP4  + SJI(I+12);                                   SMI01100
```

```
        END;                                                              SMI01110
      IF COM2 = 0 THEN  B3 = 0 ;                                          SMI01120
        ELSE  B3 = COM1 / COM2 * 100 - 100;                               SMI01130
      IF CP4 = 0 THEN  B7 = 0;      ELSE                                SMI01140
        B7 = (COM1-COM2) * CHI / (CP4 * 100);                             SMI01150
        IF PUM ¬= '      ' THEN SAN = '    ';                             SMI01160
   PUT SKIP(2) EDIT(SAN,PUM,SEE,CHI,JISU(KI-1),JISU(KI),B1,               SMI01170
         B2,B3,B5,B6,B7) (A(3),A(5),X(3),A,X(4),F(7,1),F(12,1),F(12,1),   SMI01180
                F(12,1),F(11,1),F(10,1),F(9,2),F(9,2),F(9,2));            SMI01190
      GO TO R1;                                                           SMI01200
   LA1:                                                                   SMI01210
      CLOSE FILE(DIN);                                                    SMI01220
      END TAB2;                                                           SMI01230
   TAB3: PROC;                                                            SMI01240
      ON ENDFILE(DIN) GO TO LA2;                                          SMI01250
      OPEN FILE(DIN);                                                     SMI01260
   R2:                                                                    SMI01270
      READ FILE(DIN) INTO(IND);                                           SMI01280
      IF PUM ¬= '      ' THEN GO TO R2;                                   SMI01290
      IF SEE = SNO THEN GO TO MP2;                                        SMI01300
      PUT PAGE;                                                           SMI01310
      SNO = SEE;                                                          SMI01320
    PUT SKIP(3) EDIT(CSC,'#3 TABLE OF INDEX BY INDUSTRIES BY MONTHLY')    SMI01330
                  (X(10),A,X(31),A);                                      SMI01340
      PUT SKIP(2) EDIT('CODE DIV     WEIGHT   P M - 2    PM - 1     P    SMI01350
 M    IC-VJW     VJNDW    VJJNDW     VJNDG     KI-VJW KI-VJNDW    VJJNDSMI01360
W    VJNDG') (X(1),A);                                                    SMI01370
         IF SAN ¬) '300' THEN SDATA = DATA;                               SMI01380
   MP2:                                                                   SMI01390
      IF JISU(KI-1) = 0 THEN B1 = 0;                                      SMI01400
    ELSE   B1 = JISU(KI) / JISU(KI-1) * 100 - 100;                        SMI01410
    IF SJI(KI-1) = 0 THEN  B5 = 0;                                        SMI01420
      ELSE B5 = (JISU(KI) - JISU(KI-1)) * CHI / (SJI(KI-1) * 100);        SMI01430
    IF JISU(KI-12) = 0 THEN  B2 = 0;                                      SMI01440
      ELSE   B2 = JISU(KI) / JISU(KI-12) * 100 - 100;                     SMI01450
    IF SJI(KI-12) = 0 THEN  B6 = 0;                                       SMI01460
       ELSE B6 = (JISU(KI) - JISU(KI-12)) * CH1 / (SJI(K1-12) * 100);     SMI01470
    IF JISU(KI-24)= 0 THEN  B3 = 0;                                       SMI01480
      ELSE  B3 = JISU(KI-12) / JISU(KI-24) * 100 - 100;                   SMI01490
    IF SJI(KI-24)= 0 THEN  B7 = 0;                                        SMI01500
      B7 = (JISU(KI-12) - JISU(KI-24)) * CHI /(SJ1(KI-24) * 100);         SMI01510
       COM1 = 0;   COM2 = 0;    CP4 = 0;                                   SMI01520
      DO I = K TO 13;                                                     SMI01530
        COM1 = COM1 + JISU(I+24);                                         SMI01540
        COM2 = COM2 + JISU(I+12);                                         SMI01550
        CP4  = CP4  + SJI(I+12);                                          SMI01560
      END ;                                                               SMI01570
    IF COM2 = 0 THEN  B4 = 0;    ELSE                                     SMI01580
      B4 = COM1 / COM2 * 100 - 100;                                       SMI01590
    IF CP4 = 0 THEN     B8 = 0;    ELSE                                  SMI01600
      B8 = (COM1 - COM2) * CHI /(CP4 * 100);                              SMI01610
      PUT SKIP(2) EDIT(SAN,SEE,CHI,JISU(KI-2),JISU(KI-1),JISU(KI),        SMI01620
             B1,B2,B3,B4,B5,B6,B7,B8)       (A(8),A(1),F(11,1),           SMI01630
             F(10,1),F(10,1),F(10,1),F(10,1),F(10,1),F(10,1),F(10,1),     SMI01640
             3 F(10,2),F(9,2));                                           SMI01650
```

```
            GO TO R2;                                                   SMI01660
      LA2:                                                              SMI01670
         CLOSE FILE(DIN);                                               SMI01680
      END TAB3;                                                         SMI01690
    TAB32: PROC ;                                                       SMI01700
        ON ENDFILE(DIN) GO TO LA3;                                      SMI01710
        OPEN FILE(DIN);                                                 SMI01720
          PUT PAGE;                                                     SMI01730
  PUT SKIP(3) EDIT(CSC,'#3 TABLE OF INDEX BY INDUSTRIES BY MONTHLY')    SMI01740
                  (X(10),A,X(31),A);                                    SMI01750
     PUT SKIP(2) EDIT('CODE DIV     WEIGHT   P M - 2    PM - 1     P   SMI01760
 M    IC-VJW     VJNDW    VJJNDW     VJNDG     KI-VJW KI-VJNDW    VJJNDSMI01770
W    VJNDG') (X(1),A);                                                  SMI01780
     IF SAN ¬) '300' THEN SDATA = DATA;                                 SMI01790
    R3:     READ FILE(DIN) INTO(IND);                                   SMI01800
        IF PUM ¬= '      ' THEN GO TO R3;                               SMI01810
        IF SEE ¬= '3' THEN GO TO R3;                                    SMI01820
           CP1 = MUL(KI);   CP2 = MUL(KI-1);                            SMI01830
           CP3 = SMU(KI);   CP4 = SMU(KI-1);                            SMI01840
        IF CP2 = 0 THEN  B1 = 0;                                        SMI01850
          ELSE  B1 = CP1 / CP2 * 100 - 100;                             SMI01860
        IF CP4 = 0 THEN  B5 = 0;                                        SMI01870
          ELSE  B5 = (CP1 - CP2) * CHI / (CP4 * 100);                   SMI01880
           CP2 = MUL(KI-12);     CP4 = SMU(KI-12);                      SMI01890
        IF CP2 = 0 THEN  B2 = 0;                                        SMI01900
          ELSE  B2 = CP1 / CP2 * 100 - 100;                             SMI01910
        IF CP4 = 0 THEN   B6 = 0;                                       SMI01920
          ELSE B6 = (CP1 - CP2) * CHI /(CP4 * 100);                     SMI01930
           CP2 = MUL(KI-24);                                            SMI01940
        IF CP2 = 0 THEN  B3 = 0;                                        SMI01950
           ELSE  B3 = CP1 / CP2 * 100 - 100;                            SMI01960
            CP4 = SMU(KI-24);                                           SMI01970
        IF CP4 = 0 THEN  B7 = 0;                                        SMI01980
           ELSE  B7 = (CP1 - CP2) * CHI / (CP4 * 100);                  SMI01990
            COM1 = 0;  COM2 = 0;     CP4 = 0;                           SMI02000
           DO I = K TO 13;                                              SMI02010
              COM1 = COM1 + MUL(I+24);                                  SMI02020
              COM2 = COM2 + MUL(I+12);                                  SMI02030
              CP4 = CP4 + SMU(I+12);   END;                             SMI02040
         IF COM2 = 0 THEN  B4=0;   ELSE                                 SMI02050
            B4 = COM1 / COM2 * 100 - 100;                               SMI02060
         IF CP4 = 0 THEN  B8 = 0;                                       SMI02070
           ELSE B8 = (COM1 - COM2) * CHI /(CP4 * 100);                  SMI02080
         PUT SKIP(2) EDIT(SAN,'4',CHI,MUL(KI-2),MUL(KI-1),MUL(KI),      SMI02090
                    B1,B2,B3,B4,B5,B6,B7,B8) (A(8),A(1),F(11,1),        SMI02100
                    7 F(10,1), 3 F(10,2),F(9,2));                       SMI02110
           GO TO R3;                                                    SMI02120
      LA3:                                                              SMI02130
           CLOSE FILE(DIN);                                             SMI02140
           END TAB32;                                                   SMI02150
      TAB4:  PROC;                                                      SMI02160
           ON ENDFILE(DIN) GO TO LA4 ;                                  SMI02170
           OPEN FILE(DIN);                                              SMI02180
      R4:                                                               SMI02190
           READ FILE(DIN) INTO(IND);                                    SMI02200
```

```
          IF SNO = SEE THEN GO TO MP4;                                    SMI02210
           PUT PAGE ;                                                     SMI02220
           SNO = SEE;                                                     SMI02230
           PUT SKIP(3) EDIT(CSC,'#4 TABLE OF INDEX CHANGE FROM SAMEMONTHSMI02240
  OF PREVIOUSYEAR BY INDUSTRIES') (X(10),A,X(20),A);                      SMI02250
    PUT SKIP(2) EDIT('CODE   DIVISION      WEIGHT     *PMY-2    *PMY-1   SMI02260
    *P  M        IC-VJJNDW      IC-VJNDW        VJJNDW          VJNDW')  SMI02270
        (X(2),A);                                                         SMI02280
        IF SAN ¬> '300' THEN SDATA = DATA;                                SMI02290
      MP4:                                                                SMI02300
        IF PUM ¬= '      ' THEN GO TO R4 ;                                SMI02310
        IF JISU(KI-24) = 0 THEN  B1 = 0;                                  SMI02320
          ELSE  B1 = JISU(KI-12) / JISU(KI-24)  * 100 - 100;              SMI02330
        IF SJI(KI-24) = 0 THEN  B5 = 0;                                   SMI02340
  ELSE B5 = (JISU(KI-12) - JISU(KI-24)) * CHI /(SJI(KI-24) * 100);        SMI02350
        IF JISU(KI-12) = 0 THEN  B2 = 0;                                  SMI02360
          ELSE  B2 = JISU(KI) / JISU(KI-12) * 100 - 100;                  SMI02370
        IF SJI(KI-12) = 0 THEN  B6 = 0;                                   SMI02380
          ELSE B6 = (JISU(KI) - JISU(KI-12)) * CHI /(SJI(KI-12) * 100);SMI02390
         PUT SKIP(2)                                                      SMI02400
             EDIT(SAN,SEE,CHI,JISU(KI-24),JISU(KI-12),JISU(KI),           SMI02410
              B1,B2,B5,B6)   (X(2),A,X(8),A,X(5),F(10,1),F(11,1),         SMI02420
              F(10,1),F(10,1),F(15,1),F(15,1),F(15,2),F(15,2));           SMI02430
             GO TO R4 ;                                                   SMI02440
       LA4:                                                               SMI02450
          CLOSE FILE(DIN);                                                SMI02460
       END TAB4 ;                                                         SMI02470
       TAB5: PROC;                                                        SMI02480
          ON ENDFILE(DIN) GO TO LA5;                                      SMI02490
          OPEN FILE(DIN);                                                 SMI02500
       R5:                                                                SMI02510
          READ FILE(DIN) INTO(IND);                                       SMI02520
          IF PUM ¬= '      ' THEN GO TO R5;                               SMI02530
          IF SEE = SNO THEN GOTO MP5;                                     SMI02540
            SNO = SEE;                                                    SMI02550
            PUT PAGE;                                                     SMI02560
          PUT SKIP(3) EDIT(CSC,'#5 TABLE OF CHANGE FROM SAME PERIOD OF SMI02570
 PREVIOUS YEAR') (X(10),A,X(31),A);                                       SMI02580
    PUT SKIP(2) EDIT('CODE   DIVISION      WEIGHT     JJNDG     *JNDG    SMI02590
 IND-DG        IC-VJJNDG       IC-VJNDG           GI-VJJNDG          GISMI02600
 -VJNDG')(X(2),A);                                                        SMI02610
        IF SAN ¬> '300' THEN SDATA = DATA;                                SMI02620
      MP5:  L = SUBSTR(SIM,3,2);                                          SMI02630
            COM1 = 0;  COM2 = 0;  B3 = 0 ;                                SMI02640
            CP3  = 0;  CP4  = 0;                                          SMI02650
          DO I=K TO 13;                                                   SMI02660
            COM1 = COM1 + JISU(I);                                        SMI02670
            CP3 = CP3 + SJI(I);                                           SMI02680
            COM2 = COM2 + JISU(I+12);                                     SMI02690
            CP4 = CP4 + SJI(I+12);                                        SMI02700
            B3 = B3 + JISU(I+24);                                         SMI02710
          END;                                                            SMI02720
          IF COM1 = 0 THEN DO; B1 = 0;   GO TO AAA; END;                  SMI02730
          IF CP3 = 0 THEN DO;  B5=0;  GO TO AAA; END;                     SMI02740
          IF COM2 = 0 THEN DO; B2 = 0;  GO TO AAA; END;                   SMI02750
```

```
        IF CP4 = 0 THEN DO;  B6=0;  GO TO AAA; END;                    SMI02760
         B1 = COM2 / COM1 * 100 - 100;                                 SMI02770
         B2 = B3 / COM2 * 100 - 100;                                   SMI02780
         B5 = (COM2 - COM1) * CHI / (CP3 * 100);                       SMI02790
         B6 = (B3 - COM2) * CHI / (CP4 * 100);                         SMI02800
     AAA:                                                              SMI02810
        COM1 = COM1 / L;  COM2= COM2/ L;  B3 = B3 / L;                 SMI02820
     PUT SKIP(2) EDIT(SAN,SEE,CHI,COM1,COM2,B3,B1,B2,B5,B6)            SMI02830
         (X(2),A,X(8),A,X(4),F(11,1),F(10,1),F(10,1),F(10,1),F(18,1), SMI02840
         F(15,1),F(20,2),F(17,2));                                     SMI02850
             GO TO R5;                                                 SMI02860
   LA5:                                                                SMI02870
        CLOSE FILE(DIN);                                               SMI02880
   END TAB5;                                                           SMI02890
   TAB64: PROC;                                                        SMI02900
      ON ENDFILE(DIN) GO TO LA64;                                      SMI02910
      OPEN FILE(DIN);                                                  SMI02920
   R64:                                                                SMI02930
      READ FILE(DIN) INTO(IND);                                        SMI02940
      IF SWX = '*' THEN GO TO J1;                                      SMI02950
      IF PUM ¬= '      ' THEN GO TO R64;                               SMI02960
   J1:  IF SEE = SNO THEN GO TO MP64;                                  SMI02970
      PUT PAGE;                                                        SMI02980
      SNO = SEE;                                                       SMI02990
PUT SKIP(3) EDIT(CSC,'#6 TABLE OF INDEX BY COMMODITES BY MONTH-JISU')  SMI03000
                 (X(10),A,X(33),A);                                    SMI03010
       PUT SKIP(3) EDIT('C O D E GUBUN    **1**   **2**   **3**   **4**SMI03020
   **5**   **6**   **7**   **8**   **9**   *10**   *11**   *12**   AVERSMI03030
G')(A);                                                                SMI03040
   MP64:                                                               SMI03050
      COM1 = 0;                                                        SMI03060
      DO I = K TO 13;                                                  SMI03070
      COM1 = COM1 + JISU(I+24);                                        SMI03080
      END;                                                             SMI03090
      B1 = COM1 / L;                                                   SMI03100
      IF PUM ¬= '      ' THEN SAN = '    ';                            SMI03110
      PUT SKIP(2) EDIT(SAN,PUM,SEE,(JISU(I) DO I = (K+24) TO 37))      SMI03120
                  (A(3),A(5),X(3),A,X(3), 12 F(8,1));                  SMI03130
      PUT SKIP(0) EDIT(B1)   (X(115),F(8,1));                          SMI03140
      GO TO R64;                                                       SMI03150
   LA64:                                                               SMI03160
      CLOSE FILE(DIN);                                                 SMI03170
      END TAB64;                                                       SMI03180
   TAB65: PROC;                                                        SMI03190
      ON ENDFILE(DIN) GO TO LA65;                                      SMI03200
      OPEN FILE(DIN);                                                  SMI03210
   R65:                                                                SMI03220
      READ FILE(DIN) INTO(IND);                                        SMI03230
      IF SWX = '*' THEN GO TO J2;                                      SMI03240
      IF PUM ¬= '      ' THEN GO TO R65;                               SMI03250
 J2:  IF SEE = SNO THEN GO TO MP65;                                    SMI03260
      PUT PAGE;        SNO = SEE;                                      SMI03270
  PUT SKIP(3) EDIT(CSC,'#6 TABLE OF INDEX BY COMMODITES - JUNWOLBI')   SMI03280
                 (X(10),A,X(33),A);                                    SMI03290
       PUT SKIP(3) EDIT('C O D E GUBUN    **1**   **2**   **3**   **4**SMI03300
```

```
   **5**   **6**   **7**   **8**   **9**   *10**   *11**   *12**   AVERSMI03310
G') (A);                                                          SMI03320
   MP65:                                                          SMI03330
      COM1 = 0;              KYE = 0;                             SMI03340
      DO I = K TO 13;                                             SMI03350
      IF JISU(I+23) = 0 THEN KYE(I-K+1)= 0;                       SMI03360
      ELSE KYE(I-K+1) = JISU(I+24) / J1SU(I+23) * 100 - 100;      SMI03370
      COM1 = COM1 + KYE(I-K+1);                                   SMI03380
      END;                                                        SMI03390
      B1 = COM1 / L;                                              SMI03400
      IF PUM  = '      ' THEN PUM = SAN ;                         SMI03410
      PUT SKIP(2) EDIT(PUM,SEE,(KYE(I) DO I = 1 TO 12),B1)        SMI03420
                     (X(3),A(5),X(3),A,X(3), 13 F(8,1));          SMI03430
      GO TO R65;                                                  SMI03440
   LA65:                                                          SMI03450
      CLOSE FILE(DIN);                                            SMI03460
      END TAB65;                                                  SMI03470
 TAB66: PROC;                                                     SMI03480
      ON ENDFILE(DIN) GO TO LA66;                                 SMI03490
      OPEN FILE(DIN);                                             SMI03500
   R66:                                                           SMI03510
      READ FILE(DIN) INTO(IND);                                   SMI03520
      IF SWX = '*' THEN GO TO J3;                                 SMI03530
      IF PUM ¬= '      ' THEN GO TO R66;                          SMI03540
 J3:  IF SEE = SNO THEN GO TO MP66;                               SMI03550
      PUT PAGE;          SNO = SEE;                               SMI03560
  PUT SKIP(3) EDIT(CSC,'#6 TABLE OF INDEX BY COMMODITES - JUNDONGWUL') SMI03570
                 (X(10),A,X(33),A);                               SMI03580
       PUT SKIP(3) EDIT('C O D E GUBUN    **1**   **2**   **3**   **4**SMI03590
   **5**   **6**   **7**   **8**   **9**   *10**   *11**   *12**   AVERSMI03600
G') (A);                                                          SMI03610
   MP66:                                                          SMI03620
      COM1 = 0;              KYE = 0;                             SMI03630
      DO I = K TO 13;                                             SMI03640
      IF JISU(I+12) = 0 THEN KYE(I-K+1) = 0;                      SMI03650
      ELSE KYE(I-K+1) = JISU(I+24) / JISU(I+12) * 100 - 100;      SMI03660
      COM1 = COM1 + KYE(I-K+1);                                   SMI03670
      END;                                                        SMI03680
      B1 = COM1 / L;                                              SMI03690
      IF PUM  = '      ' THEN PUM = SAN;                          SMI03700
      PUT SKIP(2) EDIT(PUM,SEE,(KYE(I) DO I = 1 TO 12),B1)        SMI03710
                     (X(3),A(5),X(3),A,X(3), 13 F(8,1));          SMI03720
      GO TO R66;                                                  SMI03730
   LA66:                                                          SMI03740
      CLOSE FILE(DIN);                                            SMI03750
      END TAB66;                                                  SMI03760
 TAB67: PROC;                                                     SMI03770
      ON ENDFILE(DIN) GO TO LA67;                                 SMI03780
      OPEN FILE(DIN);                                             SMI03790
   R67:                                                           SMI03800
      READ FILE(DIN) INTO(IND);                                   SMI03810
      IF SWX = '*' THEN GO TO J4;                                 SMI03820
      IF PUM ¬= '      ' THEN GO TO R67;                          SMI03830
 J4:  IF SEE = SNO THEN GO TO MP67;                               SMI03840
      PUT PAGE;          SNO = SEE;                               SMI03850
```

```
  PUT SKIP(3) EDIT(CSC,'#6 TABLE OF INDEX BY COMMODITES - JUNDONGKI')  SMI03860
                 (X(10),A,X(33),A);                                    SMI03870
      PUT SKIP(3) EDIT('C O D E GUBUN    **1**   **2**   **3**   **4**SMI03880
   **5**   **6**   **7**   **8**   **9**   *10**   *11**   *12**   AVERSMI03890
G') (A);                                                              SMI03900
   MP67:                                                              SMI03910
      KYE= 0 ;  COM1 = 0; COM2 = 0; B1 = 0;                           SMI03920
      DO I = K TO 13;                                                 SMI03930
      DO P = K  TO I;                                                 SMI03940
      COM1 = COM1 + JISU(P+24) ;  COM2 = COM2 + JISU(P+12);           SMI03950
      END;                                                            SMI03960
        IF COM2 = 0 THEN KYE(I-K+1) = 0;    ELSE                      SMI03970
    KYE(I-K+1) = COM1 / COM2 * 100 - 100;   B1 = B1 + KYE(I-K+1);     SMI03980
        COM1 = 0;   COM2 = 0;    END;                                 SMI03990
      B1 = B1 / L;                                                    SMI04000
      IF PUM  = '      ' THEN PUM = SAN;                              SMI04010
      PUT SKIP(2) EDIT(PUM,SEE,(KYE(I) DO I = 1 TO 12),B1)            SMI04020
                    (X(3),A(5),X(3),A,X(3), 13 F(8,1));               SMI04030
      GO TO R67;                                                      SMI04040
   LA67:                                                              SMI04050
      CLOSE FILE(DIN);                                                SMI04060
      END TAB67;                                                      SMI04070
 TABA: PROC;                                                          SMI04080
      ON ENDFILE(DIN) GO TO LAA;                                      SMI04090
      OPEN FILE(DIN);                                                 SMI04100
   RAA:                                                               SMI04110
      READ FILE(DIN) INTO(IND);                                       SMI04120
      IF SEE ¬= '3' THEN GO TO RAA;                                   SMI04130
      SEE = '4';                                                      SMI04140
      IF SWX = '*' THEN GO TO JA;                                     SMI04150
      IF PUM ¬= '      ' THEN GO TO RAA;                              SMI04160
 JA:  IF SEE = SNO THEN GO TO MPA;                                    SMI04170
      PUT PAGE;   SNO = SEE;                                          SMI04180
  PUT SKIP(3) EDIT(CSC,'#A TABLE OF INDEX BY COMMODITES - GADONGYUL')  SMI04190
                 (X(10),A,X(33),A);                                    SMI04200
      PUT SKIP(3) EDIT('C O D E GUBUN    **1**   **2**   **3**   **4**SMI04210
   **5**   **6**   **7**   **8**   **9**   *10**   *11**   *12**   AVERSMI04220
G') (A);                                                              SMI04230
   MPA:                                                               SMI04240
      KYE= 0 ;  COM1 = 0;                                             SMI04250
      DO I = K TO 13;                                                 SMI04260
    KYE(I-K+1) = MUL(I+24); KYE(I-K+1) = KYE(I-K+1) / 10;             SMI04270
    COM1 = KYE(I-K+1) + COM1;   END;                                  SMI04280
      B1 = COM1 / L;                                                  SMI04290
      IF PUM ¬= '      ' THEN SAN = '    ';                           SMI04300
      PUT SKIP(2) EDIT(SAN,PUM,'4',(KYE(I) DO I = 1 TO 12))           SMI04310
                    (X(3),A,A,X(3),A,X(3), 12 F(8,1));                SMI04320
      PUT SKIP(0) EDIT(B1) (X(115),F(8,1));                           SMI04330
      GO TO RAA;                                                      SMI04340
   LAA:                                                               SMI04350
      CLOSE FILE(DIN);                                                SMI04360
      END TABA;                                                       SMI04370
   TAB7: PROC;                                                        SMI04380
          ON ENDFILE(DIN) GO TO LA7;                                  SMI04390
          OPEN FILE(DIN);                                             SMI04400
```

```
   R7:                                                                  SMI04410
          READ FILE(DIN) INTO(IND);                                     SMI04420
          IF PUM = '      ' THEN GO TO R7;                              SMI04430
          IF SEE = SNO THEN GO TO MP7;                                  SMI04440
          PUT PAGE;                                                     SMI04450
          SNO = SEE;                                                    SMI04460
       PUT SKIP(3) EDIT(CSC,'#7 TABLE OF CHANGE BY COMMODITIES')        SMI04470
                          (X(10),A,X(36),A);                            SMI04480
       PUT SKIP(3) EDIT('CO DE DIVISION            WEIGHT   QN-PM-1  QUNSMI04490
-PM  IPM-1 IN-PM  IC-VJW  IC-VDW IC-VDG   GI-VJW   GI-VDW   GI-VDG')   SMI04500
                  (X(7),A);                                             SMI04510
         IF SAN ¬) '300' THEN SDATA = DATA;                             SMI04520
   MP7:                                                                 SMI04530
          IF JISU(KI-1) = 0 THEN  B1 = 0;                               SMI04540
           ELSE  B1 = JISU(KI) / JISU(KI-1) * 100 - 100;                SMI04550
          IF SJI(KI-1) = 0 THEN  B5 = 0 ;                               SMI04560
       ELSE B5 = (JISU(KI)-JISU(KI-1)) * CHI / (SJI(KI-1) * 100);       SMI04570
           IF JISU(KI-12) = 0 THEN  B2 = 0;                             SMI04580
              ELSE  B2 = JISU(KI) / JISU(KI-12) * 100 - 100;            SMI04590
           IF SJI(KI-12) = 0 THEN  B6 = 0;                              SMI04600
    ELSE   B6 = (JISU(KI)-JISU(KI-12)) * CHI / (SJI(KI-12) * 100);      SMI04610
          COM1 = 0;    COM2 = 0;                                        SMI04620
         DO I = K TO 13;                                                SMI04630
             COM1 = COM1 + JISU(I+24);                                  SMI04640
             COM2 = COM2 + JISU(I+12);                                  SMI04650
             CP4 = CP4 + SJI(I+12);                                     SMI04660
             END;                                                       SMI04670
    IF COM2 = 0 THEN  B3 = 0;                                           SMI04680
        ELSE B3 = COM1 / COM2 * 100 - 100;                              SMI04690
           IF CP4 = 0 THEN  B7 = 0;                                     SMI04700
    ELSE   B7 = (COM1-COM2) * CHI / (CP4 * 100);                        SMI04710
         PUT SKIP(2) EDIT(PUM,SEE,CHI,MUL(KI-1),MUL(KI),                SMI04720
             JISU(KI-1),JISU(KI),B1,B2,B3,B5,B6,B7)                     SMI04730
         (X(7),A(5),X(4),A,X(10),F(11,1),F(10),F(8),F(7,1),             SMI04740
              F(6,1),F(9,1),F(8,1),F(7,1),F(9,2),F(9,2),F(9,2));        SMI04750
              GO TO R7;                                                 SMI04760
    LA7:                                                                SMI04770
           CLOSE FILE(DIN);                                             SMI04780
             END TAB7;                                                  SMI04790
    TAB8: PROC;                                                         SMI04800
         ON ENDFILE(DIN) GO TO LA8;                                     SMI04810
         OPEN FILE(DIN);                                                SMI04820
    R8:                                                                 SMI04830
         READ FILE(DIN) INTO(IND);                                      SMI04840
         IF PUM ¬= '      ' THEN GO TO R8;                              SMI04850
         IF SNO = SEE THEN GOTO MP8;                                    SMI04860
          SNO = SEE;                                                    SMI04870
         PUT PAGE;                                                      SMI04880
    PUT SKIP(3) EDIT(CSC,'#8 TABLE OF INDEX S_A INDEX BY(B) BY MONTHLYSMI04890
(A)') (X(10),A,X(30),A);                                                SMI04900
      PUT SKIP(3) EDIT('C O D E    DIVISION         WEIGHT         PM - 2SMI04910
        PM - 1          P   M          V J W        VJNDW        VJNDG SMI04920
  ')   (X(1),A);                                                        SMI04930
        IF SAN ¬) '300' THEN SDATA = DATA;                              SMI04940
         MP8:                                                           SMI04950
```

```
            IF JISU(KI-1) = 0 THEN B1 = 0;                                  SMI04960
              ELSE B1 = JISU(KI) / JISU(KI-1) * 100 - 100;    COM2 = 0; SMI04970
            IF JISU(KI-12) = 0 THEN B2 = 0;                                 SMI04980
              ELSE B2 = JISU(KI) / JISU(KI-12) * 100 - 100;   COM1 = 0; SMI04990
          DO I = K TO 13;                                                   SMI05000
              COM1 = COM1 + JISU(I+24);                                     SMI05010
              COM2 = COM2 + JISU(I+12);                                     SMI05020
              END;                                                          SMI05030
          IF COM2 = 0 THEN B3 = 0; ELSE                                     SMI05040
              B3 = COM1 / COM2 * 100 - 100;                                 SMI05050
      PUT SKIP(2) EDIT(SAN,SEE,CHI,JISU(KI-2),JISU(KI-1),JISU(KI),          SMI05060
              B1,B2,B3) (X(5),A,X(6),A,X(3),F(16,1),F(14,1),F(15,1),        SMI05070
                   F(15,1),F(14,1),F(13,1),F(13,1));                        SMI05080
              GO TO R8;                                                     SMI05090
          LA8:                                                              SMI05100
              CLOSE FILE(DIN);                                              SMI05110
              END TAB8 ;                                                    SMI05120
          TAB9: PROC;                                                       SMI05130
              ON ENDFILE(DIN) GO TO LA9;                                    SMI05140
              OPEN FILE(DIN);                                               SMI05150
          R9:                                                               SMI05160
              READ FILE(DIN) INTO(IND);                                     SMI05170
              IF PUM = '      ' THEN GO TO R9;                              SMI05180
              IF SEE = '3' THEN GO TO LA9;                                  SMI05190
              IF SEE = SNO THEN GO TO MP9;                                  SMI05200
              PUT PAGE;   SNO = SEE;                                        SMI05210
          PUT SKIP(3) EDIT(CSC,'#9 TABLE OF QUANTITY BY MONTHLY BY COMMSMI05220
ODITIES') (X(10),A,X(32),A);                                                SMI05230
              PUT SKIP(3)  EDIT('C O D E    **1**    **2**    **3**'      SMI05240
**4**    **5**    **6**    **7**    **8**    **9**    **10**   **11**      SMI05250
**12**     TOTAL') (X(1),A);                                                SMI05260
           MP9:                                                             SMI05270
              COM1 = 0;                                                     SMI05280
              DO I = K TO 13;   COM1 = MUL(I+24) + COM1; END;               SMI05290
              PUT SKIP(2) EDIT(PUM,(MUL(1) DO I=K+24 TO 37))                SMI05300
                      (X(3),A(5), 12 F(9));                                 SMI05310
              PUT SKIP(0) EDIT(COM1)  (X(115),F(10));                       SMI05320
              GOTO R9;                                                      SMI05330
           LA9:                                                             SMI05340
          CLOSE FILE(DIN);                                                  SMI05350
           END TAB9 ;                                                       SMI05360
          TAB10: PROC;                                                      SMI05370
              DCL ML(12) FLOAT INIT((12) 0);                                SMI05380
              ON ENDFILE(DIN) GO TO LA10;                                   SMI05390
              OPEN FILE(DIN);                                               SMI05400
              PUT PAGE;                                                     SMI05410
     PUT SKIP(3) EDIT(CSC,'#10 TABLE OF GADONGYUL - WAY OF 2ND COMPUTE SMI05420
(A)') (X(10),A,X(30),A);                                                    SMI05430
              PUT SKIP(3)  EDIT('C O D E    **1**    **2**    **3**       SMI05440
**4**    **5**    **6**    **7**    **8**    **9**    **10**   **11**      SMI05450
**12**     AVE') (X(1),A);                                                  SMI05460
        R10:  READ FILE(DIN) INTO(IND);                                     SMI05470
              IF SEE ¬= '3' THEN GO TO R10;                                 SMI05480
           P = 0;                                                           SMI05490
          DO I = K TO 13;                                                   SMI05500
```

```
            P = P + 1;                                                  SMI05510
            ML(P) = JISU(I + 24) * YANG / 100 + 0.5;                    SMI05520
            B1 = ML(P) + B1;   END;                                     SMI05530
               B1 = B1 / P;                                             SMI05540
       COM1=(ML(1)+ML(2)+ML(3)+ML(4)+ML(5)+ML(6)+                      SMI05550
             ML(7)+ML(8)+ML(9)+ML(10)+ML(11)+ML(12))/L;                SMI05560
               PUT SKIP(2) EDIT(SAN,PUM,(ML(I) DO I = 1 TO 12))         SMI05570
                        (A,A, 12 F(9));                                 SMI05580
               PUT SKIP(0) EDIT(COM1)  (X(116),F(10,1));                SMI05590
               GO TO R10;                                               SMI05600
            LA10:                                                       SMI05610
            END TAB10;                                                  SMI05620
     EVE:                                                               SMI05630
      CLOSE FILE(DIN);                                                  SMI05640
       END  POTAB;                                                      SMI05650
/*                                                                      SMI05660
//LKED.SYSLMOD DD DSN=USER.LOADLIB(SMITB34),DISP=SHR                    SMI05670
//                                                                      SMI05680
```

SMICCA65

생 산 능 력 (월 간)

(CATAL PROGRAM(3))

```
//SMICCA65 JOB CLASS=A,TYPRUN=HOLD                                      SMI00010
//STEP2 EXEC PLIFCL                                                     SMI00090
//PLI.SYSIN DD *                                                        SMI00100
* PROCESS GS,NEST ;                                                     SMI00110
      POSEL:   PROC OPTIONS(MAIN);                                      SMI00120
           DCL DDD FILE RECORD KEYED SEQUENTIAL INPUT ENV(VSAM);        SMI00130
           DCL DOF FILE RECORD OUTPUT;                                  SMI00140
           DCL 1 IN,                                                    SMI00150
                3 CNO CHAR(1);                                          SMI00160
                3 SEO CHAR(1),                                          SMI00170
                3 SAN CHAR(3),                                          SMI00180
                3 PUM CHAR(5),                                          SMI00190
                3 FL1 CHAR(1),                                          SMI00200
                3 SIL PIC '9999',                                       SMI00210
                3 CHI FLOAT,                                            SMI00220
                3 YANG FIXED BIN(31),                                   SMI00230
                3 DATA(37),                                             SMI00240
                  4 MUL FIXED BIN(31),                                  SMI00250
                  4 JI  FLOAT,                                          SMI00260
                3 FL2 CHAR(1);                                          SMI00270
           ON ENDFILE(DDD) GOTO LA1;                                    SMI00280
      OPEN FILE(DDD),FILE(DOF),FILE(SYSPRINT) LINESIZE(132);            SMI00290
      RD:                                                               SMI00300
           READ FILE(DDD) INTO (IN);                                    SMI00310
           IF CNO ¬= '6' THEN GO TO RD;                                 SMI00320
           WRITE FILE(DOF) FROM(IN);                                    SMI00330
           GO TO RD;                                                    SMI00340
          LA1:                                                          SMI00350
               CLOSE FILE(DDD),FILE(DOF);                               SMI00360
          END POSEL;                                                    SMI00370
/*                                                                      SMI00380
//LKED.SYSLMOD DD DSN=USER.LOADLIB(SMICA34),DISP=SHR                    SMI00390
//                                                                      SMI00400
```

SMICTA65

생 산 능 력 (월 간)

(CATAL PROGRAM(4))

```
//SMICTA65 JOB CLASS=A,TYPRUN=HOLD                                    SMI00010
//STEP2 EXEC PLIFCL                                                   SMI00090
//PLI.SYSIN DD *                                                      SMI00100
* PROCESS GS,NEST ;                                                   SMI00110
      POSEL:   PROC OPTIONS(MAIN);                                    SMI00120
           DCL DDD FILE RECORD KEYED SEQUENTIAL INPUT ENV(VSAM);      SMI00130
           DCL DOF FILE RECORD OUTPUT;                                SMI00140
           DCL 1 IN,                                                  SMI00150
                3 CNO CHAR(1),                                        SMI00160
                3 SEO CHAR(1),                                        SMI00170
                3 SAN CHAR(3),                                        SMI00180
                3 PUM CHAR(5),                                        SMI00190
                3 FL1 CHAR(1),                                        SMI00200
                3 SIL PIC '9999',                                     SMI00210
                3 CHI FLOAT,                                          SMI00220
                3 YANG FIXED BIN(31),                                 SMI00230
                3 DATA(37),                                           SMI00240
                  4 MUL FIXED BIN(31),                                SMI00250
                  4 JI  FLOAT,                                        SMI00260
                3 FL2 CHAR(1);                                        SMI00270
           ON ENDFILE(DDD) GOTO LA1;                                  SMI00280
      OPEN FILE(DDD),FILE(DOF),FILE(SYSPRINT) LINESIZE(132);          SMI00290
      RD:                                                             SMI00300
           READ FILE(DDD) INTO (IN);                                  SMI00310
           IF CNO ¬= '6' THEN GO TO RD;                               SMI00320
           WRITE FILE(DOF) FROM(IN);                                  SMI00330
           GO TO RD;                                                  SMI00340
          LA1:                                                        SMI00350
               CLOSE FILE(DDD),FILE(DOF);                             SMI00360
          END POSEL;                                                  SMI00370
/*                                                                    SMI00380
//LKED.SYSLMOD DD DSN=USER.LOADLIB(SMITA34),DISP=SHR                  SMI00390
//                                                                    SMI00400
```

Ⅷ. Program Catalog List

SMIWMT62

(MASTER UPDATE)

내용 : 확정치조사표 사업체 물량을 C.R.T에
PUNCH한 후 SMIJN12 (확정치)에
UPDATE

```
//SMIWMT62 JOB CLASS=V,MSGCLASS=O,TYPRUN=HOLD                          SMI00010
//STEP1 EXEC PGM=IEHPROGM                                              SMI00020
//SYSPRINT DD SYSOUT=O                                                 SMI00030
//DD2    DD   UNIT=DISK,VOL=SER=BOSWK2,DISP=OLD                        SMI00040
//SYSIN   DD  *                                                        SMI00050
          SCRATCH DSNAME=SMIM66,VOL=DISK=BOSWK2                        SMI00060
/*                                                                     SMI00070
//STEP2 EXEC PLIFHG                                                    SMI00080
//PLI.SYSIN DD *                                                       SMI00090
* PROCESS S,A,GS,NEST,INCLUDE;                                          SMI00100
     LCH  : PROC OPTIONS(MAIN);                                        SMI00110
           DCL TIN FILE RECORD INPUT;                                  SMI00120
           DCL CIN FILE RECORD INPUT;                                  SMI00130
           DCL DOT FILE RECORD OUTPUT;                                 SMI00140
     DCL 1 ARTR,                                                       SMI00150
               2 TRNO CHAR(5),                                         SMI00160
               2 TRNN CHAR(5),                                         SMI00170
               2 TRON CHAR(5),                                         SMI00180
               2 TRSA CHAR(5),                                         SMI00190
               2 TRSDN CHAR(7),                                        SMI00200
               2 GFDJ,                                                 SMI00210
                 3 TRSU CHAR(2),                                       SMI00220
                 3 TRPH CHAR(1),                                       SMI00230
                 3 TRDW CHAR(1),                                       SMI00240
                 3 TRF1 CHAR(4),                                       SMI00250
                 3 TRMM CHAR(4),                                       SMI00260
                 3 TRGU CHAR(1),                                       SMI00270
               2 TNAME CHAR(32),                                       SMI00280
               2 TROS CHAR(7),                                         SMI00290
               2 TRED CHAR(1) ;                                        SMI00300
      DCL 1 MST,                                                       SMI00310
               2 MNO CHAR(5),                                          SMI00320
               2 MNN CHAR(5),                                          SMI00330
               2 MON CHAR(5),                                          SMI00340
               2 MSA CHAR(5),                                          SMI00350
               2 MDN CHAR(7),                                          SMI00360
               2 MSU CHAR(2),                                          SMI00370
               2 MPH CHAR(1),                                          SMI00380
               2 MDW CHAR(1),                                          SMI00390
               2 MF1 CHAR(4),                                          SMI00400
               2 MMM CHAR(4),                                          SMI00410
               2 MGU CHAR(1),                                          SMI00420
               2 MNAME CHAR(32),                                       SMI00430
               2 MOS CHAR(7),                                          SMI00440
               2 MED CHAR(1);                                          SMI00450
      DCL SW PIC '9' INIT(2);                                          SMI00460
      DCL BSAN CHAR(3) INIT('001');                                    SMI00470
           ON ENDFILE(TIN) GO TO LAST;                                 SMI00480
           ON ENDFILE(CIN) MNO='99999';                                SMI00490
           OPEN FILE(CIN),FILE(SYSPRINT) LINESIZE(132) PAGESIZE(66);   SMI00500
           OPEN FILE(TIN),FILE(DOT);                                   SMI00510
           OPEN           FILE(DOT);                                   SMI00520
           SW=2;                                                       SMI00530
      RD1:                                                             SMI00540
           IF SW <  3 THEN DO;                                         SMI00550
```

```
      RET:  READ FILE(TIN) INTO(ARTR);                                  SMI00560
            IF TRGU ¬= ' ' & TRGU ¬= '6' & TRGU ¬= '9' THEN DO;         SMI00570
               WRITE FILE(DOT) FROM(ARTR);                              SMI00580
               GO TO RET;                                               SMI00590
            END;                                                        SMI00600
          END;                                                          SMI00610
          IF SW >  1 THEN                                               SMI00620
          READ FILE(CIN) INTO(MST);                                     SMI00630
          IF TRNO < MNO THEN DO;                                        SMI00640
           WRITE FILE(DOT) FROM(ARTR); SW=1; GO TO RD1;                 SMI00650
          END;                                                          SMI00660
          IF TRNO > MNO THEN DO;                                        SMI00670
           SW=3;  PUT SKIP;                                             SMI00680
           IF MGU = ' ' & MMM ¬='     ' THEN DO;                        SMI00690
               WRITE FILE(DOT) FROM(MST);                               SMI00700
               CALL PRT(STRING(MST),'INSERT *');                        SMI00710
               GO TO RD1;                                               SMI00720
           END;                                                         SMI00730
           CALL PRT(STRING(MST),'NO MASTER ***');                       SMI00740
           GO TO RD1;                                                   SMI00750
          END;                                                          SMI00760
            SW=2;                                                       SMI00770
            IF MNN = '      ' THEN DO;                                  SMI00780
             IF MGU = '6' THEN DO;                                      SMI00790
                   PUT SKIP;                                            SMI00800
                   CALL PRT(STRING(MST),'NO NEWKEY ***');               SMI00810
                    GO TO RD1;                                          SMI00820
             END;                                                       SMI00830
             ELSE GO TO CHANGE;                                         SMI00840
            END;                                                        SMI00850
            ELSE IF MGU ='6' THEN DO;                                   SMI00860
              PUT SKIP; CALL PRT(STRING(ARTR),'MASTER');                SMI00870
             TRON=TRNN; TRNN=TRNO; TRNO=MNN; GO TO CHAN11; END;         SMI00880
             ELSE IF MGU='8' THEN GO TO CHAN11;                         SMI00890
             ELSE DO; CALL PRT(STRING(MST),'NO 6 KEY ***'); GO TO RD1;  SMI00900
                 END;                                                   SMI00910
  CHANGE: PUT SKIP;                                                     SMI00920
          CALL PRT(STRING(ARTR),'MASTER');                              SMI00930
  CHAN11: CALL PRT(STRING(MST),'TRANS');                                SMI00940
          IF MSA ¬='      ' THEN TRSA=MSA;                              SMI00950
          IF MDN ¬='        ' THEN TRSDN=MDN;                           SMI00960
          IF MSU ¬='   ' THEN TRSU=MSU;                                 SMI00970
          IF MPH ¬=' ' THEN DO;                                         SMI00980
             IF MPH = '0' THEN TRPH = ' '; ELSE TRPH=MPH;               SMI00990
          END;                                                          SMI01000
          IF MDW ¬=' ' THEN DO;                                         SMI01010
             IF MDW = '0' THEN TRDW = ' '; ELSE TRDW=MDW;               SMI01020
          END;                                                          SMI01030
          IF MMM ¬='     ' THEN TRMM=MMM;                               SMI01040
          IF MGU ¬=' ' THEN DO;                                         SMI01050
             IF MGU = '0' THEN TRGU = ' '; ELSE TRGU=MGU;               SMI01060
          END;                                                          SMI01070
          IF MNAME ¬='     ' THEN TNAME=MNAME;                          SMI01080
          CALL PRT(STRING(ARTR),'NEW');                                 SMI01090
             WRITE FILE(DOT) FROM(ARTR);                                SMI01100
```

```
          GOTO RD1;                                                    SMI01110
  PRT:    PROC(RRA,MOKK);                                              SMI01120
          DCL RRA CHAR(80);                                            SMI01130
     DCL 1 RTYR DEF RRA,                                               SMI01140
              2 TRNO CHAR(5),                                          SMI01150
              2 TRNN CHAR(5),                                          SMI01160
              2 TRON CHAR(5),                                          SMI01170
              2 TRSA CHAR(5),                                          SMI01180
              2 TRSDN CHAR(7),                                         SMI01190
              2 GFDJ,                                                  SMI01200
                3 TRSU CHAR(2),                                        SMI01210
                3 TRPH CHAR(1),                                        SMI01220
                3 TRDW CHAR(1),                                        SMI01230
                3 TRF1 CHAR(4),                                        SMI01240
                3 TRMM CHAR(4),                                        SMI01250
                3 TRGU CHAR(1),                                        SMI01260
              2 MNAME CHAR(32),                                        SMI01270
              2 TROS CHAR(7),                                          SMI01280
              2 TRED CHAR(1) ;                                         SMI01290
        DCL MOKK CHAR(20);                                             SMI01300
          PUT SKIP(1) EDIT(TRNO,TRNN,TRON,TRSA,TRSDN,                  SMI01310
                          STRING(GFDJ),MOKK)                           SMI01320
                            (X(15),4 A(6),A(8),A,X(39),A);             SMI01330
          CALL HPUT1('OS',0,61,32,MNAME);                              SMI01340
          END PRT;                                                     SMI01350
  % INCLUDE HANPRTS;                                                   SMI01360
  LAST:   CLOSE FILE(TIN),FILE(CIN),FILE(DOT);                         SMI01370
      END LCH;                                                         SMI01380
/*                                                                     SMI01390
//GO.TIN DD UNIT=TAPE,DISP=(OLD,KEEP),DSN=KKL20,VOL=SER=343434,        SMI01400
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200),LABEL=(2,BLP)         SMI01410
//GO.DOT DD UNIT=DISK,DISP=(NEW,KEEP),DSN=SMIM66,VOL=SER=BOSWK2,       SMI01420
//        DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200),SPACE=(80,(5000,500))   SMI01430
//GO.SYSPRINT DD SYSOUT=O                                              SMI01440
//GO.SYSOUT DD SYSOUT=O                                                SMI01450
//GO.CIN DD *                                                          SMI01460
00306                                 84089                            SMI01470
00320                                 84089                            SMI01480
00503                                 84089                            SMI01490
0061030106     321113107012  1        84088                            SMI01500
00703                                 8408 ROSKFLQURWL(WN)             SMI01510
0072112503     372013802054  1        84088                            SMI01520
0082040702     369213300887  1        84088                            SMI01530
0221525327                            84086                            SMI01540
0251525230     311622300111  1        84088                            SMI01550
0251611007     311622100781  1        84088                            SMI01560
0260910117     371032106184           84088                            SMI01570
0271400430                            84086                            SMI01580
0291000732                            84086                            SMI01590
02912                         0       8408                             SMI01600
0371926538                            84086                            SMI01610
0373082310     371033701108  1        84088                            SMI01620
0381026620                            84086                            SMI01630
03910                                 84089                            SMI01640
0451921026                            84086                            SMI01650
```

```
0472225307     372022301173  1     84088                                  SMI01660
0512190803     311533801271  1     84088                                  SMI01670
0512604929                         84086                                  SMI01680
0522020522                         84086                                  SMI01690
0541400731                         84086                                  SMI01700
05420                              84089                                  SMI01710
0621112325     372013800601  1     84088                                  SMI01720
0681550302     321173400055  1     84088                                  SMI01730
0751730112                         84086                                  SMI01740
1013511012                         84086                                  SMI01750
1022610732                         84086                                  SMI01760
1050440121                         84086                                  SMI01770
1051311543                         84086                                  SMI01780
11005                              84089                                  SMI01790
11207                              84089                                  SMI01800
12123                              8408 (WN)TPFLA                         SMI01810
12204                              8408 VKFALRMATHR(WN) RLAGORHDWKD       SMI01820
12502                         1    8408                                   SMI01830
12513                              84089                                  SMI01840
20115                              8408 (WN)TLSVUDANFTKS                  SMI01850
20720                              84089                                  SMI01860
20730                              84089                                  SMI01870
20742                              84089                                  SMI01880
2100121302     381363104344        84088                                  SMI01890
21605                              8408 GKSDIDDYDJQTK(WN)                 SMI01900
21804                              8408 (WN)DNTJD                         SMI01910
21903                              84084                                  SMI01920
21907                              84089                                  SMI01930
2533225232                         84086                                  SMI01940
26009                              84089                                  SMI01950
26104                              84089                                  SMI01960
3011106922     385471129681        84086(WN)\JALXM                        SMI01970
40207                              84089                                  SMI01980
40406                              84089                                  SMI01990
51006                              8408 CJSDNJSGHKTJD(WN)                 SMI02000
5150803517     351221113321  1     84088                                  SMI02010
60617                              84089                                  SMI02020
60717                              84089                                  SMI02030
60804                              84089                                  SMI02040
60811                              84089                                  SMI02050
61108                              84089                                  SMI02060
61505                              84089                                  SMI02070
70606                              84089                                  SMI02080
71203          35114               8408                                   SMI02090
7120803209     351111103746  1     84088                                  SMI02100
80333                              8408 (WN)TKAGHRLR&                     SMI02110
8050783114                         84086                                  SMI02120
80724                              84089                                  SMI02130
81217                              84089                                  SMI02140
82102                              8408 WPCJFWJDQL(WN)                    SMI02150
90207                              8408 EOGKSWJSWK(WN)                    SMI02160
/*                                                                        SMI02170
//                                                                        SMI02180
```

SMIWMS 62

MASTER CREATION
(RUN PROGRAM)

내용 : MASTER UPDATE (1번 STEP) 후
MASTER FILE에 물량 UPDATE 된
것을 CREATION 한다.

FILE: SMIWMS66 MCREATE D1 VM/SP RELEASE 2.1 EXPRESS PUT8302+ SLU203

```
//SMIWMS62  JOB CLASS=0,TYPRUN=HOLD                                    SMI00010
//JOBCAT DD DSN=SMICT12,DISP=SHR
//STEP1     EXEC  PGM=IDCAMS                                           DEF00020
//DD1       DD   VOL=SER=BOSWK2,UNIT=3350,DISP=OLD                     DEF00030
//SYSPRINT  DD   SYSOUT=A                                              DEF00040
//SYSIN     DD   *                                                     DEF00050
        DELETE -                                                       DEF00060
             SMISA12 -                                                 DEF00070
             FILE(DD1) -
             PURGE -
             CATALOG(SMICT12)                                          DEF00130
/*                                                                     DEF00140
//STEP2     EXEC  PGM=IDCAMS                                           DEF00020
//DD1       DD   VOL=SER=BOSWK2,UNIT=3350,DISP=OLD                     DEF00030
//SYSPRINT  DD   SYSOUT=A                                              DEF00040
//SYSIN     DD   *                                                     DEF00050
        DEFINE CLUSTER -                                               DEF00060
            (NAME(SMISA12) -                                           DEF00070
             VOLUME(BOSWK2) -                                          DEF00080
             INDEXED -                                                 DEF00090
             RECORDS(4700 100) -                                       DEF00100
             KEYS(5 0) -                                               DEF00110
             RECORDSIZE(80 80)) -                                      DEF00120
             CATALOG(SMICT12)                                          DEF00130
/*                                                                     DEF00140
//SORT EXEC PGM=SORT,PARM='SIZE(MAX)'
//SYSOUT DD SYSOUT=A
//SORTIN DD UNIT=DISK,VOL=SER=BOSWK2,DSN=SMIM66,
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200),DISP=OLD
//SORTOUT DD UNIT=DISK,DSN=SMIM66,VOL=SER=BOSWK2,
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200),DISP=OLD
//SORTWK01 DD UNIT=3350,SPACE=(CYL,(20))
//SYSIN  DD *
    SORT FIELDS=(1,5,A,6,5,A),FORMAT=CH
/*
//SS EXEC PGM=SMIMS12                                                 SMI00080
//STEPLIB DD DSN=USER.LOADLIB,DISP=SHR                                SMI00090
//SYSPRINT DD SYSOUT=A
//INF DD UNIT=DISK,DSN=SMIM66,VOL=SER=BOSWK2,DISP=OLD,
//  DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
//OTT DD DSN=SMISA12,DISP=SHR
//
```

SMIWTY66

MASTER COUNT

내용 : 전국 시 · 도별 사업체수 COUNT.

```
//SMIWTY66 JOB CLASS=5,TYPRUN=HOLD,MSGCLASS=O                          SMI00010
//JOBCAT DD DSN=SMICT12,DISP=SHR                                       SMI00020
//      EXEC PLIFHG                                                    SMI00030
//PLI.SYSIN DD *                                                       SMI00060
* PROCESS,S,GS,NEST,A,XREF,OPT(2),INCLUDE ;                            SMI00070
   /*************************************************************/
   /*****              MASTER DOBYUL COUNT PROGRAM            *****/
   /*************************************************************/
  EYECHK : PROC OPTIONS(MAIN) ;                                        SMI00080
           DCL MOUT FILE RECORD KEYED INPUT ENV(VSAM);                 SMI00090
           DCL 1 IN,                                                   SMI00100
                2 GI          CHAR(5),                                 SMI00110
                2 F1          CHAR(75) ;                               SMI00120
           DCL (I,J)         PIC'(7)9' INIT(0);                        SMI00130
           DCL C(15)         PIC'(7)9' INIT((15)0);                    SMI00140
           DCL  INN          FIXED BIN(1) INIT('1'B) ;                 SMI00150
           DCL  NO           FIXED BIN(1) INIT('0'B) ;                 SMI00160
           DCL  GG           CHAR(3) DEF GI POS(1)   ;                 SMI00170
           DCL  SGI          CHAR(5) INIT((5)' ') ;                    SMI00180
           DCL  NAME(15)    CHAR(15) INIT('TJDNF EHDQN     ',          SMI00190
                                         'TJDNF TJQN      ',          SMI00200
                                         'TJDNF SKAQN     ',          SMI00210
                                         'TJDNF QNRQN     ',          SMI00220
                                         'QN TKS          ',          SMI00230
                                         'RUD RL          ',          SMI00240
                                         'DLS CJS         ',          SMI00250
                                         'RKD DNJS        ',          SMI00260
                                         'CND QNR         ',          SMI00270
                                         'CND SKA         ',          SMI00280
                                         'WJS QNR         ',          SMI00290
                                         'WJS SKA         ',          SMI00300
                                         'RUD QNR         ',          SMI00310
                                         'AK TKS          ',          SMI00320
                                         'WJS RNR         ') ;        SMI00330
        OPEN FILE(MOUT) SEQUENTIAL INPUT,FILE(SYSPRINT) LINESIZE(120); SMI00340
        ON ENDFILE(MOUT) INN  = NO ;                                   SMI00350
          READ FILE(MOUT) INTO(IN) ;                                   SMI00360
          DO UNTIL(INN = NO) ;                                         SMI00370
             IF SGI ¬= GI THEN DO ;                                    SMI00380
                I = I + 1 ;  SGI = GI ;                                SMI00390
             IF GG < '021' THEN C(1) = C(1) + 1 ;                      SMI00400
             ELSE   IF GG < '041' THEN C(2) = C(2) + 1 ;               SMI00410
             ELSE   IF GG < '061' THEN C(3) = C(3) + 1 ;               SMI00420
             ELSE   IF GG < '101' THEN C(4) = C(4) + 1 ;               SMI00430
             ELSE   IF GG < '201' THEN C(5) = C(5) + 1 ;               SMI00440
             ELSE   IF GG < '251' THEN C(6) = C(6) + 1 ;               SMI00450
             ELSE   IF GG < '301' THEN C(7) = C(7) + 1 ;               SMI00460
             ELSE   IF GG < '401' THEN C(8) = C(8) + 1 ;               SMI00470
             ELSE   IF GG < '501' THEN C(9) = C(9) + 1 ;               SMI00480
             ELSE   IF GG < '601' THEN C(10) = C(10) + 1 ;             SMI00490
             ELSE   IF GG < '701' THEN C(11) = C(11) + 1 ;             SMI00500
             ELSE   IF GG < '801' THEN C(12) = C(12) + 1 ;             SMI00510
             ELSE   IF GG < '901' THEN C(13) = C(13) + 1 ;             SMI00520
                   ELSE C(14) = C(14) + 1 ;                            SMI00530
    END ;                                                              SMI00540
```

```
   READ FILE(MOUT) INTO(IN) ;                                          SMI00550
   END ;                                                               SMI00560
   C(15) = I ;                                                         SMI00570
   DO J = 1 TO 15 ;                                                    SMI00580
   PUT SKIP(2) EDIT(C(J)) (X(25),F(7)) ;                               SMI00590
   CALL HPUT1 ('OS',0,3,15,NAME(J)) ;                                  SMI00600
   END ;                                                               SMI00610
   CLOSE FILE(MOUT);                                                   SMI00620
  %INCLUDE HANPRTS;                                                    SMI00630
  END EYECHK;                                                          SMI00640
/*                                                                     SMI00650
//GO.SYSPRINT DD SYSOUT=O,COPIES=5                                     SMI00700
//GO.MOUT  DD DSN=SMISA12,DISP=SHR                                     SMI00710
//                                                                     SMI00720
```

SMIWOU62

OK UPDATE (RUN PROGRAM)

내용 : 확정된 물량을 확정치(SMIJN12)
FILE에 UPDATE

FILE: SMIWOU66 OKUPDATE D1 ` VM/SP RELEASE 2.1 EXPRESS PUT8302+ SLU203

```
//SMIWOU62 JOB CLASS=7,TYPRUN=HOLD                                      SMI00010
//JOBCAT DD DSN=SMICT12,DISP=SHR
//  EXEC PGM=SMIOU12                                                    SMI00020
//STEPLIB DD DSN=USER.LOADLIB,DISP=SHR                                  SMI00030
//SYSPRINT DD SYSOUT=A,COPIES=3
//SYSOUT DD SYSOUT=A,COPIES=3
//SORTWK01 DD UNIT=3350,SPACE=(CYL,50)
/*TAIN DD UNIT=TAPE,DSN=ABC,VOL=SER=123321,LABEL=(,NL),
/*    DCB=(RECFM=F,BLKSIZE=80),DISP=OLD
//EDIT3 DD DSN=SMIGM12,DISP=SHR
//SYSIN DD *
08
/*
//TAIN DD *
81014657                                                                08 1
8101465621250 48     46                                                 08
81203402     327                                                        08
81203404               2409                                             08
81507257                                                                08 1
81512257                                                                08 1
8151320221260 0                                                         08
81513257                                                                08 1
81610257                                                                08 1
8161320221270-0                                                         08
81613254                                                                08 1
81613255                                                                08 1
81613257                                                                08 1
81615253                                                                08 1
81615257                                                                08 1
81618253                                                                08 1
81618255                                                                08 1
81618257                                                                08 1
81620257                                                                08 1
81722257                                                                08 1
8181620221250-                                                          08
81816253                                                                08 1
81816255                                                                08 1
81816257                                                                08 1
81819253                                                                08 1
81819255                                                                08 1
81819257                                                                08 1
8182220221270-                                                          08
81822255                                                                08 1
81822257                                                                08 1
82002201                                                                08 1
8200220121010 3571   2693   0      0      1800   0      4464   4        08
83108255                                                                08 1
83108257                                                                08 1
8311220221080-                                                          08
83112255                                                                08 1
83112257                                                                08 1
9021420221040-                                                          08
9021420421020-                                                          08
90214253                                                                08 1
90214255                                                                08 1
```

```
90214257                                                                    08 1
9072120221060-                                                              08
9072120421040-                                                              08
9072120621170-                                                              08
90721253                                                                    08 1
90721255                                                                    08 1
90721257                                                                    08 1
91502257                                                                    08 1
91502655                                                                    08 1
9150265621250 130    110                                                    08
91503202                                                                    08 1
91503255                                                                    08 1
91503257                                                                    08 1
0131565621250 169    103                                                    08
0250420383330 2005   2578   0      1559   839    0      2185   11          08
2620620281120 0      0      0      0      0      0      0      0           08
0250321382140 0      0      0      0      0      0      0      0           08
0250321482310 225    125    0      101    0      0      249    22          08
0250321582070 32     20     0      20     0      0      32     01          08
0250321671210 397    379    110    135    0      0      531    22          08
0250321782130 190    142    0      114    0      0      218    22          08
025032188232030      0      0      0      0      0      0                  08
025032198227830      0      0      0      0      0      0                  08
0250322082078 142    75     0      86     0      0      131    01          08
0393020171070 247    1056   0      303    746    0      254    33          08
039302027212030      0      0      0      0      0      0                  08
0393060172120360     0                                                     08
914103017971 30      21                                                    08
205021       2                                                             08
105283017222                                                               08
8090120154010 0      2831   0      2831                                   08
83114201     4                                                             08
83114301     4                                                             08
83114302     4                                                             08
83114303     4                                                             08
83114304     4                                                             08
83114305     4                                                             08
83114357     4                                                             08
83114601     4                                                             08
/*
//
```

SMIWTR62

접 수 부

내용 : 매월 잠정, 확정 작업하기 위해 전월
(확정) FILE에서 각 사업체별로
LISTING .

```
//SMIWTR62 JOB CLASS=6,TYPRUN=HOLD                                      SMI00010
//JOBCAT DD DSN=SMICT12,DISP=SHR                                        SMI00020
//STEP1 EXEC PGM=SMIM365                                                SMI00030
//STEPLIB DD DSN=USER.LOADLIB,DISP=SHR                                  SMI00040
//SYSPRINT DD SYSOUT=A                                                  SMI00050
//SAFL DD DSN=SMISA12,DISP=SHR                                          SMI00060
//GMFL DD DSN=SMIJN12,DISP=SHR                                          SMI00070
//G2FL DD UNIT=SYSDA,DSN=&&AAA,DISP=(NEW,PASS),VOL=SER=SORTWK,          SMI00080
//   DCB=(RECFM=FB,LRECL=57,BLKSIZE=5700),SPACE=(CYL,(15))              SMI00090
//SYSIN DD *                                                            SMI00100
06                                                                       SMI00110
/*                                                                      SMI00120
//STEP2 EXEC PGM=SMIM365                                                SMI00130
//STEPLIB DD DSN=USER.LOADLIB,DISP=SHR                                  SMI00140
//SYSPRINT DD SYSOUT=A                                                  SMI00150
//SAFL DD DSN=SMISA12,DISP=SHR                                          SMI00160
//GMFL DD DSN=SMIGM12,DISP=SHR                                          SMI00170
//G2FL DD UNIT=SYSDA,DSN=&&CCC,DISP=(NEW,PASS),VOL=SER=SORTWK,          SMI00180
//   DCB=(RECFM=FB,LRECL=57,BLKSIZE=5700),SPACE=(CYL,(15))              SMI00190
//SYSIN DD *                                                            SMI00200
07                                                                       SMI00210
/*                                                                      SMI00220
//STEP2 EXEC PGM=SORT,PARM='SIZE(MAX)'                                  SMI00230
//SYSPRINT DD SYSOUT=A                                                  SMI00240
//SYSOUT DD SYSOUT=A                                                    SMI00250
//SORTIN DD UNIT=SYSDA,DSN=&&AAA,DISP=(OLD,DELETE),VOL=SER=SORTWK       SMI00260
//       DD UNIT=SYSDA,DSN=&&CCC,DISP=(OLD,DELETE),VOL=SER=SORTWK       SMI00270
//SORTOUT DD UNIT=SYSDA,DSN=&&BBB,DISP=(NEW,PASS),VOL=SER=SORTWK,       SMI00280
//       SPACE=(CYL,(50)),                                              SMI00290
//       DCB=(RECFM=FB,LRECL=57,BLKSIZE=5700)                           SMI00300
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(200)),VOL=SER=SORTWK               SMI00310
//SYSIN DD *                                                            SMI00320
   SORT FIELDS=(51,3,CH,A,1,5,CH,A,9,5,CH,A,56,2,CH,A,6,1,CH,A)         SMI00330
/*                                                                      SMI00340
//STEP3 EXEC PLIFCLG                                                    SMI00350
//PLI.SYSIN DD *                                                        SMI00360
* PROCESS A,S,NEST,OPT(2);                                              SMI00370
 LCH:  PROC OPTIONS(MAIN);                                              SMI00380
       DCL GMFL FILE RECORD INPUT;                                      SMI00390
       DCL 1 GMREC,                                                     SMI00400
             2  SANO     CHAR(5),                                       SMI00410
             2  CARD     CHAR(1),                                       SMI00420
             2  F001     CHAR(2),                                       SMI00430
             2  PUM      CHAR(5),                                       SMI00440
             2  F002     CHAR(1),                                       SMI00450
             2  DATA     CHAR(36),                                      SMI00460
             2  SANUP    CHAR(5),                                       SMI00470
             2  MON      CHAR(2);                                       SMI00480
       DCL 1 CARD2 DEF DATA,                                            SMI00490
             2  RANG(7) FIXED BIN(31),                                  SMI00500
             2  F003    CHAR(8) ;                                       SMI00510
       DCL 1 CARD6 DEF DATA,                                            SMI00520
             2  NUNG(2) FIXED BIN(31),                                  SMI00530
             2  F004    CHAR(28);                                       SMI00540
       DCL BSAN CHAR(3) INIT('   ');                                    SMI00550
```

```
      DCL BPUM CHAR(5) INIT('     ');                                  SMI00560
      DCL BSA  CHAR(5) INIT('     ');                                  SMI00570
      DCL BMM  CHAR(2) INIT('  ');                                     SMI00580
      DCL SASU FIXED BIN(31) INIT(0);                                  SMI00590
      ON ENDFILE(GMFL) GO TO LAST;                                     SMI00600
      OPEN FILE(GMFL);                                                 SMI00610
      OPEN FILE(SYSPRINT) LINESIZE(132) PAGESIZE(66);                  SMI00620
RE1:  READ FILE(GMFL) INTO(GMREC);                                     SMI00630
      IF BSAN ¬= SUBSTR(SANUP,1,3) THEN CALL CHSANUP;                  SMI00640
      ELSE IF BSA ¬= SANO THEN CALL CHSANO;                            SMI00650
      ELSE IF BPUM ¬= PUM THEN CALL CHPUM;                             SMI00660
      ELSE DO;                                                         SMI00670
       IF CARD = '6'                                                   SMI00680
          THEN IF BMM ¬= MON                                           SMI00690
              THEN PUT SKIP(1) EDIT(MON,NUNG(1),NUNG(2))               SMI00700
                     (X(22),A(2),X(84),2 (P 'ZZZ,ZZZ,ZZ9'));           SMI00710
              ELSE PUT SKIP(0) EDIT(NUNG(1),NUNG(2))                   SMI00720
                     (X(108),2 (P 'ZZZ,ZZZ,ZZ9'));                     SMI00730
          ELSE DO; IF CARD¬='2' THEN GO TO RE1;                        SMI00740
              PUT SKIP(1) EDIT(MON,(RANG(I) DO I=1 TO 7))              SMI00750
                            (X(22),A(2),X(1),7 (P 'ZZZ,ZZZ,ZZ9'));    SMI00760
          END;                                                         SMI00770
      END;                                                             SMI00780
      BMM = MON;                                                       SMI00790
      GO TO RE1;                                                       SMI00800
/****************************                                          SMI00810
 * * SUBROUTINE CHSANUP   * *                                          SMI00820
 ****************************/                                         SMI00830
CHSANUP: PROC;                                                         SMI00840
      PUT PAGE;                                                        SMI00850
      BSAN=SUBSTR(SANUP,1,3);                                          SMI00860
      CALL CHSANO;                                                     SMI00870
      RETURN;                                                          SMI00880
      END CHSANUP;                                                     SMI00890
/****************************                                          SMI00900
 * * SUBROUTINE CHSANO   * *                                           SMI00910
 ****************************/                                         SMI00920
CHSANO: PROC;                                                          SMI00930
      SASU=SASU+1;                                                     SMI00940
      BSA = SANO;                                                      SMI00950
      BPUM = PUM;                                                      SMI00960
      BMM = MON;                                                       SMI00970
      IF CARD = '6' THEN PUT SKIP(3) EDIT                              SMI00980
                  (SANO,SANUP,PUM,MON,NUNG(1),NUNG(2))                 SMI00990
                (X(1),A(7),A(7),A(7),A(2),X(84),2 (P 'ZZZ,ZZZ,ZZ9'));  SMI01000
      ELSE DO; IF CARD ¬= '2' THEN RETURN;                             SMI01010
        PUT SKIP(3) EDIT(SANO,SANUP,PUM,MON,(RANG(I) DO I=1 TO 7))     SMI01020
                      (X(1),A(7),A(7),A(7),A(3),7 (P 'ZZZ,ZZZ,ZZ9'));  SMI01030
      END;                                                             SMI01040
      RETURN;                                                          SMI01050
      END CHSANO;                                                      SMI01060
/****************************                                          SMI01070
 * * SUBROUTINE CHPUM    * *                                           SMI01080
 ****************************/                                         SMI01090
CHPUM: PROC;                                                           SMI01100
```

```
      BPUM = PUM; BMM=MON;                                              SMI01110
      IF CARD = '6' THEN PUT SKIP(2) EDIT                               SMI01120
                                  (SANUP,PUM,MON,NUNG(1),NUNG(2))       SMI01130
                (X(8),A(7),A(7),A(2),X(84),2 (P 'ZZZ,ZZZ,ZZ9'));        SMI01140
      ELSE DO; IF CARD ¬= '2' THEN RETURN;                              SMI01150
        PUT SKIP(2) EDIT(SANUP,PUM,MON,(RANG(I) DO I=1 TO 7))           SMI01160
                      (X(8),A(7),A(7),A(3),7 (P 'ZZZ,ZZZ,ZZ9'));        SMI01170
      END;                                                              SMI01180
      RETURN;                                                           SMI01190
      END CHPUM;                                                        SMI01200
 LAST: CLOSE FILE(GMFL);                                                SMI01210
      PUT SKIP(2) EDIT('SA-UP-CHEOI NO =',SASU)(X(30),A,F(7));          SMI01220
      END LCH;                                                          SMI01230
//GO.GMFL DD UNIT=SYSDA,DSN=&&BBB,DISP=(OLD,DELETE),VOL=SER=SORTWK,     SMI01240
//       DCB=(RECFM=FB,LRECL=57,BLKSIZE=5700)                           SMI01250
//GO.SYSPRINT DD SYSOUT=A                                               SMI01260
//GO.SYSOUT DD SYSOUT=A                                                 SMI01270
//                                                                      SMI01280
```

SMIWMC62

FILE CHANGE
(RUN PROGRAM)

내용 : 금월(SMIGM12) FILE을 전월
(SMIJN12) FILE로 잠정월을 확정
월로 만들어 준다.

FILE: SMIWMC12 CHANGE D1 VM/SP RELEASE 2.1 EXPRESS PUT8302+ SLU203

```
//SMIWMC62 JOB TYPRUN=HOLD,CLASS=1                                    SMI00010
//JOBCAT DD DSN=SMICT12,DISP=SHR                                      SMI00020
//STEP1     EXEC  PGM=IDCAMS                                          DEF00020
//DD1       DD   VOL=SER=BOSWK2,UNIT=3350,DISP=OLD                    DEF00030
//SYSPRINT  DD   SYSOUT=A                                             DEF00040
//SYSIN     DD   *                                                    DEF00050
        DELETE -                                                      DEF00060
             SMIJN12 -                                                DEF00070
             FILE(DD1) -
             PURGE -
             CATALOG(SMICT12)                                         DEF00130
/*                                                                    DEF00140
//STEP2     EXEC  PGM=IDCAMS                                          DEF00020
//DD1       DD   VOL=SER=BOSWK2,UNIT=3350,DISP=OLD                    DEF00030
//SYSPRINT  DD   SYSOUT=A                                             DEF00040
//SYSIN     DD   *                                                    DEF00050
        DEFINE CLUSTER -                                              DEF00060
            (NAME(SMIJN12) -                                          DEF00070
             VOLUME(BOSWK2) -                                         DEF00080
             INDEXED -                                                DEF00090
             RECORDS(37000 1000) -                                    DEF00100
             KEYS(8 0) -                                              DEF00110
             RECORDSIZE(50 50)) -                                     DEF00120
             CATALOG(SMICT12)                                         DEF00130
/*
//ST3 EXEC PGM=SMIMC12                                                SMI00030
//STEPLIB DD DSN=USER.LOADLIB,DISP=SHR                                SMI00040
//SYSPRINT DD SYSOUT=A
//DIREC DD DSNAME=SMIJN12,DISP=SHR                                    SMI00240
//TA DD DSN=SMIGM12,DISP=SHR                                          SMI00250
//                                                                    SMI00270
```

SMIWMV62

SUM FILE MOVE
(RUN PROGRAM)

내용 : SUN FILE (37개월)에서 금월을 전월로 MOVE시켜 잠정치, 확정치 작업을 하기 위해 작업

FILE: SMIWMV66 FILEMOVE D1 VM/SP RELEASE 2.1 EXPRESS PUT8302+ SLU203

```
//SMIWMV62 JOB TYPRUN=HOLD,CLASS=1                                    SMI00010
//JOBCAT DD DSN=SMICT12,DISP=SHR                                      SMI00020
//  EXEC PGM=SMIMV12                                                  SMI00030
//STEPLIB DD DSN=USER.LOADLIB,DISP=SHR                                SMI00040
//SYSPRINT DD SYSOUT=A                                                SMI00050
//SUMF DD DSN=SMISM12,DISP=SHR                                        SMI00060
//                                                                    SMI00070
```

SMIWCV62

CONVERT
(RUN PROGRAM)

내용 : ERROR-LIST (EDITING)작업을 하기
위해 사업체번호와 품목번호를 비교하여
전월대비를 한다.

FILE: SMIWCV66 CONVERT D1 VM/SP RELEASE 2.1 EXPRESS PUT8302+ SLU203

```
//SMIWCV62 JOB CLASS=4,TYPRUN=HOLD                                  SMI00010
//JOBCAT DD DSN=SMICT12,DISP=SHR                                    SMI00020
//STEP1     EXEC  PGM=IDCAMS                                        DEF00020
//DD1       DD   VOL=SER=BOSWK2,UNIT=3350,DISP=OLD                  DEF00030
//SYSPRINT  DD   SYSOUT=A                                           DEF00040
//SYSIN     DD   *                                                  DEF00050
        DELETE -                                                    DEF00060
             SMICV12 -                                              DEF00070
             FILE(DD1) -
             PURGE -
             CATALOG(SMICT12)                                       DEF00130
/*                                                                  DEF00140
//STEP5     EXEC  PGM=IDCAMS                                        DEF00020
//DD1       DD   VOL=SER=BOSWK2,UNIT=3350,DISP=OLD                  DEF00030
//SYSPRINT  DD   SYSOUT=A                                           DEF00040
//SYSIN     DD   *                                                  DEF00050
        DEFINE CLUSTER -                                            DEF00060
            (NAME(SMICV12) -                                        DEF00070
             VOLUME(BOSWK2) -                                       DEF00080
             INDEXED -                                              DEF00090
             RECORDS(37000 1000) -                                  DEF00100
             KEYS(11 0) -                                           DEF00110
             RECORDSIZE(53 53)) -                                   DEF00120
             CATALOG(SMICT12)                                       DEF00130
/*
//  EXEC PGM=SMICV12                                                SMI00020
//STEPLIB DD DSN=USER.LOADLIB,DISP=SHR                              SMI00030
//SYSPRINT DD SYSOUT=A
//SYSOUT DD SYSOUT=A
//SORTWK01 DD UNIT=3350,SPACE=(CYL,50)
//EDIT2 DD DSN=SMICV12,DISP=SHR
//EDIT3 DD DSN=SMIJN12,DISP=SHR
//
```

SMIWCR62

CREATION
(RUN PROGRAM)

내용 : 잠정치 (금월) 조사표를 잠정 FILE (SMIGM 12) 에 DEFINE 시켜준다.

※ 잠정조사표는 작업을 원활히 하기 위해 조사표를 ½씩 나누어 EDITING을 하는데 1차분은 SMIGM12로 하고 2차분은 SMIG212로 작업을 한다.

FILE: SMIWCR66 CREATION D1 VM/SP RELEASE 2.1 EXPRESS PUT8302+ SLU203

```
//SMIWCR62 JOB CLASS=1,TYPRUN=HOLD                                      SMI00010
//JOBCAT DD DSN=SMICT12,DISP=SHR
//STEP1     EXEC  PGM=IDCAMS                                            DEF00020
//DD1       DD   VOL=SER=BOSWK2,UNIT=3350,DISP=OLD                      DEF00030
//SYSPRINT  DD   SYSOUT=A                                               DEF00040
//SYSIN     DD   *                                                      DEF00050
        DELETE -                                                        DEF00060
             SMIG212 -                                                  DEF0007
             FILE(DD1) -
             PURGE -
             CATALOG(SMICT12)                                           DEF00130
/*                                                                      DEF00140
//DD1       DD   VOL=SER=BOSWK2,UNIT=3350,DISP=OLD                      DEF00030
//SYSPRINT  DD   SYSOUT=A                                               DEF00040
//SYSIN     DD   *                                                      DEF00050
        DELETE -                                                        DEF00060
             SMIG212 -                                                  DEF0007
             FILE(DD1) -
             PURGE -
             CATALOG(SMICT12)                                           DEF00130
/*                                                                      DEF00140
//STEP2     EXEC  PGM=IDCAMS                                            DEF00020
//DD1       DD   VOL=SER=BOSWK2,UNIT=3350,DISP=OLD                      DEF00030
//SYSPRINT  DD   SYSOUT=A                                               DEF00040
//SYSIN     DD   *                                                      DEF00050
        DEFINE CLUSTER -                                                DEF00060
            (NAME(SMIG212) -                                            DEF0007
             VOLUME(BOSWK2) -                                           DEF00080
             INDEXED -                                                  DEF00090
             RECORDS(38000 1000) -                                      DEF00100
             FREESPACE(40 40) -
             KEYS(8 0) -                                                DEF00110
             RECORDSIZE(50 50)) -                                       DEF00120
             CATALOG(SMICT12)                                           DEF00130
/*
//  EXEC PGM=SMICR12                                                    SMI00020
//STEPLIB DD DSN=USER.LOADLIB,DISP=SHR                                  SMI00030
//SYSPRINT DD SYSOUT=A
//SYSOUT DD SYSOUT=A
//SORTWK01 DD UNIT=DISK,SPACE=(CYL,50)
//TAIN DD UNIT=TAPE,DSN=TAPI,VOL=SER=REEL01,LABEL=(,NL),
//  DCB=(RECFM=FB,LRECL=80),DISP=OLD
//EDIT3 DD DSN=SMIG212,DISP=SHR
//SYSIN DD *
08
//
```

SMIWED62

EDITING
(RUN PROGRAM)

FILE: SMIWED66 EDITING D1 VM/SP RELEASE 2.1 EXPRESS PUT8302+ SLU203

```
//SMIWED62 JOB CLASS=1,TYPRUN=HOLD,TIME=1440                            SMI00010
//JOBCAT DD DSN=SMICT12,DISP=SHR
//STEP1 EXEC PGM=SMIMH65                                                SMK00020
//STEPLIB DD DSN=USER.LOADLIB,DISP=SHR                                  SMK00030
//SYSPRINT DD SYSOUT=A
//SAFL DD DSN=SMISA12,DISP=SHR
//GMFL DD DSN=SMIGM12,DISP=SHR                                          SMI04780
//G2FL DD UNIT=SYSDA,DSN=&&AAA,DISP=(NEW,PASS),VOL=SER=SORTWK,
//   DCB=(RECFM=FB,LRECL=53,BLKSIZE=5300),SPACE=(CYL,(50))
/*
//STEP2 EXEC PGM=SORT,PARM='SIZE(MAX)'
//SYSPRINT DD SYSOUT=A
//SYSOUT DD SYSOUT=A
//SORTIN DD UNIT=SYSDA,DSN=&&AAA,DISP=(OLD,DELETE),VOL=SER=SORTWK
//SORTOUT DD UNIT=SYSDA,DSN=&&BBB,DISP=(NEW,PASS),VOL=SER=SORTWK,
//       SPACE=(CYL,(50)),
//       DCB=(RECFM=FB,LRECL=53,BLKSIZE=5300)
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(200)),VOL=SER=SORTWK
//SYSIN DD *
   SORT FIELDS=(51,3,CH,A,1,8,CH,A)
/*
//STEP3 EXEC PGM=IEHPROGM
//SYSPRINT DD SYSOUT=A
//DD2     DD UNIT=DISK,VOL=SER=BOSWK2,DISP=OLD
//SYSIN   DD *
    SCRATCH DSNAME=SMIOK12,VOL=DISK=BOSWK2
/*
//STEP4 EXEC PGM=SMIED65                                                SMK00020
//STEPLIB DD DSN=USER.LOADLIB,DISP=SHR                                  SMK00030
//SYSPRINT DD SYSOUT=A
//PUMF DD DSN=SMIPM12,DISP=SHR
//EDIT3 DD UNIT=SYSDA,DSN=&&BBB,DISP=(OLD,DELETE),VOL=SER=SORTWK        SMI04780
//EDIT2 DD DSN=SMICV12,DISP=SHR                                         SMI04780
//TOUT DD UNIT=DISK,VOL=SER=BOSWK2,DSN=SMIOK12,DISP=(NEW,KEEP),
//   DCB=(RECFM=FB,LRECL=10,BLKSIZE=3000),SPACE=(10,(800,100))
//
```

SMIWOS62

OK SELECT
(RUN PROGRAM)

FILE: SMIWOS66 OKSELECT D1 VM/SP RELEASE 2.1 EXPRESS PUT8302+ SLU203

```
//SMIWOS62 JOB CLASS=V,TYPRUN=HOLD                                    SMI00010
//JOBCAT DD DSN=SMICT12,DISP=SHR                                      SMI00020
//   EXEC PGM=SMIOS12                                                 SMI00030
//STEPLIB DD DSN=USER.LOADLIB,DISP=SHR                                SMI00040
//SYSPRINT DD SYSOUT=A                                                SMI00050
//TAIN DD UNIT=DISK,VOL=SER=BOSWK2,DISP=OLD,DSN=SMIOK12,              SMI00060
//      DCB=(RECFM=FB,LRECL=10,BLKSIZE=3000)                          SMI00070
//EDIT3 DD DSN=SMIGM12,DISP=SHR                                       SMI00080
//                                                                    SMI00090
```

SMIWUP62

UPDATE
(RUN PROGRAM)

내용 : EDITING 작업후 조사표 수정분으로
잠정치 FILE(SMIGM12)에 UPDATE.

FILE: SMIWUP66 UPDATE D1 VM/SP RELEASE 2.1 EXPRESS PUT8302+ SLU203

```
//SMIWUP62 JOB CLASS=B,TYPRUN=HOLD                                  SMI00010
//JOBCAT DD DSN=SMICT12,DISP=SHR                                    SMI00020
//  EXEC PGM=SMIUP12                                                SMI00030
//STEPLIB DD DSN=USER.LOADLIB,DISP=SHR                              SMI00040
//SYSPRINT DD SYSOUT=A                                              SMI03440
//SYSOUT DD SYSOUT=A                                                SMI03440
//SORTWK01 DD UNIT=3350,SPACE=(CYL,100)                             SMI03450
//TAIN DD UNIT=TAPE,DSN=ABC,VOL=SER=123322,LABEL=(,NL),             SMI03460
//    DCB=(RECFM=F,BLKSIZE=80),DISP=OLD                             SMI03470
//EDIT3 DD DSN=SMIGM12,DISP=SHR                                     SMI03480
//SYSIN DD *                                                        SMI03490
08                                                                  SMI03500
/*                                                                  SMI03510
//                                                                  SMI03520
```

SMIWOE66

OK. ERROR 구분
(RUN PROGRAM)

내용 : EDITING 작업을 바르게 하기 위하여
ERROR 부분만 EDITING 하기 위한
준비작업

FILE: SMIWOE66 OKERROR D1 VM/SP RELEASE 2.1 EXPRESS PUT8302+ SLU203

```
//SMIWOE66 JOB CLASS=1,TYPRUN=HOLD
//JOBCAT DD DSN=SMICT12,DISP=SHR
//   EXEC PGM=SMIOE12
//STEPLIB DD DSN=USER.LOADLIB,DISP=SHR
//SYSPRINT DD SYSOUT=A
//EDIT3 DD DSN=SMIGM12,DISP=SHR
//SYSIN DD *
02204
04325
10827
03213
05324
/*
//
```

SMIWFP62

FILE MERGE
(RUN PROGRAM)

내용 : 조사표 1, 2차 접수완료분을 SMIG212,
SMIGM12로 나누어 작업하였던 것을
SMIGM12로 MERGE하여 합하는 작업

```
//SMIWFP62 JOB CLASS=1,TYPRUN=HOLD                                    SMI00010
//JOBCAT DD DSN=SMICT12,DISP=SHR                                      SMI00020
//ST3 EXEC PGM=SMIFP12                                                SMI00030
//STEPLIB DD DSN=USER.LOADLIB,DISP=SHR                                SMI00040
//SYSPRINT DD SYSOUT=A
//DIREC DD DSNAME=SMIG212,DISP=SHR                                    SMI00240
//TA    DD DSNAME=SMIGM12,DISP=SHR                                    SMI00240
//                                                                    SMI00270
```

SMIWCH62

CARD CHECK
(RUN PROGRAM)

내용 : 금월 FILE(SMIGM12)과 CONVERT
(8번 STEP)에서 생성된 FILE에서
ERROR CHECK.

FILE: SMIWCH66 CCHECK D1 VM/SP RELEASE 2.1 EXPRESS PUT8302+ SLU203

```
//SMIWCH62 JOB CLASS=B                                                  SMI00010
//JOBCAT DD DSN=SMICT12,DISP=SHR                                        SMI00020
//   EXEC PGM=SMICH12                                                   SMI02780
//STEPLIB DD DSN=USER.LOADLIB,DISP=SHR
//SYSPRINT DD SYSOUT=A
//NEW   DD DSN=SMIGM12,DISP=SHR                                         SMI02790
//OLD   DD DSN=SMICV12,DISP=SHR                                         SMI02800
//                                                                      SMI02810
```

SMIWEE62

MASTER CHECK

내용 : 사업체 MASTER에서 ERROR CHECK.

```
//SMIWEE62 JOB  CLASS=V                                                 MAS00010
//JOBCAT DD DSN=SMICT12,DISP=SHR                                        MAS00020
//    EXEC  PLIFCLG                                                     MAS00030
//PLI.SYSIN DD *                                                        MAS00040
* PROCESS GS,NEST,A,XREF,OPT(2);                                        MAS00050
  MSAT: PROC OPTIONS(MAIN);                                             MAS00060
   DCL JOSA FILE RECORD KEYED ENV(VSAM);                                MAS00070
   DCL MAST FILE RECORD KEYED ENV(VSAM);                                MAS00080
   DCL 1 INREC,                                                         MAS00090
        2 SA   CHAR(8),                                                 MAS00100
        2 FA   CHAR(42);                                                MAS00110
   DCL 1 MREC,                                                          MAS00120
        2 MA   CHAR(5),                                                 MAS00130
        2 FB   CHAR(66),                                                MAS00140
        2 SET  CHAR(1),                                                 MAS00150
        2 FC   CHAR(8);                                                 MAS00160
   DCL (ON,SW)  PIC'9' INIT(0);                                         MAS00170
   DCL ONCODE    BUILTIN;                                               MAS00180
   DCL M      FIXED   INIT(0);                                          MAS00190
   DCL WW(100) CHAR(8);                                                 MAS00200
   DCL WA  CHAR(5)  DEF SA;                                             MAS00210
   DCL (PK1,PK2)    FIXED  INIT(0);                                     MAS00220
   ON KEY(MAST) BEGIN;                                                  MAS00230
     IF ONCODE=51 THEN ON=1;                                            MAS00240
   END;                                                                 MAS00250
   ON ENDFILE(JOSA) GOTO LT;                                            MAS00260
   ON ENDFILE(MAST) GOTO QT;                                            MAS00270
   OPEN FILE(MAST) SEQUENTIAL UPDATE;                                   MAS00280
    RM: READ FILE(MAST) INTO(MREC);                                     MAS00290
       SET='0';                                                         MAS00300
      REWRITE FILE(MAST) FROM(MREC);                                    MAS00310
      GOTO RM;                                                          MAS00320
    QT: IF SW=1 THEN GOTO QQ;                                           MAS00330
    CLOSE FILE(MAST);                                                   MAS00340
   OPEN FILE(JOSA) SEQUENTIAL INPUT;                                    MAS00350
   OPEN FILE(MAST) DIRECT UPDATE;                                       MAS00360
   RT: READ FILE(JOSA) INTO(INREC);                                     MAS00370
   IF SW=0 THEN DO;                                                     MAS00380
     WW=' '; M=1; SW=1; WW(M)=SA; GOTO RT;                              MAS00390
   END;                                                                 MAS00400
   IF SUBSTR(WW(1),1,5)=WA THEN DO;                                     MAS00410
     M=M+1; WW(M)=SA;  GOTO RT;                                         MAS00420
   END;                                                                 MAS00430
   MA=SUBSTR(WW(1),1,5);                                                MAS00440
   READ FILE(MAST) INTO(MREC) KEY(MA);                                  MAS00450
   IF ON=1 THEN DO;                                                     MAS00460
     ON=0;                                                              MAS00500
     PUT SKIP EDIT(MA)(A(5));                                           MAS00510
   END;                                                                 MAS00520
   ELSE DO;                                                             MAS00530
     PK1=PK1+1; SET='1';                                                MAS00540
     REWRITE FILE(MAST) FROM(MREC) KEY(MA);                             MAS00550
   END;                                                                 MAS00560
   WW=' '; M=1; WW(M)=SA;                                               MAS00570
   GOTO RT;                                                             MAS00580
```

```
  LT:                                                                   MAS00590
   MA=SUBSTR(WW(1),1,5);                                                MAS00440
   READ FILE(MAST) INTO(MREC) KEY(MA);                                  MAS00450
   IF ON=1 THEN DO;                                                     MAS00460
     ON=0;                                                              MAS00500
     PUT SKIP EDIT(MA)(A(5));                                           MAS00510
   END;                                                                 MAS00520
   ELSE DO;                                                             MAS00530
     PK1=PK1+1; SET='1';                                                MAS00540
     REWRITE FILE(MAST) FROM(MREC) KEY(MA);                             MAS00550
   END;                                                                 MAS00560
   WW=' '; M=1; WW(M)=SA;                                               MAS00570
      CLOSE FILE(JOSA), FILE(MAST);                                     MAS00590
   OPEN FILE(MAST) SEQUENTIAL INPUT;                                    MAS00600
  RS: READ FILE(MAST) INTO(MREC);                                       MAS00610
   IF SET='0' THEN PUT SKIP EDIT(MA)(X(10),A(5));                       MAS00620
   PK2=PK2+1;                                                           MAS00630
   GOTO RS;                                                             MAS00640
  QQ: CLOSE FILE(MAST);                                                 MAS00650
  PUT SKIP(3) EDIT('JOSAPYO= ',PK1,'MASTER= ',PK2)(2 (A,F(5)));         MAS00660
 END MSAT;                                                              MAS00670
/*                                                                      MAS00680
//GO.JOSA DD DSN=SMIGM12,DISP=SHR                                       MAS00690
//GO.MAST DD DSN=SMISA12,DISP=SHR                                       MAS00700
//    EXEC  PLIFCLG                                                     BB 00030
//PLI.SYSIN DD *                                                        BB 00040
* PROCESS GS,NEST,A,XREF,OPT(2);                                        BB 00050
  MSAT: PROC OPTIONS(MAIN);                                             BB 00060
   DCL JOSA FILE RECORD KEYED ENV(VSAM);                                BB 00070
   DCL MAST FILE RECORD KEYED ENV(VSAM);                                BB 00080
   DCL 1 INREC,                                                         BB 00090
        2 SA   CHAR(8),                                                 BB 00100
        2 FA   CHAR(42);                                                BB 00110
   DCL 1 MREC,                                                          BB 00120
        2 MA   CHAR(5),                                                 BB 00130
        2 FB   CHAR(24),                                                BB 00140
        2 SET  CHAR(1),                                                 BB 00150
        2 FC   CHAR(50);                                                BB 00160
   DCL (ON,SW)  PIC'9' INIT(0);                                         BB 00170
   DCL ONCODE    BUILTIN;                                               BB 00180
   DCL M      FIXED   INIT(0);                                          BB 00190
   DCL WW      CHAR(5);                                                 BB 00200
   DCL WA   CHAR(5)  DEF SA;                                            BB 00210
   DCL CD   CHAR(1)  DEF SA POS(6);                                     BB 00220
   DCL (PK1,PK2)    FIXED  INIT(0);                                     BB 00230
   DCL (S2,S6)    CHAR(1) INIT('0');                                    BB 00240
   ON KEY(MAST) BEGIN;                                                  BB 00250
     IF ONCODE=51 THEN ON=1;                                            BB 00260
   END;                                                                 BB 00270
   ON ENDFILE(JOSA) GOTO LT;
   OPEN FILE(JOSA) SEQUENTIAL INPUT;                                    BB 00280
   OPEN FILE(MAST) DIRECT INPUT;                                        BB 00290
   RT: READ FILE(JOSA) INTO(INREC);                                     BB 00300
   IF SW=0 THEN DO;                                                     BB 00310
     WW=' '; M=1; SW=1; WW=WA;    GOTO RT;                              BB 00320
```

```
   END;                                                                 BB 00330
   IF WW=WA THEN DO;                                                    BB 00340
     IF CD='6' THEN S6='1'; ELSE IF CD='2' THEN S2='1';                 BB 00350
     GOTO RT;                                                           BB 00360
   END;                                                                 BB 00370
   MA=WW;                                                               BB 00380
   READ FILE(MAST) INTO(MREC) KEY(MA);                                  BB 00390
   IF ON=1 THEN DO;                                                     BB 00400
     ON=0;                                                              BB 00410
     PUT SKIP EDIT(MA)(A(5));                                           BB 00420
   END;                                                                 BB 00430
   ELSE DO;                                                             BB 00440
     IF SET='1' THEN DO;                                                BB 00450
       IF S2='0' THEN PUT SKIP EDIT('D2',MA)(A(3),A(5));                BB 00460
       IF S6='0' THEN PUT SKIP EDIT('D6',MA)(A(3),A(5));                BB 00470
     END;                                                               BB 00480
     IF SET='2' THEN DO;                                                BB 00490
       IF S2='1' THEN PUT SKIP EDIT('C2',MA)(A(3),A(5));                BB 00500
       IF S6='0' THEN PUT SKIP EDIT('C6',MA)(A(3),A(5));                BB 00510
     END;                                                               BB 00520
     IF SET=' ' THEN DO;                                                BB 00530
       IF S2='0' THEN PUT SKIP EDIT('P2',MA)(A(3),A(5));                BB 00540
       IF S6='1' THEN PUT SKIP EDIT('P6',MA)(A(3),A(5));                BB 00550
     END;                                                               BB 00560
   END;   S2='0'; S6='0';                                               BB 00570
   WW=' '; M=1; WW=WA;                                                  BB 00580
   GOTO RT;                                                             BB 00590
  LT: CLOSE FILE(JOSA), FILE(MAST);                                     BB 00600
 END MSAT;                                                              BB 00610
/*                                                                      BB 00620
//GO.JOSA DD DSN=SMIGM12,DISP=SHR                                       BB 00630
//GO.MAST DD DSN=SMISA12,DISP=SHR                                       BB 00640
//                                                                      BB 00650
```

SMIWJS62

승수산업삽입 (RUN PROGRAM)

FILE: SMIWJS66 SEONGSU B1 VM/SP RELEASE 2.1 EXPRESS PUT8302+ SLU203

```
//SMIWJS62 JOB CLASS=B,TYPRUN=HOLD                                  SMI00010
//JOBCAT DD DSN=SMICT12,DISP=SHR                                    SMI00020
//   EXEC PGM=SMIJS12                                               SMI00030
//STEPLIB DD DSN=USER.LOADLIB,DISP=SHR                              SMI00040
//SYSPRINT DD SYSOUT=A                                              SMI00050
//EDIT3 DD DSN=SMIGM12,DISP=SHR                                     SMI00060
//SMAST DD DSN=SMISA12,DISP=SHR                                     SMI00070
//                                                                  SMI00080
```

SMIWS662

SUMMARY (물량수정)
RUN PROGRAM

※ 매월 작업시 지수와 CARD 천공하여야 함.

FILE: SMIWS166 SUMMARY D1 VM/SP RELEASE 2.1 EXPRESS PUT8302+ SLU203

```
//SMIWS662 JOB CLASS=B,TYPRUN=HOLD                                      SMI00010
//JOBCAT DD DSN=SMICT12,DISP=SHR
//  EXEC PGM=SMIWN66                                                    SMI00020
//STEPLIB DD DSN=USER.LOADLIB,DISP=SHR                                  SMI00030
//SYSOUT DD SYSOUT=A
//SYSPRINT DD SYSOUT=A
//SORTWK01 DD UNIT=3350,SPACE=(CYL,10)
//SUMF  DD DSN=SMISM12,DISP=SHR
//EDIT2 DD DSN=SMIJN12,DISP=SHR
//EDIT3 DD DSN=SMIGM12,DISP=SHR
//SYSIN DD *
0863 1078 1377 1161 0967 1148 1078 1377 1161 0967 1144
/*
//                                                                      SMI06770
```

SMIWIX62

지수시산

(RUN PROGRAM)

내용 : 물량수정후 지수를 산출

```
//SMIWIX62 JOB CLASS=A,TYPRUN=HOLD                                      SMI00010
//JOBCAT DD DSN=SMICT12,DISP=SHR                                        SMI00020
//   EXEC PGM=SMIIX12                                                   SMI00030
//STEPLIB DD DSN=USER.LOADLIB,DISP=SHR                                  SMI00040
//SYSPRINT DD SYSOUT=A                                                  SMI00050
//SUMM DD DSN=SMISM12,DISP=SHR                                          SMI00060
//   EXEC PLIFCLG                                                       SMI00070
//PLI.SYSIN DD *                                                        SMI00080
        /*****************************************************/         SMI00090
        /*****                                            *****/        SMI00100
        /*****    JISU GYESAN (2MONTH) PROGRAM  -ORIGINAL-  *****/      SMI00110
        /*****                                            *****/        SMI00120
        /*****************************************************/         SMI00130
* PROCESS GS,NEST,A,XREF,OPT(2);                                        SMI00140
  JISURTN: PROC OPTIONS(MAIN);                                          SMI00150
 DCL SUMM FILE RECORD KEYED ENV(VSAM);                                  SMI00160
 DCL MREC   CHAR(320);                                                  SMI00170
 DCL OKEY CHAR(10) DEF MREC;                                            SMI00180
 DCL GU   CHAR(1) DEF MREC POS(2);                                      SMI00190
 DCL AREC(1500)       CHAR(320);                                        SMI00200
  ON KEY(SUMM) BEGIN;                                                   SMI00210
   IF ONCODE=52 THEN PUT EDIT(OKEY,'DUPLICATE')(A(10),A(10));           SMI00220
  END;                                                                  SMI00230
  ON ENDFILE(SUMM) GOTO LT;                                             SMI00240
  OPEN FILE(SUMM) SEQUENTIAL INPUT;  K=0;                               SMI00250
  ARR: READ FILE(SUMM) INTO(MREC);                                      SMI00260
   IF GU='A' THEN GU='P';                                               SMI00270
   ELSE IF GU='E' THEN GU='Q';                                          SMI00280
   ELSE IF GU='H' THEN GU='R';                                          SMI00290
   ELSE GOTO ARR;                                                       SMI00300
   IF SUBSTR(OKEY,3,1)='0' | SUBSTR(OKEY,3,3)='300' | SUBSTR(OKEY,3,3)  SMI00310
    ='380' | SUBSTR(OKEY,3,4)='3840' THEN GOTO ARR;                     SMI00320
    IF SUBSTR(OKEY,6,4)>'8401' & SUBSTR(OKEY,6,4)<'8409' THEN GOTO ARR; SMI00330
   K=K+1;                                                               SMI00340
   AREC(K)=MREC;                                                        SMI00350
   GOTO ARR;                                                            SMI00360
  LT: CLOSE FILE(SUMM); OPEN FILE(SUMM) DIRECT UPDATE;                  SMI00370
    DO I=1 TO K;                                                        SMI00380
     MREC=AREC(I);                                                      SMI00390
     REWRITE FILE(SUMM) FROM(MREC) KEY(OKEY);                           SMI00400
   EEX:  END;                                                           SMI00410
   CLOSE FILE(SUMM);                                                    SMI00420
 END JISURTN;                                                           SMI00430
/*                                                                      SMI00440
//GO.SUMM DD DSN=SMISM12,DISP=SHR                                       SMI00450
//        EXEC PGM=IEHPROGM                                             SMI00460
//SYSPRINT DD SYSOUT=A                                                  SMI00470
//DD2     DD UNIT=DISK,VOL=SER=BOSWK2,DISP=OLD                          SMI00480
//SYSIN   DD *                                                          SMI00490
    SCRATCH DSNAME=SMITB12,VOL=DISK=BOSWK2                              SMI00500
/*                                                                      SMI00510
//ST3 EXEC PGM=SMIPP12                                                  SMI00520
//STEPLIB DD DSN=USER.LOADLIB,DISP=SHR                                  SMI00530
//DIREC DD DSN=SMISM12,DISP=SHR                                         SMI00540
//SYSPRINT DD SYSOUT=A                                                  SMI00550
```

```
//TA DD UNIT=DISK,VOL=SER=BOSWK2,DSN=SMITB12,DISP=(NEW,KEEP),          SMI00560
//   DCB=(RECFM=FB,LRECL=320,BLKSIZE=3200),SPACE=(320,(5500,100))      SMI00570
//                                                                     SMI00580
```

SMIWT162

TABLE (RUN)
동태지수(월간)

FILE: SMIWT166 DONGTAE D1 VM/SP RELEASE 2.1 EXPRESS PUT8302+ SLU203

```
//SMIWT162 JOB CLASS=B,TYPRUN=HOLD                                      SMI00010
//JOBCAT DD DSN=SMICT12,DISP=SHR                                        SMI00020
//STEP   EXEC PGM=IEHPROGM
//SYSPRINT   DD SYSOUT=A
//DD2    DD  UNIT=DISK,VOL=SER=BOSWK2,DISP=OLD
//SYSIN  DD  *
         SCRATCH  DSNAME=SMISS32,VOL=DISK=BOSWK2
/*
//STEP1    EXEC  PGM=SMI1132                                            SMI00030
//STEPLIB DD DISP=SHR,DSN=USER.LOADLIB
//SYSPRINT  DD   SYSOUT=A,COPIES=1                                      SMI01900
//DIN   DD  DSN=SMISM12,DISP=SHR                                        SMI01870
//DUT DD  DSN=SMISS32,UNIT=3350,VOL=SER=BOSWK2,SPACE=(70,(500,90)),     SMI01880
//             DCB=(RECFM=FB,BLKSIZE=2100,LRECL=70),DISP=(NEW,KEEP)     SMI01890
//SYSOUT    DD   SYSOUT=A                                               SMI01910
/*                                                                      SMI01920
//STEP2    EXEC     PGM=SMI1232                                        NNN00030
//STEPLIB DD DISP=SHR,DSN=USER.LOADLIB
//SYSPRINT  DD   SYSOUT=A,COPIES=1                                      NNN04670
//DIN   DD  DSN=SMISM12,DISP=SHR                                        NNN04680
//SYSOUT    DD   SYSOUT=A                                               NNN04690
/*                                                                      NNN04700
//STEP3    EXEC SORT                                                    SY300020
//SORTIN DD UNIT=DISK,DSN=SMISS32,DISP=OLD,                             SY300030
//       DCB=(RECFM=FB,BLKSIZE=2100,LRECL=70),VOL=SER=BOSWK2            SY300040
//SORTOUT DD UNIT=DISK,DSN=SMISS32,DISP=(OLD,KEEP),                     SY300050
//       DCB=(RECFM=FB,BLKSIZE=2100,LRECL=70),VOL=SER=BOSWK2            SY300060
//SORTWK01 DD UNIT=DISK,SPACE=(CYL,10)                                  SY300080
//SYSIN DD *                                                            SY300090
 SORT FIELDS=(1,4,CH,A)                                                 SY300100
 RECORD TYPE=F,LENGTH=(70)                                              SY300110
 END                                                                    SY300120
/*                                                                      SY300130
//STEP4    EXEC     PGM=SMI1332                                        SMI00030
//STEPLIB DD DISP=SHR,DSN=USER.LOADLIB
//SYSPRINT  DD   SYSOUT=A,COPIES=1                                      SMI01900
//DIN DD  DSN=SMISS32,UNIT=3350,VOL=SER=BOSWK2,                         SMI01880
//         DCB=(RECFM=FB,BLKSIZE=2100,LRECL=70),DISP=(OLD,KEEP)         SMI01890
//SYSOUT    DD   SYSOUT=A                                               SMI01910
/*                                                                      SMI01920
//STEP5    EXEC     PGM=SMI1432                                        SMI00030
//STEPLIB DD DISP=SHR,DSN=USER.LOADLIB
//SYSPRINT  DD   SYSOUT=A,COPIES=1                                      SMI01900
//DIN DD  DSN=SMISS32,UNIT=3350,VOL=SER=BOSWK2,                         SMI01880
//         DCB=(RECFM=FB,BLKSIZE=2100,LRECL=70),DISP=(OLD,KEEP)         SMI01890
//SYSOUT    DD   SYSOUT=A                                               SMI01910
/*                                                                      SMI01920
//                                                                      SMI01930
```

SMIWGY62

고용 TABLE RUN PROGRAM

고용임금지수 산출

노동생산성지수 산출

```
//SMIWGY62 JOB CLASS=B,TYPRUN=HOLD                                      SMI00010
//JOBCAT DD DSN=SMICT12,DISP=SHR                                        SMI00020
//STEP1   EXEC PGM=IEHPROGM                                             SMI00030
//SYSPRINT DD SYSOUT=A                                                  SMI00040
//DD2     DD UNIT=DISK,VOL=SER=BOSWK2,DISP=OLD                          SMI00050
//SYSIN   DD *                                                          SMI00060
    SCRATCH DSNAME=SMICD235,VOL=DISK=BOSWK2                             SMI00080
    SCRATCH DSNAME=SMICD435,VOL=DISK=BOSWK2                             SMI00080
    SCRATCH DSNAME=SMICD35,VOL=DISK=BOSWK2                              SMI00080
/*                                                                      SMI00100
//STEP2 EXEC PGM=SMISA35                                                SMI00430
//STEPLIB DD DSN=USER.LOADLIB,DISP=SHR                                  SMI00440
//SYSPRINT DD SYSOUT=A                                                  SMI00450
//DIN  DD DSN=SMISM12,DISP=SHR                                          SMI00780
//DOUT DD UNIT=DISK,DSN=SMICD435,DCB=(RECFM=FB,BLKSIZE=3200,            SMI00790
//        LRECL=320),DISP=(NEW,KEEP),VOL=SER=BOSWK2,                    SMI00800
//        SPACE=(320,(50,10))                                           SMI00810
//STEP3 EXEC PGM=SMISB35                                                SMI00430
//STEPLIB DD DSN=USER.LOADLIB,DISP=SHR                                  SMI00440
//SYSPRINT DD SYSOUT=A                                                  SMI00450
//DIN  DD DSN=SMISM12,DISP=SHR                                          SMI00780
//DOUT DD UNIT=DISK,DSN=SMICD235,DCB=(RECFM=FB,BLKSIZE=3200,            SMI00790
//        LRECL=320),DISP=(NEW,KEEP),VOL=SER=BOSWK2,                    SMI00800
//        SPACE=(320,(50,10))                                           SMI00810
//STEP4   EXEC PGM=SORT,PARM='SIZE(MAX)'                                SMI00860
//SYSPRINT DD SYSOUT=A                                                  SMI00870
//SYSOUT   DD SYSOUT=A                                                  SMI00880
//SORTIN  DD UNIT=DISK,DISP=(OLD,KEEP),DSN=SMICD235,                    SMI00890
//     VOL=SER=BOSWK2,DCB=(RECFM=FB,LRECL=320,BLKSIZE=3200)             SMI00900
//SORTOUT DD UNIT=DISK,DISP=(OLD,KEEP),DSN=SMICD235,                    SMI00910
//    DCB=(RECFM=FB,LRECL=320,BLKSIZE=3200),VOL=SER=BOSWK2              SMI00920
//SORTWK01 DD UNIT=DISK,SPACE=(CYL,10)                                  SMI00930
//SYSIN   DD *                                                          SMI00940
 SORT FIELDS=(1,5,D,6,2,A),FORMAT=CH                                    SMI00950
/*                                                                      SMI00850
//STEP5   EXEC PGM=SORT,PARM='SIZE(MAX)'                                SMI00860
//SYSPRINT DD SYSOUT=A                                                  SMI00870
//SYSOUT   DD SYSOUT=A                                                  SMI00880
//SORTIN  DD UNIT=DISK,DISP=(OLD,KEEP),DSN=SMICD435,                    SMI00890
//     VOL=SER=BOSWK2,DCB=(RECFM=FB,LRECL=320,BLKSIZE=3200)             SMI00900
//SORTOUT DD UNIT=DISK,DISP=(OLD,KEEP),DSN=SMICD435,                    SMI00910
//    DCB=(RECFM=FB,LRECL=320,BLKSIZE=3200),VOL=SER=BOSWK2              SMI00920
//SORTWK01 DD UNIT=DISK,SPACE=(CYL,10)                                  SMI00930
//SYSIN   DD *                                                          SMI00940
 SORT FIELDS=(1,5,D,6,2,A),FORMAT=CH                                    SMI00950
/*                                                                      SMI00220
//STEP6 EXEC PGM=SMISC35
//STEPLIB DD DSN=USER.LOADLIB,DISP=SHR
//SYSPRINT DD SYSOUT=A                                                  23500010
//DIN DD UNIT=DISK,DSN=SMICD435,DCB=(RECFM=FB,BLKSIZE=3200,             SMI01000
//        LRECL=320),VOL=SER=BOSWK2,DISP=OLD                            SMI01010
//DOUT DD UNIT=DISK,DSN=SMICD35,DCB=(RECFM=FB,BLKSIZE=3200,             SMI01020
//        LRECL=320),DISP=(NEW,KEEP),VOL=SER=BOSWK2,                    SMI01030
//        SPACE=(320,(1000,100))                                        SMI01040
//STEP1 EXEC PGM=SMIMA35                                                XXX00080
```

```
//STEPLIB DD DSN=USER.LOADLIB,DISP=SHR                                XXX00090
//SYSPRINT DD SYSOUT=A                                                SM300010
//SYSIN DD *
9136.90137.6
/*
//DIN DD UNIT=DISK,DSN=SMICD35,DCB=(RECFM=FB,BLKSIZE=3200,            XXX05450
//        LRECL=320),VOL=SER=BOSWK2,DISP=(OLD,KEEP)                   XXX05460
//DIN2 DD UNIT=DISK,DSN=SMICD235,DCB=(RECFM=FB,BLKSIZE=3200,          XXX05470
//        LRECL=320),VOL=SER=BOSWK2,DISP=(OLD,KEEP)                   XXX05480
//SUM DD DSN=SMISM12,DISP=SHR                                         XXX05490
//STEP2   EXEC PGM=SORT,PARM='SIZE(MAX)'                              XXX05670
//SYSPRINT DD SYSOUT=A                                                XXX05680
//SYSOUT   DD SYSOUT=A                                                XXX05690
//SORTIN  DD UNIT=DISK,DISP=(OLD,KEEP),DSN=SMICD235,                  XXX05700
//     VOL=SER=BOSWK2,DCB=(RECFM=FB,LRECL=320,BLKSIZE=3200)           XXX05710
//SORTOUT DD UNIT=DISK,DISP=(OLD,KEEP),DSN=SMICD235,                  XXX05720
//    DCB=(RECFM=FB,LRECL=320,BLKSIZE=3200),VOL=SER=BOSWK2            XXX05730
//SORTWK01 DD UNIT=DISK,SPACE=(CYL,10)                                XXX05740
//SYSIN   DD *                                                        XXX05750
 SORT FIELDS=(1,10,A),FORMAT=CH                                       XXX05760
/*                                                                    XXX05770
//STEP3 EXEC PGM=SMIMB35                                              7  00020
//STEPLIB DD DSN=USER.LOADLIB,DISP=SHR                                7  00030
//SYSPRINT DD SYSOUT=A                                                7  00040
//SYSIN DD *
3408
/*
//DIN  DD DSN=SMISM12,DISP=SHR                                        7  06470
//DIN2 DD UNIT=DISK,DSN=SMICD235,DCB=(RECFM=FB,BLKSIZE=3200,          7  06490
//        LRECL=320),DISP=(OLD,KEEP),VOL=SER=BOSWK2                   7  06500
//                                                                    7  06540
```

SMIWCP62

생산능력 TABLE (월간)

(RUN)

FILE: SMIWC166 NUNGRUK D1 VM/SP RELEASE 2.1 EXPRESS PUT8302+ SLU203

```
//SMIWCP62 JOB CLASS=B,TYPRUN=HOLD                                      SMI00010
//JOBCAT  DD  DSN=SMICT12,DISP=SHR                                      SMI00020
//STEP1 EXEC PGM=IEHPROGM                                               SMI00030
//SYSPRINT DD SYSOUT=A                                                  SMI00040
//DD2    DD  UNIT=DISK,VOL=SER=BOSWK2,DISP=OLD                          SMI00050
//SYSIN   DD  *                                                         SMI00060
          SCRATCH DSNAME=SMI3434,VOL=DISK=BOSWK2                        SMI00070
/*                                                                      SMI00080
//STEP2 EXEC PGM=SMICA34                                                SMI00090
//STEPLIB DD DSN=USER.LOADLIB,DISP=SHR                                  SMI00100
//DDD DD DISP=SHR,DSN=SMISM12                                           SMI00390
//DOF DD UNIT=DISK,DISP=(NEW,KEEP),DSN=SMI3434,VOL=SER=BOSWK2,          SMI00400
//          DCB=(RECFM=FB,LRECL=320,BLKSIZE=3200),SPACE=(320,(500,50))  SMI00410
//SYSPRINT DD SYSOUT=A                                                  SMI00420
//STEP3 EXEC PGM=SORT,PARM='SIZE(MAX)'                                  SMI00460
//SYSPRINT DD SYSOUT=A                                                  SMI00470
//SYSOUT DD SYSOUT=A                                                    SMI00480
//SORTIN DD UNIT=DISK,DISP=(OLD,KEEP),DSN=SMI3434,                      SMI00490
//           DCB=(RECFM=FB,LRECL=320,BLKSIZE=3200),VOL=SER=BOSWK2       SMI00500
//SORTOUT DD UNIT=DISK,DISP=(NEW,PASS),DSN=&&SMI34,SPACE=(CYL,1),       SMI00510
//           DCB=(RECFM=FB,LRECL=320,BLKSIZE=3200),VOL=SER=SORTWK       SMI00520
//SORTWK01 DD UNIT=DISK,SPACE=(CYL,20)                                  SMI00530
//SYSIN DD *                                                            SMI00540
          SORT FIELDS=(3,8,D,2,1,A),FORMAT=CH                           SMI00550
/*                                                                      SMI00560
//STEP4 EXEC PGM=SMICB34                                                SMI00570
//STEPLIB DD DSN=USER.LOADLIB,DISP=SHR                                  SMI00580
//DIN DD UNIT=DISK,DISP=(OLD,DELETE),DSN=&&SMI34,VOL=SER=SORTWK,        SMI02160
//          DCB=(RECFM=FB,LRECL=320,BLKSIZE=3200)                       SMI02170
//DOG DD DISP=SHR,DSN=SMISM12                                           SMI02180
//SYSPRINT DD SYSOUT=A                                                  SMI02190
//STEP5 EXEC PGM=SMITA34                                                SMI00090
//STEPLIB DD DSN=USER.LOADLIB,DISP=SHR                                  SMI00100
//SYSPRINT DD SYSOUT=A
//DDD DD DISP=SHR,DSN=SMISM12                                           SMI00390
//DOF DD UNIT=DISK,DISP=(NEW,PASS),DSN=&&SMI34,VOL=SER=SORTWK,          SMI00400
//          DCB=(RECFM=FB,LRECL=320,BLKSIZE=3200),SPACE=(320,(500,50))  SMI00410
//STEP6 EXEC PGM=SMITB34                                                SMI00130
//STEPLIB DD DSN=USER.LOADLIB,DISP=SHR                                  SMI00140
//SYSPRINT DD SYSOUT=A
//DIN DD UNIT=DISK,DISP=(OLD,DELETE),DSN=&&SMI34,VOL=SER=SORTWK,        SMI00400
//          DCB=(RECFM=FB,LRECL=320,BLKSIZE=3200)                       SMI00410
//                                                                      SMI05430
```

SMIWDB66

계절조정지수 (1)

FILE: SMIWDB66 DBLOAD D1 VM/SP RELEASE 2.1 EXPRESS PUT8302+ SLU203

```
//SMIWDB66  JOB  CLASS=V,TYPRUN=HOLD                                  SMI00010
//STEP1     EXEC PGM=DFSRRC00,PARM='DLI,LOVINGME,ZVPSBA'              SMI00020
//STEPCAT   DD   DSN=ZUSERCAT,DISP=SHR                                SMI00030
//          DD   DSN=SMICT12,DISP=SHR                                 SMI00040
//STEPLIB   DD   DSN=IMSVS.RESLIB,DISP=SHR                            SMI00050
//          DD   DSN=IMSVS.PROCLIB,DISP=SHR,UNIT=DISK,VOL=SER=IMSRES  SMI00060
//          DD   DSN=IMSVS.PGMLIB,DISP=SHR,UNIT=DISK,VOL=SER=IMSRES   SMI00070
//IMS       DD   DSN=IMSVS.PSBLIB,DISP=SHR,UNIT=DISK,VOL=SER=IMSRES   SMI00080
//          DD   DSN=IMSVS.DBDLIB,DISP=SHR,UNIT=DISK,VOL=SER=IMSRES   SMI00090
//IEFRDER   DD   DUMMY                                                SMI00100
//DBFILE    DD   DSN=ZDBFILE,UNIT=DISK,DISP=SHR,VOL=SER=BOSDBD        SMI00110
//GO.SUMF   DD   DSN=SMISM12,UNIT=3350,DISP=SHR,VOL=SER=BOSWK2        SMI00120
//SYSPRINT  DD   SYSOUT=A                                             SMI00130
//SYSUDDMP  DD   SYSOUT=A                                             SMI00140
//DFSVSAMP  DD   *                                                    SMI00150
  4096,4                                                              SMI00160
  2048,4                                                              SMI00170
//GO.SYSIN  DD  *                                                     SMI00180
INPUT8406 8407 MODIFY YES LOD MQY AZZZEFGHIJKZZZZZZZ 2046 0104 ALLK 06MQSMI00190
/*                                                                    SMI00200
//                                                                    SMI00210
```

SMIQUN66

계절조정지수 (2)

FILE: SMIWD366 DONGIBI D1 VM/SP RELEASE 2.1 EXPRESS PUT8302+ SLU203

```
//SMIQUN66  JOB  CLASS=V                                                SMI00010
//          EXEC PGM=DFSRRC00,PARM='DLI,QUANTITY,ZVPSBG'                SMI00020
//STEPCAT   DD   DSN=ZUSERCAT,DISP=SHR                                  SMI00030
//STEPLIB   DD   DSN=IMSVS.RESLIB,DISP=SHR                              SMI00040
//          DD   DSN=IMSVS.PROCLIB,DISP=SHR                             SMI00050
//          DD   DSN=IMSVS.PGMLIB,DISP=SHR                              SMI00060
//IMS       DD   DSN=IMSVS.PSBLIB,DISP=SHR                              SMI00070
//          DD   DSN=IMSVS.DBDLIB,DISP=SHR                              SMI00080
//IEFRDER   DD   DUMMY                                                  SMI00090
//DBFILE    DD   DSN=ZDBFILE,UNIT=3350,DISP=SHR,VOL=SER=BOSDBD          SMI00100
//SYSPRINT  DD   SYSOUT=A,COPIES=2                                      SMI00110
//SYSUDUMP  DD   SYSOUT=A                                               SMI00120
//DFSVSAMP  DD   *                                                      SMI00130
  4096,4                                                                SMI00140
  2048,4                                                                SMI00150
//GO.SYSIN  DD  *                                                       SMI00160
PUT NPRDQFULLY 7801 8402 4 FORM2 1 79 80 83 84                          SMI00170
PUT NSHPQFULLY 7801 8402 4 FORM2 1 79 80 83 84                          SMI00180
PUT NINTQFULLY 7801 8402 4 FORM2 1 79 80 83 84                          SMI00190
PUT DSHPQFULLY 8001 8402 4 FORM2 1 81 82 83 84                          SMI00200
PUT ESHPQFULLY 8001 8402 4 FORM2 1 81 82 83 84                          SMI00210
PUT NPRDQFULLY 7801 8403 4 FORM2 1 79 80 83 84                          SMI00220
PUT NSHPQFULLY 7801 8403 4 FORM2 1 79 80 83 84                          SMI00230
PUT NINTQFULLY 7801 8403 4 FORM2 1 79 80 83 84                          SMI00240
PUT DSHPQFULLY 8001 8403 4 FORM2 1 81 82 83 84                          SMI00250
PUT ESHPQFULLY 8001 8403 4 FORM2 1 81 82 83 84                          SMI00260
/*                                                                      SMI00270
```

SMIWD166

계절조정지수 (3)

```
//SMIWD166  JOB  CLASS=B,TYPRUN=HOLD                                    SMI00010
//STEP2     EXEC PGM=DFSRRC00,PARM='DLI,SEASONED,ZVPSBA'                SMI00020
//STEPCAT   DD   DSN=ZUSERCAT,DISP=SHR                                  SMI00030
//STEPLIB   DD   DSN=IMSVS.RESLIB,DISP=SHR                              SMI00040
//          DD   DSN=IMSVS.PROCLIB,DISP=SHR,UNIT=DISK,VOL=SER=IMSRES    SMI00050
//          DD   DSN=IMSVS.PGMLIB,DISP=SHR,UNIT=DISK,VOL=SER=IMSRES     SMI00060
//IMS       DD   DSN=IMSVS.PSBLIB,DISP=SHR,UNIT=DISK,VOL=SER=IMSRES     SMI00070
//          DD   DSN=IMSVS.DBDLIB,DISP=SHR,UNIT=DISK,VOL=SER=IMSRES     SMI00080
//IEFRDER   DD   DUMMY                                                  SMI00090
//DBFILE    DD   DSN=ZDBFILE,UNIT=DISK,DISP=SHR,VOL=SER=BOSDBD          SMI00100
//SYSPRINT  DD   SYSOUT=A                                               SMI00110
//SYSUDUMP  DD   SYSOUT=A                                               SMI00120
//DFSVSAMP  DD   *                                                      SMI00130
  4096,4                                                                SMI00140
  2048,4                                                                SMI00150
//GO.SYSIN  DD  *                                                       SMI00160
PRODXNPRDIPRODS M8406 8407 3 PRIOR 002 MODIFY                           SMI00170
SHPTXNSHPISHPTS M8406 8407 3 PRIOR 002 MODIFY                           SMI00180
INVTXNINTIINVTS M8406 8407 3 DONOT 002 MODIFY                           SMI00190
PROSXNPRSIPROSS M8406 8407 3 PRIOR 002 MODIFY                           SMI00200
PROPXNPRPIPROPS M8406 8407 3 PRIOR 002 MODIFY                           SMI00210
PRACXNPRAIPRACS M8406 8407 3 PRIOR 002 MODIFY                           SMI00220
/*                                                                      SMI00230
//STEP3     EXEC PGM=DFSRRC00,PARM='DLI,JISUSUM,ZVPSBA'                 SMI00240
//STEPCAT   DD   DSN=ZUSERCAT,DISP=SHR                                  SMI00250
//STEPLIB   DD   DSN=IMSVS.RESLIB,DISP=SHR                              SMI00260
//          DD   DSN=IMSVS.PROCLIB,DISP=SHR,UNIT=DISK,VOL=SER=IMSRES    SMI00270
//          DD   DSN=IMSVS.PGMLIB,DISP=SHR,UNIT=DISK,VOL=SER=IMSRES     SMI00280
//IMS       DD   DSN=IMSVS.PSBLIB,DISP=SHR,UNIT=DISK,VOL=SER=IMSRES     SMI00290
//          DD   DSN=IMSVS.DBDLIB,DISP=SHR,UNIT=DISK,VOL=SER=IMSRES     SMI00300
//IEFRDER   DD   DUMMY                                                  SMI00310
//DBFILE    DD   DSN=ZDBFILE,UNIT=DISK,DISP=SHR,VOL=SER=BOSDBD          SMI00320
//SYSPRINT  DD   SYSOUT=A                                               SMI00330
//SYSUDUMP  DD   SYSOUT=A                                               SMI00340
//DFSVSAMP  DD   *                                                      SMI00350
  4096,4                                                                SMI00360
  2048,4                                                                SMI00370
//GO.SYSIN  DD  *                                                       SMI00380
PRODX84068407 LODMODIFY                                                 SMI00390
SHPTX84068407 LODMODIFY                                                 SMI00400
INVTX84068407 LODMODIFY                                                 SMI00410
/*                                                                      SMI00420
//STEP4     EXEC PGM=DFSRRC00,PARM='DLI,DLITABLE,ZVPSBA'                SMI00430
//STEPCAT   DD   DSN=ZUSERCAT,DISP=SHR                                  SMI00440
//STEPLIB   DD   DSN=IMSVS.RESLIB,DISP=SHR                              SMI00450
//          DD   DSN=IMSVS.PROCLIB,DISP=SHR,UNIT=DISK,VOL=SER=IMSRES    SMI00460
//          DD   DSN=IMSVS.PGMLIB,DISP=SHR,UNIT=DISK,VOL=SER=IMSRES     SMI00470
//IMS       DD   DSN=IMSVS.PSBLIB,DISP=SHR,UNIT=DISK,VOL=SER=IMSRES     SMI00480
//          DD   DSN=IMSVS.DBDLIB,DISP=SHR,UNIT=DISK,VOL=SER=IMSRES     SMI00490
//IEFRDER   DD   DUMMY                                                  SMI00500
//DBFILE    DD   DSN=ZDBFILE,UNIT=DISK,DISP=SHR,VOL=SER=BOSDBD          SMI00510
//SYSPRINT  DD   SYSOUT=A                                               SMI00520
//SYSUDDMP  DD   SYSOUT=A                                               SMI00530
//DFSVSAMP  DD   *                                                      SMI00540
  4096,4                                                                SMI00550
```

```
  2048,4                                                                SMI00560
//GO.SYSIN  DD  *                                                       SMI00570
 'PRODX' '0'  'M'  80  1  84  07  3                                     SMI00580
 'SHPTX' '0'  'M'  80  1  84  07  3                                     SMI00590
 'INVTX' '0'  'M'  80  1  84  07  3                                     SMI00600
/*                                                                      SMI00610
//STEP5     EXEC PGM=DFSRRC00,PARM='DLI,LEETABLE,ZVPSBA'               SMI00620
//STEPCAT   DD   DSN=ZUSERCAT,DISP=SHR                                 SMI00630
//STEPLIB   DD   DSN=IMSVS.RESLIB,DISP=SHR                             SMI00640
//          DD   DSN=IMSVS.PROCLIB,DISP=SHR,UNIT=DISK,VOL=SER=IMSRES   SMI00650
//          DD   DSN=IMSVS.PGMLIB,DISP=SHR,UNIT=DISK,VOL=SER=IMSRES    SMI00660
//IMS       DD   DSN=IMSVS.PSBLIB,DISP=SHR,UNIT=DISK,VOL=SER=IMSRES    SMI00670
//          DD   DSN=IMSVS.DBDLIB,DISP=SHR,UNIT=DISK,VOL=SER=IMSRES    SMI00680
//IEFRDER   DD   DUMMY                                                 SMI00690
//DBFILE    DD   DSN=ZDBFILE,UNIT=DISK,DISP=SHR,VOL=SER=BOSDBD         SMI00700
//SYSPRINT  DD   SYSOUT=A                                              SMI00710
//SYSUDUMP  DD   SYSOUT=A                                              SMI00720
//DFSVSAMP  DD   *                                                     SMI00730
  4096,4                                                               SMI00740
  2048,4                                                               SMI00750
//GO.SYSIN  DD  *                                                       SMI00760
  840784NPRDIPRODX2 PRODUCTION INDEX                                   SMI00770
  840784NSHPISHPTX2 SHIPMENT INDEX                                     SMI00780
  840784NINTIINVTX2 INVENTORY INDEX                                    SMI00790
/*                                                                     SMI00800
//STEP6     EXEC PGM=DFSRRC00,PARM='DLI,LEETABLE,ZVPSBA'               SMI00810
//STEPCAT   DD   DSN=ZUSERCAT,DISP=SHR                                 SMI00820
//STEPLIB   DD   DSN=IMSVS.RESLIB,DISP=SHR                             SMI00830
//          DD   DSN=IMSVS.PROCLIB,DISP=SHR,UNIT=DISK,VOL=SER=IMSRES   SMI00840
//          DD   DSN=IMSVS.PGMLIB,DISP=SHR,UNIT=DISK,VOL=SER=IMSRES    SMI00850
//IMS       DD   DSN=IMSVS.PSBLIB,DISP=SHR,UNIT=DISK,VOL=SER=IMSRES    SMI00860
//          DD   DSN=IMSVS.DBDLIB,DISP=SHR,UNIT=DISK,VOL=SER=IMSRES    SMI00870
//IEFRDER   DD   DUMMY                                                 SMI00880
//DBFILE    DD   DSN=ZDBFILE,UNIT=DISK,DISP=SHR,VOL=SER=BOSDBD         SMI00890
//SYSPRINT  DD   SYSOUT=A                                              SMI00900
//SYSUDUMP  DD   SYSOUT=A                                              SMI00910
//DFSVSAMP  DD   *                                                     SMI00920
  4096,4                                                               SMI00930
  2048,4                                                               SMI00940
//GO.SYSIN  DD  *                                                       SMI00950
  840784NPRPIPROPX3 OPERATION RATIO INDEX                              SMI00960
  840784NPRAIPRACX3 CAPACITY PROD INDEX                                SMI00970
  840784NPRCIDONOT3 CAPACITY INDEX                                     SMI00980
/*                                                                     SMI00990
//                                                                     SMI01000
```

SMIWT366

동태분기 TABLE

```
//SMIWT366 JOB CLASS=1,TYPRUN=HOLD                                      SMI00010
//JOBCAT DD DSN=SMICT12,DISP=SHR                                        SMI00020
//STEP2    EXEC     PLIFCLG                                             SMI00030
//PLI.SYSIN  DD     *                                                   SMI00040
* PROCESS   GS,NEST,A,OPT(2);                                           SMI00050
      (NOZERODIVIDE) :                                                  SMI00060
    TAB:   PROC   OPTIONS(MAIN);                                        SMI00070
          DCL    DIN   FILE RECORD KEYED SEQUENTIAL INPUT ENV(VSAM);    SMI00080
                   DCL    1      IN,                                   SMI00090
                                 2  CD         CHAR(1),                SMI00100
                                 2  GU         CHAR(1),                SMI00110
                                 2  SAN        CHAR(3),                SMI00120
                                 2  PUM        CHAR(5),                SMI00130
                                 2  TK         CHAR(1),                SMI00140
                                 2  SI         PIC  '(4)9',            SMI00150
                                 2  GA         FLOAT DEC(6),           SMI00160
                                 2  MUL        FIXED BIN(31),          SMI00170
                                 2  DATA(37),                          SMI00180
                                    3 DMUL     FIXED BIN(31),          SMI00190
                                    3 DGI      FLOAT DEC(6),           SMI00200
                                 2  BB         CHAR(1);                SMI00210
   DCL  AA(6)  LABEL INIT(LBL3,LBL5,LBL6,LBL7,LBL8,LBL10);              SMI00220
               DCL  SW     CHAR(1) INIT('*');                           SMI00230
               DCL  SRT(6)     FLOAT DEC(15);                           SMI00240
               DCL  SRTA       FLOAT DEC(15);                           SMI00250
               DCL  OGU        CHAR(1);                                 SMI00260
               DCL  SRTB       FLOAT DEC(15);                           SMI00270
               DCL  E          CHAR(1);                                 SMI00280
               DCL  PYO        PIC  '99' INIT(1);                       SMI00290
               DCL  A          PIC  '99' DEF STOR POS(1);               SMI00300
               DCL  B          PIC  '99' DEF STOR POS(3);               SMI00310
               DCL  G          PIC  '99' DEF STOR POS(5);               SMI00320
               DCL  C          PIC  '99';                               SMI00330
               DCL  D          PIC  '99';                               SMI00340
               DCL  L          PIC  '99';                               SMI00350
               DCL  STOR       CHAR(6);                                 SMI00360
               DCL  SRTC       FLOAT DEC(15);                           SMI00370
               DCL  SRTD       FLOAT DEC(15);                           SMI00380
               DCL  DGIA       FLOAT DEC(15);                           SMI00390
               DCL  DGIB       FLOAT DEC(15);                           SMI00400
               DCL  DGIC       FLOAT DEC(15);                           SMI00410
               DCL  DGIE       FLOAT DEC(15);                           SMI00420
               DCL  SGIA       FLOAT DEC(15);                           SMI00430
               DCL  SGIB       FLOAT DEC(15);                           SMI00440
               DCL  SGIC       FLOAT DEC(15);                           SMI00450
               DCL  SGIE       FLOAT DEC(15);                           SMI00460
               DCL  SG         FLOAT DEC(15);                           SMI00470
               DCL  DG         FLOAT DEC(15);                           SMI00480
               DCL  1     SAVR,                                         SMI00490
                          2 SAR(37),                                    SMI00500
                            3 SMUL     FIXED BIN(31),                   SMI00510
                            3 SGI      FLOAT DEC(6);                    SMI00520
     DISPLAY ('*****************************************');             SMI00530
     DISPLAY ('*** PLEASE  KEY IN NUMBER SIX  CHARACTERS ***');         SMI00540
     DISPLAY ('*****************************************')              SMI00550
```

```
            REPLY(STOR);                                                SMI00560
   OP:     OPEN FILE(DIN),FILE(SYSPRINT) LINESIZE(132);                 SMI00570
               ON ENDFILE(DIN) GOTO LAST;                               SMI00580
           SW='*';                                                      SMI00590
         IF PYO=2 THEN OGU='J';                                         SMI00600
            ELSE IF PYO=3 THEN OGU='M';                                 SMI00610
            ELSE OGU='A';                                               SMI00620
       IF A=37 THEN E='P';                                              SMI00630
          ELSE E='F';                                                   SMI00640
  RD: READ FILE(DIN) INTO(IN);                                          SMI00650
           IF SUBSTR(SAN,1,1)  ='0' THEN  GOTO RR;                      SMI00660
              GOTO AA(PYO);                                             SMI00670
 LBL3: IF GU='A' | GU>'D' &  GU<'J' THEN DO;                            SMI00680
        IF PUM¬='00000' THEN GOTO RD;                                   SMI00690
        IF SW¬= '*' THEN GOTO R4;                                       SMI00700
           CALL PUT4;                                                   SMI00710
 PUT4:     PROC;                                                        SMI00720
           PUT PAGE;                                                    SMI00730
           PUT SKIP(3) EDIT('# 3.  TABLE  OF CHANGE  FROM SAME MONTH  OFSMI00740
  PREVIOUS  YEAR BY   INDUSTRIES')(X(23),A(76),X(33));                  SMI00750
           PUT SKIP    EDIT((76)'-')(X(23),A(76),X(33));                SMI00760
           PUT SKIP(2) EDIT('MONTH=',G,E,GU)(X(5),A(6),F(2),A,A);       SMI00770
           PUT SKIP(3) EDIT((115)'-')(A(115));                          SMI00780
           PUT SKIP   EDIT('   INC  -   DEC   (%)        DAE - CHONG  GISMI00790
 - YU - DO')(X(61),A(54));                                              SMI00800
          PUT SKIP   EDIT((54)'-')(X(61),A(54));                        SMI00810
          PUT SKIP   EDIT('          CODE              PMY-2          PMY-1SMI00820
            PM           B/A           C/B            B/A            C/B')SMI00830
   (A(115));                                                            SMI00840
           PUT SKIP    EDIT((115)'-')(A(115));                          SMI00850
           SW='$';                                                      SMI00860
           END PUT4;                                                    SMI00870
    R4:    IF OGU¬=GU THEN DO;                                          SMI00880
             CALL PUT4; END;                                            SMI00890
           OGU=GU;                                                      SMI00900
           SRT=0;                                                       SMI00910
           CALL AVPA;                                                   SMI00920
           SRTA=0; SRTB=0; SRTC=0; SRTD=0;                              SMI00930
           CALL AVPB;                                                   SMI00940
       PUT SKIP(2) EDIT(SAN,(SRT(I) DO I=1 TO 3),SRTA,SRTB,SRTC,SRTD)   SMI00950
                     (X(10),A(7),3 F(14,1),2 F(14,1),2 F(14,2));        SMI00960
       GOTO RD; END;                                                    SMI00970
       GOTO RD;                                                         SMI00980
    LBL5:                                                               SMI00990
      IF GU < 'J' | GU > 'L' THEN GOTO RD;                              SMI01000
      IF SW¬= '*' THEN GOTO R5;                                         SMI01010
      CALL PUT5;                                                        SMI01020
 PUT5: PROC;                                                            SMI01030
      PUT PAGE;                                                         SMI01040
      PUT SKIP(3) EDIT('# 5.  TABLE   OF  INDEX   BY  SPECIAL GROUPS(USSMI01050
ER) BY  MONTHLY')(X(35),A(62),X(35));                                   SMI01060
      PUT SKIP EDIT((62)'-')(X(35),A(62),X(35));                        SMI01070
          PUT SKIP(2) EDIT('MONTH=',G,E,GU)(X(5),A(6),F(2),A,A);        SMI01080
          PUT SKIP(3) EDIT((117)'-')(A(117));                           SMI01090
          PUT SKIP    EDIT('        INC  -  DEC  (%)             DAE-CHSMI01100
```

```
ONG   GI-YU-DO')(X(57),A(57));                                          SMI01110
          PUT SKIP    EDIT((60)'-')(X(57),A(60));                       SMI01120
          PUT SKIP    EDIT('SCODE      AVPY-2    AVPY-1      PM-2      SMI01130
 PM-1       PM       VJW     VJNDW     VJNDG       VJW     VJNDW    VJNSMI01140
DG')(A(117));                                                           SMI01150
          PUT SKIP    EDIT((117)'-')(A(117));                           SMI01160
          SW='$';                                                       SMI01170
          END PUT5;                                                     SMI01180
  R5:    IF OGU¬=GU THEN DO;                                            SMI01190
          CALL PUT5; END;                                               SMI01200
          OGU=GU;                                                       SMI01210
          SRT=0; SRTA=0; SRTB=0; SRTC=0;                                SMI01220
          L=A-8;                                                        SMI01230
          DG=0; SG=0;                                                   SMI01240
          CALL AAA;                                                     SMI01250
          DGIA=DG; SGIA=SG; DG=0; SG=0;                                 SMI01260
          L=A-5;                                                        SMI01270
          CALL AAA;                                                     SMI01280
          DGIB=DG; SGIB=SG; DG=0; SG=0;                                 SMI01290
          L=A-2;                                                        SMI01300
          CALL AAA;                                                     SMI01310
          DGIC=DG; SGIC=SG; DG=0; SG=0;                                 SMI01320
          L=A-14;                                                       SMI01330
          CALL AAA;                                                     SMI01340
          DGIE=DG; SGIE=SG;                                             SMI01350
          SRTA=0; SRTB=0; SRTC=0;                                       SMI01360
          CALL  INC;                                                    SMI01370
          SRTA=0; SRTB=0; SRTC=0;                                       SMI01380
          CALL  GYU;                                                    SMI01390
          SRTA=0; SRTB=0; SRTC=0;                                       SMI01400
          CALL  AVPY;                                                   SMI01410
      PUT SKIP(2) EDIT(GU,SAN,SRTA,SRTB,DGIA,DGIB,DGIC,(SRT             SMI01420
              (I)  DO I = 1 TO 3),(SRT(J) DO J = 4 TO 6))               SMI01430
              (A(1),A(7),5 F(10,1), 3 F(10,1),3 F(10,2));               SMI01440
          GOTO RD ;                                                     SMI01450
    LBL6:                                                               SMI01460
      IF GU < 'M' | GU > 'O' THEN GOTO RD;                              SMI01470
      IF SW¬= '*' THEN GOTO R6;                                         SMI01480
      CALL PUT6;                                                        SMI01490
 PUT6: PROC;                                                            SMI01500
      PUT PAGE;                                                         SMI01510
      PUT SKIP(3) EDIT('# 6.  TABLE  OF INDEX  BY SPECIAL GROUP(STRUCTUSMI01520
RE) BY  MONTHLY')(X(35),A(62),X(35));                                   SMI01530
      PUT SKIP EDIT((62)'-')(X(35),A(62),X(35));                        SMI01540
      PUT SKIP(2) EDIT('MONTH',G,E,GU)(X(5),A(6),F(2),A,A);             SMI01550
          PUT SKIP(3) EDIT((117)'-')(A(117));                           SMI01560
          PUT SKIP    EDIT('        INC  -  DEC  (%)            DAE-CHSMI01570
ONG   GI-YU-DO')(X(57),A(57));                                          SMI01580
          PUT SKIP    EDIT((60)'-')(X(57),A(60));                       SMI01590
          PUT SKIP    EDIT('SCODE      AVPY-2    AVPY-1      PM-2      SMI01600
 PM-1       PM       VJW     VJNDW     VJNDG       VJW     VJNDW    VJNSMI01610
DG')(A(117));                                                           SMI01620
          PUT SKIP    EDIT((117)'-')(A(117));                           SMI01630
          SW='$';                                                       SMI01640
          END PUT6;                                                     SMI01650
```

```
 R6:     IF OGU¬=GU THEN DO;                                          SMI01660
          CALL PUT6; END;                                             SMI01670
          OGU=GU;                                                     SMI01680
          L=A-8;                                                      SMI01690
          DG=0; SG=0;                                                 SMI01700
          CALL AAA;                                                   SMI01710
          DGIA=DG; SGIA=SG; DG=0; SG=0;                               SMI01720
          L=A-5;                                                      SMI01730
          CALL AAA;                                                   SMI01740
          DGIB=DG; SGIB=SG; DG=0; SG=0;                               SMI01750
          L=A-2;                                                      SMI01760
          CALL AAA;                                                   SMI01770
          DGIC=DG; SGIC=SG; DG=0; SG=0;                               SMI01780
          SRT=0; SRTA=0; SRTB=0; SRTC=0;                              SMI01790
          L=A-14;                                                     SMI01800
          CALL AAA;                                                   SMI01810
          DGIE=DG; SGIE=SG;                                           SMI01820
          SRTA=0; SRTB=0; SRTC=0;                                     SMI01830
          CALL  INC;                                                  SMI01840
          SRTA=0; SRTB=0; SRTC=0;                                     SMI01850
          CALL  GYU;                                                  SMI01860
          SRTA=0; SRTB=0; SRTC=0;                                     SMI01870
          CALL  AVPY;                                                 SMI01880
  PUT SKIP(2) EDIT(GU,SAN,SUBSTR(PUM,1,1),SRTA,SRTB,DGIA,DGIB,        SMI01890
 DGIC,(SRT(I)  DO I = 1 TO 3),(SRT(J) DO J = 4 TO 6))                 SMI01900
 (A,A,A(6),5 F(10,1), 3 F(10,1),3 F(10,2));                           SMI01910
          GOTO RD ;                                                   SMI01920
 LBL7:                                                                SMI01930
          IF GU='A' | GU>'B' & GU<'J' THEN DO;                        SMI01940
          IF SW¬='*' THEN GOTO R7;                                    SMI01950
          CALL PUT7;                                                  SMI01960
 PUT7:    PROC;                                                       SMI01970
          PUT PAGE;                                                   SMI01980
          PUT SKIP(3) EDIT('# 7.  TABLE  OF CHANGE  FROM PREVIOUS MONTHSMI01990
 OF INDEX & QUANITITY BY COMMODITIES')(X(30),A);                      SMI02000
          PUT SKIP    EDIT((76)'-')(X(30),A);                         SMI02010
          PUT SKIP(2) EDIT('MONTH=',G,E,GU)(X(5),A(6),F(2),A,A);      SMI02020
          PUT SKIP(3) EDIT((121)'-')(A(121));                         SMI02030
          PUT SKIP   EDIT('              QUANTITY                  INC SMI02040
-  DEC(%)            INDEX                 INC - DEC (%)        ')    SMI02050
             (X(8),A);                                                SMI02060
         PUT SKIP   EDIT((113)'-')(X(8),A);                           SMI02070
         PUT SKIP   EDIT('    CODE          PM-2       PM-1        PM SMI02080
  B/A       C/B    WEIGHT       PM-2        PM-1       PM         B/A SMI02090
 C/B')(A);                                                            SMI02100
          PUT SKIP    EDIT((121)'-')(A(121));                         SMI02110
          SW='$';                                                     SMI02120
          END PUT7;                                                   SMI02130
  R7:     IF OGU¬=GU THEN DO;                                         SMI02140
            CALL PUT7; END;                                           SMI02150
          OGU=GU;                                                     SMI02160
          SRT=0; SRTA=0; SRTB=0; SRTC=0;                              SMI02170
          L=A-8;                                                      SMI02180
          SG=0;                                                       SMI02190
          CALL BBB;                                                   SMI02200
```

```
          SRT(5)=SG;                                                    SMI02210
          SG=0;                                                         SMI02220
          L=A-5;                                                        SMI02230
          SG=0;                                                         SMI02240
          CALL BBB;                                                     SMI02250
          SRT(6)=SG;                                                    SMI02260
          SG=0;                                                         SMI02270
          L=A-2;                                                        SMI02280
          SG=0;                                                         SMI02290
          CALL BBB;                                                     SMI02300
          SRTA=SG;                                                      SMI02310
          SG=0; DG=0;                                                   SMI02320
          L=A-8;                                                        SMI02330
          CALL AAA;                                                     SMI02340
          DGIA=DG;                                                      SMI02350
          DG=0;                                                         SMI02360
          L=A-5;                                                        SMI02370
          CALL AAA;                                                     SMI02380
          DGIB=DG;                                                      SMI02390
          DG=0;                                                         SMI02400
          L=A-2;                                                        SMI02410
          CALL AAA;                                                     SMI02420
          DGIC=DG;                                                      SMI02430
          DG=0;                                                         SMI02440
          CALL CCA;                                                     SMI02450
          CALL CGA;                                                     SMI02460
       PUT SKIP(2) EDIT(SAN,PUM,SRT(5),SRT(6),SRTA,SRT(1),SRT(2),GA,    SMI02470
          DGIA,DGIB,DGIC,SRT(3),SRT(4))(A(3),A(5),X(3),                 SMI02480
           3 F(10), 8 F(10,1));                                         SMI02490
       GOTO RD; END;                                                    SMI02500
       GOTO RD;                                                         SMI02510
     CCA: PROC ;                                                        SMI02520
          IF SRT(6) = 0 THEN SRT(1) = 0;                                SMI02530
           ELSE SRT(1) =(SRTA / SRT(6)) * 100 - 100;                    SMI02540
          IF SRT(5) = 0 THEN SRT(2) = 0;                                SMI02550
           ELSE SRT(2) =(SRTA / SRT(5)) * 100 - 100;                    SMI02560
          END CCA ;                                                     SMI02570
     CGA:  PROC ;                                                       SMI02580
          IF  DGIA = 0 THEN SRT(3) = 0;                                 SMI02590
          ELSE SRT(3) =  (DGIC / DGIA) * 100 - 100;                     SMI02600
          IF  DGIB = 0 THEN SRT(4) = 0;                                 SMI02610
          ELSE SRT(4) =  (DGIC / DGIB) * 100 - 100;                     SMI02620
      END CGA;                                                          SMI02630
LBL8:                                                                   SMI02640
          IF GU='A' | GU>'B' & GU<'J' THEN DO;                          SMI02650
          IF SW¬='*' THEN GOTO R8;                                      SMI02660
          CALL PUT8;                                                    SMI02670
PUT8:     PROC;                                                         SMI02680
          PUT PAGE;                                                     SMI02690
          PUT SKIP(3) EDIT('# 8.  TABLE  OF CHANGE  FROM SAME MONTH OF  SMI02700
PREVIOUS YEAR OF INDEX & QUANITITY BY COMMODITIES')(X(20),A);           SMI02710
          PUT SKIP    EDIT((93)'-')(X(20),A);                           SMI02720
          PUT SKIP(2) EDIT('MONTH=',G,E,GU)(X(5),A(6),F(2),A,A);        SMI02730
          PUT SKIP(3) EDIT((121)'-')(A(121));                           SMI02740
          PUT SKIP   EDIT('                QUANTITY                INC -SMI02750
```

```
 DEC(%)                INDEX                INC - DEC (%)     ')            SMI02760
            (X(21),A);                                                      SMI02770
        PUT SKIP   EDIT((100)'-')(X(21),A);                                 SMI02780
        PUT SKIP   EDIT('      CODE        WEIGHT    PY-2PM     PY-1PM      SMI02790
  PYPM        B/A        C/B     PY-2PM     PY-1PM      PYPM        B/A     SMI02800
  C/B')(A);                                                                 SMI02810
        PUT SKIP    EDIT((121)'-')(A(121));                                 SMI02820
        SW='$';                                                             SMI02830
        END PUT8;                                                           SMI02840
   R8:  IF OGU¬=GU THEN DO;                                                 SMI02850
          CALL PUT8; END;                                                   SMI02860
        OGU=GU;                                                             SMI02870
        SRT=0; SRTA=0; SRTB=0; SRTC=0;                                      SMI02880
        L=A-26;                                                             SMI02890
        SG=0;                                                               SMI02900
        CALL BBB;                                                           SMI02910
        SRT(5)=SG;                                                          SMI02920
        SG=0;                                                               SMI02930
        L=A-14;                                                             SMI02940
        SG=0;                                                               SMI02950
        CALL BBB;                                                           SMI02960
        SRT(6)=SG;                                                          SMI02970
        SG=0;                                                               SMI02980
        L=A-2;                                                              SMI02990
        SG=0;                                                               SMI03000
        CALL BBB;                                                           SMI03010
        SRTA=SG;                                                            SMI03020
        SG=0; DG=0;                                                         SMI03030
        L=A-26;                                                             SMI03040
        CALL AAA;                                                           SMI03050
        DGIA=DG;                                                            SMI03060
        DG=0;                                                               SMI03070
        L=A-14;                                                             SMI03080
        CALL AAA;                                                           SMI03090
        DGIB=DG;                                                            SMI03100
        DG=0;                                                               SMI03110
        L=A-2;                                                              SMI03120
        CALL AAA;                                                           SMI03130
        DGIC=DG;                                                            SMI03140
        DG=0;                                                               SMI03150
        CALL CCB;                                                           SMI03160
        CALL KGB;                                                           SMI03170
      PUT SKIP(2) EDIT(SAN,PUM,GA,SRT(5),SRT(6),SRTA,SRT(1),SRT(2),         SMI03180
        DGIA,DGIB,DGIC,SRT(3),SRT(4))(A(3),A,X(3),                          SMI03190
        F(10,1), 3 F(10), 7 F(10,1));                                       SMI03200
      GOTO RD; END;                                                         SMI03210
      GOTO RD;                                                              SMI03220
    CCB: PROC ;                                                             SMI03230
        IF SRT(6) = 0 THEN SRT(1) = 0;                                      SMI03240
         ELSE SRT(1) =(SRTA / SRT(6)) * 100 - 100;                          SMI03250
        IF SRT(5) = 0 THEN SRT(2) = 0;                                      SMI03260
         ELSE SRT(2) =(SRTA / SRT(5)) * 100 - 100;                          SMI03270
        END CCB ;                                                           SMI03280
    KGB:  PROC ;                                                            SMI03290
        IF DGIA = 0 THEN SRT(3) = 0;                                        SMI03300
```

```
          ELSE  SRT(3) =  (DGIC / DGIA) * 100 - 100;                    SMI03310
          IF DGIB = 0 THEN SRT(4) = 0;                                  SMI03320
          ELSE  SRT(4) =  (DGIC / DGIB) * 100 - 100;                    SMI03330
        END KGB;                                                        SMI03340
      LBL10:                                                            SMI03350
          IF GU ='A' | GU >'D' & GU('J' THEN DO;                        SMI03360
      IF SW¬= '*' THEN GOTO R10;                                        SMI03370
      CALL PUT10;                                                       SMI03380
  PUT10: PROC;                                                          SMI03390
      PUT PAGE;                                                         SMI03400
          PUT SKIP(3) EDIT('# 10. TABLE   OF   INDEX  BY  INDUSTRIES    SMI03410
   BY    MONTHLY ')(X(30),A(60));                                       SMI03420
          PUT SKIP    EDIT((60)'-')(X(30),A(60));                       SMI03430
          PUT SKIP(2) EDIT('MONTH=',G,E,GU)(X(5),A(6),F(2),A,A);        SMI03440
          PUT SKIP(3) EDIT((117)'-')(A(117));                           SMI03450
          PUT SKIP    EDIT('        INC  -  DEC  (%)             DAE-CHSMI03460
 ONG   GI-YU-DO')(X(57),A(57));                                         SMI03470
          PUT SKIP    EDIT((60)'-')(X(57),A(60));                       SMI03480
          PUT SKIP    EDIT('CODE        AVPY-2    AVPY-1      PM-2      SMI03490
  PM-1       PM       VJW     VJNDW     VJNDG       VJW     VJNDW     VJSMI03500
 DG')(A(117));                                                          SMI03510
          PUT SKIP    EDIT((117)'-')(A(117));                           SMI03520
          SW='$' ;                                                      SMI03530
          END PUT10;                                                    SMI03540
   R10:   IF OGU¬=GU THEN DO;                                           SMI03550
          CALL PUT10; END;                                              SMI03560
          OGU=GU;                                                       SMI03570
          IF PUM ¬='00000' THEN GOTO RD;                                SMI03580
          L=A-8;                                                        SMI03590
          DG=0; SG=0;                                                   SMI03600
          CALL AAA;                                                     SMI03610
          DGIA=DG; SGIA=SG; DG=0; SG=0;                                 SMI03620
          L=A-5;                                                        SMI03630
          CALL AAA;                                                     SMI03640
          DGIB=DG; SGIB=SG; DG=0; SG=0;                                 SMI03650
          L=A-2;                                                        SMI03660
          CALL AAA;                                                     SMI03670
          DGIC=DG; SGIC=SG; DG=0; SG=0;                                 SMI03680
          SRT=0; SRTA=0; SRTB=0; SRTC=0;                                SMI03690
          L=A-14;                                                       SMI03700
          CALL AAA;                                                     SMI03710
          DGIE=DG; SGIE=SG;                                             SMI03720
          SRTA=0; SRTB=0; SRTC=0;                                       SMI03730
          CALL INC;                                                     SMI03740
          SRTA=0; SRTB=0; SRTC=0;                                       SMI03750
          CALL GYU;                                                     SMI03760
          SRTA=0; SRTB=0; SRTC=0;                                       SMI03770
          CALL AVPY;                                                    SMI03780
          PUT SKIP(2) EDIT(SAN,SRTA,SRTB,DGIA,DGIB,DGIC,(SRT            SMI03790
              (I)  DO I = 1 TO 3),(SRT(J) DO J = 4 TO 6))               SMI03800
              (A(7),5 F(10,1),3 F(10,1),3 F(10,2));                     SMI03810
          GOTO RD ; END;                                                SMI03820
          GOTO RD;                                                      SMI03830
      INC: PROC ;                                                       SMI03840
          IF DGIB=0 THEN SRT(1)=0;                                      SMI03850
```

```
        ELSE SRT(1) = (DGIC / DGIB) * 100 - 100;                    SMI03860
        IF DGIA=0 THEN SRT(2)=0;                                    SMI03870
        ELSE SRT(2) = (DGIC / DGIE) * 100 - 100 ;                   SMI03880
        C = (A-B) + 1;                                              SMI03890
        DO  J = C TO A;                                             SMI03900
            SRTA = SRTA + DGI(J) ;                                  SMI03910
        END ;                                                       SMI03920
        SRTA = SRTA / B ;                                           SMI03930
        C = (A-B) - 11;                                             SMI03940
        D = (C+B) - 1;                                              SMI03950
        DO  J = C TO D ;                                            SMI03960
            SRTB = SRTB + DGI(J) ;                                  SMI03970
        END;                                                        SMI03980
        SRTB = SRTB / B ;                                           SMI03990
        IF SRTB=0 THEN SRT(3)=0;                                    SMI04000
        ELSE  SRT(3) = (SRTA / SRTB) * 100 - 100 ;                  SMI04010
        END INC ;                                                   SMI04020
    GYU:  PROC ;                                                    SMI04030
        IF SGIB=0 THEN SRT(4)=0;                                    SMI04040
        ELSE SRT(4) = ((DGIC - DGIB) * GA) / (SGIB * 100);          SMI04050
        IF SGIA=0 THEN SRT(5)=0;                                    SMI04060
        ELSE SRT(5) = ((DGIC - DGIE) * GA) / (SGIE * 100);          SMI04070
        C = (A-B) +1;                                               SMI04080
        DO  J = C TO A ;                                            SMI04090
            SRTA = SRTA + DGI(J);                                   SMI04100
        END;                                                        SMI04110
        SRTA = SRTA / B;                                            SMI04120
        C=(A-B) - 11;                                               SMI04130
        D=(C+B) - 1;                                                SMI04140
        DO  J = C TO D ;                                            SMI04150
            SRTB = SRTB + DGI(J);                                   SMI04160
            SRTC = SRTC + SGI(J) ;                                  SMI04170
        END ;                                                       SMI04180
        SRTB = SRTB / B ;  SRTC = SRTC / B ;                        SMI04190
        IF SRTC=0 THEN SRT(6)=0;                                    SMI04200
        ELSE  SRT(6) = ((SRTA - SRTB)* GA) /  (SRTC * 100);         SMI04210
        END  GYU ;                                                  SMI04220
   AVPY:    PROC ;                                                  SMI04230
            L=(A-(B+23));                                           SMI04240
        DO  J = 1 TO 12 ;                                           SMI04250
            SRTA = SRTA + DGI(L);                                   SMI04260
            L=L+1;                                                  SMI04270
        END ;                                                       SMI04280
        SRTA = SRTA / 12;                                           SMI04290
            L=(A-(B+11));                                           SMI04300
        DO  J = 1 TO 12 ;                                           SMI04310
              SRTB = SRTB + DGI(L);                                 SMI04320
              L=L+1;                                                SMI04330
        END ;                                                       SMI04340
        SRTB = SRTB / 12;                                           SMI04350
        END AVPY;                                                   SMI04360
   AVPA:    PROC ;                                                  SMI04370
            L=A-26;                                                 SMI04380
        DO  J = 1 TO 3 ;                                            SMI04390
            SRT(1) = SRT(1) + DGI(L);                               SMI04400
```

```
            SRT(4) = SRT(4) + SGI(L);                                   SMI04410
            L=L+1;                                                      SMI04420
        END ;                                                           SMI04430
        SRT(1) = SRT(1) / 3;                                            SMI04440
        SRT(4) = SRT(4) / 3;                                            SMI04450
            L=A-14;                                                     SMI04460
        DO  J = 1 TO 3 ;                                                SMI04470
              SRT(2) = SRT(2) + DGI(L);                                 SMI04480
              SRT(5) = SRT(5) + SGI(L);                                 SMI04490
              L=L+1;                                                    SMI04500
        END ;                                                           SMI04510
        SRT(2) = SRT(2) / 3;                                            SMI04520
        SRT(5) = SRT(5) / 3;                                            SMI04530
        L=A-2;                                                          SMI04540
        DO  J = 1 TO 3 ;                                                SMI04550
              SRT(3) = SRT(3) + DGI(L);                                 SMI04560
              L=L+1;                                                    SMI04570
        END ;                                                           SMI04580
        SRT(3) = SRT(3) / 3;                                            SMI04590
        END AVPA;                                                       SMI04600
    AVPB: PROC;                                                         SMI04610
        IF SRT(1)=0 THEN SRTA=0;                                        SMI04620
        ELSE SRTA=(SRT(2)/SRT(1)) * 100 - 100;                          SMI04630
        IF SRT(2)=0 THEN SRTB=0;                                        SMI04640
        ELSE SRTB=(SRT(3)/SRT(2)) * 100 - 100;                          SMI04650
        IF SRT(4)=0 THEN SRTC=0;                                        SMI04660
        ELSE SRTC=((SRT(2)-SRT(1)) * GA) / (SRT(4)*100);                SMI04670
        IF SRT(5)=0 THEN SRTD=0;                                        SMI04680
        ELSE SRTD=((SRT(3)-SRT(2)) * GA) / (SRT(5)*100);                SMI04690
        END AVPB;                                                       SMI04700
      AAA: PROC;                                                        SMI04710
        DO J = 1 TO 3;                                                  SMI04720
           DG = DG + DGI(L);                                            SMI04730
           SG = SG + SGI(L);                                            SMI04740
           L=L+1;                                                       SMI04750
        END;                                                            SMI04760
        DG=DG/3;                                                        SMI04770
        SG=SG/3;                                                        SMI04780
        END AAA;                                                        SMI04790
      BBB: PROC;                                                        SMI04800
        DO J = 1 TO 3;                                                  SMI04810
           SG = SG + DMUL(L);                                           SMI04820
           L=L+1;                                                       SMI04830
        END;                                                            SMI04840
        SG=SG/3;                                                        SMI04850
        END BBB;                                                        SMI04860
      RR:                                                               SMI04870
        SAR=DATA; GOTO AA(PYO);                                         SMI04880
      LAST:  CLOSE FILE(DIN);                                           SMI04890
             PYO=PYO+1;                                                 SMI04900
             IF PYO < 7 THEN GOTO OP;                                   SMI04910
      END  TAB ;                                                        SMI04920
/*                                                                      SMI04930
//GO.SYSPRINT  DD   SYSOUT=A,COPIES=1                                   SMI04940
//GO.DIN   DD  DSN=SMISM12,DISP=SHR                                     SMI04950
```

```
//GO.SYSOUT    DD    SYSOUT=A                                    SMI04960
/*                                                               SMI04970
//                                                               SMI04980
```

SMIWB362

고용분기 TABLE

FILE: SMIWB366 GYBUNGI D1 VM/SP RELEASE 2.1 EXPRESS PUT8302+ SLU203

```
//SMIWB362 JOB CLASS=B,TYPRUN=HOLD                                      SMI00010
//JOBCAT DD DSN=SMICT12,DISP=SHR                                        SMI00020
//STEP1 EXEC PLIFCLG                                                    SMI00030
//PLI.SYSIN DD *                                                        SMI00040
* PROCESS GS,NEST,OPT(2);                                               SMI00050
  (NOZERODIVIDE) :                                                      SMI00060
 MON_TAB : PROC OPTIONS(MAIN);                                          SMI00070
  DCL DIN FILE RECORD INPUT ;                                           SMI00080
  DCL DIN2 FILE RECORD INPUT ;                                          SMI00090
  DCL 1 IN,                                                             SMI00100
      2 CARD        CHAR(1),                                            SMI00110
      2 F1          CHAR(1),                                            SMI00120
      2 SANUP       CHAR(3),                                            SMI00130
      2 GUBUN       CHAR(1),                                            SMI00140
      2 BUN         CHAR(1),                                            SMI00150
      2 F2          CHAR(4),                                            SMI00160
      2 YYMM        CHAR(4),                                            SMI00170
      2 WEIGHT      FLOAT DEC(6),                                       SMI00180
      2 POINT       FIXED BIN(31),                                      SMI00190
      2 DATA(37),                                                       SMI00200
         3 MMM      FIXED BIN(31),                                      SMI00210
         3 GISU     FLOAT DEC(6) ,                                      SMI00220
      2 F3          CHAR(1) ;                                           SMI00230
  DCL YY           CHAR(2) DEF YYMM POS(1) ;                            SMI00240
  DCL MM           CHAR(2) DEF YYMM POS(3) ;                            SMI00250
  DCL (A,I,J,K,L,M,N,NN,W,B,T,C)    PIC'99' INIT(0);                   SMI00260
  DCL ASAN(45)      CHAR(3)  INIT((45)0) ;                              SMI00270
  DCL AA(45,4)      FLOAT DEC(16) INIT((180)0);                         SMI00280
  DCL TAB3(45,17)   FLOAT DEC(16) INIT((765)0);                         SMI00290
  DCL TAB4(2,45,15)   FLOAT DEC(16) INIT((1350)0);                      SMI00300
  DCL TAB6(2,45,11)   FLOAT DEC(16) INIT((990)0);                       SMI00310
  DCL (TJMGISU,TWEIGHT)   FLOAT DEC(16) INIT(0);                        SMI00320
  DCL (TJYGISU21,TJYGISU22)   FLOAT DEC(16) INIT(0);                    SMI00330
  DCL (JYGISU21,JYGISU22)   FLOAT DEC(16) INIT(0);                      SMI00340
  DCL MUL(37)               FLOAT DEC(16) INIT((37)0);                  SMI00350
  DCL GGGG                  FLOAT DEC(16) INIT(0);                      SMI00360
  DCL (LSS,LSW,MSW)    PIC'9' INIT(0);                                  SMI00370
  DCL ( ZZ1,ZZ2   )    PIC'9999999V9' INIT(0);                          SMI00380
  DCL SSANUP        CHAR(3) INIT('$$$') ;                               SMI00390
  OPEN FILE(DIN) ;                                                      SMI00400
  OPEN FILE(SYSPRINT) LINESIZE(130) ;                                   SMI00410
  ON ENDFILE(DIN2) BEGIN ;                                              SMI00420
         CLOSE FILE(DIN2) ;                                             SMI00430
         IF LSS = 1 THEN DO ;    LSS = 0 ;  GOTO  RR ;   END ;          SMI00440
         GOTO R1 ;     END ;                                            SMI00450
  ON ENDFILE(DIN) BEGIN ;                                               SMI00460
        CLOSE FILE(DIN) ;                                               SMI00470
        OPEN  FILE(DIN);                                                SMI00480
        LSW = LSW + 1 ;                                                 SMI00490
        IF LSW = 1 THEN  DO ;                                           SMI00500
                                SSANUP = '$$$' ;                        SMI00510
                                GOTO RR ;      END ;                    SMI00520
           CALL TAB3_RTN ;                                              SMI00530
           ASAN(1) = '***' ;                                            SMI00540
                    CALL WRT3_RTN ;                                     SMI00550
```

```
                    CALL WRT4_RTN ;                                      SMI00560
                    CALL WRT6_RTN ;                                      SMI00570
                    GOTO LAST ;                                          SMI00580
        END ;                                                            SMI00590
2 : READ FILE (DIN2) INTO(IN) ;                                          SMI00600
        NN = NN + 1 ;                                                    SMI00610
     IF NN = 1 THEN DO ;                                                 SMI00620
     IF SUBSTR(YYMM,3,2) ='03' | SUBSTR(YYMM,3,2) = '06' |               SMI00630
        SUBSTR(YYMM,3,2) ='09' | SUBSTR(YYMM,3,2) = '12' THEN B = 37;    SMI00640
     ELSE B = 36 ;                                                       SMI00650
       PUT DATA(B) ;                                                     SMI00660
      END ;                                                              SMI00670
        AA(NN,1) =  (GISU(B-3)+GISU(B-4)+GISU(B-5)) / 3  ;               SMI00680
        AA(NN,2) =  (GISU(B)+GISU(B-1)+GISU(B-2)) / 3  ;                 SMI00690
        IF AA(NN,1) = 0 THEN  AA(NN,3) = 0 ;                             SMI00700
    ELSE  AA(NN,3) =  AA(NN,2) / AA(NN,1) * 100 - 100 ;                  SMI00710
    GGGG = (GISU(B-12)+GISU(B-13)+GISU(B-14)) / 3 ;                      SMI00720
    AA(NN,4) = AA(NN,2) / GGGG * 100 - 100 ;                             SMI00730
       GOTO R2 ;                                                         SMI00740
*111111111111111111  FRIST  READ START 111111111111111111*/              SMI00750
1 : READ FILE(DIN) INTO(IN);                                             SMI00760
    IF CARD ¬= '4'  THEN GOTO R1 ;                                       SMI00770
    IF GUBUN ¬= '1' THEN GOTO R1 ;                                       SMI00780
        IF SSANUP ¬= SANUP THEN  DO ;                                    SMI00790
            N = N + 1 ;                                                  SMI00800
            ASAN(N) = SANUP ;                                            SMI00810
            SSANUP  = SANUP ;    END ;                                   SMI00820
    GOTO R1 ;                                                            SMI00830
*1111111END FRIST READ  &  SECOND READ START 2222222222222*/             SMI00840
************ END SECOND READ & THREE READ START **************/          SMI00850
R : READ FILE(DIN) INTO(IN);                                             SMI00860
    IF CARD ¬= '4' THEN GOTO RR ;                                        SMI00870
     MUL = MMM ;                                                         SMI00880
    IF SSANUP ¬= SANUP THEN  DO ;                                        SMI00890
            IF W >= 1 THEN CALL TAB3_RTN ;                               SMI00900
            W = W + 1 ;                                                  SMI00910
            SSANUP = SANUP ; END ;                                       SMI00920
        IF GUBUN = '1' THEN DO ;                                         SMI00930
             IF BUN = '2' THEN DO ;                                      SMI00940
             TAB3(W,1) = (MUL(B-12) + MUL(B-13) + MUL(B-14)) / 3;        SMI00950
             TAB3(W,4) = (MUL(B-3)+MUL(B-4)+MUL(B-5)) / 3  ;             SMI00960
             TAB3(W,7) = (MUL(B)+MUL(B-1)+MUL(B-2)) / 3  ;               SMI00970
                              END ;                                      SMI00980
        ELSE IF BUN = '3' THEN DO ;                                      SMI00990
             TAB3(W,10)  = (GISU(B)+GISU(B-1)+GISU(B-2)) / 3;            SMI01000
             TAB3(W,2) = (MUL(B-3)+MUL(B-4)+MUL(B-5)) / 3 ;              SMI01010
             TAB3(W,5) = (MUL(B)+MUL(B-1)+MUL(B-2)) / 3 ;                SMI01020
        GGGG = (MUL(B-12)+MUL(B-13)+MUL(B-14)) / 3 ;                     SMI01030
       IF GGGG = 0 THEN TAB3(W,12) = 0;                                  SMI01040
      ELSE TAB3(W,12) = TAB3(W,5) / GGGG * 100 - 100 ;                   SMI01050
                              END ;                                      SMI01060
        ELSE IF BUN = '4' THEN DO ;                                      SMI01070
                TAB3(W,13)  = (GISU(B)+GISU(B-1)+GISU(B-2)) / 3  ;       SMI01080
                TAB3(W,3) = (MUL(B-3)+MUL(B-4)+MUL(B-5)) / 3 ;           SMI01090
                TAB3(W,6) = (MUL(B)+MUL(B-1)+MUL(B-2)) / 3 ;             SMI01100
```

```
      GGGG = (MUL(B-12)+MUL(B-13)+MUL(B-14)) / 3 ;                    SMI01110
      IF GGGG = 0 THEN TAB3(W,15) = 0;                                SMI01120
    ELSE TAB3(W,15) = TAB3(W,6) / GGGG * 100 - 100 ;                  SMI01130
                              END ;                                   SMI01140
      ELSE IF BUN = '5' THEN DO ;                                     SMI01150
               TAB6(1,W,7) = WEIGHT ;                                 SMI01160
               TAB6(1,W,8) = (GISU(B-3)+GISU(B-4)+GISU(B-5)) / 3 ;    SMI01170
               TAB6(1,W,9) = (GISU(B)+GISU(B-1)+GISU(B-2)) / 3 ;      SMI01180
            IF TAB6(1,W,8) = 0 THEN TAB6(1,W,10) = 0 ;                SMI01190
        ELSE   TAB6(1,W,10) = TAB6(1,W,9) / TAB6(1,W,8) * 100 - 100;SMI01200
           GGGG = (GISU(B-12)+GISU(B-13)+GISU(B-14)) / 3 ;            SMI01210
            IF GGGG = 0 THEN TAB6(1,W,11) = 0;                        SMI01220
        ELSE   TAB6(1,W,11) = TAB6(1,W,9) / GGGG * 100 - 100 ;        SMI01230
                              END ;                                   SMI01240
      ELSE IF BUN = '6' THEN DO ;                                     SMI01250
          TAB6(1,W,1) = WEIGHT ;                                      SMI01260
          TAB6(1,W,2) = (GISU(B-12)+GISU(B-13)+GISU(B-14)) / 3 ;      SMI01270
          TAB6(1,W,3) = (GISU(B-3)+GISU(B-4)+GISU(B-5)) / 3 ;         SMI01280
          TAB6(1,W,4) = (GISU(B)+GISU(B-1)+GISU(B-2)) / 3 ;           SMI01290
      IF TAB6(1,W,3) = 0 THEN TAB6(1,W,5) = 0 ;                       SMI01300
    ELSE TAB6(1,W,5) = TAB6(1,W,4) / TAB6(1,W,3) * 100 - 100 ;        SMI01310
      IF TAB6(1,W,2) = 0 THEN TAB6(1,W,6) = 0 ;                       SMI01320
    ELSE TAB6(1,W,6) = TAB6(1,W,4) / TAB6(1,W,2) * 100 - 100 ;        SMI01330
                              END ;                                   SMI01340
      ELSE IF BUN = '7' THEN DO ;                                     SMI01350
          TAB6(2,W,7) = WEIGHT ;                                      SMI01360
          TAB6(2,W,8) = (GISU(B-3)+GISU(B-4)+GISU(B-5)) / 3 ;         SMI01370
          TAB6(2,W,9) = (GISU(B)+GISU(B-1)+GISU(B-2)) / 3 ;           SMI01380
      IF TAB6(2,W,8) = 0 THEN TAB6(2,W,10) = 0 ;                      SMI01390
    ELSE TAB6(2,W,10) = TAB6(2,W,9) / TAB6(2,W,8) * 100 -100 ;        SMI01400
      GGGG  = (GISU(B-12)+GISU(B-13)+GISU(B-14)) / 3 ;                SMI01410
      IF GGGG = 0 THEN TAB6(2,W,11) = 0 ;                             SMI01420
    ELSE TAB6(2,W,11) = TAB6(2,W,9) / GGGG * 100  - 100 ;             SMI01430
                              END ;                                   SMI01440
      ELSE IF BUN = '8' THEN DO ;                                     SMI01450
          TAB6(2,W,1) = WEIGHT ;                                      SMI01460
          TAB6(2,W,2) = (GISU(B-12)+GISU(B-13)+GISU(B-14)) / 3 ;      SMI01470
          TAB6(2,W,3) = (GISU(B-3)+GISU(B-4)+GISU(B-5)) / 3 ;         SMI01480
          TAB6(2,W,4) = (GISU(B)+GISU(B-1)+GISU(B-2)) / 3 ;           SMI01490
          TAB6(2,W,5) = TAB6(2,W,4) / TAB6(2,W,3) * 100 - 100;        SMI01500
          TAB6(2,W,6) = TAB6(2,W,4) / TAB6(2,W,2) * 100 - 100;        SMI01510
                              END ;                                   SMI01520
                        END ;                                         SMI01530
  ELSE IF GUBUN = '2' THEN DO ;                                       SMI01540
          TAB4(1,W,1) = WEIGHT ;                                      SMI01550
      IF BUN = '4' THEN DO ;                                          SMI01560
  IF W = 1 THEN TJYGISU21 = (GISU(B-12)+GISU(B-13)+GISU(B-14)) / 3;   SMI01570
    TAB4(1,W,2) = (MUL(B-3)+MUL(B-4)+MUL(B-5)) / 3 ;                  SMI01580
    TAB4(1,W,3) = (GISU(B-3)+GISU(B-4)+GISU(B-5)) / 3 ;               SMI01590
    TAB4(1,W,4) = (MUL(B)+MUL(B-1)+MUL(B-2)) / 3 ;                    SMI01600
    TAB4(1,W,5) = (GISU(B)+GISU(B-1)+GISU(B-2)) / 3 ;                 SMI01610
   IF TAB4(1,W,3) = 0 THEN TAB4(1,W,6) = 0 ;                          SMI01620
   ELSE  TAB4(1,W,6) = TAB4(1,W,5) / TAB4(1,W,3) * 100 - 100 ;        SMI01630
   GGGG =  (GISU(B-12) + GISU(B-13)+GISU(B-14)) / 3 ;                 SMI01640
      IF GGGG = 0 THEN TAB4(1,W,7) = 0 ;                              SMI01650
```

```
      ELSE  TAB4(1,W,7) = TAB4(1,W,5) / GGGG * 100 - 100 ;                SMI01660
            TAB4(1,W,8) = ((TAB4(1,W,5) - GGGG) * TAB4(1,W,1))            SMI01670
                        / (TJYGISU21 * TAB4(1,1,1)) * 100 ;               SMI01680
                                  END ;                                   SMI01690
        ELSE IF BUN = '5' THEN DO ;                                       SMI01700
    IF W = 1 THEN TJYGISU22 = (GISU(B-12)+GISU(B-13)+GISU(B-14)) / 3;     SMI01710
            TAB4(1,W,9) = (MUL(B-3)+MUL(B-4)+MUL(B-5)) / 3 ;              SMI01720
            TAB4(1,W,10) = (GISU(B-3)+GISU(B-4)+GISU(B-5)) / 3 ;          SMI01730
            TAB4(1,W,11) = (MUL(B)+MUL(B-1)+MUL(B-2)) / 3 ;               SMI01740
            TAB4(1,W,12) = (GISU(B)+GISU(B-1)+GISU(B-2)) / 3 ;            SMI01750
          IF TAB4(1,W,10) = 0 THEN TAB4(1,W,13) = 0;                      SMI01760
            TAB4(1,W,13) = TAB4(1,W,12) / TAB4(1,W,10) * 100 - 100 ;      SMI01770
            GGGG = (GISU(B-12)+GISU(B-13)+GISU(B-14)) / 3 ;               SMI01780
          IF GGGG = 0 THEN TAB4(1,W,14) = 0 ;                             SMI01790
            TAB4(1,W,14) = TAB4(1,W,12) / GGGG  * 100 - 100 ;             SMI01800
            TAB4(1,W,15) = ((TAB4(1,W,12) - GGGG) * TAB4(1,W,1))          SMI01810
                        / (TJYGISU22 * TAB4(1,1,1)) * 100 ;               SMI01820
                    END ;         END ;                                   SMI01830
    ELSE DO ;                                                             SMI01840
             TAB4(2,W,1) = WEIGHT ;                                       SMI01850
        IF BUN = '4' THEN DO ;                                            SMI01860
    IF W = 1 THEN JYGISU21 = (GISU(B-12)+GISU(B-13)+GISU(B-14)) / 3;      SMI01870
            TAB4(2,W,2) = (MUL(B-3)+MUL(B-4)+MUL(B-5)) / 3 ;              SMI01880
            TAB4(2,W,3) = (GISU(B-3)+GISU(B-4)+GISU(B-5)) / 3 ;           SMI01890
            TAB4(2,W,4) = (MUL(B)+MUL(B-1)+MUL(B-2)) / 3 ;                SMI01900
            TAB4(2,W,5) = (GISU(B)+GISU(B-1)+GISU(B-2)) / 3 ;             SMI01910
          IF TAB4(2,W,3) = 0 THEN TAB4(2,W,6) = 0 ;                       SMI01920
      ELSE  TAB4(2,W,6) = TAB4(2,W,5) / TAB4(2,W,3) * 100 - 100 ;         SMI01930
          GGGG = (GISU(B-12)+GISU(B-13)+GISU(B-14)) / 3 ;                 SMI01940
          IF GGGG = 0 THEN TAB4(2,W,7) = 0 ;                              SMI01950
      ELSE  TAB4(2,W,7) = TAB4(2,W,5) / GGGG   * 100 - 100 ;              SMI01960
            TAB4(2,W,8) = ((TAB4(2,W,5) - GGGG) * TAB4(2,W,1))            SMI01970
                        / (JYGISU21 * TAB4(2,1,1)) * 100 ;                SMI01980
                                  END ;                                   SMI01990
        ELSE IF BUN = '5' THEN DO ;                                       SMI02000
    IF W = 1 THEN JYGISU22 = (GISU(B-12)+GISU(B-13)+GISU(B-14)) / 3;      SMI02010
            TAB4(2,W,9) = (MUL(B-3)+MUL(B-4)+MUL(B-5)) / 3 ;              SMI02020
            TAB4(2,W,10) = (GISU(B-3)+GISU(B-4)+GISU(B-5)) / 3 ;          SMI02030
            TAB4(2,W,11) = (MUL(B)+MUL(B-1)+MUL(B-2)) / 3 ;               SMI02040
            TAB4(2,W,12) = (GISU(B)+GISU(B-1)+GISU(B-2)) / 3 ;            SMI02050
          IF TAB4(2,W,10) = 0 THEN TAB4(2,W,13) = 0 ;                     SMI02060
      ELSE TAB4(2,W,13) = TAB4(2,W,12) / TAB4(2,W,10) * 100 - 100 ;       SMI02070
          GGGG =  (GISU(B-12)+GISU(B-13)+GISU(B-14)) / 3 ;                SMI02080
          IF GGGG = 0 THEN TAB4(2,W,14) = 0 ;                             SMI02090
      ELSE TAB4(2,W,14) = TAB4(2,W,12) / GGGG * 100 - 100 ;               SMI02100
            TAB4(2,W,15) = ((TAB4(2,W,12) - GGGG) * TAB4(2,W,1))          SMI02110
                        / (JYGISU22 * TAB4(2,1,1)) * 100 ;                SMI02120
                                  END ;                                   SMI02130
                END ;                                                     SMI02140
    GOTO RR ;                                                             SMI02150
TAB3_RTN : PROC ;                                                         SMI02160
            TAB3(W,8) = TAB3(W,7) - TAB3(W,4) ;                           SMI02170
            TAB3(W,9) = TAB3(W,7) - TAB3(W,1) ;                           SMI02180
            IF TAB3(W,2) = 0 THEN TAB3(W,11) = 0 ;                        SMI02190
     ELSE   TAB3(W,11) = TAB3(W,5) / TAB3(W,2) * 100 - 100 ;              SMI02200
```

```
            IF TAB3(W,3) = 0 THEN TAB3(W,14) = 0 ;                      SMI02210
      ELSE  TAB3(W,14) = TAB3(W,6) / TAB3(W,3) * 100 - 100 ;            SMI02220
            IF TAB3(W,4) = 0 THEN TAB3(W,16) = 0 ;                      SMI02230
      ELSE  TAB3(W,16) = TAB3(W,7) / TAB3(W,4) * 100 - 100 ;            SMI02240
            IF TAB3(W,1) = 0 THEN TAB3(W,17) = 0 ;                      SMI02250
      ELSE  TAB3(W,17) = TAB3(W,7) / TAB3(W,1) * 100 - 100 ;            SMI02260
  END TAB3_RTN ;                                                        SMI02270
  WRT3_RTN : PROC ;                                                     SMI02280
      CALL HEAD3 ;                                                      SMI02290
     DO I = 1 TO W ;                                                    SMI02300
            PUT SKIP(2) EDIT(ASAN(I),(TAB3(I,J) DO J = 1 TO 17))        SMI02310
               (COL(1),A,F(10),(2)((2) F(7),F(10)),(2) F(8),            SMI02320
                (7) F(6,1),F(8,1),F(6,1));                              SMI02330
     END ;                                                              SMI02340
  END WRT3_RTN ;                                                        SMI02350
  WRT4_RTN : PROC ;                                                     SMI02360
     DO I = 1 TO 2 ;                                                    SMI02370
         CALL HEAD4 ;                                                   SMI02380
         DO J = 1 TO W;                                                 SMI02390
            PUT SKIP(2) EDIT(ASAN(J),(TAB4(I,J,K) DO K = 1 TO 15))      SMI02400
               (COL(2),A(5),F(8,1),  2 (F(10), F(7,1)),3 F(7,1) ,       SMI02410
               X(3),2 (F(10), F(7,1)),3 F(7,1))  ;                      SMI02420
     END ;    END ;                                                     SMI02430
  END WRT4_RTN ;                                                        SMI02440
  WRT6_RTN : PROC ;                                                     SMI02450
     DO I = 1 TO 2 ;                                                    SMI02460
         CALL HEAD6 ;                                                   SMI02470
         DO J = 1 TO W ;                                                SMI02480
         PUT SKIP(2) EDIT(ASAN(J),(TAB6(I,J,L) DO L = 1 TO 6),          SMI02490
            (AA(J,L) DO L = 1 TO 4),(TAB6(I,J,L) DO L = 7 TO 11))       SMI02500
               (COL(2),A(3),F(8,1),(5) F(7,1), 2 F(8,1),                SMI02510
                F(7,1),F(7,1),F(12,1),2 F(7,1),2 F(9,1)) ;              SMI02520
     END ;   END ;                                                      SMI02530
  END WRT6_RTN ;                                                        SMI02540
 HEAD3 : PROC ;                                                         SMI02550
            PUT PAGE ;                                                  SMI02560
            PUT            SKIP(3) EDIT ('#3  KOYONG DONG HAING',       SMI02570
                           'YY / ',YY,'    MM / ',MM)                   SMI02580
                           ( X(52),A,COL(110),A,A(2),A,A) ;             SMI02590
            PUT            SKIP(1) EDIT((21)'_') (X(52),A(21)) ;        SMI02600
            PUT            SKIP(2) EDIT((130)'_') (COL(1),A(130)) ;     SMI02610
            PUT      SKIP(2) EDIT(      '  INC. DEC    ACC.INC.DEC(%)   SMI02620
 SEP.INC.DEC(%)    EMP.INC.DEC(%) ') ( COL(66),A) ;                     SMI02630
            PUT   SKIP(1) EDIT('SAN  JAN.',(66)'_')                     SMI02640
                           (A,COL(65),A(66));                           SMI02650
            PUT            SKIP(2) EDIT('    DONG.EMP.  J.ACC  J.SEP    SMI02660
   J.EMP  G.ACC  G.SEP   GUM EMP    PM-1   PM-2  PM-1  PM-2  PM-3   PM-SMI02670
1  PM-2  PM-3  PM-1    PM-2 ') (COL(1),A) ;                             SMI02680
            PUT            SKIP(1) EDIT((130)'_') (COL(1),A(130)) ;     SMI02690
 END HEAD3 ;                                                            SMI02700
 HEAD4 : PROC ;                                                         SMI02710
            PUT PAGE ;                                                  SMI02720
            PUT            SKIP(3) EDIT ('#4  NOMIAL & REAL WAGE INDEXSMI02730
','YY / ',YY,'    MM / ',MM)                                            SMI02740
                           ( X(52),A,COL(110),A,A(2),A,A) ;             SMI02750
```

```
        IF I = 1 THEN PUT SKIP(0) EDIT('( G. TOTAL )') (COL(91),A) ;    SMI02760
             ELSE PUT SKIP(0) EDIT('( CONTRACTUAL )') (COL(91),A) ;     SMI02770
             PUT            SKIP(1) EDIT((34)'_') (X(52),A(34)) ;      SMI02780
             PUT            SKIP(1) EDIT((129)'_') (X(1),A(129)) ;     SMI02790
             PUT            SKIP(2) EDIT('NOMIRAL   WAGE') (COL(30),A);SMI02800
             PUT            SKIP(0) EDIT('REAL   WAGE') (COL(93),A);   SMI02810
             PUT            SKIP(1) EDIT((116)'_') (COL(15),A(116)) ;  SMI02820
             PUT            SKIP(1) EDIT('SAN   WEIGHT         PM - 1  SMI02830
         P   M          INC.DEC(%)                PM - 1           P  MSMI02840
            INC.DEC(%)')              (X(1),A) ;                        SMI02850
             PUT            SKIP(1) EDIT((116)'_') (COL(15),A(116)) ;  SMI02860
             PUT            SKIP(1) EDIT('WAGE   INDEX     WAGE   INDEXSMI02870
   PM-1   PM-2   PM-3        WAGE   INDEX     WAGE   INDEX   PM-1   PM-SMI02880
 2   PM-3')                     (COL(20),A) ;                           SMI02890
             PUT            SKIP(1) EDIT((129)'_') (X(1),A(129)) ;     SMI02900
  END HEAD4 ;                                                           SMI02910
  HEAD6 : PROC ;                                                        SMI02920
             PUT PAGE ;                                                 SMI02930
             PUT            SKIP(3) EDIT ('#6  UPJONGBYEL  KOANGKONGUP SMI02940
  SANGSANSENG JIGU','YY / ',YY,'    MM / ',MM)                          SMI02950
                            ( X(43),A,COL(110),A,A(2),A,A) ;            SMI02960
        IF I = 1   THEN PUT SKIP(0) EDIT('( MAN . DAY )') (COL(91),A); SMI02970
             ELSE PUT SKIP(0) EDIT('( MAN . HOUR )') (COL(91),A) ;      SMI02980
             PUT            SKIP(0) EDIT((47)'_') (X(43),A(47)) ;      SMI02990
             PUT            SKIP(2) EDIT((124)'_') (X(1),A(124)) ;     SMI03000
             PUT            SKIP(2) EDIT('LABOR  PRODUTIVE  INDEX      SMI03010
                OUT-PUT  INDEX                    IN-PUT  INDEX')      SMI03020
                            (COL(17),A) ;                               SMI03030
             PUT            SKIP(1) EDIT((121)'_') (COL(5),A(121)) ;   SMI03040
             PUT            SKIP(1) EDIT('SAN                          SMI03050
       INC. DEC(%)                  INC. DEC(%)                        SMI03060
       INC. DEC(%)   ')          (COL(2),A) ;                          SMI03070
             PUT            SKIP(1) EDIT((15)'_',(15)'_',(20)'_')      SMI03080
                     (COL(35),A(16),COL(66),A(16),COL(106),A(20)) ;    SMI03090
             PUT            SKIP(1) EDIT('    WEIGHT    PY    PM-1    PSMI03100
 M    PM-1   PM-2     PM-1    PM    PM-1   PM-2      WEIGHT   PM-1     SMI03110
 PM     PM-1    PM-2  ')          (COL(2),A) ;                         SMI03120
             PUT        SKIP(1) EDIT((124)'_') (COL(2),A(124)) ;       SMI03130
  END HEAD6 ;                                                           SMI03140
  LAST  :                                                               SMI03150
          CLOSE   FILE(DIN),FILE(DIN2), FILE(SYSPRINT);                 SMI03160
          END MON_TAB ;                                                 SMI03170
/*                                                                      SMI03180
//GO.DIN  DD DSN=SMISM12,DISP=SHR                                       SMI03190
//GO.DIN2 DD UNIT=DISK,DSN=SMICD235,DCB=(RECFM=FB,BLKSIZE=3200,         SMI03200
//        LRECL=320),DISP=(OLD,KEEP),VOL=SER=BOSWK2                     SMI03210
//GO.SYSPRINT DD SYSOUT=A                                               SMI03220
//GO.SYSOUT DD SYSOUT=A                                                 SMI03230
/*                                                                      SMI03240
//                                                                      SMI03250
```

SMIWC366

생산능력 분기별 TABLE

FILE: SMIWC366 6BUNGI D1 VM/SP RELEASE 2.1 EXPRESS PUT8302+ SLU203

```
//SMIWC366 JOB CLASS=1,TYPRUN=HOLD                                      SMI00010
//JOBCAT  DD  DSN=SMICT12,DISP=SHR                                      SMI00020
//STEP1 EXEC PLIFCLG                                                    SMI00020
//PLI.SYSIN DD *                                                        SMI00040
* PROCESS S,GS,NEXT,XREF,A,OPT(2) ;                                      SMI00050
  IDENX:        PROC      OPTIONS(MAIN) ;                               SMI00060
                DCL  INF  FILE  RECORD  INPUT ;                         SMI00070
                DCL  1    IN,                                           SMI00080
                     2    CARD   CHAR(1),                               SMI00090
                     2    GUBUN  CHAR(1),                               SMI00100
                     2    SANUP  CHAR(3),                               SMI00110
                     2    F1     CHAR(5),                               SMI00120
                     2    F2     CHAR(1),                               SMI00130
                     2    YYM    PIC '99',                              SMI00140
                     2    YMM    PIC '99',                              SMI00140
                     2    WEIGHT FLOAT,                                 SMI00150
                     2    POINT  FIXED  BIN(31),                        SMI00160
                     2    DATA(37),                                     SMI00170
                       3  MUL    FIXED  BIN(31),                        SMI00180
                       3  GISU   FLOAT,                                 SMI00190
                     2    F3     CHAR(1) ;                              SMI00200
                 DCL MMM(37)     FLOAT ;                                SMI00210
                 DCL  (SW,I,J,K)          PIC'99'  INIT(0) ;            SMI00220
                 DCL  (CCC1,CCC2,CCC3)    FLOAT;                        SMI00220
                 DCL  (LSW,MMYY)          PIC'99'  INIT(1) ;            SMI00230
                 DCL  WSANUP            CHAR(3)  INIT((3)' ') ;         SMI00240
                 DCL  SGU               CHAR(1)  INIT((1)' ') ;         SMI00250
                 DCL CTAB(10)      FLOAT DEC(16) INIT((10)0) ;          SMI00260
                 DCL DTAB(12)      FLOAT DEC(16) INIT((12)0) ;          SMI00270
          DCL (J1,J2,J3,JT1,JT2,JT3,TWEIGHT)    FLOAT DEC(16) INIT(0) ; SMI00280
           OPEN FILE(INF);                                              SMI00290
           OPEN FILE(SYSPRINT) LINESIZE(132) ;                          SMI00300
           ON   ENDFILE(INF)  BEGIN ;                                   SMI00310
                     CLOSE FILE(INF) ;                                  SMI00320
                     OPEN  FILE(INF) ;                                  SMI00330
                     LSW = LSW + 1 ;                                    SMI00340
                     IF LSW < 5 THEN GO TO RD ;                         SMI00350
                     GO TO  LAST ;   END ;                              SMI00360
  RD:      READ FILE(INF)   INTO(IN) ;                                  SMI00370
           IF   CARD ¬= '6'   THEN GO TO RD ;                           SMI00380
           MMYY=YMM; IF MMYY = 1 THEN MMYY = 13;
           K = 14 - MMYY;
           JN=MOD(MMYY,3);
           MMM = MUL ;                                                  SMI00390
           IF   GUBUN ¬= SGU   THEN  DO ;                               SMI00400
                    TWEIGHT = WEIGHT ;                                  SMI00410
        J1  = (GISU(32-JN) + GISU(33-JN) + GISU(34-JN)) / 3 ;           SMI00420
        J2  = (GISU(23-JN) + GISU(24-JN) + GISU(25-JN)) / 3 ;           SMI00430
        JT1 = (MMM(32-JN) + MMM(33-JN) + MMM(34-JN)) / 3 ;              SMI00450
        JT2 = (MMM(23-JN) + MMM(24-JN) + MMM(25-JN)) / 3 ;              SMI00460
        JT3 = (MMM(23-JN) + MMM(24-JN) + MMM(25-JN)) / 3 ;              SMI00470
           CCC3 = 0;
           DO I = 38 - MMYY TO 37 - JN;
              CCC3 = CCC3 + GISU(I-12);
           END;
```

```
                    SGU = GUBUN ;    PUT PAGE ;                          SMI00480
                                END ;                                    SMI00490
        IF LSW = 1 THEN DO ;                                             SMI00500
        IF   F1 = (5)' '  THEN  GO  TO  RD ;                             SMI00510
        CCC1 = 0;
        CCC2 = 0;
        DO I = 38 - MMYY TO 37 - JN;
           CCC1 = CCC1 + GISU(I);
           CCC2 = CCC2 + GISU(I-12);
        END;
        CTAB(1) = WEIGHT ;                                               SMI00520
        CTAB(2) = (GISU(23-JN) + GISU(24-JN) + GISU(25-JN)) / 3 ;        SMI00530
        CTAB(3) = (GISU(32-JN) + GISU(33-JN) + GISU(34-JN)) / 3 ;        SMI00540
        CTAB(4) = (GISU(35-JN) + GISU(36-JN) + GISU(37-JN)) / 3 ;        SMI00550
         IF CTAB(3)=0 THEN CTAB(5)=0;
ELSE    CTAB(5) = CTAB(4) / CTAB(3) * 100 - 100 ;                        SMI00560
          IF CTAB(2)=0 THEN CTAB(6)=0;
ELSE     CTAB(6) = CTAB(4) / CTAB(2) * 100 - 100 ;                       SMI00570
        CTAB(7) = CCC1/CCC2*100-100 ;                                    SMI00580
   CTAB(8) = ((CTAB(4) - CTAB(3)) * WEIGHT ) / (J1 * TWEIGHT ) * 100;    SMI00590
   CTAB(9) = ((CTAB(4) - CTAB(2)) * WEIGHT ) / (J2 * TWEIGHT ) * 100;    SMI00600
   CTAB(10) = ((CCC1-CCC2) * WEIGHT)/(CCC3*100);                         SMI00580
       PUT SKIP(2) EDIT(SANUP,F1,GUBUN,(CTAB(I) DO I = 1 TO 10 ))        SMI00620
             (A(5),A(7),A(3), 10 F(9,1)) ;                               SMI00630
       CTAB = 0 ;                                                        SMI00640
       END ;                                                             SMI00650
       ELSE IF LSW = 2 THEN DO ;                                         SMI00660
        IF   F1 ¬= (5)' '  THEN  GO  TO  RD ;                            SMI00670
        DTAB(1) = WEIGHT ;                                               SMI00680
        CCC1 = 0;
        CCC2 = 0;
        DO I = 38 - MMYY TO 37 - JN;
           CCC1 = CCC1 + GISU(I);
           CCC2 = CCC2 + GISU(I-12);
        END;
       DO I = K  TO K+11;                                                SMI00690
        DTAB(2) = DTAB(2) + GISU(I) ;                                    SMI00700
        DTAB(3) = DTAB(3) + GISU(I+12) ;                                 SMI00710
       END ;                                                             SMI00720
        DTAB(2) = DTAB(2) / 12 ;                                         SMI00730
        DTAB(3) = DTAB(3) / 12 ;                                         SMI00740
        DTAB(4) = (GISU(23-JN) + GISU(24-JN) + GISU(25-JN)) / 3 ;        SMI00750
        DTAB(5) = (GISU(32-JN) + GISU(33-JN) + GISU(34-JN)) / 3 ;        SMI00760
        DTAB(6) = (GISU(35-JN) + GISU(36-JN) + GISU(37-JN)) / 3 ;        SMI00770
        DTAB(7) = DTAB(6) / DTAB(5) * 100 - 100 ;                        SMI00780
        DTAB(8) = DTAB(6) / DTAB(4) * 100 - 100 ;                        SMI00790
        DTAB(9) = CCC1    / CCC2    * 100 - 100 ;                        SMI00800
   DTAB(10) = ((DTAB(6) - DTAB(5)) * WEIGHT ) / (J1 * TWEIGHT ) * 100;SMI00810
   DTAB(11) = ((DTAB(6) - DTAB(4)) * WEIGHT ) / (J2 * TWEIGHT ) * 100;SMI00820
   DTAB(12) = ((CCC1 - CCC2) * WEIGHT ) / (CCC3 * TWEIGHT ) * 100;       SMI00830
       PUT SKIP(2) EDIT(SANUP,GUBUN,(DTAB(I) DO I = 1 TO 12 ))           SMI00840
             (A(5),A(3), 12 F(9,1)) ;                                    SMI00850
       DTAB = 0 ;                                                        SMI00860
       END ;                                                             SMI00870
   ELSE IF LSW = 3 THEN DO ;                                             SMI00880
```

```
        IF   F1 = (5)' '  THEN  GO  TO  RD ;                          SMI00890
        IF   GUBUN ¬= '3' THEN  GO  TO  RD ;                          SMI00900
        CTAB(1) = WEIGHT ;                                            SMI00910
        CTAB(2) = (MMM(23-JN) + MMM(24-JN) + MMM(25-JN)) / 3 ;        SMI00920
        CTAB(3) = (MMM(32-JN) + MMM(33-JN) + MMM(34-JN)) / 3 ;        SMI00930
        CTAB(4) = (MMM(35-JN) + MMM(36-JN) + MMM(37-JN)) / 3 ;        SMI00940
    IF CTAB(3)=0 THEN CTAB(5)=0;
    ELSE  CTAB(5) = CTAB(4) / CTAB(3) * 100 - 100 ;                   SMI00950
    IF CTAB(2)=0 THEN CTAB(6)=0;
    ELSE  CTAB(6) = CTAB(4) / CTAB(2) * 100 - 100 ;                   SMI00960
    IF CTAB(2)=0 THEN CTAB(7)=0;
    ELSE  CTAB(7) = CTAB(4) / CTAB(2) * 100 - 100 ;                   SMI00970
    CTAB(8) = ((CTAB(4) - CTAB(3)) * WEIGHT ) / (JT1 * TWEIGHT ) * 100; SMI00980
    CTAB(9) = ((CTAB(4) - CTAB(2)) * WEIGHT ) / (JT2 * TWEIGHT ) * 100; SMI00990
    CTAB(10) = ((CTAB(4) - CTAB(2)) * WEIGHT ) / (JT3 * TWEIGHT ) * 100;SMI01000
      DO   I = 1 TO 10 ;                                              SMI01010
           CTAB(I) = CTAB(I) / 10 ;                                   SMI01020
       END ;                                                          SMI01030
         PUT SKIP(2) EDIT(SANUP,F1,GUBUN,(CTAB(I) DO I = 1 TO 10 ))   SMI01040
               (A(5),A(7),A(3), 10 F(9,1)) ;                          SMI01050
         CTAB = 0 ;                                                   SMI01060
      END ;                                                           SMI01070
     ELSE DO ;                                                        SMI01080
        IF   F1 ¬= (5)' '  THEN  GO  TO  RD ;                         SMI01090
        IF   GUBUN ¬= '3' THEN  GO  TO  RD ;                          SMI01100
        DTAB(1) = WEIGHT ;                                            SMI01110
       DO I = K TO K+11 ;                                             SMI01120
        DTAB(2) = DTAB(2) + MMM(I) ;                                  SMI01130
        DTAB(3) = DTAB(3) + MMM(I+12) ;                               SMI01140
       END ;                                                          SMI01150
        DTAB(2) = DTAB(2) / 12 ;                                      SMI01160
        DTAB(3) = DTAB(3) / 12 ;                                      SMI01170
        DTAB(4) = (MMM(23-JN) + MMM(24-JN) + MMM(25-JN)) / 3 ;        SMI01180
        DTAB(5) = (MMM(32-JN) + MMM(33-JN) + MMM(34-JN)) / 3 ;        SMI01190
        DTAB(6) = (MMM(35-JN) + MMM(36-JN) + MMM(37-JN)) / 3 ;        SMI01200
        DTAB(7) = DTAB(6) / DTAB(5) * 100 - 100 ;                     SMI01210
        DTAB(8) = DTAB(6) / DTAB(4) * 100 - 100 ;                     SMI01220
        DTAB(9) = DTAB(6) / DTAB(4) * 100 - 100 ;                     SMI01230
    DTAB(10) = ((DTAB(6) - DTAB(5)) * WEIGHT ) / (JT1 * TWEIGHT ) * 100;SMI01240
    DTAB(11) = ((DTAB(6) - DTAB(4)) * WEIGHT ) / (JT2 * TWEIGHT ) * 100;SMI01250
    DTAB(12) = ((DTAB(6) - DTAB(4)) * WEIGHT ) / (JT3 * TWEIGHT ) * 100;SMI01260
      DO   I = 1 TO 12 ;                                              SMI01270
           DTAB(I) = DTAB(I) / 10 ;                                   SMI01280
      END ;                                                           SMI01290
         PUT SKIP(2) EDIT(SANUP,GUBUN,(DTAB(I) DO I = 1 TO 12 ))      SMI01300
               (A(5),A(3), 12 F(9,1)) ;                               SMI01310
         DTAB = 0 ;                                                   SMI01320
      END ;                                                           SMI01330
         GOTO  RD ;                                                   SMI01340
     LAST:                                                            SMI01350
              CLOSE FILE(INF),FILE(SYSPRINT) ;                        SMI01360
              END IDENX ;                                             SMI01370
/*                                                                    SMI01380
//GO.SYSPRINT DD SYSOUT=A
//GO.INF DD DSN=SMISM12,DISP=SHR                                      SMI00400
```

//

SMI05430

SMIWGM62

BACK-UP (전월과 금월)

UTILITY PROGRAM

FILE: SMIWBK66 JNBACKUP D1 VM/SP RELEASE 2.1 EXPRESS PUT8302+ SLU203

```
//SMIWGM62 JOB CLASS=1,TYPRUN=HOLD                                    SMI00010
//JOBCAT DD DSN=SMICT12,DISP=SHR                                      SMI00020
//ST3 EXEC PGM=IDCAMS                                                 SMI00030
//DIREC DD DSNAME=SMIGM12,DISP=SHR                                    SMI00240
//TA DD UNIT=TAPE,DSN=SMIE312,DCB=(RECFM=FB,LRECL=50,BLKSIZE=3200),   SMI00250
//   DISP=(NEW,KEEP)                                                  SMI00260
//SYSPRINT DD SYSOUT=A
//SYSIN  DD *
         REPRO  INFILE (DIREC)-
               OUTFILE (TA)
//                                                                    SMI00270
```

SMIWSM62

SUM FILE BACK-UP

UTILITY PROGRAM

FILE: SMIWBK66 SMBACKUP D1 VM/SP RELEASE 2.1 EXPRESS PUT8302+ SLU203

```
//SMIWSM62 JOB CLASS=1,TYPRUN=HOLD                                      SMI00010
//JOBCAT DD DSN=SMICT12,DISP=SHR                                        SMI00020
//ST3 EXEC PGM=IDCAMS                                                   SMI00030
//DIREC DD DSNAME=SMISM12,DISP=SHR                                      SMI00240
//TA DD UNIT=TAPE,DSN=SMIE312,DCB=(RECFM=FB,LRECL=320,BLKSIZE=3200),    SMI00250
//   DISP=(NEW,KEEP)                                                    SMI00260
//SYSPRINT DD SYSOUT=A
//SYSIN  DD *
         REPRO  INFILE (DIREC)-
                OUTFILE (TA)
//                                                                      SMI00270
```

SMIWSA62

MASTER BACK-UP

※ SORT PROGRAM이다.

FILE: SMIWBK66 MTBACKUP D1 VM/SP RELEASE 2.1 EXPRESS PUT8302+ SLU203

```
//SMIWSA62 JOB CLASS=1,TYPRUN=HOLD                                  SMI00010
//SORT EXEC PGM=SORT,PARM='SIZE(MAX)'                               SMI00020
//SYSPRINT DD SYSOUT=A                                              SMI00030
//SYSOUT DD SYSOUT=A                                                SMI00040
//SORTIN DD UNIT=DISK,DISP=(OLD,KEEP),DSN=SMIM66,VOL=SER=BOSWK2,    SMI00050
//        DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)                      SMI00060
//SORTOUT DD UNIT=TAPE,DISP=(NEW,KEEP),DSN=KKL20,                   SMI00070
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)                    SMI00080
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(350)),VOL=SER=SORTWK           SMI00090
//SYSIN DD *                                                        SMI00100
  SORT FIELDS=(1,5,A),FORMAT=CH                                     SMI00110
/*                                                                  SMI00120
//                                                                  SMI00130
```

SMICMV12

보정 FILE MOVE

SOURCE PROGRAM

```
//SMICMV12 JOB CLASS=1,TYPRUN=HOLD                                      SMI00010
//  EXEC PLIFCL                                                         SMI00020
//PLI.SYSIN DD *                                                        SMI00030
* PROCESS GS,NEST,A,XREF OPT(2);                                        SMI00040
 SUMFILE:PROC OPTIONS(MAIN);                                            SMI00050
  DCL SUMF FILE RECORD KEYED ENV(VSAM);                                 SMI00060
 DCL 1 SREC,                                                            SMI00070
      2 OKEY        CHAR(10),                                           SMI00080
      2 SPC         CHAR(1),                                            SMI00090
      2 OYEAR       PIC'99',                                            SMI00100
      2 OMONTH      PIC'99',                                            SMI00110
      2 WEIGHT      FLOAT,                                              SMI00120
      2 BASEM  FIXED BIN(31),                                           SMI00130
      2 DATAO(37),                                                      SMI00140
        3 ODATA FIXED BIN(31),                                          SMI00150
        3 JDATA FLOAT,                                                  SMI00160
      2 SIGN        CHAR(1);                                            SMI00170
 DCL ONCODE      BUILTIN;                                               SMI00180
 ON KEY(SUMF) BEGIN;                                                    SMI00190
   IF ONCODE=52 THEN PUT EDIT(OKEY)(A(12));                             SMI00200
 END;  TOT=0;                                                           SMI00210
 ON ENDFILE(SUMF) GOTO LT;                                              SMI00220
 OPEN FILE(SUMF) SEQUENTIAL UPDATE;                                     SMI00230
 RT: READ FILE(SUMF) INTO(SREC);                                        SMI00240
   IF TOT=0 THEN PUT SKIP EDIT(OKEY,OYEAR,OMONTH,(JDATA(I) DO I=1 TO    SMI00250
     12))(A(12),2 P'99',12 F(8,1));                                     SMI00260
   IF OMONTH=12 THEN DO;                                                SMI00270
     OYEAR=OYEAR+1;  OMONTH=1; END;                                     SMI00280
   ELSE OMONTH=OMONTH+1;                                                SMI00290
   DO I=1 TO 36;                                                        SMI00300
     ODATA(I)=ODATA(I+1); JDATA(I)=JDATA(I+1);                          SMI00310
   END;                                                                 SMI00320
   ODATA(37)=0;  JDATA(37)=0;                                           SMI00330
   IF TOT=0 THEN PUT SKIP EDIT(OKEY,OYEAR,OMONTH,(JDATA(I) DO I=1 TO    SMI00340
     12))(A(12),2 P'99',12 F(8,1));                                     SMI00350
   REWRITE FILE(SUMF) FROM(SREC);                                       SMI00360
   TOT=TOT+1;                                                           SMI00370
   GOTO RT;                                                             SMI00380
  LT: CLOSE FILE(SUMF);                                                 SMI00390
   PUT SKIP EDIT('TOT RECORD= ',TOT)(A,F(10));                          SMI00400
 END SUMFILE;                                                           SMI00410
/*                                                                      SMI00420
//LKED.SYSLMOD DD DSN=USER.LOADLIB(SMIMV12),DISP=SHR                    SMI00430
//                                                                      SMI00440
```

SMICSM62

보정·물량 CATAL

FILE: SMICSM62 BJCATAL A0 VM/SP RELEASE 2.1 EXPRESS PUT8302+ SLU203

```
//SMICSM62 JOB CLASS=7,TYPRUN=HOLD                                    SMI00010
//  EXEC PLIFCL                                                       SMI00020
//PLI.SYSIN DD *                                                      SMI00030
* PROCESS GS,NEST,A,XREF,OPT(2);                                      SMI00040
  (NOZERODIVIDE) :                                                    SMI00050
 SUMMARY: PROC OPTIONS(MAIN);                                         SMI00060
 DCL CARD                      CHAR(80);                             SMI00070
 DCL ID                        CHAR(1),                              SMI00080
     CMONTH                    PIC'99',                              SMI00090
     CCARD                     CHAR(1);                              SMI00100
 DCL PRICE(12)                 FLOAT;            /*** PRICE(10) ***/  SMI00110
 DCL SUMF FILE RECORD KEYED ENV(VSAM);                               SMI00120
 DCL 1 SUMR,                                                         SMI00130
      2 OKEY                   CHAR(10),                             SMI00140
      2 FILK                   CHAR(13),                             SMI00150
      2 DATAJ(37),                                                   SMI00160
        3 ODATA                FIXED BIN(31),                        SMI00170
        3 JDATA                FLOAT,                                SMI00180
      2 OFIL                   CHAR(1);                              SMI00190
 DCL SYSEOF                    BIT(1) INIT('1'B);                    SMI00200
 DCL (CD2CH,CD5V)              BIT(1) INIT('1'B);                    SMI00210
 DCL EDIT2 FILE RECORD KEYED ENV(VSAM);                              SMI00220
 DCL EDIT3 FILE RECORD KEYED ENV(VSAM);                              SMI00230
 DCL ONCODE                    BUILTIN;                              SMI00240
 DCL 1 INREC,                                                        SMI00250
      2 SANUM                  CHAR(5),                              SMI00260
      2 CD                     CHAR(1),                              SMI00270
      2 INUM                   CHAR(2),                              SMI00280
      2 INDATA                 CHAR(40),                             SMI00290
      2 FA                     CHAR(2);                              SMI00300
 DCL YUGO1                     CHAR(1);                              SMI00310
 DCL 1 DTT2 DEF INDATA,                                              SMI00320
      2 JAEPUM                 CHAR(5),                              SMI00330
      2 YUGO2                  CHAR(1),                              SMI00340
      2 DAT2(7)                FIXED BIN(31),                        SMI00350
      2 WHY(2)                 CHAR(1);                              SMI00360
 DCL 1 DTT3 DEF INDATA,                                              SMI00370
      2 WONJAE                 CHAR(4),                              SMI00380
      2 YUGO3                  CHAR(1),                              SMI00390
      2 GU                     CHAR(1),                              SMI00400
      2 DAT3(2)                FIXED BIN(31);                        SMI00410
 DCL DAT4(10)                  FIXED BIN(31)  DEF  INDATA;           SMI00420
 DCL DAT5(7)                   FIXED BIN(31)  DEF  INDATA;           SMI00430
 DCL 1 DTT6 DEF INDATA,                                              SMI00440
      2 CAPACT                 CHAR(5),                              SMI00450
      2 YUGO6                  CHAR(1),                              SMI00460
      2 DAT6(2)                FIXED BIN(31);                        SMI00470
 DCL SORTREC                   CHAR(89);                             SMI00480
 DCL 1 SRT DEF SORTREC,                                              SMI00490
      2 SKEY                   CHAR(15),                             SMI00500
      2 SDATA(17)              FIXED BIN(31),                        SMI00510
      2 SRATE                  PIC'99',                              SMI00520
      2 SSIZE                  CHAR(1),                              SMI00530
      2 SGRT(2)                CHAR(1),                              SMI00540
      2 SYUGO                  CHAR(1);                              SMI00550
```

```
DCL INDUST                    CHAR(3);                                  SMI00560
DCL OPUM                      CHAR(5),                                  SMI00570
    OGU                       CHAR(1),                                  SMI00580
    KYUGO                     CHAR(1);                                  SMI00590
DCL ANU                       CHAR(10),                                 SMI00600
    KNU                       CHAR(10);                                 SMI00610
DCL TAB                       CHAR(1);                                  SMI00620
DCL SIZE                      CHAR(1);                                  SMI00630
DCL SNUM                      CHAR(5);                                  SMI00640
DCL RATE                      PIC'99';                                  SMI00650
DCL RETURN_CODE               FIXED BIN(15);                            SMI00660
DCL PEUP                      BIT(1);                                   SMI00670
DCL INN                       CHAR(2) DEF SORTREC POS(70);              SMI00680
DCL IL(2)                     CHAR(2);                                  SMI00690
DCL MDAT(3,17)                FLOAT DEC(16);                            SMI00700
DCL QDATA(3,17)               FLOAT DEC(16);                            SMI00710
DCL ZB                        BIT(1) INIT('1'B);                        SMI00720
DCL PDATA(3,17)               FLOAT DEC(16),                            SMI00730
    TDATA(3,17)               FLOAT DEC(16),                            SMI00740
    ADATA(3,17)               FLOAT DEC(16),                            SMI00750
    SDT(3,17)                 FLOAT DEC(16),                            SMI00760
    FDATA(3,17)               FLOAT DEC(16);                            SMI00770
DCL (WON,OWON)                CHAR(7);                                  SMI00780
DCL MONTH                     PIC'99';                                  SMI00790
DCL SINDUST                   CHAR(3);                                  SMI00800
DCL FST                       BIT(1) INIT('0'B);                        SMI00810
DCL RTYPE                     BIT(1) INIT('1'B);                        SMI00820
ON KEY(SUMF) BEGIN;                                                     SMI00830
  IF ONCODE=51 THEN DO;                                                 SMI00840
    PUT EDIT(OKEY,'INV')(A(11),A(4));  END;                             SMI00850
END;                                                                    SMI00860
ON ENDFILE(SYSIN) GOTO LLT;                                             SMI00870
RRC:  GET SKIP EDIT(CMONTH,CCARD,ID)(P'99',A(1),A(1));                  SMI00880
IF CCARD='2' & ID='3' THEN DO;                                          SMI00890
   GET EDIT((PRICE(I) DO I=1 TO 12))(12 (X(1),P'999V9'));   /** 10 **/ SMI00900
   PRICE(1)=PRICE(1)*0.65+PRICE(2)*0.35;                                SMI00910
   PRICE(2)=PRICE(6)*0.65+PRICE(7)*0.35;                                SMI00920
 END;                                                                   SMI00930
 OPEN FILE(EDIT2) SEQUENTIAL INPUT;                                     SMI00940
 OPEN FILE(EDIT3) SEQUENTIAL INPUT;                                     SMI00950
 OPEN FILE(SUMF) DIRECT UPDATE;                                         SMI00960
 OPEN FILE(SYSPRINT) LINESIZE(132);                                     SMI00970
 READ FILE(EDIT3) INTO(INREC);                                          SMI00980
 CALL PLISRTD(' SORT FIELDS=(1,15,CH,A) ',                              SMI00990
  ' RECORD TYPE=F,LENGTH=(89) ', 45000, RETURN_CODE, E15X, E35X);       SMI01000
 SUBSTR(SKEY,1,5)='99999';                                              SMI01010
 CALL E35X(SORTREC);                                                    SMI01020
 CALL PLIRETC(RETURN_CODE);                                             SMI01030
E15X: PROC RETURNS(CHAR(89));                                           SMI01040
 ON ENDFILE(EDIT2) GOTO EN15;                                           SMI01050
 ON ENDFILE(EDIT3) RTYPE='0'B;                                          SMI01060
 DO WHILE(RTYPE);                                                       SMI01070
   IF CD='1' THEN DO;                                                   SMI01080
 IF SUBSTR(INDATA,11,2)='  ' THEN SUBSTR(INDATA,11,2)='01';             SMI01090
 IF SUBSTR(INDATA,11,1)<'0' | SUBSTR(INDATA,11,1)>'9'                   SMI01100
```

```
        THEN SUBSTR(INDATA,11,1)='0';                                   SMI01110
  IF SUBSTR(INDATA,12,1)('0' | SUBSTR(INDATA,12,1))'9'                  SMI01120
        THEN SUBSTR(INDATA,12,1)='0';                                   SMI01130
      INDUST=SUBSTR(INDATA,1,3);  RATE=SUBSTR(INDATA,11,2);             SMI01140
      IF RATE=0 THEN RATE=1;                                            SMI01150
      SIZE=SUBSTR(INDATA,6,1);  TAB=SUBSTR(INDATA,8,1);                 SMI01160
      MONTH=CMONTH;             YUGO1=SUBSTR(INDATA,7,1);               SMI01170
      READ FILE(EDIT3) INTO(INREC);                                     SMI01180
    END;                                                                SMI01190
   ELSE DO;   IF CD='2' THEN CD2CH='0'B;                                SMI01200
    IF CCARD='4' & CD2CH THEN GOTO AR;                                  SMI01210
    IF CD=CCARD THEN DO;                                                SMI01220
      CALL SORTRTN;                                                     SMI01230
      IF CD5V THEN READ FILE(EDIT3) INTO(INREC);                        SMI01240
      CD5V='1'B;                                                        SMI01250
      CALL PLIRETC(12);                                                 SMI01260
      RETURN(SORTREC);                                                  SMI01270
    END;                                                                SMI01280
  AR:  READ FILE(EDIT3) INTO(INREC);                                    SMI01290
   END;                                                                 SMI01300
  END;                                                                  SMI01310
  RTT: IF CD5V THEN READ FILE(EDIT2) INTO(INREC);                       SMI01320
    CD5V='1'B;                                                          SMI01330
    IF CD='1' THEN DO;                                                  SMI01340
  IF SUBSTR(INDATA,11,2)='  ' THEN SUBSTR(INDATA,11,2)='01';            SMI01350
  IF SUBSTR(INDATA,11,1)('0' | SUBSTR(INDATA,11,1))'9'                  SMI01360
        THEN SUBSTR(INDATA,11,1)='0';                                   SMI01370
  IF SUBSTR(INDATA,12,1)('0' | SUBSTR(INDATA,12,1))'9'                  SMI01380
        THEN SUBSTR(INDATA,12,1)='0';                                   SMI01390
      INDUST=SUBSTR(INDATA,1,3); RATE=SUBSTR(INDATA,11,2);              SMI01400
      IF RATE=0 THEN RATE=1;                                            SMI01410
      SIZE=SUBSTR(INDATA,6,1);  TAB=SUBSTR(INDATA,8,1);                 SMI01420
      MONTH=CMONTH-1;           YUGO1=SUBSTR(INDATA,7,1);               SMI01430
    END;                                                                SMI01440
    ELSE DO;  IF CD='2' THEN CD2CH='0'B;                                SMI01450
     IF CCARD='4' & CD2CH THEN GOTO RTT;                                SMI01460
     IF CD=CCARD THEN DO;                                               SMI01470
       CALL SORTRTN;                                                    SMI01480
       CALL PLIRETC(12);                                                SMI01490
       RETURN(SORTREC);                                                 SMI01500
     END;                                                               SMI01510
    END;                                                                SMI01520
    GOTO RTT;                                                           SMI01530
  EN15:   CALL PLIRETC(8);                                              SMI01540
 END E15X;                                                              SMI01550
 E35X: PROC(SRTREC);                                                    SMI01560
 DCL SRTREC          CHAR(89);                                          SMI01570
  SORTREC=SRTREC;                                                       SMI01580
  IF CCARD='2' THEN CALL CD2RTN;                                        SMI01590
  ELSE IF CCARD='3' THEN CALL CD3RTN;                                   SMI01600
  ELSE IF CCARD='4' THEN CALL CD4RTN;                                   SMI01610
  ELSE IF CCARD='6' THEN CALL CD6RTN;                                   SMI01620
  CALL PLIRETC(4);                                                      SMI01630
 END E35X;                                                              SMI01640
 SORTRTN: PROC;                                                         SMI01650
```

```
  SORTREC=' ';                                                          SMI01660
  IF CD='2' THEN DO;                                                    SMI01670
    IF RATE=1 THEN SKEY=JAEPUM || '1' || SANUM || MONTH ;               SMI01680
              ELSE SKEY=JAEPUM || '2' || SANUM || MONTH ;               SMI01690
    INN=INUM;                                                           SMI01700
    IF SUBSTR(JAEPUM,1,4)='2201' | SUBSTR(JAEPUM,1,4)='5206' THEN DO;   SMI01710
      SDATA(1)=DAT2(2);  SDATA(2)=0;                                    SMI01720
      DO I=3 TO 6;                                                      SMI01730
        SDATA(I)=DAT2(I);                                               SMI01740
      END;                                                              SMI01750
    END;                                                                SMI01760
    ELSE DO;                                                            SMI01770
      DO I=1 TO 6;                                                      SMI01780
        SDATA(I)=DAT2(I+1);                                             SMI01790
      END;                                                              SMI01800
    END;                                                                SMI01810
    SRATE=RATE;                                                         SMI01820
    SSIZE=SIZE;                                                         SMI01830
    SGRT=WHY;                                                           SMI01840
    SYUGO=YUGO2;                                                        SMI01850
  END;                                                                  SMI01860
  ELSE IF CD='3' THEN DO;                                               SMI01870
      IF INDUST='312' THEN INDUST='311';                                SMI01880
    SKEY=INDUST || WONJAE || SANUM || GU || MONTH;                      SMI01890
    SDATA(1)=DAT3(1);                                                   SMI01900
    SDATA(2)=DAT3(2);                                                   SMI01910
    SYUGO=YUGO3; INN=INUM;                                              SMI01920
  END;                                                                  SMI01930
  ELSE IF CD='4' THEN DO;                                               SMI01940
      CD2CH='1'B;                                                       SMI01950
      IF INDUST='312' THEN INDUST='311';                                SMI01960
      IF INDUST='410' THEN INDUST='400';                                SMI01970
    IF RATE=1 THEN SKEY=INDUST || '1' || SANUM || MONTH;                SMI01980
              ELSE SKEY=INDUST || '2' || SANUM || MONTH;                SMI01990
    DO I=1 TO 10;                                                       SMI02000
      SDATA(I)=DAT4(I);                                                 SMI02010
    END;                                                                SMI02020
    SRATE=RATE;                                                         SMI02030
    SYUGO=YUGO1;                                                        SMI02040
    IF RTYPE THEN READ FILE(EDIT3) INTO(INREC);                         SMI02050
    ELSE READ FILE(EDIT2) INTO(INREC);                                  SMI02060
    IF CD='5' THEN DO;                                                  SMI02070
     CD5V='1'B;                                                         SMI02080
    DO I=1 TO 7;                                                        SMI02090
      SDATA(I+10)=DAT5(I);                                              SMI02100
    END; END;                                                           SMI02110
    ELSE DO;  CD5V='0'B;                                                SMI02120
     DO I=1 TO 7;                                                       SMI02130
       SDATA(I+10)=0;                                                   SMI02140
     END;  END;                                                         SMI02150
  END;                                                                  SMI02160
  ELSE IF CD='6' THEN DO;                                               SMI02170
    SKEY=CAPACT || SANUM || MONTH;                                      SMI02180
    SYUGO=YUGO6;  INN=INUM;                                             SMI02190
    SDATA(1)=DAT6(1);                                                   SMI02200
```

```
    SDATA(2)=DAT6(2);                                                   SMI02210
  END;                                                                  SMI02220
 END SORTRTN;                                                           SMI02230
 CD2RTN: PROC ;                                                         SMI02240
  OGU=GU;  OPUM=JAEPUM;                                                 SMI02250
  JAEPUM=SUBSTR(SKEY,1,5);                                              SMI02260
  GU=SUBSTR(SKEY,6,1);                                                  SMI02270
  SANUM=SUBSTR(SKEY,7,5);                                               SMI02280
  MONTH=SUBSTR(SKEY,12,2);                                              SMI02290
  IF RTYPE='0'B THEN DO;                                                SMI02300
    PUT PAGE EDIT('PRODUCTS BY ESTABLISHMENT (',CMONTH,')')             SMI02310
                 (X(50),A,F(2),A);  RATE=SRATE; KYUGO=SYUGO;            SMI02320
    OGU=GU;  OPUM=JAEPUM;  RTYPE='1'B;  PUT SKIP; SNUM=SANUM;           SMI02330
  END;                                                                  SMI02340
  ANU=OPUM || SNUM;  KNU=JAEPUM || SANUM;                               SMI02350
  IF ANU¬=KNU THEN DO;                                                  SMI02360
   IF ID='1' THEN DO;                                                   SMI02370
    DO I=1 TO 8;                                                        SMI02380
     IF PDATA(1,I)=0 THEN PDATA(3,I)=0;                                 SMI02390
     ELSE PDATA(3,I)=PDATA(2,I)/PDATA(1,I)*100-100;                     SMI02400
    END;                                                                SMI02410
    IF FST='0'B THEN                                                    SMI02420
      PUT SKIP(2) EDIT(OPUM,SNUM,RATE,KYUGO,SIZE,IL(1),                 SMI02430
    (PDATA(1,J) DO J=1,4 TO 8),PDATA(3,1),WHY(1),WHY(2))                SMI02440
  (2 A(7),P'99',X(2),2 A(3),A(3),6 F(10),F(10,1),X(20),A(2),A(1));      SMI02450
    ELSE PUT SKIP(2) EDIT(SNUM,RATE,KYUGO,SIZE,IL(1),(PDATA(1,J)        SMI02460
    DO J=1,4 TO 8),PDATA(3,1),WHY(1),WHY(2))(X(7),A(7),P'99',X(2),      SMI02470
    2 A(3),A(3),6 F(10),F(10,1),X(20),A(2),A(1));                       SMI02480
 PUT SKIP EDIT(IL(2),(PDATA(2,J) DO J=1,4 TO 8),(PDATA(3,K) DO K=5,     SMI02490
     6,8))(X(24),A(3),6 F(10),3 F(10,1));                               SMI02500
    FST='1'B;                                                           SMI02510
   END;                                                                 SMI02520
   IF RATE=1 THEN OGU='1'; ELSE OGU='2';                                SMI02530
   IF SRATE=1 THEN GU='1'; ELSE GU='2';                                 SMI02540
   IF SNUM='06108' THEN DO; SDT=PDATA; GOTO TAKJU;  END;                SMI02550
   IF OGU='1' THEN TDATA=TDATA+PDATA;                                   SMI02560
   ELSE DO;                                                             SMI02570
     FDATA=FDATA+PDATA;                                                 SMI02580
     ADATA=ADATA+PDATA*RATE;                                            SMI02590
   END;                                                                 SMI02600
   TAKJU: PDATA=0;  SNUM=SANUM; IL=' ';                                 SMI02610
  IF OPUM¬=JAEPUM THEN DO;                                              SMI02620
   IF ID='1' THEN DO;                                                   SMI02630
     CALL JTPUT('S_TOTAL',FDATA);                                       SMI02640
     CALL JTPUT('R*TOTAL',ADATA);                                       SMI02650
   END;                                                                 SMI02660
   TDATA=TDATA+ADATA;                                                   SMI02670
   CALL JTPUT('A_TOTAL',TDATA);                                         SMI02680
   IF OPUM='13050' THEN DO;                                             SMI02690
     CALL JTPUT('A_TOTAL',SDT); TDATA=SDT;                              SMI02700
   END;                                                                 SMI02710
   IF SUBSTR(OPUM,5,1)='0' & ID='3' THEN CALL IDX2;                     SMI02720
   IF SUBSTR(OPUM,5,1)¬='0' THEN DO;                                    SMI02730
    IF OPUM='83378' | OPUM='82118' | OPUM='62068' | OPUM='62088' |      SMI02740
       OPUM='41088' | OPUM='41138' | OPUM='41148' | OPUM='41158' |      SMI02750
```

```
      OPUM='41168' | OPUM='51278' | OPUM='55098' | OPUM='55108' |        SMI02760
      OPUM='69098' | OPUM='69108' | OPUM='69118' | OPUM='81078' |        SMI02770
      OPUM='81108' | OPUM='81138' | OPUM='82068' | OPUM='82078' |        SMI02780
      OPUM='82098' | OPUM='82108' | OPUM='82138' | OPUM='82208' |        SMI02790
      OPUM='82248' | OPUM='82308' | OPUM='82358' | OPUM='82438' |        SMI02800
      OPUM='82198' | OPUM='82178' | OPUM='83048' | OPUM='82428' |        SMI02810
      OPUM='82278' | OPUM='82168' | OPUM='82258' | OPUM='82268' |        SMI02820
      OPUM='85138' | OPUM='85128' | OPUM='82448' |                      SMI02830
 OPUM='82368' | SUBSTR(OPUM,1,2)='01' | SUBSTR(OPUM,1,4)='2201' THEN;   SMI02840
    ELSE DO;                                                            SMI02850
     QDATA=QDATA+TDATA; ZB='0'B; END; END;                              SMI02860
 IF ZB='0'B & SUBSTR(OPUM,1,4)¬=SUBSTR(JAEPUM,1,4) THEN DO;             SMI02870
  CALL JTPUT('0_TOTAL',QDATA);  ZB='1'B; SUBSTR(OPUM,5,1)='0';          SMI02880
    IF  OPUM = '55080'  THEN  DO;          /*************************/ SMI02890
      QDATA = QDATA - TDATA ;              /**   PUMMOK = '55080'   **/ SMI02900
      QDATA(1,*) = QDATA(1,*) + TDATA(1,*) * PRICE(11);   /**********/ SMI02910
      QDATA(2,*) = QDATA(2,*) + TDATA(2,*) * PRICE(12);   /**********/ SMI02920
    END;                                   /*************************/ SMI02930
  TDATA=QDATA;                                                          SMI02940
  IF ID='3' THEN CALL IDX2;  QDATA=0;                                   SMI02950
 END;                                                                   SMI02960
  NN: IF SUBSTR(OPUM,1,1)¬=SUBSTR(JAEPUM,1,1) THEN PUT PAGE;            SMI02970
  FDATA=0; ADATA=0; TDATA=0; FST='0'B; OPUM=JAEPUM; OGU=GU;             SMI02980
 END;                                                                   SMI02990
 ELSE IF OGU¬=GU & ID='1' THEN DO;                                      SMI03000
  CALL JTPUT('S_TOTAL',TDATA); OGU=GU; PUT SKIP;                        SMI03010
 END;                                                                   SMI03020
 END;                                                                   SMI03030
 RATE=SRATE; SIZE=SSIZE; WHY=SGRT; KYUGO=SYUGO;                         SMI03040
 OGU=GU;                                                                SMI03050
 IF CMONTH=MONTH THEN I=2;                                              SMI03060
 ELSE I=1;                                                              SMI03070
 IL(I)=INN;                                                             SMI03080
 PDATA(I,1)=SDATA(1);                                                   SMI03090
 DO J=2 TO 6;                                                           SMI03100
   PDATA(I,J+2)=SDATA(J);                                               SMI03110
 END;                                                                   SMI03120
 IF SSIZE='1' THEN PDATA(I,2)=PDATA(I,1);                               SMI03130
 ELSE PDATA(I,3)=PDATA(I,1);                                            SMI03140
JTPUT: PROC(TITLE,DATA);                                                SMI03150
DCL TITLE          CHAR(7);                                             SMI03160
DCL DATA(3,17)     FLOAT DEC(16);                                       SMI03170
DCL PTIT           CHAR(5) INIT((5)' ');                                SMI03180
 DO I=1 TO 8;                                                           SMI03190
  IF DATA(1,I)=0 THEN DATA(3,I)=0;                                      SMI03200
  ELSE DATA(3,I)=DATA(2,I)/DATA(1,I)*100-100;                           SMI03210
 END;                                                                   SMI03220
 IF SUBSTR(TITLE,1,1)='A' THEN PTIT=OPUM;                               SMI03230
 IF SUBSTR(TITLE,1,1)='S' THEN PUT SKIP;                                SMI03240
 PUT SKIP(2) EDIT(PTIT,TITLE,(DATA(1,I) DO I=1 TO 8),(DATA(3,J) DO J=1 SMI03250
 TO 3))(A(6),A(7),X(1),8 F(10),3 F(10,1));                              SMI03260
 PUT SKIP EDIT((DATA(2,I) DO I=1 TO 8),(DATA(3,J) DO J=5,6,8))          SMI03270
 (X(14),8 F(10),3 F(10,1));                                             SMI03280
END JTPUT;                                                              SMI03290
IDX2: PROC;                                                             SMI03300
```

```
 IF OPUM='22010' THEN DO;                                                SMI03310
    TDATA(1,*)=TDATA(1,*)/PRICE(1)*100;                                  SMI03320
    TDATA(2,*)=TDATA(2,*)/PRICE(2)*100;  END;                            SMI03330
 IF OPUM='52060' THEN DO;                                                SMI03340
   TDATA(1,*)=TDATA(1,*)/PRICE(3)*100;                                   SMI03350
   TDATA(2,*)=TDATA(2,*)/PRICE(8)*100;  END;                             SMI03360
 IF OPUM='82440' THEN DO;                                                SMI03370
   TDATA(1,*)=TDATA(1,*)/PRICE(4)*100;                                   SMI03380
   TDATA(2,*)=TDATA(2,*)/PRICE(9)*100;  END;                             SMI03390
 IF OPUM='85120' THEN DO;                                                SMI03400
   TDATA(1,*)=TDATA(1,*)/PRICE(5)*100;                                   SMI03410
   TDATA(2,*)=TDATA(2,*)/PRICE(10)*100;  END;                            SMI03420
 IF SUBSTR(OPUM,1,1)='0' THEN INDUST='2' || SUBSTR(OPUM,2,1) || '0';     SMI03430
 ELSE IF SUBSTR(OPUM,1,2)='99' THEN INDUST='400';                        SMI03440
 ELSE INDUST='3' || SUBSTR(OPUM,1,2);                                    SMI03450
 OKEY='2A' || INDUST || OPUM;                                            SMI03460
 READ FILE(SUMF) INTO(SUMR) KEY(OKEY);                                   SMI03470
 ODATA(36)=TDATA(1,1); ODATA(37)=TDATA(2,1);                             SMI03480
 REWRITE FILE(SUMF) FROM(SUMR) KEY(OKEY);                                SMI03490
 OKEY='2B' || INDUST || OPUM;                                            SMI03500
 READ FILE(SUMF) INTO(SUMR) KEY(OKEY);                                   SMI03510
 ODATA(36)=TDATA(1,1); ODATA(37)=TDATA(2,1);                             SMI03520
 REWRITE FILE(SUMF) FROM(SUMR) KEY(OKEY);                                SMI03530
 OKEY='2C' || INDUST || OPUM;                                            SMI03540
 READ FILE(SUMF) INTO(SUMR) KEY(OKEY);                                   SMI03550
 ODATA(36)=TDATA(1,2); ODATA(37)=TDATA(2,2);                             SMI03560
 REWRITE FILE(SUMF) FROM(SUMR) KEY(OKEY);                                SMI03570
 OKEY='2D' || INDUST || OPUM;                                            SMI03580
 READ FILE(SUMF) INTO(SUMR) KEY(OKEY);                                   SMI03590
 ODATA(36)=TDATA(1,3); ODATA(37)=TDATA(2,3);                             SMI03600
 REWRITE FILE(SUMF) FROM(SUMR) KEY(OKEY);                                SMI03610
 OKEY='2E' || INDUST || OPUM;                                            SMI03620
 READ FILE(SUMF) INTO(SUMR) KEY(OKEY);                                   SMI03630
 ODATA(36)=TDATA(1,5)+TDATA(1,6); ODATA(37)=TDATA(2,5)+TDATA(2,6);       SMI03640
 REWRITE FILE(SUMF) FROM(SUMR) KEY(OKEY);                                SMI03650
 OKEY='2F' || INDUST || OPUM;                                            SMI03660
 READ FILE(SUMF) INTO(SUMR) KEY(OKEY);                                   SMI03670
 ODATA(36)=TDATA(1,5); ODATA(37)=TDATA(2,5);                             SMI03680
 REWRITE FILE(SUMF) FROM(SUMR) KEY(OKEY);                                SMI03690
 OKEY='2G' || INDUST || OPUM;                                            SMI03700
 READ FILE(SUMF) INTO(SUMR) KEY(OKEY);                                   SMI03710
 ODATA(36)=TDATA(1,6); ODATA(37)=TDATA(2,6);                             SMI03720
 REWRITE FILE(SUMF) FROM(SUMR) KEY(OKEY);                                SMI03730
 OKEY='2H' || INDUST || OPUM;                                            SMI03740
 READ FILE(SUMF) INTO(SUMR) KEY(OKEY);                                   SMI03750
 ODATA(36)=TDATA(1,8); ODATA(37)=TDATA(2,8);                             SMI03760
 REWRITE FILE(SUMF) FROM(SUMR) KEY(OKEY);                                SMI03770
END IDX2;                                                                SMI03780
END CD2RTN;                                                              SMI03790
CD3RTN: PROC;                                                            SMI03800
 WON=SUBSTR(SKEY,1,7);                                                   SMI03810
 SANUM=SUBSTR(SKEY,8,5);                                                 SMI03820
 GU=SUBSTR(SKEY,13,1);                                                   SMI03830
 MONTH=SUBSTR(SKEY,14,2);                                                SMI03840
 IF RTYPE='0'B THEN DO;                                                  SMI03850
```

```
    PUT PAGE EDIT('RAW MATERIAL BY ESTABLISHMENT (',CMONTH,')')          SMI03860
                 (X(28),A,F(2),A);                                       SMI03870
    PUT SKIP;                                                            SMI03880
    OWON=WON; RTYPE='1'B;  YUGO3=SYUGO; SNUM=SANUM; OGU=GU;              SMI03890
  END;                                                                   SMI03900
  IF WON || SANUM || GU¬=OWON || SNUM || OGU THEN DO;                    SMI03910
     IF PDATA(J,1)=0 THEN PDATA(J,3)=0;                                  SMI03920
     ELSE PDATA(J,3)=PDATA(J,2)/PDATA(J,1)*100-100;                      SMI03930
     IF PDATA(J,4)=0 THEN PDATA(J,6)=0;                                  SMI03940
     ELSE PDATA(J,6)=PDATA(J,5)/PDATA(J,4)*100-100;                      SMI03950
    IF ID='3' THEN                                                       SMI03960
    PUT SKIP EDIT(SUBSTR(OWON,1,3),SUBSTR(OWON,4,4),SNUM,OGU,YUGO3,      SMI03970
      (PDATA(J,I) DO I=1 TO 6))(A(5),2 A(7),A(4),A(3),                   SMI03980
       2(2 F(10),F(10,1)));                                              SMI03990
  SNUM=SANUM;  OGU=GU;  PDATA=0;                                         SMI04000
  IF OWON¬=WON THEN DO;                                                  SMI04010
    TDATA(3,*)=TDATA(1,*)+TDATA(2,*);                                    SMI04020
   DO K=1 TO 3;                                                          SMI04030
    IF TDATA(K,1)=0 THEN TDATA(K,3)=0;                                   SMI04040
    ELSE TDATA(K,3)=TDATA(K,2)/TDATA(K,1)*100-100;                       SMI04050
    IF TDATA(K,4)=0 THEN TDATA(K,6)=0;                                   SMI04060
    ELSE TDATA(K,6)=TDATA(K,5)/TDATA(K,4)*100-100;                       SMI04070
   END;                                                                  SMI04080
    PUT SKIP(2) EDIT(SUBSTR(OWON,1,3),SUBSTR(OWON,4,4),                  SMI04090
                     'TOTAL','1',(TDATA(1,I) DO I=1 TO 6))               SMI04100
      (A(5),A(7),A(7),A(4),X(3),2(2 F(10),F(10,1)));                     SMI04110
    PUT SKIP EDIT('2',(TDATA(2,I) DO I=1 TO 6))(X(19),A(4),              SMI04120
              X(3),2(2 F(10),F(10,1)));                                  SMI04130
    PUT SKIP EDIT('3',(TDATA(3,I) DO I=1 TO 6))(X(19),A(4),              SMI04140
             X(3),2(2 F(10),F(10,1)));                                   SMI04150
    IF ID='3' THEN CALL IDX3;                                            SMI04160
    IF SUBSTR(OWON,2,1)¬=SUBSTR(WON,2,1) THEN PUT PAGE;                  SMI04170
    PUT SKIP;                                                            SMI04180
    TDATA=0; OWON=WON;                                                   SMI04190
  END;                                                                   SMI04200
  END;                                                                   SMI04210
  YUGO3=SYUGO;                                                           SMI04220
  IF CMONTH=MONTH THEN I=2;                                              SMI04230
  ELSE I=1;                                                              SMI04240
  IF GU='1' THEN J=1;                                                    SMI04250
  ELSE J=2;                                                              SMI04260
  PDATA(J,I)=SDATA(1);                                                   SMI04270
  PDATA(J,I+3)=SDATA(2);                                                 SMI04280
  TDATA(J,I)=TDATA(J,I)+PDATA(J,I);                                      SMI04290
  TDATA(J,I+3)=TDATA(J,I+3)+PDATA(J,I+3);                                SMI04300
  IDX3: PROC;                                                            SMI04310
  DCL Q   PIC'9';                                                        SMI04320
   DO Q=1 TO 3;                                                          SMI04330
    OKEY='31' || WON || Q;                                               SMI04340
    READ FILE(SUMF) INTO(SUMR) KEY(OKEY);                                SMI04350
    ODATA(36)=TDATA(Q,1); ODATA(37)=TDATA(Q,4);                          SMI04360
    REWRITE FILE(SUMF) FROM(SUMR) KEY(OKEY);                             SMI04370
    OKEY='32' || WON || Q;                                               SMI04380
    READ FILE(SUMF) INTO(SUMR) KEY(OKEY);                                SMI04390
    ODATA(36)=TDATA(Q,2); ODATA(37)=TDATA(Q,5);                          SMI04400
```

```
    REWRITE FILE(SUMF) FROM(SUMR) KEY(OKEY);                                SMI04410
   END;                                                                     SMI04420
  END IDX3;                                                                 SMI04430
 END CD3RTN;                                                                SMI04440
 CD4RTN: PROC;                                                              SMI04450
  INDUST=SUBSTR(SKEY,1,3);                                                  SMI04460
  GU=SUBSTR(SKEY,4,1);                                                      SMI04470
  SANUM=SUBSTR(SKEY,5,5);                                                   SMI04480
  MONTH=SUBSTR(SKEY,10,2);                                                  SMI04490
  IF RTYPE='0'B THEN DO ;                                                   SMI04500
    PUT PAGE EDIT('WORKER & SALARY BY ESTABLISHMENT (',CMONTH,')')          SMI04510
      (X(46),A,F(2),A);  YUGO3=SYUGO;  RATE=SRATE;                          SMI04520
    OGU=GU; SINDUST=INDUST; RTYPE='1'B; PUT SKIP; SNUM=SANUM;               SMI04530
  END;                                                                      SMI04540
   INDUST=TRANSLATE(INDUST,'0',' ');                                        SMI04550
  IF INDUST || SANUM¬=SINDUST || SNUM THEN DO;                              SMI04560
   IF ID='1' THEN DO;                                                       SMI04570
    DO I=1 TO 4;                                                            SMI04580
     PDATA(3,I)=PDATA(2,I)-PDATA(1,I);                                      SMI04590
     IF PDATA(1,I+10)+PDATA(1,I+6)=0 THEN PDATA(3,I+10)=0;  ELSE            SMI04600
     PDATA(3,I+10)=(PDATA(2,I+10)+PDATA(2,I+6))/(PDATA(1,I+10)+PDATA(1,     SMI04610
     I+6))*100-100;                                                         SMI04620
    END;                                                                    SMI04630
    PDATA(3,5)=PDATA(2,5)-PDATA(1,5);                                       SMI04640
    IF PDATA(1,1)=0 THEN PDATA(3,7)=0;                                      SMI04650
    ELSE PDATA(3,7)=PDATA(2,1)/PDATA(1,1)*100-100;                          SMI04660
    IF PDATA(1,3)=0 THEN PDATA(3,8)=0;                                      SMI04670
    ELSE PDATA(3,8)=PDATA(2,3)/PDATA(1,3)*100-100;                          SMI04680
    IF PDATA(1,4)=0 THEN PDATA(3,9)=0;                                      SMI04690
    ELSE PDATA(3,9)=PDATA(2,4)/PDATA(1,4)*100-100;                          SMI04700
    IF PDATA(1,5)=0 THEN DO; PDATA(3,10)=0;                                 SMI04710
      EDATA(3,6)=0;  PDATA(3,15)=0;  END;                                   SMI04720
    ELSE DO; PDATA(3,10)=PDATA(2,5)/PDATA(1,5)*100-100;                     SMI04730
     PDATA(3,6)=PDATA(2,6)/PDATA(1,5)*100;                                  SMI04740
     PDATA(3,15)=PDATA(2,15)/PDATA(1,5)*100;                                SMI04750
    END;                                                                    SMI04760
    IF PDATA(1,16)=0 THEN PDATA(3,16)=0;                                    SMI04770
    ELSE PDATA(3,16)=PDATA(2,16)/PDATA(1,16)*100-100;                       SMI04780
    IF PDATA(1,17)=0 THEN PDATA(3,17)=0;                                    SMI04790
    ELSE PDATA(3,17)=PDATA(2,17)/PDATA(1,17)*100-100;                       SMI04800
   PUT SKIP(2) EDIT(SINDUST,SNUM,RATE,YUGO3,((PDATA(J,K) DO K=1,7,11,       SMI04810
     6) DO J=1,2,3))(2 A(7),P'99',X(2),A(3),9 F(9),3 F(9,1));               SMI04820
    PUT SKIP EDIT(((PDATA(J,K) DO K=2,8,12,15) DO J=1,2),(PDATA(3,I)        SMI04830
    DO I=2,12,15))(X(21),8 F(9),F(9),  X(9),2 F(9,1));                      SMI04840
    PUT SKIP EDIT(((PDATA(J,K) DO K=3,9,13,16) DO J=1,2),(PDATA(3,I)        SMI04850
    DO I=3,8,13,16))(X(21),9 F(9),3 (F(9,1)));                              SMI04860
    PUT SKIP EDIT(((PDATA(J,K) DO K=4,17) DO J=1,2),(PDATA(3,I)             SMI04870
   DO I=4,9,17))(X(21),2 (F(9),X(18),F(9)),F(9),F(9,1),X(9),F(9,1));        SMI04880
    PUT SKIP EDIT(((PDATA(J,K) DO K=5,10,14) DO J=1,2,3))                   SMI04890
    (X(21),2(3 F(9),X(9)),F(9),2 F(9,1));                                   SMI04900
   END ;                                                                    SMI04910
   IF OGU='1' THEN TDATA=TDATA+PDATA;                                       SMI04920
   ELSE DO;                                                                 SMI04930
     FDATA=FDATA+PDATA;                                                     SMI04940
     ADATA=ADATA+PDATA* RATE;                                               SMI04950
```

```
   END;                                                                 SMI04960
   PDATA=0;  SNUM=SANUM;                                                SMI04970
  IF SINDUST¬=INDUST THEN DO;                                           SMI04980
    IF ID='1' THEN DO;                                                  SMI04990
      CALL WTPUT('S_TOTAL',FDATA);                                      SMI05000
      CALL WTPUT('R*TOTAL',ADATA);                                      SMI05010
    END;                                                                SMI05020
    TDATA=TDATA+ADATA;                                                  SMI05030
    CALL WTPUT('A_TOTAL',TDATA);                                        SMI05040
    IF ID='3' THEN CALL IDX4;                                           SMI05050
    IF SUBSTR(SINDUST,2,1)¬=SUBSTR(INDUST,2,1) THEN PUT PAGE;           SMI05060
    FDATA=0; ADATA=0; TDATA=0; SINDUST=INDUST;  OGU=GU;                 SMI05070
  END;                                                                  SMI05080
  ELSE IF OGU¬=GU & ID='1' THEN DO; CALL WTPUT('S_TOTAL',TDATA);        SMI05090
    OGU=GU; PUT SKIP; END;                                              SMI05100
  END;                                                                  SMI05110
  RATE=SRATE;  YUGO3=SYUGO;  OGU=GU;                                    SMI05120
  IF CMONTH=MONTH THEN I=2;                                             SMI05130
  ELSE I=1;                                                             SMI05140
  PDATA(I,*)=SDATA(*);                                                  SMI05150
 WTPUT : PROC(TITLE,DATA);                                              SMI05160
 DCL TITLE            CHAR(7);                                          SMI05170
 DCL DATA(3,17)       FLOAT DEC(16);                                    SMI05180
 DCL PTIT             CHAR(5) INIT((5)' ');                             SMI05190
    DO I=1 TO 4;                                                        SMI05200
    DATA(3,I)=DATA(2,I)-DATA(1,I);                                      SMI05210
     IF DATA(1,I+10)+DATA(1,I+6)=0 THEN DATA(3,I+10)=0;  ELSE           SMI05220
     DATA(3,I+10)=(DATA(2,I+10)+DATA(2,I+6))/(DATA(1,I+10)+DATA(1,      SMI05230
     I+6))*100-100;                                                     SMI05240
    END;                                                                SMI05250
    DATA(3,5)=DATA(2,5)-DATA(1,5);                                      SMI05260
    IF DATA(1,1)=0 THEN DATA(3,7)=0;                                    SMI05270
    ELSE DATA(3,7)=DATA(2,1)/DATA(1,1)*100-100;                         SMI05280
    IF DATA(1,3)=0 THEN DATA(3,8)=0;                                    SMI05290
    ELSE DATA(3,8)=DATA(2,3)/DATA(1,3)*100-100;                         SMI05300
    IF DATA(1,4)=0 THEN DATA(3,9)=0;                                    SMI05310
    ELSE DATA(3,9)=DATA(2,4)/DATA(1,4)*100-100;                         SMI05320
    IF DATA(1,5)=0 THEN DO; DATA(3,10)=0;                               SMI05330
      DATA(3,6)=0;  DATA(3,15)=0;  END;                                 SMI05340
    ELSE DO; DATA(3,10)=DATA(2,5)/DATA(1,5)*100-100;                    SMI05350
     DATA(3,6)=DATA(2,6)/DATA(1,5)*100;                                 SMI05360
     DATA(3,15)=DATA(2,15)/DATA(1,5)*100;                               SMI05370
    END;                                                                SMI05380
    IF DATA(1,16)=0 THEN DATA(3,16)=0;                                  SMI05390
    ELSE DATA(3,16)=DATA(2,16)/DATA(1,16)*100-100;                      SMI05400
    IF DATA(1,17)=0 THEN DATA(3,17)=0;                                  SMI05410
    ELSE DATA(3,17)=DATA(2,17)/DATA(1,17)*100-100;                      SMI05420
 IF SUBSTR(TITLE,1,1)='A' THEN PTIT=SINDUST;                            SMI05430
  PUT SKIP(2) EDIT(PTIT,TITLE,((DATA(J,K) DO K=1,7,11,6) DO J=1,2,3))   SMI05440
   (A(7),A(7),X(7),9 F(9),3 F(9,1));                                    SMI05450
  PUT SKIP EDIT(((DATA(J,K) DO K=2,8,12,15) DO J=1,2),(DATA(3,I) DO I=  SMI05460
   2,12,15))(X(21),9 F(9),X(9),2 F(9,1));                               SMI05470
  PUT SKIP EDIT(((DATA(J,K) DO K=3,9,13,16) DO J=1,2),(DATA(3,I) DO I=  SMI05480
   3,8,13,16))(X(21),9 F(9),3 F(9,1));                                  SMI05490
  PUT SKIP EDIT(((DATA(J,K) DO K=4,17) DO J=1,2),(DATA(3,I) DO I=4,9,   SMI05500
```

```
  17))(X(21),2 (F(9),X(18),F(9)),F(9),F(9,1),X(9),F(9,1));
 PUT SKIP EDIT(((DATA(J,K) DO K=5,10,14) DO J=1,2,3))
  (X(21),2(3 F(9),X(9)),F(9),2 F(9,1));
END WTPUT;
IDX4:PROC;
CALL JUPD('1','1',1,3);
CALL JUPD('1','2',5,4);
CALL JAPD('1','3',6);
CALL JAPD('1','4',15);
CALL JAPD('1','5',16);
CALL JAPD('1','7',17);
CALL JAPD('2','1',7);
CALL JAPD('2','2',9);
CALL JAPD('2','3',10);
CALL JAPD('3','1',7);
CALL JAPD('3','2',9);
CALL JAPD('3','3',10);
JUPD: PROC(A,B,C,D);
DCL A     CHAR(1);
DCL B     CHAR(1);
DCL C     PIC'99';
DCL D     PIC'99';
 OKEY='40' || SINDUST || A || B;
 READ FILE(SUMF) INTO(SUMR) KEY(OKEY);
 ODATA(36)=TDATA(1,C); JDATA(36)=TDATA(1,D);
 ODATA(37)=TDATA(2,C); JDATA(37)=TDATA(2,D);
 REWRITE FILE(SUMF) FROM(SUMR) KEY(OKEY);
END JUPD;
JAPD: PROC(A,B,C);
DCL A     CHAR(1);
DCL B     CHAR(1);
DCL C     PIC'99';
 OKEY='40' || SINDUST || A || B;
 READ FILE(SUMF) INTO(SUMR) KEY(OKEY);
 IF A='2' | A='1' THEN DO;
 ODATA(36)=TDATA(1,C); ODATA(37)=TDATA(2,C);
 JDATA(36)=0;          JDATA(37)=0;          END;
 ELSE DO;
 ODATA(36)=TDATA(1,C)-TDATA(1,C+4); ODATA(37)=TDATA(2,C)-TDATA(2,C+4);
 JDATA(36)=0;     JDATA(37)=0;
 END;
 REWRITE FILE(SUMF) FROM(SUMR) KEY(OKEY);
END JAPD;
END IDX4;
END CD4RTN;
CD6RTN: PROC ;
 CAPACT=SUBSTR(SKEY,1,5);
 SANUM=SUBSTR(SKEY,6,5);
 MONTH=SUBSTR(SKEY,11,2);
 IF RTYPE='0'B THEN DO;
   PUT PAGE EDIT('CAPACITY BY ESTABLISHMENT (',CMONTH,')')
      (X(23),A,F(2),A);
   OPUM=CAPACT; RTYPE='1'B; PUT SKIP; SNUM=SANUM; YUGO3=SYUGO;
 END;
 CAPACT=TRANSLATE(CAPACT,'0',' ');
```

```
  IF OPUM || SNUM¬=CAPACT || SANUM THEN DO;                                SMI06060
   IF ID='1' THEN DO;                                                      SMI06070
    IF PDATA(1,1)=0 THEN PDATA(1,3)=0;                                     SMI06080
    ELSE PDATA(1,3)=PDATA(1,2)/PDATA(1,1)*100-100;                         SMI06090
    IF PDATA(1,4)=0 THEN PDATA(1,6)=0;                                     SMI06100
    ELSE PDATA(1,6)=PDATA(1,5)/PDATA(1,4)*100-100;                         SMI06110
    PUT SKIP EDIT(OPUM,SNUM,YUGO3,IL(1),IL(2),(PDATA(1,J) DO               SMI06120
      J=1 TO 6))(2 A(7),A(3),2 A(4),2(2 F(10),F(10,1)));                   SMI06130
   END ;                                                                   SMI06140
    TDATA=TDATA+PDATA;                                                     SMI06150
    PDATA=0;  SNUM=SANUM;  IL=' ';                                         SMI06160
  IF OPUM¬=CAPACT THEN DO;                                                 SMI06170
    TDATA(1,3)=TDATA(1,2)/TDATA(1,1)*100-100;                              SMI06180
    TDATA(1,6)=TDATA(1,5)/TDATA(1,4)*100-100;                              SMI06190
    TDATA(1,7)=TDATA(1,4)/TDATA(1,1)*100;                                  SMI06200
    TDATA(1,8)=TDATA(1,5)/TDATA(1,2)*100;                                  SMI06210
    TDATA(1,9)=TDATA(1,8)-TDATA(1,7);                                      SMI06220
    PUT SKIP(2) EDIT(OPUM,'TOTAL',(TDATA(1,I) DO I=1 TO 9))                SMI06230
      (A(14),A(11),2(2 F(10),F(10,1)),3 F(10,1));                          SMI06240
    IF ID='3' THEN CALL IDX6;                                              SMI06250
    IF SUBSTR(OPUM,1,1)¬=SUBSTR(CAPACT,1,1) THEN PUT PAGE;                 SMI06260
    TDATA=0; OPUM=CAPACT;  PUT SKIP;                                       SMI06270
  END;                                                                     SMI06280
  END;                                                                     SMI06290
  YUGO3=SYUGO;                                                             SMI06300
  IF CMONTH=MONTH THEN I=2;                                                SMI06310
  ELSE I=1;                                                                SMI06320
  IL(I)=INN;                                                               SMI06330
  PDATA(1,I)=SDATA(1);                                                     SMI06340
  PDATA(1,I+3)=SDATA(2);                                                   SMI06350
  IDX6: PROC;                                                              SMI06360
  IF SUBSTR(OPUM,1,1)='0' THEN INDUST='2' || SUBSTR(OPUM,2,1) || '0';      SMI06370
  ELSE IF SUBSTR(OPUM,1,2)='99' THEN INDUST='4';                           SMI06380
  ELSE INDUST='3' || SUBSTR(OPUM,1,2);                                     SMI06390
  OKEY='61' || INDUST || OPUM;                                             SMI06400
  READ FILE(SUMF) INTO(SUMR) KEY(OKEY);                                    SMI06410
  ODATA(36)=TDATA(1,1); ODATA(37)=TDATA(1,2);                              SMI06420
  REWRITE FILE(SUMF) FROM(SUMR) KEY(OKEY);                                 SMI06430
  OKEY='62' || INDUST || OPUM;                                             SMI06440
  READ FILE(SUMF) INTO(SUMR) KEY(OKEY);                                    SMI06450
  ODATA(36)=TDATA(1,4);  ODATA(37)=TDATA(1,5);                             SMI06460
  REWRITE FILE(SUMF) FROM(SUMR) KEY(OKEY);                                 SMI06470
 END IDX6;                                                                 SMI06480
 END CD6RTN;                                                               SMI06490
 CLOSE FILE(SUMF),FILE(EDIT2),FILE(EDIT3);                                 SMI06500
  GOTO RRC;                                                                SMI06510
 LLT:                                                                      SMI06520
 END SUMMARY;                                                              SMI06530
/*                                                                         SMI06540
//LKED.SYSLMOD DD DSN=USER.LOADLIB(SMISM62),DISP=SHR                       SMI06550
//                                                                         SMI06560
```

SMICIX22

보정지수산출 CATAL

FILE: SMICIX62 BJCATAL A0 VM/SP RELEASE 2.1 EXPRESS PUT8302+ SLU203

```
//SMICIX22 JOB  CLASS=6,TYPRUN=HOLD                                    SMI00010
// EXEC  PLIFCL                                                        SMI00020
//PLI.SYSIN DD *                                                       SMI00030
* PROCESS GS,NEST,S,A,XREF,OPT(2);                                     SMI00040
  JISURTN: PROC  OPTIONS(MAIN);                                        SMI00050
 /*****************************************************************/   SMI00060
 /*                  GWANG_DONG BOJUNG JAKUP                      */   SMI00070
 /*   84.5.25 START        84.6.22 FINISHED       BY  UHM  DOO-YONG. */   SMI00080
 /*   STATEMENT THAT HAVE COMMENT( /*        ) MEANS BOJUNG JAKUP  */   SMI00090
 /*   SUMM FILE EXCHANGE BOJUNG FILE  (  SUMM ==> BOJUNG )         */   SMI00100
 /*****************************************************************/   SMI00110
 DCL  ORIGIN  FILE  RECORD  KEYED  ENV(VSAM); /*  ORIGINAL FILE     */   SMI00120
 DCL  BOJUNG  FILE  RECORD  KEYED  ENV(VSAM); /*  SUMM ==> BOJUNG   */   SMI00130
 DCL  BUPUM382(8,1)  FLOAT(16)  INIT(         /**********************/   SMI00140
                                   1263,     /**********************/   SMI00150
                                   3238,     /*                    */   SMI00160
                                    758,     /*  BUPUM382  MEANS   */   SMI00170
                                    505,     /*  1 PUMOK BUGAGACHI */   SMI00180
                                   3083,     /*                    */   SMI00190
                                    512,     /*                    */   SMI00200
                                   2571,     /*                    */   SMI00210
                                   2503);    /**********************/   SMI00220
 DCL  BUPUM383(8,2)  FLOAT(16)  INIT(         /**********************/   SMI00230
                                 1027,2107,  /**********************/   SMI00240
                                 2635,5396,  /*                    */   SMI00250
                                 1027,2107,  /*  BUPUM383  MEANS   */   SMI00260
                                    0,   0,  /*  2 PUMOK BUGAGACHI */   SMI00270
                                 2356,5542,  /*                    */   SMI00280
                                 2356, 718,  /*                    */   SMI00290
                                    0,4824,  /*                    */   SMI00300
                                  774, 485); /**********************/   SMI00310
 DCL  BUPUM385(8,2)  FLOAT(16)  INIT(         /**********************/   SMI00320
                                  645, 333,  /*                    */   SMI00330
                                 1657, 851,  /*                    */   SMI00340
                                  454, 333,  /*  BUPUM385  MEANS   */   SMI00350
                                  191;   0,  /*  2 PUMOK BUGAGACHI */   SMI00360
                                 1664, 862,  /*                    */   SMI00370
                                   98, 794,  /*                    */   SMI00380
                                 1566,  68,  /*                    */   SMI00390
                                 1268, 500); /**********************/   SMI00400
 DCL  BUSO382(8)  FLOAT(16)  INIT(            /*  BUSO382 SO-BUNRYU */   SMI00410
                29073, 81240, 15352, 13721,  /*         BUGAGACHI  */   SMI00420
                79184, 66638, 12546, 84061); /**********************/   SMI00430
 DCL BUSO383(8) FLOAT(16) INIT(               /*  BUSO383 SO-BUNRYU */   SMI00440
                70358,152485, 55629, 14729,  /*         BUGAGACHI  */   SMI00450
               212746,102952,109794,292730); /**********************/   SMI00460
 DCL BUSO385(8) FLOAT(16) INIT(               /*  BUSO385 SO-BUNRYU */   SMI00470
                10565, 26581,  6651,  3914,  /*         BUGAGACHI  */   SMI00480
                24891, 10727, 14164, 17984); /**********************/   SMI00490
 DCL BONUM382(1) CHAR(5) INIT                 /* BONUM382 1 PUMMOK  */   SMI00500
                         ('82440');           /**********************/   SMI00510
 DCL BONUM383(2) CHAR(5) INIT                 /* BONUM383 2 PUMMOK  */   SMI00520
                          ('83390','83400');  /**********************/   SMI00530
 DCL BONUM385(2) CHAR(5) INIT                 /* BONUM385 2 PUMMOK  */   SMI00540
                          ('85120','85130');  /**********************/   SMI00550
```

```
DCL BUGASAING(5) FLOAT(16)   INIT                /* BUGAGACHI OF        */  SMI00560
                 (69737,239797,335639,           /* NAIGU SAINGSANJAI   */  SMI00570
                   167510,72287) ;               /**********************/  SMI00580
DCL BUGASOBI(5)  FLOAT(16)   INIT                /* BUGAGACHI OF        */  SMI00590
                 (54517,170324,239242,           /* NAIGU SOBIJAI       */  SMI00600
                   82756,87568) ;                /**********************/  SMI00610
DCL 1 SUMREC,                                                                SMI00620
     2 OKEY        CHAR(10),                                                 SMI00630
     2 SPC         CHAR(1),                                                  SMI00640
     2 OYEAR       PIC'99',                                                  SMI00650
     2 OMONTH      PIC'99',                                                  SMI00660
     2 WEIGHT      FLOAT,                                                    SMI00670
     2 BASEM  FIXED BIN(31),                                                 SMI00680
     2 DATAO(37),                                                            SMI00690
       3 ODATA FIXED BIN(31),                                                SMI00700
       3 JDATA FLOAT,                                                        SMI00710
     2 SIGN        CHAR(1);                                                  SMI00720
DCL  1 KKEY DEF OKEY,                                                        SMI00730
     2 CD    CHAR(1),                                                        SMI00740
     2 GU    CHAR(1),                                                        SMI00750
     2 INDUST CHAR(3),                                                       SMI00760
     2 PUM   CHAR(5);                                                        SMI00770
DCL YUJISU(9,43,37)      FLOAT DEC(16);                                      SMI00780
DCL IYUL(456,75)         FLOAT DEC(16);  /** ORIGIN PUMOK 451 **/           SMI00790
                                         /** BOJUNG PUMOK   5 **/           SMI00800
                                         /** TOTAL  PUMOK 456 **/           SMI00810
DCL SPCJISU(9,5,37)      FLOAT DEC(16);                                      SMI00820
DCL HRJISU(28,3,37)      FLOAT DEC(16);                                      SMI00830
DCL BASE                 FLOAT DEC(16);                                      SMI00840
DCL DATA(37)             FLOAT DEC(16);                                      SMI00850
DCL JISU                 PIC'99999V9';                                       SMI00860
DCL A10(25) CHAR(5) INIT('62050','81090','81100','81150','82070',            SMI00870
 '82090','82100','82120','82130','82140','82150','82160','82170',            SMI00880
 '82190','82200','82250','82270','82280','82290','82300','82310',            SMI00890
 '82340','82350','84060','84070');                                           SMI00900
DCL GUB(20) CHAR(1) INIT('A','B','C','D','E','F','G','H','I','J','K',         SMI00910
    'L','M','N','O','P','Q','R','S','T');                                    SMI00920
DCL HRCD(28)  CHAR(4) INIT('2311','2313','2314','2321','2322','2323',         SMI00930
  '2324','2331','2332','1341','2342','1351','1352','1353','1354',            SMI00940
  '2355','2356','1361','1362','1369','1371','1372','1381','1382',            SMI00950
  '1383','1384','1385','2390');                                              SMI00960
DCL OGU                  CHAR(1);                                            SMI00970
DCL OINDUST              CHAR(3);                                            SMI00980
DCL SOSUM(37)          FLOAT DEC(16),                                        SMI00990
    JUSUM(37)          FLOAT DEC(16),                                        SMI01000
    DASUM(37)          FLOAT DEC(16),                                        SMI01010
    CHSUM(37)          FLOAT DEC(16);                                        SMI01020
DCL Q                    PIC'99';                                            SMI01030
DCL EXCP(3,37)         FLOAT DEC(16);                                        SMI01040
DCL YUCD(43) CHAR(3) INIT('210','230','290','200','311','313','314',         SMI01050
  '310','321','322','323','324','320','331','332','330','341','342',         SMI01060
  '340','351','352','353','354','355','356','350','361','362','369',         SMI01070
  '360','371','372','370','381','382','383','384','385','380','390',         SMI01080
  '300','400','000');                                                        SMI01090
DCL SPCD(9)  CHAR(3) INIT('000','800','810','820','821','822','900',         SMI01100
```

```
   '910','920');                                                        SMI01110
 DCL INT                  BIT(1) INIT('1'B);                            SMI01120
 YUJISU=0; IYUL=0; SPCJISU=0; HRJISU=0; BASE=0; DATA=0;                 SMI01130
 SOSUM=0; JUSUM=0; DASUM=0; CHSUM=0;  EXCP=0;                           SMI01140
  ON KEY(BOJUNG) BEGIN;                                                 SMI01150
   IF ONCODE=51 THEN PUT EDIT(OKEY,'NOT FOUND')(A(10),A(10));           SMI01160
  END;                                                                  SMI01170
  OPEN FILE(BOJUNG) SEQUENTIAL UPDATE;                                  SMI01180
  OPEN FILE(ORIGIN) DIRECT UPDATE;      /********  BOJUNG    ******/    SMI01190
  READ FILE(BOJUNG) INTO(SUMREC);                                       SMI01200
  OGU=GU;  J=0; I=0; K = 0;  BASE=BASEM;                                SMI01210
   DATA(*)=ODATA(*) ;  CALL MUL10;                                      SMI01220
  DO UNTIL(GU='I');  I=I+1;                                             SMI01230
    DO WHILE(OGU=GU);                                                   SMI01240
      IF SUBSTR(PUM,5,1)='0' & PUM¬='00000' THEN DO;                    SMI01250
        IF INT THEN DO; OINDUST=INDUST; INT='0'B; END;                  SMI01260
        IF BASE=0 THEN DATA=0;                                          SMI01270
        ELSE DATA=DATA/BASE*100;                                        SMI01280
      DO LEE = 1 TO 37 ;                                                SMI01290
        JISU=DATA(LEE)+0.05; DATA(LEE)=JISU; JDATA(LEE)=JISU;           SMI01300
      END;                                                              SMI01310
        REWRITE FILE(BOJUNG) FROM(SUMREC);                              SMI01320
        IF I=8 THEN DO;                                                 SMI01330
          J=J+1;                                                        SMI01340
        DO LEE = 1 TO 37;                                               SMI01350
          IYUL(J,LEE+1)=DATA(LEE);                                      SMI01360
        END;                                                            SMI01370
          IYUL(J,1)=WEIGHT;                                             SMI01380
        END;                                                            SMI01390
        ELSE IF I=5  THEN DO;                                           SMI01400
          J=J+1;                                                        SMI01410
        DO LEE = 1 TO 37;                                               SMI01420
          IYUL(J,LEE+38)=DATA(LEE);                                     SMI01430
        END;                                                            SMI01440
        END;                                                            SMI01450
        IF OINDUST¬=INDUST THEN DO;                                     SMI01460
         IF SUBSTR(OINDUST,3,1)='0' THEN CALL CALRTN(JUSUM,DASUM);      SMI01470
         ELSE DO ;                                                      SMI01480
          IF OINDUST = '382' THEN SOSUM = 0;          /*  BOJUNG     */ SMI01490
            ELSE IF OINDUST = '383' THEN SOSUM = 0;   /*  YUJISU = 0 */ SMI01500
              ELSE IF OINDUST = '385' THEN SOSUM = 0; /*             */ SMI01510
          CALL CALRTN(SOSUM,JUSUM);                                     SMI01520
          IF SUBSTR(OINDUST,1,2)¬=SUBSTR(INDUST,1,2) THEN               SMI01530
            CALL CALRTN(JUSUM,DASUM); END;                              SMI01540
            IF SUBSTR(OINDUST,1,1)¬=SUBSTR(INDUST,1,1) THEN             SMI01550
            CALL CALRTN(DASUM,CHSUM);                                   SMI01560
          OINDUST=INDUST;                                               SMI01570
        END;                                                            SMI01580
 IF SUBSTR(INDUST,1,1)='2' | INDUST='390' THEN JUSUM=JUSUM+DATA*WEIGHT; SMI01590
        ELSE IF SUBSTR(INDUST,1,1)='3' THEN SOSUM=SOSUM+DATA*WEIGHT;    SMI01600
        ELSE DASUM=DASUM+DATA*WEIGHT;                                   SMI01610
        /*************************************************************/ SMI01620
        /******************** SPECIAL JACKUP ************************/   SMI01630
        /*************************************************************/ SMI01640
        IF I = 1 THEN L = 1 ;                        /*  SAING SAN ******/ SMI01650
```

```
      ELSE IF I = 5 THEN L =2 ;                    /*  CHUL HA    ******/  SMI01660
      ELSE IF I = 8 THEN L =3 ;                    /*  JAI GO     ******/  SMI01670
      ELSE IF I = 6 THEN L =4 ;                    /*  NAI SU     ******/  SMI01680
      ELSE IF I = 7 THEN L =5 ;                    /*  SU CHUL    ******/  SMI01690
      IF I=1 | I=5 | I=8 | I=6 | I=7 THEN DO ; /*******************/  SMI01700
      IF SPC¬=' ' THEN DO;                         /*******************/  SMI01710
        Q=SPC;                                     /*******************/  SMI01720
        SPCJISU(Q,L,*)=SPCJISU(Q,L,*)+DATA(*)*WEIGHT;  /************/  SMI01730
      END; END ;                                   /************/  SMI01740
      /*************************************************************/  SMI01750
      /********************** SPECIAL ***************************/  SMI01760
      /*************************************************************/  SMI01770
      IF I=1 | I=5 | I=8 THEN DO;                                      SMI01780
      IF PUM>'84010' & PUM<'84090' THEN EXCP(L,*)=EXCP(L,*)+DATA(*)*   SMI01790
       WEIGHT;                                                         SMI01800
    END;   END;                                                        SMI01810
    READ FILE(BOJUNG) INTO(SUMREC);                                    SMI01820
    DATA(*)=ODATA(*);                                                  SMI01830
    BASE=BASEM;  CALL MUL10;                                           SMI01840
  END;                                                                 SMI01850
  CALL CALRTN(DASUM,CHSUM);                                            SMI01860
  CALL CALRTN(CHSUM,CHSUM); CHSUM=0;                                   SMI01870
  OGU=GU; J=0;  K=0;  INT='1'B;                                        SMI01880
END;                                                                   SMI01890
L=0;  I=9; OINDUST=INDUST;                                             SMI01900
DO WHILE(GU='I');                                                      SMI01910
  IF PUM¬='00000' THEN DO;                                             SMI01920
    IF INT THEN DO;  INT='0'B; OINDUST=INDUST; END;                    SMI01930
    L=L+1;  IF WEIGHT=0 THEN GOTO JYY;                                 SMI01940
  DO LEE = 1 TO 37;                                                    SMI01950
    IF IYUL(L,LEE+38)=0 THEN DATA(LEE)=0;                              SMI01960
    ELSE DATA(LEE)=IYUL(L,LEE+1)/IYUL(L,LEE+38)*100;                   SMI01970
  END;                                                                 SMI01980
  DO LEE = 1 TO 37;                                                    SMI01990
    JISU=DATA(LEE)+0.05; JDATA(LEE)=JISU; DATA(LEE)=JISU;              SMI02000
 END;                                                                  SMI02010
    REWRITE FILE(BOJUNG) FROM(SUMREC);                                 SMI02020
  END;                                                                 SMI02030
 JYY: READ FILE(BOJUNG) INTO(SUMREC);                                  SMI02040
END;                                                                   SMI02050
 CALL CALRTN(DASUM,CHSUM);                                             SMI02060
 CALL CALRTN(CHSUM,CHSUM);                                             SMI02070
CLOSE FILE(BOJUNG);                                                    SMI02080
OPEN FILE(BOJUNG) DIRECT UPDATE; J=0;                                  SMI02090
DO Q=1 TO 8;  K=0;                                                     SMI02100
  IF Q = 1 | Q = 5 | Q = 8 THEN J = J + 1;                             SMI02110
  DO I = 1 TO 43;                                                      SMI02120
  /*****************************************************************/SMI02130
  /*  ( SANUP,JUNGONGEP,CHELDOSUNBAG_JEOI )  JISU BOJUNG ROUTINE    */SMI02140
  /*****************************************************************/SMI02150
    IF I = 35 THEN DO ;            /* PUMOK = 382                   */SMI02160
      DATA = 0 ;                   /* ............................  */SMI02170
      DO LK = 1 TO 1 ;                                /* 1 PUMOK    */SMI02180
        OKEY ='2'||GUB(Q)||YUCD(I)||BONUM382(LK);     /* BUGAGACHI  */SMI02190
        READ FILE(BOJUNG) INTO(SUMREC) KEY(OKEY) ;    /* NUJUK      */SMI02200
```

```
        DATA = DATA + BUPUM382(Q,LK) * JDATA / 100 ; /*  ROUTINE   */SMI02210
      END ;                                                          SMI02220
      SUBSTR(OKEY,6,5) = '00000';                                    SMI02230
      READ FILE(ORIGIN) INTO(SUMREC) KEY(OKEY) ;      /*   BOJUNG   */SMI02240
      DATA=(DATA+BUSO382(Q)*JDATA/100)/BUSO382(Q)*100; /*  JISU    */SMI02250
      YUJISU(Q,35,*) = DATA(*) * WEIGHT ;      /* 382 BUNRYU YUJISU */SMI02260
      YUJISU(Q,39,*)=YUJISU(Q,39,*)+YUJISU(Q,35,*);  /* 380 BUNRYU */SMI02270
      YUJISU(Q,41,*)=YUJISU(Q,41,*)+YUJISU(Q,35,*);  /* 300 BUNRYU */SMI02280
      YUJISU(Q,43,*)=YUJISU(Q,43,*)+YUJISU(Q,35,*);  /* 000 BUNRYU */SMI02290
      READ FILE(BOJUNG) INTO(SUMREC) KEY(OKEY) ;                     SMI02300
      GOTO RRK ;                                                     SMI02310
    END ;                                                            SMI02320
    IF I = 36 THEN DO ;            /* PUMOK = 383                   */SMI02330
      DATA = 0 ;                   /* ............................. */SMI02340
      DO LK = 1 TO 2 ;                                /* 2 PUMOK    */SMI02350
        OKEY ='2'||GUB(Q)||YUCD(I)||BONUM383(LK);     /* BUGAGACHI  */SMI02360
        READ FILE(BOJUNG) INTO(SUMREC) KEY(OKEY) ;    /* NUJUK      */SMI02370
        DATA = DATA + BUPUM383(Q,LK) * JDATA / 100 ; /*  ROUTINE   */SMI02380
      END ;                                                          SMI02390
      SUBSTR(OKEY,6,5) = '00000';                                    SMI02400
      READ FILE(ORIGIN) INTO(SUMREC) KEY(OKEY) ;      /*  BOJUNG    */SMI02410
      DATA=(DATA+BUSO383(Q)*JDATA/100)/BUSO383(Q)*100; /*  JISU    */SMI02420
      YUJISU(Q,36,*) = DATA(*) * WEIGHT ;      /* 383 BUNRYU YUJISU */SMI02430
      YUJISU(Q,39,*)=YUJISU(Q,39,*)+YUJISU(Q,36,*);  /* 380 BUNRYU */SMI02440
      YUJISU(Q,41,*)=YUJISU(Q,41,*)+YUJISU(Q,36,*);  /* 300 BUNRYU */SMI02450
      YUJISU(Q,43,*)=YUJISU(Q,43,*)+YUJISU(Q,36,*);  /* 000 BUNRYU */SMI02460
      READ FILE(BOJUNG) INTO(SUMREC) KEY(OKEY) ;                     SMI02470
      GOTO RRK ;                                                     SMI02480
    END ;                                                            SMI02490
    IF I = 38 THEN DO ;            /* PUMOK = 385                   */SMI02500
      DATA = 0 ;                   /* ............................. */SMI02510
      DO LK = 1 TO 2 ;                                /* 2 PUMOK    */SMI02520
        OKEY ='2'||GUB(Q)||YUCD(I)||BONUM385(LK);     /* BUGAGACHI  */SMI02530
        READ FILE(BOJUNG) INTO(SUMREC) KEY(OKEY) ;    /*  NUJUK     */SMI02540
        DATA = DATA + BUPUM385(Q,LK) * JDATA / 100 ; /*  ROUTINE   */SMI02550
      END ;                                                          SMI02560
      SUBSTR(OKEY,6,5) = '00000';                                    SMI02570
      READ FILE(ORIGIN) INTO(SUMREC) KEY(OKEY) ;      /*   BOJUNG   */SMI02580
      DATA=(DATA+BUSO385(Q)*JDATA/100)/BUSO385(Q)*100; /*   JISU   */SMI02590
      YUJISU(Q,38,*) = DATA(*) * WEIGHT ;    /* 385 BUNRYU YUJISU   */SMI02600
      YUJISU(Q,39,*)=YUJISU(Q,39,*)+YUJISU(Q,38,*);  /* 380 BUNRYU */SMI02610
      YUJISU(Q,41,*)=YUJISU(Q,41,*)+YUJISU(Q,38,*);  /* 300 BUNRYU */SMI02620
      YUJISU(Q,43,*)=YUJISU(Q,43,*)+YUJISU(Q,38,*);  /* 000 BUNRYU */SMI02630
      READ FILE(BOJUNG) INTO(SUMREC) KEY(OKEY) ;                     SMI02640
      GOTO RRK ;                                                     SMI02650
    END ;                                                            SMI02660
  /*************************************************************/    SMI02670
    OKEY='2' || GUB(Q) || YUCD(I);                                   SMI02680
    OKEY=TRANSLATE(OKEY,'0',' ');                                    SMI02690
    READ FILE(BOJUNG) INTO(SUMREC) KEY(OKEY);                        SMI02700
    IF WEIGHT=0 THEN DATA=0;                                         SMI02710
    ELSE DATA(*)=YUJISU(Q,I,*)/WEIGHT;                               SMI02720
RRK :                                                   /* BOJUNG */ SMI02730
  DO LEE = 1 TO 37;                                                  SMI02740
    JISU=DATA(LEE)+0.05; JDATA(LEE)=JISU;                            SMI02750
```

```
    END;                                                                SMI02760
      PUT SKIP EDIT(OKEY,WEIGHT,(JDATA(Z) DO Z=35 TO 37))               SMI02770
         (A(10),4 F(10,1));                                             SMI02780
      REWRITE FILE(BOJUNG) FROM(SUMREC) KEY(OKEY);                      SMI02790
      IF Q=1 | Q=5 | Q=8 THEN DO;                                       SMI02800
      IF SUBSTR(INDUST,3,1)¬='0'  THEN DO;                              SMI02810
        K=K+1;                                                          SMI02820
        HRJISU(K,J,*)=YUJISU(Q,I,*);                                    SMI02830
      END;                                                              SMI02840
      ELSE IF INDUST='390'  THEN DO;                                    SMI02850
        K=K+1;                                                          SMI02860
        HRJISU(K,J,*)=YUJISU(Q,I,*);                                    SMI02870
      END;                                                              SMI02880
     IF YUCD(I)='000' | YUCD(I)='300' | YUCD(I)='380' | YUCD(I)='384'   SMI02890
      THEN DO;                                                          SMI02900
      OKEY='2' || GUB(J+15) || YUCD(I);                                 SMI02910
      OKEY=TRANSLATE(OKEY,'0',' ');                                     SMI02920
      READ FILE(BOJUNG) INTO(SUMREC) KEY(OKEY);                         SMI02930
      IF WEIGHT=0 THEN DATA=0;                                          SMI02940
      ELSE DATA(*)=(YUJISU(Q,I,*)-EXCP(J,*))/WEIGHT;                    SMI02950
    DO LEE = 1 TO 37;                                                   SMI02960
      JISU=DATA(LEE)+0.05; JDATA(LEE)=JISU;                             SMI02970
    END;                                                                SMI02980
      REWRITE FILE(BOJUNG) FROM(SUMREC) KEY(OKEY);                      SMI02990
     END;                                                               SMI03000
     END;                                                               SMI03010
      IF Q=5 THEN YUJISU(9,I,*)=DATA(*);                                SMI03020
      IF Q=8 THEN DO;                                                   SMI03030
        YUJISU(8,I,*)=DATA(*);                                          SMI03040
        IF SUBSTR(INDUST,1,1)='0' THEN DO;                              SMI03050
      OKEY='2I' || YUCD(I);                                             SMI03060
      OKEY=TRANSLATE(OKEY,'0',' ');                                     SMI03070
      READ FILE(BOJUNG) INTO(SUMREC) KEY(OKEY);                         SMI03080
      IF WEIGHT=0 THEN DATA=0;                                          SMI03090
      ELSE DATA=CHSUM/WEIGHT;                                           SMI03100
    DO LEE = 1 TO 37;                                                   SMI03110
      JISU=DATA(LEE)+0.05; JDATA(LEE)=JISU;                             SMI03120
    END;                                                                SMI03130
      REWRITE FILE(BOJUNG) FROM(SUMREC) KEY(OKEY);                      SMI03140
        END;                                                            SMI03150
      ELSE IF SUBSTR(INDUST,2,1)='0' THEN DO;                           SMI03160
      OKEY='2I' || YUCD(I);                                             SMI03170
      OKEY=TRANSLATE(OKEY,'0',' ');                                     SMI03180
      READ FILE(BOJUNG) INTO(SUMREC) KEY(OKEY);                         SMI03190
      IF WEIGHT=0 THEN DATA=0;                                          SMI03200
      ELSE DATA=DASUM/WEIGHT;                                           SMI03210
    DO LEE = 1 TO 37;                                                   SMI03220
      JISU=DATA(LEE)+0.05; JDATA(LEE)=JISU;                             SMI03230
    END;                                                                SMI03240
      REWRITE FILE(BOJUNG) FROM(SUMREC) KEY(OKEY);                      SMI03250
      CHSUM=CHSUM+DATA*WEIGHT;  DASUM=0;                                SMI03260
        END;                                                            SMI03270
        ELSE DO;                                                        SMI03280
          IF SUBSTR(INDUST,1,1)='2' | SUBSTR(INDUST,1,1)='4' |          SMI03290
             SUBSTR(INDUST,1,2)='39' THEN DO;                           SMI03300
```

```
      OKEY='2I' || YUCD(I);                                          SMI03310
      OKEY=TRANSLATE(OKEY,'0',' ');                                  SMI03320
      READ FILE(BOJUNG) INTO(SUMREC) KEY(OKEY);                      SMI03330
     DO LEE = 1 TO 37;                                               SMI03340
      IF YUJISU(9,I,LEE)=0 THEN DATA(LEE)=0;                         SMI03350
      ELSE DATA(LEE)=YUJISU(8,I,LEE)/YUJISU(9,I,LEE)*100;            SMI03360
     END;                                                            SMI03370
    DO LEE = 1 TO 37;                                                SMI03380
      JISU=DATA(LEE)+0.05; JDATA(LEE)=JISU;                          SMI03390
    END;                                                             SMI03400
      REWRITE FILE(BOJUNG) FROM(SUMREC) KEY(OKEY);                   SMI03410
          DASUM=DASUM+DATA*WEIGHT;                                   SMI03420
          END;                                                       SMI03430
          ELSE IF SUBSTR(INDUST,3,1)='0' THEN DO;                    SMI03440
      OKEY='2I' || YUCD(I);                                          SMI03450
      OKEY=TRANSLATE(OKEY,'0',' ');                                  SMI03460
      READ FILE(BOJUNG) INTO(SUMREC) KEY(OKEY);                      SMI03470
      IF WEIGHT=0 THEN DATA=0;                                       SMI03480
      ELSE DATA(*)=JUSUM/WEIGHT;                                     SMI03490
    DO LEE = 1 TO 37;                                                SMI03500
      JISU=DATA(LEE)+0.05; JDATA(LEE)=JISU;                          SMI03510
    END;                                                             SMI03520
      REWRITE FILE(BOJUNG) FROM(SUMREC) KEY(OKEY);                   SMI03530
        DASUM=DASUM+DATA*WEIGHT;  JUSUM=0;                           SMI03540
          END;                                                       SMI03550
          ELSE DO;                                                   SMI03560
      OKEY='2I' || YUCD(I);                                          SMI03570
      OKEY=TRANSLATE(OKEY,'0',' ');                                  SMI03580
      READ FILE(BOJUNG) INTO(SUMREC) KEY(OKEY);                      SMI03590
     DO LEE = 1 TO 37;                                               SMI03600
      IF YUJISU(9,I,LEE)=0 THEN DATA(LEE)=0;                         SMI03610
      ELSE DATA(LEE)=YUJISU(8,I,LEE)/YUJISU(9,I,LEE)*100;            SMI03620
     END;                                                            SMI03630
    DO LEE = 1 TO 37;                                                SMI03640
      JISU=DATA(LEE)+0.05; JDATA(LEE)=JISU;                          SMI03650
    END;                                                             SMI03660
      REWRITE FILE(BOJUNG) FROM(SUMREC) KEY(OKEY);                   SMI03670
       JUSUM=JUSUM+DATA*WEIGHT;                                      SMI03680
          END;                                                       SMI03690
         END;                                                        SMI03700
       END;                                                          SMI03710
    END;                                                             SMI03720
  END ;    K=0;                                                      SMI03730
  /******************* SPECIAL_JISU BOJUNG ROUTINE *******************/SMI03740
    DO Q = 1 TO 5 ;                    /*                           */SMI03750
       IF Q =1 THEN DO                 /*  Q MEANS                  */SMI03760
               L1 = 10 ;               /*    SUBSCRIPT OF           */SMI03770
               L2 = 1 ;                /*        SPCJISU(*,Q,*)     */SMI03780
               END  ;                  /*  OR  SUBSCRIPT OF         */SMI03790
       ELSE IF Q = 2 THEN DO ;         /*      BUGASAING(Q)         */SMI03800
               L1 = 11 ;               /*  OR  SUBSCRIPT OF         */SMI03810
               L2 = 5 ;                /*      BUGASOBI(Q)          */SMI03820
               END  ;                  /*                           */SMI03830
       ELSE IF Q = 3 THEN DO ;         /*  L1 MEANS GUB NO OF SPECIAL */SMI03840
               L1 = 12 ;               /*    BUNRYU IN ORIGIN FILE  */SMI03850
```

```
            L2 = 8 ;                  /* (10=J,11=K,12=L,19=S,20=T)   */SMI03860
            END  ;                    /*                              */SMI03870
      ELSE IF Q = 4 THEN DO ;         /*  L2 MEANS GUB NO             */SMI03880
            L1 = 19 ;                 /*            IN BOJUNG FILE    */SMI03890
            L2 = 6 ;                  /* (1=A,5=E,8=H,6=F,7=G)        */SMI03900
            END  ;                    /*  OR  SUBSCRIPT OF            */SMI03910
      ELSE IF Q = 5 THEN DO ;         /*      BUPUM382(L2,LK)         */SMI03920
            L1 = 20 ;                 /*      BUPUM383(L2,LK)         */SMI03930
            L2 = 7 ;                  /*      BUPUM385(L2,LK)         */SMI03940
            END  ;                    /*******************************/SMI03950
                        /*****************************************/SMI03960
      DATA = 0 ;                                        /*          */SMI03970
      DO LK = 1 TO 1 ;                                  /*  NAIGU-  */SMI03980
         OKEY = '2'||GUB(L2)||'382'||BONUM382(LK) ;     /* SAINGSAN */SMI03990
         READ FILE(BOJUNG) INTO(SUMREC) KEY(OKEY) ;     /* SPC JISU */SMI04000
         DATA = DATA + BUPUM382(L2,LK) * JDATA/100 ;    /*          */SMI04010
      END ;                                             /*          */SMI04020
      DO LK = 1 TO 2 ;                                  /*          */SMI04030
         OKEY = '2'||GUB(L2)||'385'||BONUM385(LK) ;     /*          */SMI04040
         READ FILE(BOJUNG) INTO(SUMREC) KEY(OKEY) ;     /*          */SMI04050
         DATA = DATA + BUPUM385(L2,LK) * JDATA/100 ;    /*          */SMI04060
      END ;                                             /*          */SMI04070
      OKEY = '2'||GUB(L1)||'810'||'00000' ;             /*          */SMI04080
      READ FILE(ORIGIN) INTO(SUMREC) KEY(OKEY) ;        /*          */SMI04090
      DATA = ( DATA + BUGASAING(Q) * JDATA /100 )       /*          */SMI04100
                  / BUGASAING(Q) * 100 ;                /*          */SMI04110
      SPCJISU(3,Q,*) = DATA (*) * WEIGHT ;              /*          */SMI04120
                        /*****************************************/SMI04130
      DATA = 0 ;                                        /*          */SMI04140
      DO LK = 1 TO 2 ;                                  /*  NAIGU-  */SMI04150
         OKEY = '2'||GUB(L2)||'383'||BONUM383(LK) ;     /*  SOBIJAI */SMI04160
         READ FILE(BOJUNG) INTO(SUMREC) KEY(OKEY) ;     /* SPC JISU */SMI04170
         DATA = DATA + BUPUM383(L2,LK) * JDATA / 100 ;  /*          */SMI04180
      END ;                                             /*          */SMI04190
      OKEY = '2'||GUB(L1)||'910'||'00000' ;             /*          */SMI04200
      READ FILE(ORIGIN) INTO(SUMREC) KEY(OKEY) ;        /*          */SMI04210
      DATA = ( DATA + BUGASOBI(Q) * JDATA / 100 )       /*          */SMI04220
                  / BUGASOBI(Q) * 100 ;                 /*          */SMI04230
      SPCJISU(8,Q,*) = DATA (*) * WEIGHT ;              /*          */SMI04240
   END ;                                                               SMI04250
/*************************************************************/        SMI04260
SPCJISU(7,*,*)=SPCJISU(8,*,*)+SPCJISU(9,*,*);                          SMI04270
SPCJISU(4,*,*)=SPCJISU(5,*,*)+SPCJISU(6,*,*);                          SMI04280
SPCJISU(2,*,*)=SPCJISU(3,*,*)+SPCJISU(4,*,*);                          SMI04290
SPCJISU(1,*,*)=SPCJISU(2,*,*)+SPCJISU(7,*,*);                          SMI04300
DO Q=1 TO 3;                                                           SMI04310
K=K+1;                                                                 SMI04320
   DO I=1 TO 28 ;                                                      SMI04330
      OKEY='2' || GUB(K+12) || HRCD(I);                                SMI04340
      OKEY=TRANSLATE(OKEY,'0',' ');                                    SMI04350
      READ FILE(BOJUNG) INTO(SUMREC) KEY(OKEY);                        SMI04360
      IF WEIGHT=0 THEN DATA=0;                                         SMI04370
      ELSE DATA(*)=HRJISU(I,K,*)/WEIGHT;                               SMI04380
   DO LEE = 1 TO 37;                                                   SMI04390
     JISU=DATA(LEE)+0.05; JDATA(LEE)=JISU;                             SMI04400
```

```
    END;                                                                SMI04410
       REWRITE FILE(BOJUNG) FROM(SUMREC) KEY(OKEY);                     SMI04420
       IF SUBSTR(HRCD(I),1,1)='1' THEN SOSUM(*)=SOSUM(*)+HRJISU(I,K,*); SMI04430
       ELSE JUSUM(*)=JUSUM(*)+HRJISU(I,K,*);                            SMI04440
    END;                                                                SMI04450
    OKEY='2' || GUB(K+12) || '1';                                       SMI04460
    OKEY=TRANSLATE(OKEY,'0',' ');                                       SMI04470
    READ FILE(BOJUNG) INTO(SUMREC) KEY(OKEY);                           SMI04480
    IF WEIGHT=0 THEN DATA=0;                                            SMI04490
    ELSE DATA=SOSUM/WEIGHT;                                             SMI04500
    DO LEE = 1 TO 37;                                                   SMI04510
      JISU=DATA(LEE)+0.05; JDATA(LEE)=JISU;                             SMI04520
    END;                                                                SMI04530
    DASUM=DASUM+SOSUM;                                                  SMI04540
    REWRITE FILE(BOJUNG) FROM(SUMREC) KEY(OKEY);                        SMI04550
    OKEY='2' || GUB(K+12) || '2';                                       SMI04560
    OKEY=TRANSLATE(OKEY,'0',' ');                                       SMI04570
    READ FILE(BOJUNG) INTO(SUMREC) KEY(OKEY);                           SMI04580
    IF WEIGHT=0 THEN DATA=0;                                            SMI04590
    ELSE DATA=JUSUM/WEIGHT;                                             SMI04600
    DO LEE = 1 TO 37;                                                   SMI04610
      JISU=DATA(LEE)+0.05; JDATA(LEE)=JISU;                             SMI04620
    END;                                                                SMI04630
    DASUM=DASUM+JUSUM;                                                  SMI04640
    REWRITE FILE(BOJUNG) FROM(SUMREC) KEY(OKEY);                        SMI04650
    OKEY='2' || GUB(K+12) ;                                             SMI04660
    OKEY=TRANSLATE(OKEY,'0',' ');                                       SMI04670
    READ FILE(BOJUNG) INTO(SUMREC) KEY(OKEY);                           SMI04680
    IF WEIGHT=0 THEN DATA=0;                                            SMI04690
    ELSE DATA=DASUM/WEIGHT;                                             SMI04700
    DO LEE = 1 TO 37;                                                   SMI04710
      JISU=DATA(LEE)+0.05; JDATA(LEE)=JISU;                             SMI04720
    END;                                                                SMI04730
    REWRITE FILE(BOJUNG) FROM(SUMREC) KEY(OKEY);                        SMI04740
    SOSUM=0;  JUSUM=0;  DASUM=0;                                        SMI04750
  END;                                                                  SMI04760
    /*****************************************************/             SMI04770
    /************** SPECIAL JAKUP ************************/              SMI04780
    /*****************************************************/             SMI04790
    DO K = 1 TO 5 ;                                                     SMI04800
       IF K < 4 THEN L = K + 9 ;                                        SMI04810
                ELSE L = K +15 ;                                        SMI04820
    DO I=1 TO 9;                                                        SMI04830
      OKEY='2' || GUB(L) || SPCD(I);                                    SMI04840
      OKEY=TRANSLATE(OKEY,'0',' ');                                     SMI04850
      READ FILE(BOJUNG) INTO(SUMREC) KEY(OKEY);                         SMI04860
      IF WEIGHT=0 THEN DATA=0;                                          SMI04870
      ELSE DATA(*)=SPCJISU(I,K,*)/WEIGHT;                               SMI04880
      DO LEE = 1 TO 37;                                                 SMI04890
        JISU=DATA(LEE)+0.05; JDATA(LEE)=JISU;                           SMI04900
      END;                                                              SMI04910
      REWRITE FILE(BOJUNG) FROM(SUMREC) KEY(OKEY);                      SMI04920
    END;                                                                SMI04930
    END;                                                                SMI04940
    /*****************************************************/             SMI04950
```

```
    /******************* SPECIAL ************************/                SMI04960
    /****************************************************/                SMI04970
  CLOSE FILE(BOJUNG);                                                     SMI04980
  CLOSE FILE(ORIGIN);              /******   BOJUNG   ********/          SMI04990
 CALRTN: PROC(ASUM,BSUM);                                                 SMI05000
  DCL ASUM(37)  FLOAT DEC(16);                                            SMI05010
  DCL BSUM(37)  FLOAT DEC(16);                                            SMI05020
  K=K+1;                                                                  SMI05030
  YUJISU(I,K,*)=ASUM(*);                                                  SMI05040
  BSUM=BSUM+ASUM;                                                         SMI05050
  ASUM=0;                                                                 SMI05060
 END CALRTN;                                                              SMI05070
 MUL10: PROC;                                                             SMI05080
  DO M=1 TO 25;                                                           SMI05090
    IF A10(M)=PUM THEN DATA=DATA*10;                                      SMI05100
  END;                                                                    SMI05110
 END MUL10;                                                               SMI05120
 LASS:                                                                    SMI05130
 END JISURTN;                                                             SMI05140
/*                                                                        SMI05150
//LKED.SYSLMOD DD DSN=USER.LOADLIB(SMIBJ62),DISP=SHR                      SMI05160
//                                                                        SMI05170
```

SMIWMV62

보정 FILE MOVE RUN

이하 보정 6개품목

VTR, 전자레인지, 전자계측기, 전자복사기, 컴퓨터 및 주변기기, 특수운동화의 보정지수 산출키 위해 작업

```
//SMIWMV62 JOB CLASS=3                                          SMI00010
//JOBCAT DD DSN=SMICT12,DISP=SHR                                SMI00020
//  EXEC PGM=SMIMV12                                            SMI00030
//STEPLIB DD DSN=USER.LOADLIB,DISP=SHR                          SMI00040
//SYSPRINT DD SYSOUT=A                                          SMI00050
//SUMF DD DSN=BOJUNG62,DISP=SHR                                 SMI00060
//                                                              SMI00070
```

SMIWSM62

보정지수물량 RUN

```
//SMIWSM62 JOB CLASS=3                                          SMI00010
//JOBCAT DD DSN=SMICT12,DISP=SHR                                SMI00020
//  EXEC PGM=SMISM62                                            SMI00030
//STEPLIB DD DSN=USER.LOADLIB,DISP=SHR                          SMI00040
//SYSOUT DD SYSOUT=A                                            SMI00050
//SYSPRINT DD SYSOUT=A                                          SMI00060
//SORTWK01 DD UNIT=3350,SPACE=(CYL,10)                          SMI00070
//SUMF  DD DSN=BOJUNG62,DISP=SHR                                SMI00080
//EDIT2 DD DSN=SMIJN12,DISP=SHR                                 SMI00090
//EDIT3 DD DSN=SMIGM12,DISP=SHR                                 SMI00100
//SYSIN DD *                                                    SMI00110
0823 1078 1377 1161 0967 1148 1078 1377 1161 0967 1144 0031 0031 SMI00120
/*   (1)  (2)  (3)  (4)  (5)  (6)  (7)  (8)  (9)  (10) (11) (12) SMI00130
//                                                               SMI00140
```

전월지수

금월지수

"55080" 특수운동화 전월지수

"55080" 특수운동화 금월지수

월 (MONTH)

CARD

※ 21 : 사업체별 생산량
23 : 생산 TOTAL
41 : 사업체별 고용
43 : 고용 TOTAL
61 : 사업체별 능력
63 : 능력 TOTAL
22 : 총 괄

ROUTINE 업무 SMIWSM66과 DATA가 같아야 한다.

전월 금월 지수에서 (1), (2), (6), (7)은 "22010" (의약품)이며

(3), (8)은 "52060"

(4), (9)은 "82440"

(5), (10)은 "85120"

(11), (12)은 "55082"(특수운동화, 전월, 금월지수)이다.

※ 특수운동화 (55080)에 승수산출방법

55080 ⟨ 55081 (일반운동화)
55082 (특수운동화)

"55082"의 물량에 승수를 곱해서

"55081"의 물량을 더하여 산출

SMIWIX62

보정지수산출 RUN

```
//SMIWIX62 JOB CLASS=A,TYPRUN=HOLD                                      SMI00010
//JOBCAT DD DSN=SMICT12,DISP=SHR                                        SMI00020
//   EXEC PGM=SMIBJ62                                                   SMI00030
//STEPLIB DD DSN=USER.LOADLIB,DISP=SHR                                  SMI00040
//SYSPRINT DD SYSOUT=A                                                  SMI00050
//ORIGIN DD DSN=SMISM12,DISP=SHR                                        SMI00060
//BOJUNG DD DSN=BOJUNG62,DISP=SHR                                       SMI00070
//   EXEC PLIFCLG                                                       SMI00080
//PLI.SYSIN DD *                                                        SMI00090
* PROCESS GS,NEST,A,XREF,OPT(2);                                        SMI00100
  JISURTN: PROC OPTIONS(MAIN);                                          SMI00110
 DCL BOJUNG FILE RECORD KEYED ENV(VSAM);                                SMI00120
 DCL MREC   CHAR(320);                                                  SMI00130
 DCL OKEY CHAR(10) DEF MREC;                                            SMI00140
 DCL GU   CHAR(1) DEF MREC POS(2);                                      SMI00150
 DCL AREC(1500)       CHAR(320);                                        SMI00160
  ON KEY(BOJUNG) BEGIN;                                                 SMI00170
   IF ONCODE=52 THEN PUT EDIT(OKEY,'DUPLICATE')(A(10),A(10));           SMI00180
  END;                                                                  SMI00190
  ON ENDFILE(BOJUNG) GOTO LT;                                           SMI00200
  OPEN FILE(BOJUNG) SEQUENTIAL INPUT;  K=0;                             SMI00210
  ARR: READ FILE(BOJUNG) INTO(MREC);                                    SMI00220
   IF GU='A' THEN GU='P';                                               SMI00230
   ELSE IF GU='E' THEN GU='Q';                                          SMI00240
   ELSE IF GU='H' THEN GU='R';                                          SMI00250
   ELSE GOTO ARR;                                                       SMI00260
   IF SUBSTR(OKEY,3,1)='0' | SUBSTR(OKEY,3,3)='300' | SUBSTR(OKEY,3,3)  SMI00270
    ='380' | SUBSTR(OKEY,3,4)='3840' THEN GOTO ARR;                     SMI00280
    IF SUBSTR(OKEY,6,4)>'8401' & SUBSTR(OKEY,6,4)<'8409' THEN GOTO ARR; SMI00290
   K=K+1;                                                               SMI00300
   AREC(K)=MREC;                                                        SMI00310
   GOTO ARR;                                                            SMI00320
  LT: CLOSE FILE(BOJUNG); OPEN FILE(BOJUNG) DIRECT UPDATE;              SMI00330
    DO I=1 TO K;                                                        SMI00340
     MREC=AREC(I);                                                      SMI00350
     REWRITE FILE(BOJUNG) FROM(MREC) KEY(OKEY);                         SMI00360
   EEX:  END;                                                           SMI00370
   CLOSE FILE(BOJUNG);                                                  SMI00380
 END JISURTN;                                                           SMI00390
/*                                                                      SMI00400
//GO.BOJUNG DD DSN=BOJUNG62,DISP=SHR                                    SMI00410
//                                                                      SMI00420
```

SMIWBK62

보정 BACK-UP

UTILITY PROGRAM

FILE: SMIWBK62 BJBACKUP A0 VM/SP RELEASE 2.1 EXPRESS PUT8302+ SLU203

```
//SMIWBK62 JOB CLASS=1,TYPRUN=HOLD                                    SMI00010
//JOBCAT DD DSN=SMICT12,DISP=SHR                                      SMI00020
//ST3 EXEC PGM=IDCAMS                                                 SMI00030
//DIREC DD DSNAME=BOJUNG62,DISP=SHR                                   SMI00240
//TA DD UNIT=TAPE,DSN=BOJUNG,DCB=(RECFM=FB,LRECL=320,BLKSIZE=3200),   SMI00250
//   DISP=(NEW,KEEP)                                                  SMI00260
//SYSPRINT DD SYSOUT=A
//SYSIN  DD *
         REPRO  INFILE (DIREC)-
                OUTFILE (TA)
//                                                                    SMI00270
```

SMISPC62

보정 특수분류 LIST

FILE: SMISPC62 SPECIAL A1 VM/SP RELEASE 2.1 EXPRESS PUT8302+ SLU203

```
//SMISPC62 JOB  CLASS=6,TYPRUN=HOLD
//JOBCAT  DD  DSN=SMICT12,DISP=SHR
// EXEC  PLIFCLG
//PLI.SYSIN DD *
* PROCESS GS,NEST,S,A,XREF,OPT(2);                                      AA 00010
  JISURTN: PROC OPTIONS(MAIN);                                          AA 00020
     DCL  SUMM  FILE  RECORD  KEYED  ENV(VSAM);                         AA 00030
     DCL  1  SUMREC,                                                    AA 00040
          2  OKEY                    CHAR(10),                         AA 00050
          2  SPC                     CHAR(1),                          AA 00060
          2  OYEAR                   PIC'99',                          AA 00070
          2  OMONTH                  PIC'99',                          AA 00080
          2  WEIGHT                  FLOAT,                            AA 00090
          2  BASEM                   FIXED BIN(31),                    AA 00100
          2  DATAO(37),                                                AA 00110
             3  ODATA                FIXED BIN(31),                    AA 00120
             3  JDATA                FLOAT,                            AA 00130
          2  SIGN                    CHAR(1);                          AA 00140
     DCL  1  KKEY  DEF  OKEY,                                           AA 00160
          2  CD                      CHAR(1),                          AA 00170
          2  GU                      CHAR(1),                          AA 00180
          2  INDUST                  CHAR(3),                          AA 00190
          2  PUM                     CHAR(5);                          AA 00200
     DCL  WSPC(5,9)                  FLOAT DEC(16);                    AA 00230
     DCL  DATA(37)                   FLOAT DEC(16);                    AA 00270
     DCL  JISU                       PIC'99999V9';                     AA 00280
     DCL  GUB(5)                     CHAR(1)   INIT(
                                     'J','K','L','S','T');
     DCL  Q                          PIC'99';                          AA 00410
     DCL  SPCD(9)                    CHAR(3) INIT(
                                     '000','800','810','820','821',
                                     '822','900','910','920');
     WSPC = 0 ;
     ON KEY(SUMM) BEGIN;
     IF ONCODE=51 THEN PUT EDIT(OKEY,'NOT FOUND')(A(10),A(10));
     END;
     OPEN FILE(SUMM) SEQUENTIAL INPUT ;                                 AA 00580
     OPEN  FILE(SYSPRINT)  LINESIZE(132)  PAGESIZE(66);
     ON  ENDFILE(SUMM)  GOTO  PKP;
  RD : READ FILE(SUMM) INTO(SUMREC);                                    AA 01080
     IF CD > '2' THEN GOTO PKP ;
        IF GU ='A' THEN L = 1 ;
           ELSE IF GU ='E' THEN L = 2 ;
             ELSE IF GU ='H' THEN L = 3 ;
               ELSE IF GU ='F' THEN L = 4 ;
                 ELSE IF GU ='G' THEN L = 5 ;
                   ELSE GOTO RD ;
                     IF SUBSTR(PUM,5,1)='0' & PUM¬='00000'
                                          THEN GOTO TYT;
                       ELSE GOTO  RD;
  TYT : IF SPC¬=' ' THEN DO;                                            AA 00990
           Q=SPC;                                                       AA 01000
           WSPC(L,Q) = WSPC(L,Q) + WEIGHT ;                             AA 01010
           END ;
        GOTO RD ;
```

```
  PKP :                                                                 AA 01030
        CLOSE  FILE(SUMM);                                              AA 01450
        OPEN  FILE(SUMM)  DIRECT  UPDATE;                               AA 01460
        WSPC(*,7) = WSPC(*,8) + WSPC(*,9) ;                             AA 01720
        WSPC(*,4) = WSPC(*,5) + WSPC(*,6) ;                             AA 01720
        WSPC(*,2) = WSPC(*,3) + WSPC(*,4) ;                             AA 01720
        WSPC(*,1) = WSPC(*,2) + WSPC(*,7) ;                             AA 01720
          DO K = 1 TO 5 ;
             DO I = 1 TO 9 ;                                            AA 01780
               OKEY = '2' || GUB(K) || SPCD(I);                         AA 01790
               OKEY = TRANSLATE(OKEY,'0',' ');
               READ  FILE(SUMM)  INTO(SUMREC)  KEY(OKEY);
               WEIGHT = WSPC(K,I) ;
               REWRITE  FILE(SUMM)  FROM(SUMREC)  KEY(OKEY);
               CALL  PRINT ;
             END;                                                       AA 01850
          END;                                                          AA 01850
  LAST : CLOSE FILE (SUMM);
  PRINT: PROC ;
             PUT SKIP(2) EDIT (OKEY,SPC,OYEAR,OMONTH,WEIGHT,BASEM,SIGN,
                              ((ODATA(M),JDATA(M))DO  M = 1 TO 7))
                           (4(A),F(6,1),F(7),A,X(10),7(F(7),F(6,1)));
             PUT  SKIP  EDIT (((ODATA(M),JDATA(M))DO M=8 TO 17))
                             (10(F(7),F(6,1)));
             PUT  SKIP  EDIT (((ODATA(M),JDATA(M))DO M=18 TO 27))
                             (10(F(7),F(6,1)));
             PUT  SKIP  EDIT (((ODATA(M),JDATA(M))DO M=28 TO 37))
                             (10(F(7),F(6,1)));
         END  PRINT;
 END JISURTN;                                                           AA 02250
/*
//GO.SUMM DD  UNIT=DISK,DSN=BOJUNG62,DISP=SHR
//GO.SYSPRINT DD  SYSOUT=A
//GO.SYSOUT   DD  SYSOUT=A
//
```

SMITUK62

보정 특수분류 내수·수출 LIST

```
//SMITUK62 JOB  CLASS=6,TYPRUN=HOLD
//JOBCAT  DD  DSN=SMICT12,DISP=SHR      /****************************/
// EXEC  PLIFCLG                        /**   SPECIAL  WEIGHT  SUM  **/
//PLI.SYSIN DD *                        /**          'S'  'T'       **/
* PROCESS GS,NEST,S,A,XREF,OPT(2);      /****************************/  AA 00010
  JISURTN: PROC OPTIONS(MAIN);                                          AA 00020
     DCL  SUMM  FILE  RECORD  KEYED  ENV(VSAM);                         AA 00030
     DCL  1  SUMREC,                                                    AA 00040
          2  OKEY                       CHAR(10),                      AA 00050
          2  SPC                        CHAR(1),                       AA 00060
          2  OYEAR                      PIC'99',                       AA 00070
          2  OMONTH                     PIC'99',                       AA 00080
          2  WEIGHT                     FLOAT,                         AA 00090
          2  BASEM                      FIXED BIN(31),                 AA 00100
          2  DATAO(37),                                                AA 00110
             3  ODATA                   FIXED BIN(31),                 AA 00120
             3  JDATA                   FLOAT,                         AA 00130
          2  SIGN                       CHAR(1);                       AA 00140
     DCL  1  KKEY  DEF  OKEY,                                           AA 00160
          2  CD                         CHAR(1),                       AA 00170
          2  GU                         CHAR(1),                       AA 00180
          2  INDUST                     CHAR(3),                       AA 00190
          2  PUM                        CHAR(5);                       AA 00200
     DCL  WSPC(2,9)                     FLOAT DEC(16);                 AA 00230
     DCL  DATA(37)                      FLOAT DEC(16);                 AA 00270
     DCL  JISU                          PIC'99999V9';                  AA 00280
     DCL  GUB(2)                        CHAR(1)   INIT('S','T');
     DCL  Q                             PIC'99';                       AA 00410
     DCL  SPCD(9)                       CHAR(3) INIT(
                                        '000','800','810','820','821',
                                        '822','900','910','920');
     WSPC = 0 ;
     ON KEY(SUMM) BEGIN;
     IF ONCODE=51 THEN PUT EDIT(OKEY,'NOT FOUND')(A(10),A(10));
     END;
     OPEN FILE(SUMM) SEQUENTIAL INPUT ;                                 AA 00580
     OPEN  FILE(SYSPRINT)  LINESIZE(132)  PAGESIZE(66);
     ON  ENDFILE(SUMM)  GOTO  PKP;
  RD : READ FILE(SUMM) INTO(SUMREC);                                    AA 01080
     IF CD > '2' THEN GOTO PKP ;
        IF GU ='F' THEN L = 1 ;
           ELSE IF GU ='G' THEN L = 2 ;
                   ELSE GOTO RD ;
                     IF SUBSTR(PUM,5,1)='0' & PUM¬='00000'
                                          THEN GOTO TYT;
                       ELSE GOTO  RD;
  TYT : IF SPC¬=' ' THEN DO;                                            AA 00990
           Q=SPC;                                                       AA 01000
           WSPC(L,Q) = WSPC(L,Q) + WEIGHT ;                             AA 01010
           END ;
        GOTO RD ;
  PKP :                                                                 AA 01030
        CLOSE  FILE(SUMM);                                              AA 01450
        OPEN  FILE(SUMM)  DIRECT  UPDATE;                               AA 01460
        WSPC(*,7) = WSPC(*,8) + WSPC(*,9) ;                             AA 01720
```

```
        WSPC(*,4) = WSPC(*,5) + WSPC(*,6) ;                                AA 01720
        WSPC(*,2) = WSPC(*,3) + WSPC(*,4) ;                                AA 01720
        WSPC(*,1) = WSPC(*,2) + WSPC(*,7) ;                                AA 01720
          DO K = 1 TO 2 ;
             DO I = 1 TO 9 ;                                               AA 01780
               OKEY = '2' || GUB(K) || SPCD(I);                            AA 01790
               OKEY = TRANSLATE(OKEY,'0',' ');
               READ  FILE(SUMM)  INTO(SUMREC)  KEY(OKEY);
               WEIGHT = WSPC(K,I) ;
               REWRITE  FILE(SUMM)  FROM(SUMREC)  KEY(OKEY);
               CALL  PRINT ;
             END;                                                          AA 01850
          END;                                                             AA 01850
  LAST : CLOSE FILE (SUMM);
  PRINT: PROC ;
             PUT SKIP(2) EDIT (OKEY,SPC,OYEAR,OMONTH,WEIGHT,BASEM,SIGN,
                                 ((ODATA(M),JDATA(M))DO  M = 1 TO 7))
                              (4(A),F(6,1),F(7),A,X(10),7(F(7),F(6,1)));
             PUT  SKIP  EDIT (((ODATA(M),JDATA(M))DO M=8 TO 17))
                              (10(F(7),F(6,1)));
             PUT  SKIP  EDIT (((ODATA(M),JDATA(M))DO M=18 TO 27))
                              (10(F(7),F(6,1)));
             PUT  SKIP  EDIT (((ODATA(M),JDATA(M))DO M=28 TO 37))
                              (10(F(7),F(6,1)));
         END  PRINT;
 END JISURTN;                                                              AA 02250
/*
//GO.SUMM DD  UNIT=DISK,DSN=BOJUNG62,DISP=SHR
//GO.SYSPRINT DD  SYSOUT=A
//GO.SYSOUT   DD  SYSOUT=A
//
```

SMIGAJ62

보정 가중치 기준물량 CHANGE

FILE: SMIGAJ62 GAJUNG A1、VM/SP RELEASE 2.1 EXPRESS PUT8302+ SLU203

```
//SMIGAJ62 JOB CLASS=C                                                  SMI00010
//JOBCAT DD DSN=SMICT12,DISP=SHR   /*********************************/  SMI00020
// EXEC PLIFCLG                    /*       SPECIAL JISU            */  SMI00030
//PLI.SYSIN DD *                   /*         GAJUNGCHI  INSERT     */  SMI00040
* PROCESS GS,NEST,OPT(2);          /*********************************/  SMI00050
  GAJUNG: PROC  OPTIONS(MAIN);                                          SMI00060
      DCL  SUMM  FILE  RECORD  KEYED  ENV(VSAM);                        SMI00070
      DCL  1  SUMREC,                                                   SMI00080
           2  SKEY                    CHAR(10),                        SMI00090
           2  SPEC                    CHAR(1),                         SMI00100
           2  YEAR                    PIC'99',                         SMI00110
           2  MONTH                   PIC'99',                         SMI00120
           2  WEIGHT                  FLOAT,                           SMI00130
           2  BASE                    FIXED BIN(31),                   SMI00140
           2  DATAO(37),                                               SMI00150
              3  MUL                  FIXED BIN(31),                   SMI00160
              3  JISU                 FLOAT,                           SMI00170
           2  SIGN                    CHAR(1);                         SMI00180
      DCL  1  KKEY  DEF  SKEY,                                          SMI00190
           2  CD                      CHAR(1),                         SMI00200
           2  GU                      CHAR(1),                         SMI00210
           2  INDUST                  CHAR(3),                         SMI00220
           2  PUM                     CHAR(5);                         SMI00230
      DCL  ARRAY(9)  FLOAT  INIT(    81,    81,    81,     0,          SMI00240
                                     81,    81,     0,     0,     0);   SMI00250
      DCL  CODE(9)  CHAR(10)  INIT                                      SMI00260
                        ('2A38484060','2B38484060','2C38484060',        SMI00270
                         '2D38484060','2E38484060','2F38484060',        SMI00280
                         '2G38484060','2H38484060','2I38484060');       SMI00290
      ON  KEY(SUMM)  BEGIN;                                             SMI00300
      IF  ONCODE = 51  THEN  PUT  EDIT(SKEY,'NOT FOUND')(A(10),A(10));  SMI00310
      END;                                                              SMI00320
      OPEN  FILE(SUMM)  DIRECT  UPDATE;                                 SMI00330
      OPEN  FILE(SYSPRINT)  PAGESIZE(66)  LINESIZE(132);                SMI00340
      DO  K  =  1  TO  9;                                               SMI00350
        SKEY = CODE(K) ;                                                SMI00360
        READ  FILE(SUMM)  INTO(SUMREC)  KEY(SKEY);                      SMI00370
        BASE = ARRAY(K) ;                                               SMI00380
        REWRITE  FILE(SUMM)  FROM(SUMREC)  KEY(SKEY);                   SMI00390
        CALL  PRINT ;                                                   SMI00400
      END ;                                                             SMI00410
      CLOSE  FILE(SUMM);                                                SMI00420
  PRINT: PROC ;                                                         SMI00430
          PUT SKIP(2) EDIT (SKEY,SPEC,YEAR,MONTH,WEIGHT,BASE,SIGN,      SMI00440
                           ((MUL(M),JISU(M))DO  M = 1 TO 7))            SMI00450
                           (4(A),F(6,1),F(7),A,X(10),7(F(7),F(6,1)));   SMI00460
          PUT  SKIP  EDIT  (((MUL(M),JISU(M))DO M=8 TO 17))             SMI00470
                           (10(F(7),F(6,1)));                           SMI00480
          PUT  SKIP  EDIT  (((MUL(M),JISU(M))DO M=18 TO 27))            SMI00490
                           (10(F(7),F(6,1)));                           SMI00500
          PUT  SKIP  EDIT  (((MUL(M),JISU(M))DO M=28 TO 37))            SMI00510
                           (10(F(7),F(6,1)));                           SMI00520
        END  PRINT;                                                     SMI00530
   END  GAJUNG;                                                         SMI00540
/*                                                                      SMI00550
```

FILE: SMIGAJ62 GAJUNG A1 VM/SP RELEASE 2.1 EXPRESS PUT8302+ SLU203

```
//GO.SUMM   DD  UNIT=DISK,DSN=BOJUNG62,DISP=SHR,VOL=SER=BOSWK2          SMI00560
//GO.SYSPRINT DD SYSOUT=A                                               SMI00570
//GO.SYSOUT   DD SYSOUT=A                                               SMI00580
//                                                                      SMI00590
```

원재료 T.T.T. TAPE DESIGN

4 (byte)	2	2	1	8	8
원재료번호	연 도	월	국 산	사 용 량	재 고 량
			수 입 여 부	(FLOAT(16))	(FLOAT(16))

국산=1
수입=2

BK SIZE	3200
RECORD LENGTH	25

보관 TAPE 명세서

TAPE NO	내 용	명 칭
3699	80 ~ 82 원재료 Merge	84 SMISOWDR 228φ82
4237	83 년 원재료 Merge	83 SMIMGWDR 2283φ112
6561	84.1월~8월 원재료 Merge	84 SMISOWDR 22φ1φ8
2974	80.1월~84.8월 원재료 Merge	84 SMISOWDR 228φ84φ8

SMITTA22

원재료 '80～'82 TTT

```
//SMITTA22 JOB CLASS=2                                                  SMI00010
//TTT   EXEC PLIFCLG                                                    SMI00020
//PLI.SYSIN DD *                                                        SMI00030
* PROCESS GS,NEST,OPT(2) ;                                              SMI00040
    HH : PROC OPTIONS (MAIN) ;                                          SMI00050
         DCL TPI FILE RECORD INPUT ;                                    SMI00060
         DCL TPO FILE RECORD OUTPUT ;                                   SMI00070
          DCL  1 TI                        ,                            SMI00080
               2 F1          CHAR ( 1)     ,                            SMI00090
               2 YR          CHAR ( 2)     ,                            SMI00100
               2 F2          CHAR (47)     ,                            SMI00110
               2 WON         CHAR ( 4)     ,                            SMI00120
               2 GU          CHAR (1)      ,                            SMI00130
               2 F3          CHAR ( 5)     ,                            SMI00140
               2 MDATA(12)                 ,                            SMI00150
               3 F4          CHAR (12)     ,                            SMI00160
               3 SA          FLOAT( 6)     ,                            SMI00170
               3 F5          CHAR ( 4)     ,                            SMI00180
               3 JAI         FLOAT( 6)     ,                            SMI00190
               3 F6          CHAR ( 4)     ,                            SMI00200
               2 F7          CHAR (243)    ,                            SMI00210
               2 BUN         CHAR (1)      ;                            SMI00220
          DCL SW CHAR(1) INIT('0') ;                                    SMI00230
         DCL 1 TO ,                                                     SMI00240
             2 WON_NO CHAR (4) ,                                        SMI00250
             2 YEAR   CHAR (2) ,                                        SMI00260
             2 MON    CHAR (2) ,                                        SMI00270
             2 GUBUN  CHAR (1) ,                                        SMI00280
             2 SAYONG FLOAT(16),                                        SMI00290
             2 JAIGO  FLOAT(16);                                        SMI00300
             DCL MTIT(12) CHAR(2) INIT('01','02','03','04','05','06',   SMI00310
                                       '07','08','09','10','11','12');  SMI00320
             DCL L FIXED (9) INIT(0) ;                                  SMI00330
         ON ENDFILE(TPI) GOTO LAST ;                                    SMI00340
         OPEN FILE(TPI),FILE(TPO) ;                                     SMI00350
     RD :      READ FILE(TPI) INTO(TI) ;                                SMI00360
               IF BUN¬= '2' THEN GOTO RD ;                              SMI00370
                     WON_NO = WON ;                                     SMI00380
                     YEAR = YR ;                                        SMI00390
                     GUBUN  = GU  ;                                     SMI00400
                     DO K = 1 TO 12 ;                                   SMI00410
                     MON = MTIT(K) ;                                    SMI00420
                     SAYONG    =  SA(K) ;                               SMI00430
                     JAIGO     =  JAI(K) ;                              SMI00440
                     WRITE FILE(TPO) FROM(TO) ;                         SMI00450
                     L = L + 1 ;                                        SMI00460
                     END ;                                              SMI00470
               GOTO RD ;                                                SMI00480
     LAST :          CLOSE FILE(TPI),FILE(TPO) ;                        SMI00490
                     PUT SKIP EDIT (L) (F(10) ) ;                       SMI00500
                     END HH ;                                           SMI00510
/*                                                                      SMI00520
//GO.TPI   DD UNIT=TAPE,DSN=HH,VOL=SER=100001,LABEL=(2,BLP),            SMI00530
//            DCB=(RECFM=FB,BLKSIZE=3200,LRECL=640),DISP=(OLD,KEEP)     SMI00540
//         DD UNIT=AFF=TPI,DSN=HH,VOL=SER=100002,LABEL=(2,BLP)          SMI00550
```

FILE: SMITTA22 TT808182 A1 VM/SP RELEASE 2.1 EXPRESS PUT8302+ SLU203

```
//          DD UNIT=AFF=TPI,DSN=HH,VOL=SER=100003,LABEL=(2,BLP)          SMI00560
//GO.TPO    DD UNIT=TAPE,DSN=HO,DISP=(NEW,KEEP),                         SMI00570
//             DCB=(RECFM=FB,BLKSIZE=3200,LRECL=25)                      SMI00580
//GO.SYSOUT DD SYSOUT=A                                                  SMI00590
//GO.SYSPRINT DD SYSOUT=A                                                SMI00600
//                                                                       SMI00610
```

SMITTT22

원재료 '83～'84 TTT

FILE: SMITTT22 TTT8384 A1 VM/SP RELEASE 2.1 EXPRESS PUT8302+ SLU203

```
//SMITTT22 JOB CLASS=B,TYPRUN=HOLD                                      SMI00010
//TTT   EXEC PLIFCLG                                                    SMI00020
//PLI.SYSIN DD *                                                        SMI00030
* PROCESS GS,NEST,OPT(2) ;                                              SMI00040
    HH : PROC OPTIONS (MAIN) ;                                          SMI00050
         DCL TPI FILE RECORD INPUT ;                                    SMI00060
         DCL TPO FILE RECORD OUTPUT ;                                   SMI00070
         DCL 1 TI ,                                                     SMI00080
             2 F1   CHAR (5)  ,                                         SMI00090
             2 CD   CHAR (1)  ,                                         SMI00100
             2 F2   CHAR (2)  ,                                         SMI00110
             2 WON  CHAR (4)  ,                                         SMI00120
             2 F3   CHAR (1)  ,                                         SMI00130
             2 GU   CHAR (1)  ,                                         SMI00140
             2 SA   FIXED BIN (31) ,                                    SMI00150
             2 JAI  FIXED BIN (31) ,                                    SMI00160
             2 F4   CHAR (28)  ;                                        SMI00170
         DCL 1 TO ,                                                     SMI00180
             2 WON_NO CHAR (4) ,                                        SMI00190
             2 YEAR   CHAR (2) INIT('83') ,                             SMI00200
             2 MON    CHAR (2) INIT('06') ,                             SMI00210
             2 GUBUN  CHAR (1) ,                                        SMI00220
             2 SAYONG FLOAT(16),                                        SMI00230
             2 JAIGO  FLOAT(16);                                        SMI00240
             DCL L FIXED (9) INIT(0) ;                                  SMI00250
         ON ENDFILE(TPI) GOTO LAST ;                                    SMI00260
         OPEN FILE(TPI),FILE(TPO) ;                                     SMI00270
     RD :      READ FILE(TPI) INTO(TI) ;                                SMI00280
               IF CD ¬= '3' THEN GOTO RD ;                              SMI00290
                     WON_NO = WON ;                                     SMI00300
                     GUBUN  = GU  ;                                     SMI00310
                     SAYONG = SA  ;                                     SMI00320
                     JAIGO  = JAI ;                                     SMI00330
               WRITE FILE(TPO) FROM(TO) ;                               SMI00340
               L = L + 1 ;                                              SMI00350
               GOTO RD ;                                                SMI00360
     LAST :          CLOSE FILE(TPI),FILE(TPO) ;                        SMI00370
                     PUT SKIP EDIT (L) (F(10) ) ;                       SMI00380
                     END HH ;                                           SMI00390
/*                                                                      SMI00400
//GO.TPI    DD UNIT=TAPE,DSN=HH,VOL=SER=123321,LABEL=(2,BLP),           SMI00410
//             DCB=(RECFM=FB,BLKSIZE=3200,LRECL=50),DISP=(OLD,KEEP)     SMI00420
//GO.TPO    DD UNIT=TAPE,DSN=HO,DISP=(NEW,KEEP),                        SMI00430
//             DCB=(RECFM=FB,BLKSIZE=3200,LRECL=25)                     SMI00440
//GO.SYSOUT DD SYSOUT=A                                                 SMI00450
//GO.SYSPRINT DD SYSOUT=A                                               SMI00460
//                                                                      SMI00470
```

SMICUS22

원재료 SORT 및 MERGE

FILE: SMICUS22 SOMER A1 VM/SP RELEASE 2.1 EXPRESS PUT8302+ SLU203

```
//SMICUS22 JOB CLASS=B,TYPRUN=HOLD                                    SMI00010
//SRT EXEC SORT,PARM='SIZE(MAX)'                                      SMI00020
//SYSOUT  DD  SYSOUT=A                                                SMI00030
//SYSPRINT DD SYSOUT=A                                                SMI00040
//SORTIN DD UNIT=TAPE,VOL=SER=SOR,DSN=HH,LABEL=(2,BLP),               SMI00050
//       DISP=(OLD,KEEP),                                             SMI00060
//       DCB=(RECFM=FB,LRECL=25,BLKSIZE=3200)                         SMI00070
//       DD UNIT=AFF=SORTIN,DSN=LL,VOL=SER=SOR1,LABEL=(2,BLP)         SMI00080
//SORTOUT DD UNIT=TAPE,DSN=HH,DISP=(NEW,KEEP),                        SMI00090
//        DCB=(RECFM=FB,LRECL=25,BLKSIZE=3200)                        SMI00100
//SORTWK01 DD UNIT=DISK,SPACE=(CYL,(250)),VOL=SER=SORTWK              SMI00110
//SYSIN    DD *                                                       SMI00120
 SORT FIELDS=(1,8,A),FORMAT=CH,FILSZ=E2300                            SMI00130
/*                                                                    SMI00140
//                                                                    SMI00150
```

SMITLC22

원재료 TABLE

```
//SMITLC22 JOB CLASS=B,TYPRUN=HOLD                                      SMI00010
//TABLE1 EXEC PLIFCLG                                                   SMI00020
//PLI.SYSIN DD *                                                        SMI00030
* PROCESS GS,NEST,OPT(2) ;                                              SMI00040
     HH : PROC OPTIONS(MAIN) ;                                          SMI00050
          DCL  TPI FILE RECORD INPUT ;            /*********************/SMI00060
          DCL  1 TI                        ,      /**  80 - 84 = YEAR **/SMI00070
               2 WON_NO       CHAR ( 4)    ,      /**   1 - 8   MONTH **/SMI00080
               2 YEAR         CHAR ( 2)    ,      /**   1 - 6   MONTH **/SMI00090
               2 MON          CHAR ( 2)    ,      /**   1 - 12  MONTH **/SMI00100
               2 S            PIC  '9'     ,      /** WONJAIRYO SAYONG */SMI00110
               2 SAYONG       FLOAT(16)    ,      /*********************/SMI00120
               2 JAIGO        FLOAT(16);                                 SMI00130
          DCL SW CHAR(1) INIT('0') ;                                    SMI00140
          DCL GI CHAR(4) ;                                              SMI00150
          DCL K  PIC '99';                                              SMI00160
          DCL ARR(3,2,14) FLOAT(16) INIT((84)0) ;                       SMI00170
          DCL TITLE(14) CHAR( 9) INIT(                                  SMI00180
                                 ' 80. 1- 6',                           SMI00190
                                 ' 80. 1- 8',                           SMI00200
                                 ' 80. 1-12',                           SMI00210
                                 ' 81. 1- 6',                           SMI00220
                                 ' 81. 1- 8',                           SMI00230
                                 ' 81. 1-12',                           SMI00240
                                 ' 82. 1- 6',                           SMI00250
                                 ' 82. 1- 8',                           SMI00260
                                 ' 82. 1-12',                           SMI00270
                                 ' 83. 1- 6',                           SMI00280
                                 ' 83. 1- 8',                           SMI00290
                                 ' 83. 1-12',                           SMI00300
                                 ' 84. 1- 6',                           SMI00310
                                 ' 84. 1- 8') ;                         SMI00320
          ON ENDFILE(TPI) GOTO LAST ;                                   SMI00330
          OPEN FILE(TPI),FILE(SYSPRINT) LINESIZE(132) PAGESIZE(66) ;    SMI00340
       PUT PAGE ;                                                       SMI00350
       PUT SKIP(5) EDIT('TABLE 1       ---SAYONG--- ')                  SMI00360
                       (X(50),A  ) ;                                    SMI00370
       PUT SKIP(3) EDIT('NO',(TITLE(K)DO K=1 TO 14))                    SMI00380
                       (X(2),A,X(2),14 A) ;                             SMI00390
       PUT SKIP(1) ;                                                    SMI00400
    RD :   READ FILE(TPI) INTO(TI) ;                                    SMI00410
           IF S < 1 | S > 2 THEN GOTO RD ;                              SMI00420
           IF WON_NO < '501 ' | WON_NO > '907 ' THEN GOTO RD ;          SMI00430
                 IF SW = '1' THEN GOTO COM ;                            SMI00440
                 GI = WON_NO ;                                          SMI00450
                 SW = '1' ;                                             SMI00460
                 GOTO NUJUK ;                                           SMI00470
    COM :  IF GI = WON_NO THEN GOTO NUJUK ;                             SMI00480
           CALL PUMOK ;                                                 SMI00490
           GI = WON_NO ;                                                SMI00500
    NUJUK :                                                             SMI00510
                IF YEAR  = '80' THEN K = 1 ;                            SMI00520
           ELSE IF YEAR  = '81' THEN K = 4 ;                            SMI00530
           ELSE IF YEAR  = '82' THEN K = 7 ;                            SMI00540
           ELSE IF YEAR  = '83' THEN K = 10;                            SMI00550
```

```
                                ELSE K = 13;                                SMI00560
            IF MON <= '06' THEN DO ;                                        SMI00570
            ARR(S,1,K) = ARR(S,1,K) + SAYONG   ;                            SMI00580
            ARR(S,1,K+1) = ARR(S,1,K+1) + SAYONG   ;                        SMI00590
            ARR(S,1,K+2) = ARR(S,1,K+2) + SAYONG   ;                        SMI00600
            END ;                                                           SMI00610
            ELSE IF MON <= '08' THEN DO ;                                   SMI00620
            ARR(S,1,K+1) = ARR(S,1,K+1) + SAYONG   ;                        SMI00630
            ARR(S,1,K+2) = ARR(S,1,K+2) + SAYONG   ;                        SMI00640
            END ;                                                           SMI00650
            ELSE IF MON <= '12' THEN DO ;                                   SMI00660
            ARR(S,1,K+2) = ARR(S,1,K+2) + SAYONG   ;                        SMI00670
            END ;                                                           SMI00680
            GOTO RD ;                                                       SMI00690
    LAST :  CLOSE FILE(TPI) ;                                               SMI00700
            CALL PUMOK ;                                                    SMI00710
    PUMOK : PROC ;                                                          SMI00720
            ARR(3,1,*) = ARR(1,1,*) + ARR(2,1,*) ;                          SMI00730
            DO M = 1 TO 3 ;                                                 SMI00740
            DO K = 4 TO 14 ;                                                SMI00750
            IF ARR(M,1,K-3) = 0 THEN ARR(M,2,K) = 0 ; ELSE                  SMI00760
            ARR(M,2,K) = ARR(M,1,K) / ARR(M,1,K-3) * 100 -100 ;             SMI00770
            END ;                                                           SMI00780
            END ;                                                           SMI00790
            PUT SKIP(3) EDIT(GI,(ARR(1,1,K)DO K=1 TO 14))                   SMI00800
                            ( A,X(2),14 F(9));                              SMI00810
            PUT SKIP    EDIT((ARR(1,2,K)DO K=4 TO 14))(X(33),11 F(9,2));    SMI00820
            PUT SKIP(2) EDIT((ARR(2,1,K)DO K=1 TO 14))(X(6),14 F(9)) ;      SMI00830
            PUT SKIP    EDIT((ARR(2,2,K)DO K=4 TO 14))(X(33),11 F(9,2));    SMI00840
            PUT SKIP(2) EDIT((ARR(3,1,K)DO K=1 TO 14))(X(6),14 F(9)) ;      SMI00850
            PUT SKIP    EDIT((ARR(3,2,K)DO K=4 TO 14))(X(33),11 F(9,2));    SMI00860
            ARR = 0 ;                                                       SMI00870
            END PUMOK ;                                                     SMI00880
                        END HH ;                                            SMI00890
/*                                                                          SMI00900
//GO.TPI    DD UNIT=TAPE,DSN=HH,VOL=SER=123321,LABEL=(2,BLP),               SMI00910
//             DCB=(RECFM=FB,BLKSIZE=3200,LRECL=25),DISP=(OLD,KEEP)         SMI00920
//GO.SYSOUT DD SYSOUT=A                                                     SMI00930
//GO.SYSPRINT DD SYSOUT=A                                                   SMI00940
//                                                                          SMI00950
```

SMITLD22

원재료 TABLE

```
//SMITLD22 JOB CLASS=B,TYPRUN=HOLD                                      SMI00010
//TABLE1 EXEC PLIFCLG                                                   SMI00020
//PLI.SYSIN DD *                                                        SMI00030
* PROCESS GS,NEST,OPT(2) ;                                              SMI00040
     HH : PROC OPTIONS(MAIN) ;                                          SMI00050
          DCL  TPI FILE RECORD INPUT ;            /*********************/SMI00060
          DCL  1 TI                      ,        /**  80 - 84 = YEAR **/SMI00070
               2 WON_NO      CHAR ( 4)   ,        /**   1 - 8   MONTH **/SMI00080
               2 YEAR        CHAR ( 2)   ,        /**   1 - 6   MONTH **/SMI00090
               2 MON         CHAR ( 2)   ,        /**   1 - 12  MONTH **/SMI00100
               2 S           PIC  '9'    ,        /** WONJAIRYO JAIGO **/SMI00110
               2 SAYONG      FLOAT(16)   ,        /********************/SMI00120
               2 JAIGO       FLOAT(16);                                 SMI00130
          DCL SW CHAR(1) INIT('0') ;                                    SMI00140
          DCL GI CHAR(4) ;                                              SMI00150
          DCL K  PIC '99';                                              SMI00160
          DCL ARR(3,2,14) FLOAT(16) INIT((84)0) ;                       SMI00170
          DCL TITLE(14) CHAR( 9) INIT(                                  SMI00180
                                 ' 80. 6   ',                           SMI00190
                                 ' 80. 8   ',                           SMI00200
                                 ' 80. 12  ',                           SMI00210
                                 ' 81. 6   ',                           SMI00220
                                 ' 81. 8   ',                           SMI00230
                                 ' 81. 12  ',                           SMI00240
                                 ' 82. 6   ',                           SMI00250
                                 ' 82. 8   ',                           SMI00260
                                 ' 82. 12  ',                           SMI00270
                                 ' 83. 6   ',                           SMI00280
                                 ' 83. 8   ',                           SMI00290
                                 ' 83. 12  ',                           SMI00300
                                 ' 84. 6   ',                           SMI00310
                                 ' 84. 8   ') ;                         SMI00320
          ON ENDFILE(TPI) GOTO LAST ;                                   SMI00330
          OPEN FILE(TPI),FILE(SYSPRINT) LINESIZE(132) PAGESIZE(66) ;    SMI00340
       PUT PAGE ;                                                       SMI00350
       PUT SKIP(5) EDIT('TABLE 2     ---JAIGO--- ' )                    SMI00360
                     (X(50),A  ) ;                                      SMI00370
       PUT SKIP(3) EDIT('NO',(TITLE(K)DO K=1 TO 14))                    SMI00380
                     (X(2),A,X(2),14 A) ;                               SMI00390
       PUT SKIP(1) ;                                                    SMI00400
    RD :  READ FILE(TPI) INTO(TI) ;                                     SMI00410
          IF S < 1 | S > 2 THEN GOTO RD ;                               SMI00420
          IF WON_NO < '501 ' | WON_NO > '906 ' THEN GOTO RD ;           SMI00430
                IF SW = '1' THEN GOTO COM ;                             SMI00440
                GI = WON_NO ;                                           SMI00450
                SW = '1' ;                                              SMI00460
                GOTO NUJUK ;                                            SMI00470
    COM :  IF GI = WON_NO THEN GOTO NUJUK ;                             SMI00480
           CALL PUMOK ;                                                 SMI00490
           GI = WON_NO ;                                                SMI00500
    NUJUK :                                                             SMI00510
                IF MON = '06' THEN K = 1 ;                              SMI00520
           ELSE IF MON = '08' THEN K = 2 ;                              SMI00530
           ELSE IF MON = '12' THEN K = 3 ;                              SMI00540
                                ELSE GOTO RD ;                          SMI00550
```

```
            K = K + ( YEAR - 80 ) * 3 ;                                  SMI00560
           ARR(S,1,K) = ARR(S,1,K) + JAIGO   ;                           SMI00570
           GOTO RD ;                                                     SMI00580
    LAST : CLOSE FILE(TPI) ;                                             SMI00590
           CALL PUMOK ;                                                  SMI00600
   PUMOK : PROC ;                                                        SMI00610
           ARR(3,1,*) = ARR(1,1,*) + ARR(2,1,*) ;                        SMI00620
           DO M = 1 TO 3 ;                                               SMI00630
           DO K = 4 TO 14 ;                                              SMI00640
           IF ARR(M,1,K-3) = 0 THEN ARR(M,2,K) = 0 ; ELSE                SMI00650
           ARR(M,2,K) = ARR(M,1,K) / ARR(M,1,K-3) * 100 -100 ;           SMI00660
           END ;                                                         SMI00670
           END ;                                                         SMI00680
           PUT SKIP(3) EDIT(GI,(ARR(1,1,K)DO K=1 TO 14))                 SMI00690
                            ( A,X(2),14 F(9));                           SMI00700
           PUT SKIP    EDIT((ARR(1,2,K)DO K=4 TO 14))(X(33),11 F(9,2));  SMI00710
           PUT SKIP(2) EDIT((ARR(2,1,K)DO K=1 TO 14))(X(6),14 F(9)) ;    SMI00720
           PUT SKIP    EDIT((ARR(2,2,K)DO K=4 TO 14))(X(33),11 F(9,2));  SMI00730
           PUT SKIP(2) EDIT((ARR(3,1,K)DO K=1 TO 14))(X(6),14 F(9)) ;    SMI00740
           PUT SKIP    EDIT((ARR(3,2,K)DO K=4 TO 14))(X(33),11 F(9,2));  SMI00750
           ARR = 0 ;                                                     SMI00760
           END PUMOK ;                                                   SMI00770
                       END HH ;                                          SMI00780
/*                                                                       SMI00790
//GO.TPI   DD UNIT=TAPE,DSN=HH,VOL=SER=123321,LABEL=(2,BLP),             SMI00800
//            DCB=(RECFM=FB,BLKSIZE=3200,LRECL=25),DISP=(OLD,KEEP)       SMI00810
//GO.SYSOUT DD SYSOUT=A                                                  SMI00820
//GO.SYSPRINT DD SYSOUT=A                                                SMI00830
//                                                                       SMI00840
```

〈別　添〉

계절조정지수 산출작업

1. 계절지수 산출작업(매년)

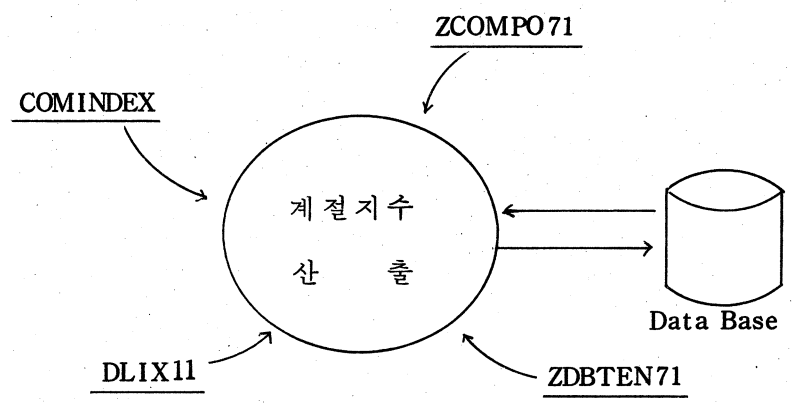
1. 계절지수 산출작업 (매년)

1.1 요 지

당국에서는 매년 년말 12월 확정치를 기준으로 대상계열에 대한 계절지수를 산출하여, 향후 1년간 고정 사용한다.

1.2 작업의 흐름도

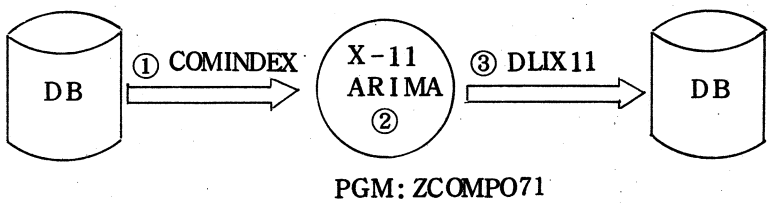
1.2.1 Orerflow



1.2.2 detail flow

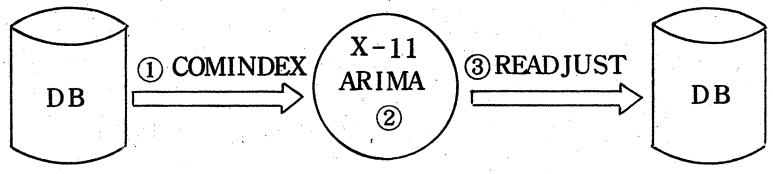
Stepl

매년 12월 확정치 기준 계절조정지수 Load



STEP 2

X-11, ARIMA를 이용 1년치 data의 예측(forecasting)



PGM: ZDBTEN71

1.2.3 과정별 Program의 기능

STEP 1. ① COMINDEX Program — PL/I

X-11 작업준비 과정, 즉 X-11을 이용키 위한 Input format으로 DB 자료를 DISK로 Reading한다.

② ZCOMPO71 Program — FORTRAN Package

DISK자료를 Input으로 계절조정지수 산출

③ DLIX11 Program — PL/I

계산된 최종조정 지수의 DB loading, 조정지수 Updating

STEP 2. ① 설명생략

② ZDBTEN71 Program — FORTRAN Package

계절 factor를 통한 1년치 data의 예측(forecasting)

③ ②의 처리결과의 DB Loading

1.3 작업방법

1.3.1 프로그램별 Options의 사용방법

STEP	No.	내 용	설 명
STEP1	(1)	DBNAME	처리대상 Data 이름
	(2)	DBKEY	〃 Key
Program :COMINDEX	(3)	주 기	"M", "Q", "Y"의 표시 여기서는 "M"만 사용
	(4)	기 간	처리코자 하는 기간 여기서는 가장 처음 실려 있는 때부터 현재(12월말)까지의 모든 Data를말함
	(5)	사전조정지수	"2"- 사전조정 지수사용 "b"- 사전조정 지수사용를 사용치 않음.
	(6)	Level	해당 Key의 Level
Program :DLIX11	(1)	기 간	12월말 현재의 년월
	(2)	사전조정지수	"ADJUST"→ 사전조정 "DONOT" → Not
	(3)	처리여부	"MODIFY", "INSERT"
	(4)	DBNAME	최종 조정지수에 붙여지는 DBNAME
	(5)	주 기	"M", "Q", "Y"여기서는 "M"만 사용
	(6)	Key수	⑦의 Key수
	(7)	DB Key	대상 DB Key
Program :STEP2	(1)	기 간	forecasting된 기간 (1년 연장된 년월)
program :READJUST	(2)	DBLOAD 여부	"LOAD"면 DB Load "CHECK"면 Print
		기타 Options 사용은 DLIX11 참조	

프로그램 : 계절지수 산출작업(매년)

RUN PROGRAM

① 12월 확정치 기준 계절 조정지수 산정 Fixed 작업

② 1년치 Data의 예측(Forecasting) 작업

③ X-11, ARIMA Package 이용

```
//ZSSCOP71  JOB  CLASS=V,MSGLEVEL=(1,1),PRTY=13,MSGCLASS=X
//STEP1     EXEC PGM=DFSRRC00,PARM= DLI,COMINDEX,ZVPSBA
//STEPCAT   DD   DSN=ZUSERCAT,DISP=SHR
//STEPLIB   DD   DSN=IMSVS.RESLIB,DISP=SHR
//          DD   DSN=IMSVS.PROCLIB,DISP=SHR
//          DD   DSN=IMSVS.PGMLIB,DISP=SHR
//IMS       DD   DSN=IMSVS.PSBLIB,DISP=SHR
//          DD   DSN=IMSVS.DBDLIB,DISP=SHR
//IEFRDER   DD   DUMMY
//DBFILE    DD   DSN=ZDBFILE,UNIT=3350,DISP=SHR,VOL=SER=BDSDBD
//SYSPRINT  DD   SYSOUT=X
//DFSVSAMP  DD   *
  4096,4
  2048,4
//GO.CARDIN DD  *
NOCOMP 560                    1.00791391                50        2
M1                  01   71       1
NPRDI00000  M70018312    1                                  1.229
NPRDI2      M70018312    1                                  1.229
NPRDI21     M70018312    1                                  1.229
NPRDI23     M70018312    1                                  1.229
NPRDI29     M70018312    1                                  1.229
NPRDI3      M70018312    1                                  1.229
NPRDI31     M70018312    1                                  1.229
NPRDI32     M70018312    1                                  1.229
NPRDI33     M70018312    1                                  1.229
NPRDI34     M70018312    1                                  1.229
NPRDI35     M70018312    1                                  1.229
NPRDI36     M70018312    1                                  1.229
NPRDI37     M70018312    1                                  1.229
NPRDI38     M70018312    1                                  1.229
NPRDI39     M70018312    1                                  1.229
NPRDI4      M70018312    1                                  1.229
-------     ---------
 (1) (2)   (3)  (4)
Z
/*
//GO.REGION DD  UNIT=3350,DSN=COMPJI,DISP=(NEW,KEEP),VOL=SER=SORTWK,
//          DCB=(RECFM=F,LRECL=260),SPACE=(CYL,50)
/*
//STEP2    EXEC PGM=ZCOMPO71
//STEPLIB   DD DSN=USER.LOADLIB,DISP=SHR
//SYSPRINT  DD SYSOUT=A
//FT05F001  DD DSN=COMPJI,DISP=(OLD,KEEP),UNIT=3350,VOL=SER=SORTWK
//FT06F001  DD SYSOUT=A
//FT08F001 DD DSN=ADJUST,DISP=(NEW,KEEP),UNIT=3350,VOL=SER=SORTWK,
//          DCB=(RECFM=FB,LRECL=260,BLKSIZE=260),SPACE=(CYL,50)
/*
//STEP3     EXEC PGM=DFSRRC00,PARM= DLI,DLIX11,ZVPSBA
//STEPCAT   DD   DSN=ZUSERCAT,DISP=SHR
//STEPLIB   DD   DSN=IMSVS.RESLIB,DISP=SHR
//          DD   DSN=IMSVS.PROCLIB,DISP=SHR
//          DD   DSN=IMSVS.PGMLIB,DISP=SHR
//IMS       DD   DSN=IMSVS.PSBLIB,DISP=SHR
//          DD   DSN=IMSVS.DBDLIB,DISP=SHR
```

```
//IEFRDER    DD   DUMMY
//DBFILE     DD   DSN=ZDBFILE,UNIT=3350,DISP=SHR,VOL=SER=BDSDBD
//SYSPRINT   DD   SYSOUT=X
//SYSUDUMP   DD   SYSOUT=A
//DFSVSAMP   DD   *
  4096,4
  2048,4
//GO.SYSIN   DD  *
   83 12    ADJUST     MODIFY
   -----    ------     ------
    (1)      (2)        (3)
   PRODX    M    16
   -----    -    --
    (4)    (5)  (6)
   00    2    21    23    29    3    31    32    33    34    35    36    37    38
   39    4
/*
//ADJFILE   DD  DSN=ADJUST,UNIT=3350,VOL=SER=SORTWK,DISP=(OLD,KEEP),
//              DCB=(RECFM=FB,LRECL=260,BLKSIZE=260)
/*
//SCRT1    EXEC PGM=IEHPROGM
//SYSPRINT DD SYSOUT=A
//DD3      DD UNIT=DISK,VOL=SER=SORTWK,DISP=OLD
//SYSIN    DD *
 SCRATCH DSNAME=COMPJI,VOL=DISK=SORTWK
 SCRATCH DSNAME=ADJUST,VOL=DISK=SORTWK
/*
//SCRT2    EXEC PGM=IEHPROGM
//SYSPRINT DD SYSOUT=A
//DD1      DD UNIT=SYSDA,VOL=SER=SORTWK,DISP=OLD
//SYSIN    DD *
 SCRATCH VTOC,VOL=SYSDA=SORTWK,SYS
/*
//
```

```
//ZSSC0271  JOB  CLASS=V,MSGLEVEL=(1,1),PRTY=13
//STEP1     EXEC PGM=DFSRRC00,PARM= DL1,COMINDEX,ZVPS8A
//STEPCAT   DD   DSN=ZUSERCAT,DISP=SHR
//STEPLIB   DD   DSN=IMSVS.RESLIB,DISP=SHR
//          DD   DSN=IMSVS.PROCLIB,DISP=SHR
//          DD   DSN=IMSVS.PGMLIB,DISP=SHR
//IMS       DD   DSN=IMSVS.PSBLIB,DISP=SHR
//          DD   DSN=IMSVS.DBDLIB,DISP=SHR
//IEFRDER   DD   DUMMY
//DBFILE    DD   DSN=ZDBFILE,UNIT=3350,DISP=SHR,VOL=SER=BOSDBD
//SYSPRINT  DD   SYSOUT=A
//DFSVSAMP  DD   *
  4096,4
  2048,4
//GO.CARDIN DD  *
NOCOMP 560                  1.00791391              50       2
M1                  01   71       1
NPRDI311    M70018312    1                                  1.229
NPRDI313    M70018312    1                                  1.229
NPRDI314    M70018312    1                                  1.229
NPRDI321    M70018312    1                                  1.229
NPRDI322    M70018312    1                                  1.229
NPRDI323    M70018312    1                                  1.229
NPRDI324    M70018312    1                                  1.229
NPRDI331    M70018312    1                                  1.229
NPRDI332    M70018312    1                                  1.229
NPRDI341    M70018312    1                                  1.229
NPRDI342    M70018312    1                                  1.229
NPRDI351    M70018312    1                                  1.229
NPRDI352    M70018312    1                                  1.229
NPRDI353    M70018312    1                                  1.229
NPRDI354    M70018312    1                                  1.229
NPRDI355    M70018312    1                                  1.229
NPRDI356    M70018312    1                                  1.229
--------    ---------
 (1) (2)   (3) (4)
Z
/*
//GO.REGION DD  UNIT=3350,DSN=COMPJI,DISP=(NEW,KEEP),VOL=SER=SORTWK,
//          DCB=(RECFM=F,LRECL=260),SPACE=(CYL,50)
/*
//STEP2    EXEC PGM=ZDBTEN71,TIME=90
//STEPLIB   DD DSN=USER.LOADLIB,DISP=SHR
//SYSPRINT  DD SYSOUT=A
//FT05F001  DD DSN=COMPJI,DISP=(OLD,KEEP),UNIT=3350,VOL=SER=SORTWK
//FT06F001  DD SYSOUT=A
//FT08F001 DD DSN=ADJUST,DISP=(NEW,KEEP),UNIT=3350,VOL=SER=SORTWK,
//          DCB=(RECFM=FB,LRECL=260,BLKSIZE=260),SPACE=(CYL,50)
/*
//STEP3     EXEC PGM=DFSRRC00,PARM= DLI,READJUST,ZVPS8A
//STEPCAT   DD   DSN=ZUSERCAT,DISP=SHR
//STEPLIB   DD   DSN=IMSVS.RESLIB,DISP=SHR
//          DD   DSN=IMSVS.PROCLIB,DISP=SHR
//          DD   DSN=IMSVS.PGMLIB,DISP=SHR
//IMS       DD   DSN=IMSVS.PSBLIB,DISP=SHR
```

```
//            DD   DSN=IMSVS.DBDLIB,DISP=SHR
//IEFRDER     DD   DUMMY
//DBFILE      DD   DSN=ZDBFILE,UNIT=3350,DISP=SHR,VOL=SER=BDSDBD
//SYSPRINT    DD   SYSOUT=A
//SYSUDUMP    DD   SYSOUT=A
//DFSVSAMP    DD   *
  4096,4
  2048,4
//GO.SYSIN    DD  *
   84 12    MODIFY     LOAD
   -----               ----
    (1)                (2)
   PRODS    M    17
   311    313    314    321    322    323    324    331    332    341    342
   351    352    353    354    355    356
/*
//ADJFILE   DD  DSN=ADJUST,UNIT=3350,VOL=SER=SORTWK,DISP=(OLD,KEEP),
//              DCB=(RECFM=FB,LRECL=260,BLKSIZE=260)
/*
//SCRT1    EXEC PGM=IEHPROGM
//SYSPRINT DD SYSOUT=A
//DD3      DD UNIT=DISK,VOL=SER=SORTWK,DISP=OLD
//SYSIN    DD *
 SCRATCH DSNAME=COMPJI,VOL=DISK=SORTWK
 SCRATCH DSNAME=ADJUST,VOL=DISK=SORTWK
/*
//SCRT2    EXEC PGM=IEHPROGM
//SYSPRINT DD SYSOUT=A
//DD1      DD UNIT=SYSDA,VOL=SER=SORTWK,DISP=OLD
//SYSIN    DD *
 SCRATCH VTOC,VOL=SYSDA=SORTWK,SYS
/*
//
```

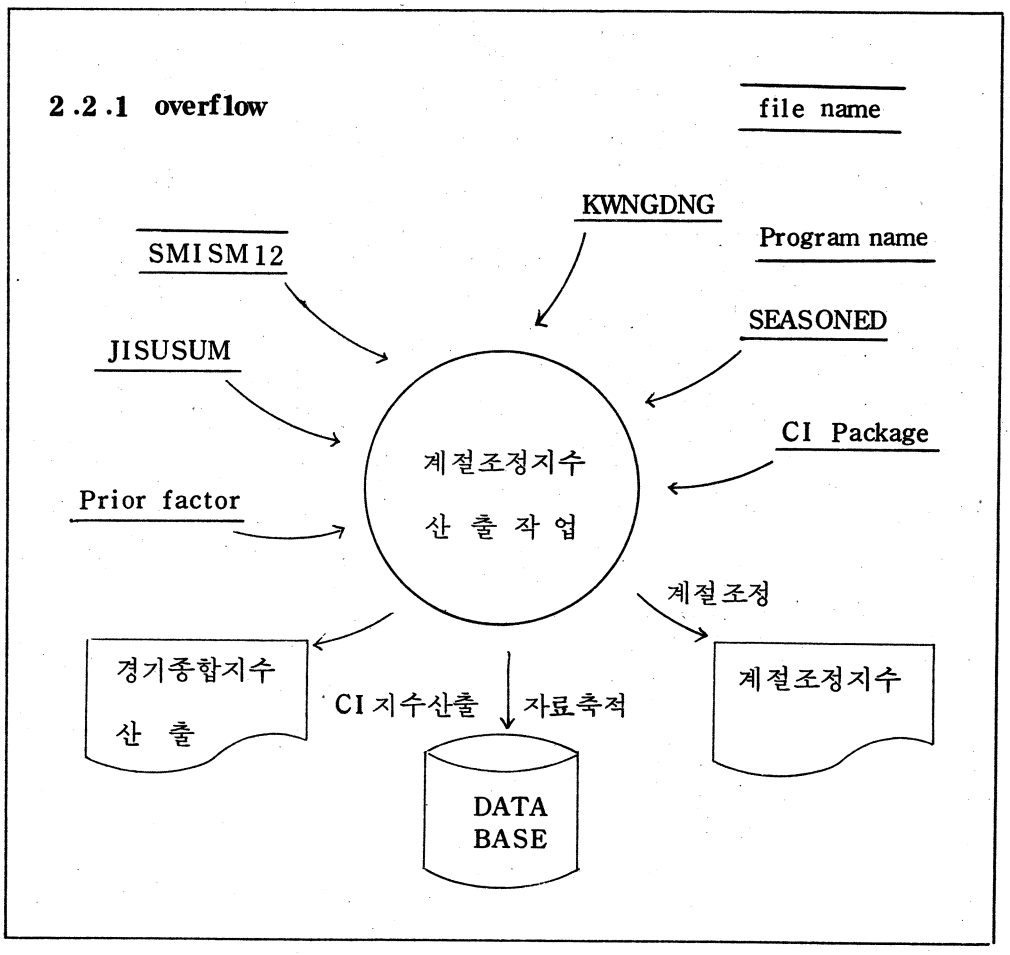
2. 계절조정지수 산정을 위한 매월작업

2. 계절조정지수 산정을 위한 매월작업

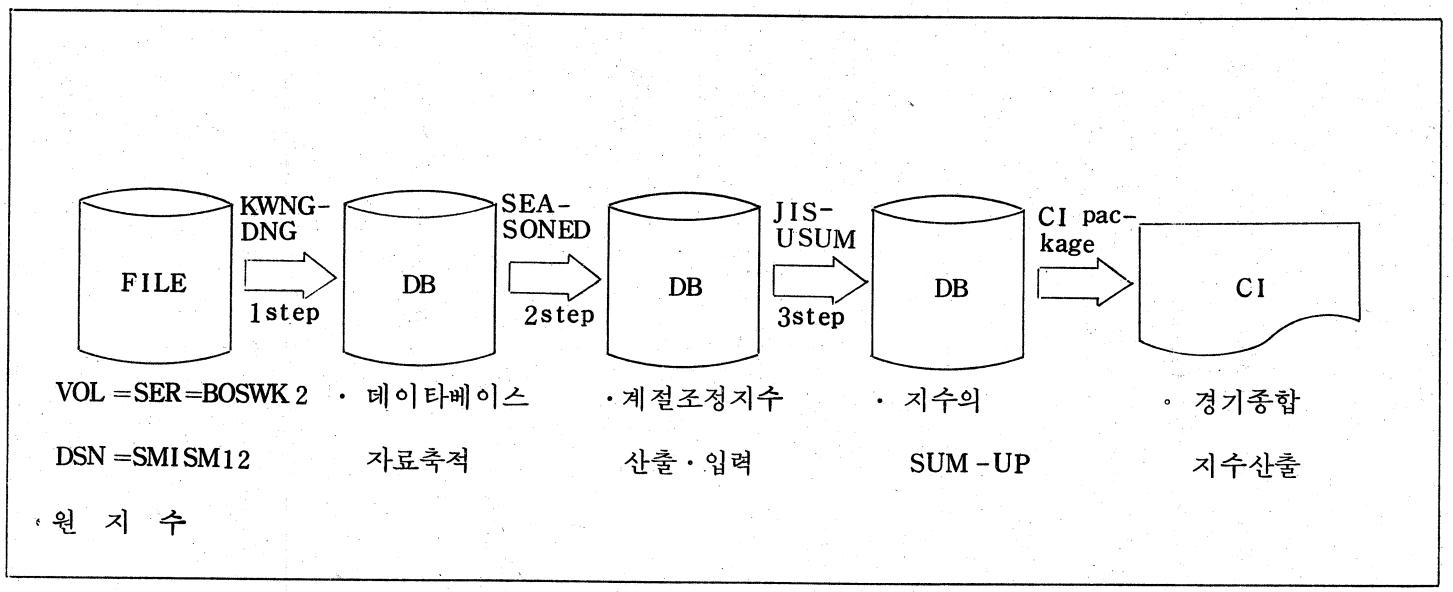
2.1 요 지

본 작업과정에서는 매년 재조정작업을 거쳐 산정 고정된 계절지수와 매월별의 사전조정지수(Prior factor)를 이용하여 매월 원지수를 가공 계절조정지수를 산출하는데 그 목적이 있다.

2.2 작업의 흐름도



2.2.2 detail flow



2.2.3 과정별 program의 기능

STEP 1. KWNGDNG program - PL/I

본 program은 Summary file SMISM12로부터 지수 및 물량 Data를 Data Base로 축적시키는데 목적이 있다. 지수는 Data별로 다소 틀리지만 소분류까지 load하고 있으며, 현재 load 대상 Data는 다음과 같다.

현재 축적되고 있는 내용

구분	내 용	축적대상과 해당 DB 이름	구분	내 용	축적대상과 해당 DB 이름
2A	생 산 지 수 (부가가치)	J = NPRDI Q = NPRDQ S = PRODS X = PRODX	2J	특수분류생산지수	J = NPRSI S = PROSS X = PROSX
			416	노동생산성지수	J = LAPDI
2E	출 하 지 수	J = NSHPI Q = NSHPQ S = SHPTS X = SHPTX	424	명목임금지수	J = NOWAI
			425	실질임금지수	J = REWAI
			61	생산능력지수	J = NPRCI Q = NPRCQ
2F	시판출하지수	J = DSHPI Q = DSHPQ	62	생산능력생산지수	J = NPRAI Q = NPRAQ S = PRACI X = PRACX
2G	수출출하지수	J = ESHPI Q = ESHPQ			
2H	재 고 지 수	J = NINTI Q = NINTQ S = INVTS X = INVTX	63	가 동 율 지 수	J = NPRPI Q = NPRPQ S = PROPS X = PROPX
2I	재 고 율 지 수	J = SHINI	이상 13개 자료가, 33개 Data Base name으로 축적된다.		

* J=지 수

Q=물 량

S=계절지수

X=계절조정된 지수

* 여기에서 계절지수(××××S)는 계절조정지수(××××X)를 산출키 위하여 X-11, ARIMA package에 의하여 기 산출되어진 지수를 말한다.

STEP2. SEASONED program—PL/I

계절조정지수 부분을 산출하는 과정을 처리한다. 즉 KWN-GDNG 프로그램에 의해 원지수(××××I)물량(××××Q)이 Load된후 이를 다시 본 프로그램에 의해 X-11 ARIMA에 의해 기 산출되어져 있는(년1회) 계절지수(××××S)와 사전조정지수(매월조정, Prior Factor)등을 이용 계절조정지수(××××X)를 산출한다.

STEP3. JISUSUM program—PL/I

1, 2 step에서 처리된 계절조정지수 Ratio of Sum 방식에 의해 처리하는 과정이다.

이상과 같은 STEP 3이 매월별 광공업 동태조사의 원지수 물량자료의 Data Base 축적작업 및 계절조정지수 산출을 위한 작업과정이다. 여기서 생성된 자료 및 타 기관에서 수집된 자료를 이용해 추후 CI package에 의해 경기종합지수(CI)가 산출되는 바, 여기서는 따로 설명치 않기로 한다.

2.3 작업방법

2.3.1 프로그램별 options의 사용방법

STEP	No.	내 용	설 명
STEP 1 program :KWNGDNG	(1)	처 리 여 부	"MODIFY"는 기존있는 Data에 추가하라는 뜻이고, "INSERT"는 처음으로 Data를 Load하는 경우에 사용되며 이밖의 표식에 대해서는 아무처리도 하지 않는다.
	(2)	기 간 표 시	Load코자 하는 기간년월일
	(3)	카 드 구 분	Summary file 즉 "SMISM12"에서 사용되는 Card 구분표식
	(4)	DBNAME	Load Data에 주어지는 이름
	(5)	"J1S", "MUL, "ALL"	"JIS" – 지수만 Load "MUL" – 물량만 〃 "ALL" – 모 두 〃
STEP 2	(1)	DBNAME	(1) 계절조정지수에 주어질 이름

STEP	No.	내 용	설 명
program : SEASONED	(2)		(2) 원지수 이름
	(3)		(3) 계절지수 이름
	(4)	주 기	"M", "Q", "Y", 여기서는 사실상 "M"만 사용됨.
	(5)	기 간 표 시	기간의 년월일
	(6)	Level	현재 지수는 소분류까지 처리하고 있음(3 Level)
	(7)	사전조정지수	사전조정지수의 사용여부
	(8)	처리개월수	(5) 참조
	(9)	처 리 여 부	STEP 1의 (1) 참조
STEP 3 program : JISUSUM	(1)	DBNAME	SUM - UP 대상
	(2)	기 간 표 시	기간의 년월일
	(3)	Load 여 부	"LOD", "PRT" 즉 Load 와 PRINT의 선택
	(4)	처 리 여 부	STEP 2의 (8) 참조

프로그램 : 계절조정지수 산정을 위한 매월작업

RUN PROGRAM

매월 광공업동태조사 자료의 처리과정 결과로 산출되어지는 원지수와 물량자료의 Data Base 축적과 원지수를 이용 계절 조정지수를 산정키 위한 작업

```
FILE  RUN       KWNGDNG  A1  VM/SP RELEASE 2.1 EXPRESS PUT8302+ SLU203

//ZNPSPY88   JOB  CLASS=V,PRTY=13,MSGCLASS=X
//STEP1      EXEC IMSRUN,MBR=KWNGDNG,PSB=ZVPSBA,SOUT=X
//STEPCAT    DD   DSN=ZUSERCAT,DISP=SHR
//           DD   DSN=SMICT12,DISP=SHR
//SUMF   DD   DSN=SMISM12,UNIT=3350,DISP=SHR,VOL=SER=BOSWK2
//SYSIN  DD  *
MODIFY 84098410 2 A  NPRDI  JIS
MODIFY 84098410 2 E  NSHPI  JIS
MODIFY 84098410 2 F  DSHPI  JIS
MODIFY 84098410 2 G  ESHPI  JIS
MODIFY 84098410 2 H  NINTI  JIS
MODIFY 84098410 2 I  SHINI  JIS
MODIFY 84098410 2 J  NPRSI  JIS
MODIFY 84098410 2 K  NSHSI  JIS
MODIFY 84098410 4 15 LAINT  JIS
MODIFY 84098410 6 1  NPRCI  JIS
MODIFY 84098410 6 2  NPRAI  JIS
MODIFY 84098410 6 3  NPRPI  JIS
------ -------- ---  ----- ---
 (1)      (2)   (3)   (4)  (5)
/*
//STEP2      EXEC IMSRUN,MBR=SEASONED,PSB=ZVPSBA,SOUT=X
//SYSIN  DD  *
PRODXNPRDIPRODS M8409 8410 3 PRIOR 002 MODIFY
SHPTXNSHPISHPTS M8409 8410 3 PRIOR 002 MODIFY
INVTXNINTIINVTS M8409 8410 3 DONOT 002 MODIFY
PROSXNPRSIPROSS M8409 8410 3 PRIOR 002 MODIFY
PROPXNPRPIPROPS M8409 8410 3 PRIOR 002 MODIFY
PRACXNPRAIPRACS M8409 8410 3 PRIOR 002 MODIFY
--------------- ---------- - ----- --- ------
 (1)  (2)  (3) (4)   (5)  (6) (7)  (8)  (9)
/*
//STEP3      EXEC IMSRUN,MBR=JISUSUM,PSB=ZVPSBA,SOUT=X
//GO.SYSIN   DD  *
PRODX84098410 LODMODIFY
SHPTX84098410 LODMODIFY
INVTX84098410 LODMODIFY
------------- ---------
 (1)    (2)   (3) (4)
//STEP4      EXEC IMSRUN,MBR=LEETABLE,PSB=ZVPSBG,SOUT=X
//SYSIN  DD  *
  841083NPRPIPROPX3 OPERATION RATIO INDEX
  841083NPRAIPRACX3 CAPACITY PROD
  841083NPRCIDONOT3 CAPACITY INDEX
  ----------------- ---------------------
  (1)(2) (3) (4) (5)       (6)
/*
//
```

계절조정지수 산출을 위한
SOURCE PROGRAMS LIST

프로그램 : KWNGDNG

기능 : 계절 조정지수 산정작업을 위한

Data의 DB Load 과정

```
//ZNPKW088  JOB CLASS=V,PRTY=13,MSGCLASS=X
//        EXEC   IMSPLI,MBR=KWNGDNG,SOUT=X
//SYSIN       DD *
* PROCESS     INCLUDE,GS,S  ;
  ROME :   PROC(PCBP)  OPTIONS(MAIN) ;
  /*==========================================================*/
  DCL SUMF FILE SEQUENTIAL RECORD KEYED ENV(VSAM) ;
  DCL PCBP POINTER  ;
  DCL 1 INPUT_RECORD ,
            2 CODE  CHAR(1) , 2 INREC CHAR(10) , 2 YR     PIC'99' ,
            2 MN    PIC'99' , 2 WGT   FLOAT(6) , 2 UN_USE FIXED BIN(31),
            2 DATA(37) ,
              3 QIN  FIXED BIN(31) ,   3 JIS  FLOAT(6) ,
            2 UN_USED CHAR(1)  ;
  DCL 1 GIGAN   ,
            2 FROMY  PIC'99' , 2 FROMM  PIC'99' ,
            2 TOY    PIC'99' , 2 TOM    PIC'99' ;
  DCL   (MGIGAN,QGIGAN,YGIGAN)  LIKE  GIGAN ;
   /*-----------------------------------------------*/
  DCL   (MFIRST,QFIRST,YFIRST,
            MLAST ,QLAST ,YLAST )  FIXED        ;
  DCL 1  VARIABLES_USING_IN_THIS_PROCESS,
            2 FLAG            BIT(1)  INIT('0'B)   ,
            2 MQY             CHAR(1) INIT('M')    ,
            2 AIM             CHAR(2)          ,
            2 FUNCT           CHAR(6) INIT('MODIFY')    ,
            2 GUBUN_CHECK     CHAR(2) INIT('  ')   ,
            2 MIDDATA(1)      FLOAT   BIN(53) ,
            2 QDATA           CHAR(2) INIT('OK')   ,
            2 YDATA           CHAR(2) INIT('OK')   ,
            2 DBNAME    CHAR(5) ,  2 DBKEY     CHAR(9) ,
            2 WHAT      CHAR(3) ,  2 CARD      CHAR(1) ;
  DCL 1  VARIABLES_IN_DBLDING_ROUTINE   ,
            2 HTITLE          CHAR(30)        ,
            2 HUNIT           CHAR(11)        ,
            2 HWGT            FLOAT INIT(0)        ;
  DCL     PSW       PIC'9'    INIT(1)      ;
/*****************************************************************/
   DBKEY = '000000000'     ;
   ON  ENDFILE(SYSIN)   FLAG = '1'B;
   GET SKIP EDIT(FUNCT,GIGAN.FROMY,GIGAN.FROMM,GIGAN.TOY,GIGAN.TOM,
                CARD,GUBUN_CHECK,DBNAME,WHAT)
             (A(6),X(1),4 F(2),X(1),A(1),X(1),A(2),X(1),A(5),X(1),A(3));
   READ FILE(SUMF) INTO(INPUT_RECORD) ;
   /*****************************************************************/
    /****      MAIN  ROUTINE                                    *****/
   /*****************************************************************/
  DO   WHILE(FLAG='0'B)    ;
    IF   FUNCT='MODIFY'| FUNCT='INSERT' THEN CALL ACTION;
    GET SKIP EDIT(FUNCT,GIGAN.FROMY,GIGAN.FROMM,GIGAN.TOY,GIGAN.TOM,
             CARD,GUBUN_CHECK,DBNAME,WHAT)
             (A(6),X(1),4 F(2),X(1),A(1),X(1),A(2),X(1),A(5),X(1),A(3));
  END  ;
   /*****************************************************************/
 /*==       ACTION                                                ==*/
```

```
/*****************************************************************/
ACTION : PROCEDURE    ;
ON  ENDFILE(SUMF)  BEGIN; FLAG = '1'B; RETURN; END;
AIM = '  ' ;
/*===============================================================*/
IF  CODE¬= CARD THEN DO WHILE(CODE¬= CARD) ;
    READ FILE(SUMF)  INTO(INPUT_RECORD)   ; END ;
    IF CARD = '2' | CARD = '6' THEN  AIM = SUBSTR(INREC,1,1) ;
                                ELSE  AIM = SUBSTR(INREC,5,2) ;
IF (CARD='2' | CARD='6') & GUBUN_CHECK ¬= AIM THEN DO
    UNTIL(GUBUN_CHECK= AIM) ;
      READ FILE(SUMF)  INTO(INPUT_RECORD)   ;
      AIM = SUBSTR(INREC,1,1) ;
    END;
/*===============================================================*/
/*=========================================*//*   MONTH LOAD INQUIRY */
MGIGAN= GIGAN ;
MFIRST= 37- ((YR- GIGAN.FROMY)*12+ (MN-GIGAN.FROMM)) ;
MLAST = 37- ((YR- GIGAN.TOY  )*12+ (MN-GIGAN.TOM  )) ;
/*===============================================================*/
/*=========================================*//* QUATER LOAD INQUIRY */
IF MOD(GIGAN.TOM,3)¬=0 THEN DO;
      DO UNTIL(MOD(GIGAN.TOM,3)= 0) ; GIGAN.TOM =GIGAN.TOM +1 ; END ;
      IF YR= GIGAN.TOY &GIGAN.TOM >MN THEN DO ;
         GIGAN.TOM= GIGAN.TOM- 3 ;
         IF GIGAN.TOM= 0 THEN DO ;
            GIGAN.TOM=12 ; GIGAN.TOY= GIGAN.TOY- 1 ; END ;
         END   ;
   END   ;
IF MOD(GIGAN.FROMM,3)¬= 1 THEN DO UNTIL(MOD(GIGAN.FROMM,3)=1) ;
   GIGAN.FROMM= GIGAN.FROMM- 1 ;   END ;
IF (GIGAN.TOY<GIGAN.FROMY) |
   ((YR=GIGAN.FROMY& YR=GIGAN.TOY) &(GIGAN.FROMM>GIGAN.TOM |
   MOD(MGIGAN.TOM,3)¬=0)) THEN DO;
   QDATA='NO'; YDATA='NO' ;  GOTO NEXT ; END ;
IF CARD¬='4' THEN DO;
   QFIRST= 37- ((YR- GIGAN.FROMY)*12+ (MN-GIGAN.FROMM)) ;
   QLAST = 37- ((YR- GIGAN.TOY  )*12+ (MN-GIGAN.TOM  )) ;
   GIGAN.FROMM= GIGAN.FROMM/ 3+ 1 ;
   GIGAN.TOM  = GIGAN.TOM  / 3    ;
   QGIGAN= GIGAN ;
   END;
/*===============================================================*/
/*=========================================*//*  YEAR LOAD INQUIRY */
IF  GIGAN.TOY=GIGAN.FROMY & GIGAN.TOM¬=4 THEN DO;
    YDATA='NO'; GOTO NEXT; END;
IF  GIGAN.TOM¬= 4 THEN  GIGAN.TOY= GIGAN.TOY- 1;
IF  GIGAN.TOY < GIGAN.FROMY THEN DO; YDATA='NO'; GOTO NEXT; END;
GIGAN.TOM=12 ;
GIGAN.FROMM=1;
YFIRST= 37- ((YR- GIGAN.FROMY)*12+ (MN-GIGAN.FROMM)) ;
YLAST = 37- ((YR- GIGAN.TOY  )*12+ (MN-GIGAN.TOM  )) ;
GIGAN.FROMM, GIGAN.TOM= 0 ;
YGIGAN= GIGAN ;
/*===============================================================*/
```

```
/*=  TRANSFER TO DBLOADING_PART    =*/
/*==================================*/
NEXT :
SELECT(CODE) ;
  WHEN('4')  DO ;
    DO  WHILE(CODE='4') ;
      IF AIM='15' THEN GOTO LOOP ; ELSE
      IF AIM='16' THEN DBNAME='LAPDI' ; ELSE
      IF AIM='24' THEN DBNAME='NOWAI' ; ELSE
      IF AIM='25' THEN DBNAME='REWAI' ; ELSE GOTO LOOP ;
      DBKEY= SUBSTR(INREC,2,3) ;
      CALL DBLOAD ;
    LOOP  :
    READ FILE(SUMF)  INTO(INPUT_RECORD)   ;
    AIM= SUBSTR(INREC,5,2) ;
    END    ;
  END    ;
  /*-----------------------------------*/
  OTHERWISE  DO ;
   DO WHILE(GUBUN_CHECK=AIM) ;
    INREC= TRANSLATE(INREC,'0',' ') ;
    DBKEY= SUBSTR(INREC,2,3)|| SUBSTR(INREC,7,2) ;
    IF ( SUBSTR(DBKEY,1,1)='2' | SUBSTR(DBKEY,1,2)='39' )
         & SUBSTR(DBKEY,4,2)¬='00'
         THEN SUBSTR(DBKEY,3,1)='1' ; ELSE
    IF SUBSTR(DBKEY,1,2)='4' & SUBSTR(DBKEY,4,2)¬='00'
         THEN SUBSTR(DBKEY,2,4)='1101' ;
    IF CODE¬='6' | (CODE='6' & SUBSTR(DBKEY,4,2)='00') THEN
         CALL  DBLOAD  ;
    READ FILE(SUMF)  INTO(INPUT_RECORD)   ;
    AIM= SUBSTR(INREC,1,1) ;
   END  ;
  END  ;
END  ;
END ACTION ;
  /*****************************************************************/
/*==      DBLOAD                                                  ==*/
/*****************************************************************/
DBLOAD : PROC ;
 DCL IN_DATA(37)  FLOAT(16) ;
 IN_DATA= 0      ;
 GIGAN  =MGIGAN ;
 MQY    = 'M'   ;
/*=================================================================*/
/*=================================*//*  MONTH   LOAD REAL   */
IF CARD= '6' & DBKEY='300' THEN DBKEY='000';
IF CARD= '4' THEN IF AIM= '15' THEN WHAT='MUL' ;
                                ELSE WHAT='JIS' ;
IF WHAT¬='MUL' THEN DO ;
   DO I = MFIRST TO MLAST ;
      IN_DATA(I-MFIRST+1)= DATA(I).JIS*10+ 0.050001 ;
   END   ;
   CALL LOAD(PCBP,FUNCT,DBNAME,DBKEY,MQY,MGIGAN,IN_DATA,HTITLE,
            HUNIT,HWGT)    ;
   PUT  LIST('M_STAT_CODE= ',M_STAT_CODE);
```

```
 END   ;
 IF WHAT¬='JIS' THEN
 IF AIM¬='I' & AIM¬='J' & AIM ¬='K' & AIM¬='16' & AIM¬='24' & AIM¬='25'
    &  SUBSTR(INREC,5,5)¬='00000' THEN       DO ;
       IF CARD¬='4' THEN SUBSTR(DBNAME,5,1) = 'Q' ;
       DO I = MFIRST TO MLAST ;
            IN_DATA(I-MFIRST+1)= DATA(I).QIN ;
            END   ;
       CALL LOAD(PCBP,FUNCT,DBNAME,DBKEY,MQY,MGIGAN,IN_DATA,HTITLE,
                 HUNIT,HWGT)    ;
       PUT  LIST('M_STAT_CODE= ',M_STAT_CODE);
     END   ;
/*==============================================================*/
/*=========================================*//*  QUATER  LOAD REAL  */
IF QDATA='NO' THEN RETURN ;
IF CARD ='4'  THEN GOTO BELOW;
  MQY   = 'Q'    ;
  GIGAN = QGIGAN ;
  IN_DATA  = 0 ;
  IF CARD¬= '4' THEN SUBSTR(DBNAME,5,1) ='I' ;
  II= 1 ; J= QFIRST ;
  IF WHAT¬='MUL' THEN DO ;
     DO UNTIL (J> QLAST) ;
         MIDDATA=0 ; I= 1 ;
         DO I=1 TO 3;
            MIDDATA(1)= MIDDATA(1)+ DATA(J).JIS*10 ;
            J = J + 1 ;
            END   ;
         IN_DATA(II)= MIDDATA(1)/3 +0.050001 ;
         II= II+ 1 ;
         END ;
     CALL LOAD(PCBP,FUNCT,DBNAME,DBKEY,MQY,QGIGAN,
                     IN_DATA,HTITLE,HUNIT,HWGT) ;
     PUT  LIST('M_STAT_CODE= ',M_STAT_CODE);
     END ;
/*==============================================================*/
  IF WHAT¬='JIS' THEN
    IF AIM¬='I' & AIM¬='J' & AIM ¬='K' & AIM¬='16' & AIM¬='24' &
       AIM¬='25' &  SUBSTR(INREC,5,5)¬='00000' THEN       DO ;
       II= 1 ; J= QFIRST ; IN_DATA= 0 ;
    IF CARD¬='4' THEN SUBSTR(DBNAME,5,1) = 'Q' ;
    SELECT(DBNAME) ;
       WHEN('INVTQ') DO UNTIL(J>QLAST) ;
            IN_DATA(II)= DATA(J+2).QIN+ 0.50001 ;
            J= J+ 3 ;
            END ;
       OTHERWISE      DO UNTIL(J>QLAST) ;
            DO I = 1 TO 3 ;
               IN_DATA(II)= IN_DATA(II)+ DATA(J).QIN ;
               J= J+ 1 ;
               END    ;
            II= II+ 1 ;
            END ;
      END   ;
      CALL LOAD(PCBP,FUNCT,DBNAME,DBKEY,MQY,QGIGAN,IN_DATA,HTITLE,
```

```
                HUNIT,HWGT)     ;
      PUT  LIST('M_STAT_CODE= ',M_STAT_CODE);
    END   ;
 /*==================================================================*/
 /*=========================================*//*  YEAR    LOAD REAL  */
  BELOW :
   IF YDATA='NO' THEN RETURN ;
   MQY   = 'Y'    ;
   GIGAN = YGIGAN ;
   IN_DATA  = 0 ;
   IF CARD¬= '4' THEN SUBSTR(DBNAME,5,1) ='I' ;
      II= 1 ; J= YFIRST ;
   IF WHAT¬='MUL' THEN DO ;
      DO UNTIL (J> YLAST) ;
         MIDDATA=0 ;
         DO I =1 TO 12 ;
            MIDDATA(I)= MIDDATA(I)+ DATA(J).JIS*10 ;
            J = J + 1 ;
         END   ;
         IN_DATA(II)= MIDDATA(I)/12 +0.050001 ;
         II= II+ 1 ;
         END ;
      CALL LOAD(PCBP,FUNCT,DBNAME,DBKEY,MQY,YGIGAN,
                IN_DATA,HTITLE,HUNIT,HWGT) ;
      PUT  LIST('M_STAT_CODE= ',M_STAT_CODE);
      END ;
 /*==================================================================*/
   IF WHAT¬='JIS' THEN
      IF AIM¬='I' & AIM¬='J' & AIM ¬='K' & AIM¬='16' & AIM¬='24' &
         AIM¬='25' &  SUBSTR(INREC,5,5)¬='00000' THEN      DO ;
         IF CARD¬='4' THEN SUBSTR(DBNAME,5,1) = 'Q' ;
            II= 1 ; J= YFIRST ; MIDDATA=0; IN_DATA= 0 ;
         SELECT(DBNAME) ;
           WHEN('INVTQ') DO UNTIL(J>YLAST) ;
                IN_DATA(II)= DATA(J+11).QIN+ 0.50001 ;
                J= J+ 12;
                END ;
           OTHERWISE     DO UNTIL(J>YLAST) ;
                DO I = 1 TO 12  ;
                   IN_DATA(II)= IN_DATA(II)+ DATA(J).QIN ;
                   J= J+ 1 ;
                   END    ;
                II= II+ 1 ;
                END ;
           END   ;
           CALL LOAD(PCBP,FUNCT,DBNAME,DBKEY,MQY,YGIGAN,IN_DATA,HTITLE,
                     HUNIT,HWGT)     ;
           PUT  LIST('M_STAT_CODE= ',M_STAT_CODE);
    END   ;
  END DBLOAD ;
 /*****************************************************************/
   /*=============================================================*/
  %INCLUDE LOADS ;
  END ROME ;
/*
```

//

프로그램 : SEASONED

기능 : 매월 계정지수 산출과정

```
//ZDBTAS71  JOB CLASS=V,PRTY=13                                          ADJ00010
//            EXEC IMSDLI,MBR=SEASONED                                   ADJ00020
//SYSLIB      DD DSN=IMSVS.SSSLIB,DISP=SHR                               ADJ00030
//SYSPRINT    DD SYSOUT=A,DCB=(LRECL=121,RECFM=FBA,BLKSIZE=605)          ADJ00040
//SYSIN     DD  *                                                        ADJ00050
* PROCESS GS,NEST,INCLUDE                                                ADJ00060
 MICHIN  PROC(PCBP) OPTIONS(MAIN)                                        ADJ00070
      DCL ONCODE BUILTIN                                                 ADJ00080
       INCLUDE DBDCLL                                                    COM00060
      DCL  CODE CHAR(9),                                                 ADJ00100
           SAVE(50) CHAR(9),
          DBNAME  CHAR(5),                                               ADJ00240
          DBKEY   CHAR(9),                                               ADJ00250
          LEVEL   PIC 9  ,                                               ADJ00260
          HTITLE CHAR(30) ,                                              ADJ00270
          HUNIT  CHAR(11) ,                                              ADJ00280
          HWGT  FLOAT,                                                   ADJ00290
          TRUE   CHAR(5),                                                ADJ00300
          ORIGN  CHAR(5),                                                ADJ00310
          FACTOR CHAR(5),                                                ADJ00320
          SEASON CHAR(5),                                                ADJ00330
          FSEQ   CHAR(1),                                                ADJ00340
         STARTY PIC 99 , STARTM PIC 99 , LEVELIN PIC 9 ,                 ADJ00350
         YEAR PIC 99 , MONTH PIC 99 ,
          INDATA(360) FLOAT(16),                                         ADJ00360
          PSW PIC 9  INIT(1),                                            ADJ00370
          MQY CHAR(1),                                                   ADJ00380
          FIRST FIXED,
          FUNCT CHAR(6),                                                 ADJ00390
          1 GIGAN,                                                       ADJ00400
           2 FROMY PIC 99  ,                                             ADJ00410
           2 FROMM PIC 99  ,                                             ADJ00420
           2 TOY   PIC 99  ,                                             ADJ00630
           2 TOM   PIC 99  ,                                             ADJ00440
           WANT   PIC 999 ,                                              ADJ00450
           A TABLE(50,360) FLOAT(16),                                    ADJ00460
           D TABLE(50,360) FLOAT(16),                                    ADJ00470
           P TABLE(360) FLOAT(16),                                       ADJ00480
           FINAL(360)   PIC 999999                                       ADJ00490
       DCL VALUE PIC 9 ,                                                 ADJ00500
           ROOL FIXED INIT(1),
           (I,J,ROOT,T) FIXED                                            ADJ00510
     ON ENDFILE(SYSIN) GOTO LAST                                         ADJ00520
     OPEN FILE(SYSPRINT) LINESIZE(132)                                   ADJ00530
   REREAD                                                                ADJ00540
      GET SKIP EDIT(TRUE,ORIGN,SEASON,FSEQ,STARTY,STARTM,YEAR,MONTH,     ADJ00550
             LEVELIN,FACTOR,WANT,FUNCT)(A(5),A(5),A(5),X(1),A(1),        ADJ00560
        2 F(2),X(1),2 F(2),X(1),F(1),X(1),A(5),X(1),F(3),X(1),A(6))      ADJ00570
    AGAIN                                                                ADJ00590
        DBNAME=ORIGN                                                     ADJ00600
        KEY=ALLZERO                                                      ADJ00620
        LOWLEV=LEVELIN                                                   ADJ00630
        PLEVEL=0                                                         ADJ00640
        DBKEY=ALLZERO                                                    ADJ00650
        FREQ=FSEQ
```

```
        ROOL=1
        CALL LOOPED                                                     ADJ00660
        DBNAME=SEASON                                                   ADJ00670
        KEY=ALLZERO
        LOWLEV=LEVELIN
        PLEVEL=0
        DBKEY=ALLZERO
        FREQ=FSEQ
        ROOL=1
        CALL LOOPED                                                     ADJ00680
   NEXT                                                                 ADJ00710
        IF FACTOR= PRIOR  THEN DO                                       ADJ00720
        DBNAME= PRFCT                                                   ADJ00730
        KEY=ALLZERO                                                     ADJ00740
        PLEVEL=0                                                        ADJ00760
        DBKEY=ALLZERO                                                   ADJ00770
        FREQ=FSEQ                                                       ADJ00780
        CALL DBGET                                                      ADJ00790
        IF PLEVEL 0 THEN
        DO   CALL FACTORA
        END
        END  /****** PRIOR END *****/
    NEXT1
        CALL ADJUST
        GOTO REREAD                                                     ADJ00810
    LOOPED  PROC
        /****** DB READ PROC ********/
     LOOP
          CALL DBGET
          IF PLEVEL   0 THEN
          DO
              CALL SUBSAVE
              GOTO LOOP
          END
      END  /******* LOPPED PROC END ********/
    SUBSAVE  PROC                                                       ADJ00820
      /****** THIS PROC IS A SUBROUTINE FOR DATA SAVING ********/       ADJ00830
       CODE=SUBSTR(KEY,1,9)                                             ADJ00880
       SELECT(CODE)
        WHEN( 211000000 )  RETURN
        WHEN( 231000000 )  RETURN
        WHEN( 291000000 )  RETURN
        WHEN( 391000000 )  RETURN
        WHEN( 410000000 )  RETURN
        WHEN( 411000000 )  RETURN
        OTHERWISE GO TO ALPHA
       END
    ALPHA
       IF DBNAME=ORIGN THEN DO
        SAVE(ROOL)=CODE
        FIRST=ROOL
        END
       IF ONCODE= 51 THEN  DO                                           ADJ00890
        IF CODE =ALLZERO                                                ADJ00900
          THEN PUT EDIT(CODE, NAME NOT FOUND )(SKIP,A(10),A)            ADJ00910
```

```
        END                                                          ADJ00920
       IF ONCODE=52 THEN                                             ADJ00930
         PUT EDIT(CODE, DUPLICATE NAME )(SKIP,A(10),A)
       I=(YEAR-INITY)*12 +(MONTH-INITM) +1
      ROOT=I - WANT + 1
        DO J=ROOT TO I
        IF DBNAME=ORIGN THEN
         A TABLE(ROOL,J-ROOT+1)=DATA(J)
        ELSE IF DBNAME=SEASON THEN
         D TABLE(ROOL,J-ROOT+1)=DATA(J)
        END  /**** ROOT ****/
     OUTED
        ROOL=ROOL+1
       END  /****** SUBSAVE ROUTINE END ********/
      FACTORA  PROC                                                  ADJ00940
       CODE=SUBSTR(KEY,1,9)
       IF CODE = 000000000  THEN GOTO DOWN
       IF ONCODE= 51 THEN DO
        IF CODE =ALLZERO THEN
         PUT EDIT(CODE, NAME NOT FOUND )(SKIP,X(2),A(10),A)
       END
       IF ONCODE= 52 THEN
         PUT EDIT(CODE, DUPLICATE NAME )(SKIP,X(2),A(10),A)
       I=(YEAR-INITY)*12 +(MONTH-INITM) + 1
      ROOT=I- WANT + 1
         DO J=ROOT TO I
         P TABLE(J-ROOT+1)=DATA(J)                                   ADJ00970
        END  /**** ROOT ******/                                      ADJ00980
      DOWN
       END  /******FACTOR PROC **********/                           ADJ00990
     ADJUST  PROC                                                    ADJ01010
     /**** THIS PROC IS ANOTHER ONE FOR COMPUTING   DBLOADING ******/ ADJ01020
     /***** SEASONAL ADJUSTED INDEX  ******/                          ADJ01030
     DO J=1 TO FIRST
       DO ROOL=1 TO WANT
       IF ROOL WANT THEN GOTO TOWN
       ELSE DO
       IF A TABLE(J,ROOL)=0
          D TABLE(J,ROOL)=0  THEN GOTO TOWN
       ELSE DO
       IF FACTOR= PRIOR  THEN                                        ADJ01120
    FINAL(ROOL)=(A TABLE(J,ROOL)/P TABLE(ROOL))*10000/D TABLE(J,ROOL) ADJ01130
                 *1000 + 0.5                                         ADJ01140
     ELSE FINAL(ROOL)=(A TABLE(J,ROOL)/D TABLE(J,ROOL))*10000 +0.5   ADJ01150
       INDATA(ROOL) = FINAL(ROOL)                                    ADJ01190
       END  /****** ELSE END *********/                              ADJ01200
       END  /****** ELSE END *********/
       END  /****** ROOL * LOOP END ********/
    TOWN
      DBNAME=TRUE                                                    ADJ01210
      DBKEY=SAVE(J)
      GIGAN.FROMY=STARTY
      GIGAN.FROMM=STARTM
      GIGAN.TOY=YEAR
      GIGAN.TOM=MONTH
```

```
      MQY=FSEQ
     /******* CALL PRT  **********/                                    ADJ01230
          CALL LOAD(PCBP,FUNCT,DBNAME,DBKEY,MQY,GIGAN.INDATA,          ADJ01240
                HTITLE,HUNIT,HWGT)                                     ADJ01250
      END  /******* WANT  J LOOP END ******/                           ADJ01260
   END   /******* ADJUST PROC ROUTINE END *********/
   PRT  PROC                                                           ADJ01300
      PUT SKIP EDIT(DBNAME,DBKEY,MQY,GIGAN.FROMY,GIGAN.FROMM,          ADJ01310
               GIGAN.TOY,GIGAN.TOM)
              (X(5),A(5),A(10),A(1),X(1),4 (X(1),F(2)))                ADJ01320
      PUT SKIP EDIT((INDATA(I) DO I=1 TO 2))(X(2),2 (X(2),F(8,0)))     ADJ01330
   END  /*** LIST PRINTING *****/                                      ADJ01350
   INCLUDE DBGETLEE
   INCLUDE LOADS                                                       ADJ01530
  LAST                                                                 ADJ01540
    CLOSE FILE(SYSPRINT)                                               ADJ01550
 END   /*****MICHIN NOM ******/                                        ADJ01560
/*                                                                     ADJ01570
//                                                                     ADJ01580
```

프로그램 : JISUSUM

기능 : 계절조정지수의 Summation 과정

```
//ZDBSUM71  JOB CLASS=V,PRTY=13                                          SUM00010
//            EXEC IMSPLI,MBR=SUMJISUR                                   SUM00020
//SYSLIB      DD DSN=IMSVS.SSSLIB,DISP=SHR                               SUM00030
//SYSPRINT    DD SYSOUT=A,DCB=(LRECL=121,RECFM=FBA,BLKSIZE=605)          SUM00040
//SYSIN     DD  *                                                        SUM00050
* PROCESS GS,NEST,INCLUDE                                                SUM00060
 GOLBIN  PROC(PCBP) OPTIONS(MAIN)                                        SUM00070
       INCLUDE DBDCLL                                                    SUM00080
      DCL ONCODE BUILTIN,                                                SUM00090
          DBNAME  CHAR(5),                                               SUM00100
          DNAME   CHAR(5),                                               SUM00110
          DBKEY   CHAR(9),                                               SUM00120
          LEVEL   PIC 9  ,                                               SUM00130
          LEVELIN PIC 9  ,                                               SUM00140
          HTITLE CHAR(30) ,                                              SUM00150
          HUNIT  CHAR(11) ,                                              SUM00160
          HWGT  FLOAT,                                                   SUM00170
          CHANGE CHAR(5),                                                SUM00180
          INDATA(360) FLOAT(16),                                         SUM00190
          PSW PIC 9  INIT(1),                                            SUM00200
          MQY CHAR(1),                                                   SUM00210
          FUNCT CHAR(6),                                                 SUM00220
           (FY,FM,TY,TM) PIC 99 ,                                        SUM00230
          1 GIGAN,                                                       SUM00240
           2 FROMY PIC 99 ,                                              SUM00250
           2 FROMM PIC 99 ,                                              SUM00260
           2 TOY   PIC 99 ,                                              SUM00270
           2 TOM   PIC 99 ,                                              SUM00280
          (FY1,FY2,FY3,FM1,FM2,FM3) PIC 99 ,                             SUM00290
          CHCODE   CHAR(1),                                              SUM00300
          WGT1     PIC 99999.9 ,                                         SUM00310
          WEIGHT   CHAR(3),                                              SUM00320
          WGTOLD(3)  PIC 99999.9 ,                                       SUM00330
          NONO FIXED,                                                    SUM00340
          CODE CHAR(6)                                                   SUM00350
         DCL CHECK CHAR(4)                                               SUM00360
    ON ENDFILE(SYSIN) GOTO LAST                                          SUM00370
    OPEN FILE(SYSPRINT) LINESIZE(132)                                    SUM00380
  ROLL                                                                   SUM00390
    GET SKIP EDIT(DNAME,FY,FM,TY,TM,FUNCT,LEVELIN,CHECK,FY1,FM1,         SUM00400
              FY2,FM2,FY3,FM3,WEIGHT,WGTOLD(1),WGTOLD(2),WGTOLD(3))      SUM00410
      (A(5),4 F(2),A(6),F(1),X(1),A(4),X(1),2 F(2),X(1),2 F(2),X(1),     SUM00420
       2 F(2),X(1),A(3),3 F(7,1))                                        SUM00430
     CALL SUMJISU                                                        SUM00440
     GOTO ROLL                                                           SUM00450
     SUMJISU  PROC                                                       SUM00460
     /****** ADJUSTED INDEXES WILL BE PROCESSING FOR SUMMATION *****/    SUM00470
     /****** PROCESSSING FLOWS *************/                            SUM00480
        DCL I      FIXED,                                                SUM00490
            JJ     FIXED, AA FIXED,                                      SUM00500
            K      FIXED,                                                SUM00510
            NOM1   FIXED, NOM FIXED,                                     SUM00520
            ST     FIXED,                                                SUM00530
            LT     FIXED,                                                SUM00540
            ROOT   FIXED,ROOT1 FIXED,                                    SUM00550
```

```
          ROOT2 FIXED,ROOT3 FIXED,                                  SUM00560
          WANT  FIXED, WANT1 FIXED,                                 SUM00570
          SUMOLD(360) FLOAT(16),                                    SUM00580
          SUMNEW(360) FLOAT(16)                                     SUM00590
     SUMOLD=0                                                       SUM00600
     SUMNEW=0                                                       SUM00610
     KEY=ALLZERO                                                    SUM00620
     DBNAME=DNAME                                                   SUM00630
     LOWLEV=LEVELIN                                                 SUM00640
     PLEVEL=0                                                       SUM00650
     DBKEY=ALLZERO                                                  SUM00660
     FREQ= M                                                        SUM00670
     CALL LOOPED                                                    SUM00680
     CALL DBLOAD                                                    SUM00690
     GO TO NEWEND                                                   SUM00700
  LOOPED  PROC                                                      SUM00710
     /****** DB READ PROC ********/                                  SUM00720
   LOOP                                                             SUM00730
        CALL DBGET                                                  SUM00740
        IF PLEVEL    0 THEN                                         SUM00750
        DO                                                          SUM00760
            CALL COUNTA                                             SUM00770
            GOTO LOOP                                               SUM00780
        END                                                         SUM00790
    END   /******* LOPPED PROC END ********/                        SUM00800
    COUNTA  PROC                                                    SUM00810
    /****** INDEXES ARE FORCED TO THE SUMMATION BY WEIGHT *******/  SUM00820
      CODE=SUBSTR(KEY,1,6)                                          SUM00830
      CHCODE=SUBSTR(KEY,1,1)                                        SUM00840
      IF SUBSTR(CODE,2,1)  = 0  THEN RETURN                         SUM00850
  JUMP                                                              SUM00860
      IF ONCODE=51 THEN DO                                          SUM00870
       IF CODE =ALLZERO                                             SUM00880
       THEN PUT EDIT(CODE, NAME NOT FOUND )(SKIP,A(10),A)           SUM00890
      END                                                           SUM00900
      IF ONCODE=52 THEN                                             SUM00910
      PUT EDIT(CODE, DUPLICATE NAME )(SKIP,A(10),A)                 SUM00920
      WANT=(TY-INITY)*12 +(TM-INITM)+1                              SUM00930
      IF FY1=00 THEN FY1=INITY                                      SUM00940
      IF FM1=00 THEN FM1=INITM                                      SUM00950
      GIGAN_TOY=TY      GIGAN_TOM=TM                                SUM00960
      ROOT1=(FY1-INITY)*12+ (FM1-INITM)+1                           SUM00970
      ROOT2=(FY2-INITY)*12+ (FM2-INITM)+1                           SUM00980
      ROOT3=(FY3-INITY)*12+ (FM3-INITM)+1                           SUM00990
      IF WEIGHT= NEW  THEN GO TO NEWING                             SUM01000
                      ELSE GO TO OLDING                             SUM01010
 OLDING                                                             SUM01020
      SELECT(CHCODE)                                                SUM01030
       WHEN( 2 )                                                    SUM01040
       DO    WGT1=WGTOLD(1)   GO TO NEXT   END                      SUM01050
       WHEN( 3 )                                                    SUM01060
       DO    WGT1=WGTOLD(2)   GO TO NEXT   END                      SUM01070
       WHEN( 4 )                                                    SUM01080
       DO    WGT1=WGTOLD(3)   GO TO NEXT   END                      SUM01090
       OTHERWISE GO TO NEWING                                       SUM01100
```

```
        END                                                                 SUM01110
    NEXT                                                                    SUM01120
        DO NOM1= ROOT1 TO ROOT2                                             SUM01130
        SUMOLD(NOM1-ROOT1+1)=SUMOLD(NOM1-ROOT1+1) + DATA(NOM1)*WGT1         SUM01140
        END                                                                 SUM01150
    NEWING                                                                  SUM01160
        IF WEIGHT= OLD  THEN GO TO NEXT6                                    SUM01170
        ELSE DO                                                             SUM01180
        DO NOM=ROOT3 TO WANT                                                SUM01190
        SUMNEW(NOM-ROOT3+1)=SUMNEW(NOM-ROOT3+1) + DATA(NOM)*WGT             SUM01200
        END                                                                 SUM01210
        END                                                                 SUM01220
     NEXT6                                                                  SUM01230
        SELECT(WEIGHT)                                                      SUM01240
         WHEN( OLD ) ROOT=ROOT2-ROOT1+1                                     SUM01250
         WHEN( NEW ) ROOT=WANT-ROOT3+1                                      SUM01260
         OTHERWISE   ROOT=WANT-ROOT1+1                                      SUM01270
        END                                                                 SUM01280
        WANT1=ROOT2-ROOT1+1                                                 SUM01290
       ENDCO                                                                SUM01300
      END   /********* COUNTA PROC END ********/                            SUM01310
           /********** NOW DBDATA SUCCESSFULLY READ ********/               SUM01320
    DBLOAD  PROC                                                            SUM01330
        INDATA=0                                                            SUM01340
       SELECT(WEIGHT)                                                       SUM01350
        WHEN( OLD )                                                         SUM01360
        DO                                                                  SUM01370
        DO ST=1 TO ROOT                                                     SUM01380
        INDATA(ST)=SUMOLD(ST)/10000 + 0.49999                               SUM01390
        END                                                                 SUM01400
        END  /****** WHEN END *******/                                       SUM01410
        WHEN( NEW )                                                         SUM01420
        DO                                                                  SUM01430
        DO ST=1 TO ROOT                                                     SUM01440
        INDATA(ST)=SUMNEW(ST)/10000 + 0.49999                               SUM01450
        END                                                                 SUM01460
        END  /****** WHEN END *******/                                       SUM01470
        OTHERWISE                                                           SUM01480
        DO                                                                  SUM01490
        DO ST=1 TO WANT1                                                    SUM01500
        INDATA(ST)=SUMOLD(ST)/10000 + 0.49999                               SUM01510
        END                                                                 SUM01520
        DO LT=(WANT1+1) TO ROOT                                             SUM01530
        INDATA(LT)=SUMNEW(LT-WANT1)/10000 + 0.49999                         SUM01540
        END                                                                 SUM01550
        END  /****** WHEN END *******/                                       SUM01560
       END   /******* SELECT END **********/                                SUM01570
        IF FY=00 THEN FY=FY1                                                SUM01580
        IF FM=00 THEN FM=FM1                                                SUM01590
        GIGAN.FROMY=FY                                                      SUM01600
        GIGAN.FROMM=FM                                                      SUM01610
        CODE= 000000                                                        SUM01620
        MQY= M                                                              SUM01630
        IF CHECK= PRTS  THEN CALL PRT                                       SUM01640
        ELSE DO                                                             SUM01650
```

```
        CALL LOAD(PCBP,FUNCT,DNAME,CODE,MQY,GIGAN,INDATA,                SUM01660
                 HTITLE,HUNIT,HWGT)                                      SUM01670
         END                                                             SUM01680
        END  /***** DBLOAD PROC END *******/                             SUM01690
    PRT  PROC                                                            SUM01700
  PUT SKIP EDIT(DNAME,CODE,MQY,GIGAN.FROMY,GIGAN.FROMM,GIGAN.TOY,        SUM01710
                 GIGAN.TOM)                                              SUM01720
               (X(5),A(5),A(9),A(1),X(1),2 F(2),X(1),2 F(2))             SUM01730
      PUT SKIP EDIT((INDATA(I) DO I=1 TO 360))(SKIP,X(2),12 (X(2),       SUM01740
                     F(8,0)))                                            SUM01750
    END  /*** LIST PRINTING *****/                                       SUM01760
                /** INDEXES ARE DESCENDED BY THIS ROUTINE **/            SUM01770
      NEWEND                                                             SUM01780
    END   /******** SUMJISU END ********/                                SUM01790
   INCLUDE DBGETLEE                                                      SUM01800
   INCLUDE LOADS                                                         SUM01810
  LAST                                                                   SUM01820
     CLOSE FILE(SYSPRINT)                                                SUM01830
 END   /*****GOLBIN NOM ******/                                          SUM01840
/*                                                                       SUM01850
//                                                                       SUM01860
```

프로그램 : LEETABLE

기능 : 매월 계절 조정작업 결과에 따
른 증감비, 전년대비, 동기비 등을
이 프로그램을 통해 제공하고 있다.

```
//ZUPTAB71 JOB CLASS=V,PRTY=13
//         EXEC IMSPLI,MBR=LEETABLE
//SYSLIB   DD DSN=IMSVS.SSSLIB,DISP=SHR
//SYSPRINT DD SYSOUT=A,DCB=(RECFM=FBA,LRECL=121,BLKSIZE=605)
//SYSIN    DD   *
* PROCESS NEST,INCLUDE
    DANCE  PROC(PCBP) OPTIONS(MAIN)
      DCL ONCODE BUILTIN                                                GNP00070
      DCL PCBP   POINTER                                                GNP00080
      DCL 1 PCBNAME BASED(PCBP),                                        GNP00090
           2 NAME          CHAR(8),                                     GNP00100
           2 SEGLEV        CHAR(2),                                     GNP00110
           2 STATUS        CHAR(2),                                     GNP00120
           2 M PROC OPT    CHAR(4),                                     GNP00130
           2 FILLER        FIXED BINARY(31,0),                          GNP00140
           2 M SEG NAME    CHAR(8),                                     GNP00150
           2 M LEN KFB     FIXED BINARY(31,0),                          GNP00160
           2 M NO SENSEG   FIXED BINARY(31,0),                          GNP00170
           2 M KEY FB      CHAR(60)                                     GNP00180
      DCL 1 DIREC,                                                      GNP00190
            2 DBNAME     CHAR(5),                                       GNP00200
            2 DBKEY      CHAR(9),                                       GNP00210
            2 FROMY      PIC 99 ,                                       GNP00220
            2 FROMM      PIC 99 ,                                       GNP00230
            2 TOY        PIC 99 ,                                       GNP00240
            2 TOM        PIC 99  ,                                      GNP00250
            2 UPLEV      PIC 9 ,                                        GNP00260
            2 LOWLEV     PIC 9  INIT(9),                                GNP00270
            2 FREQ       CHAR,                                          GNP00280
            2 DUMMY      CHAR                                           GNP00290
      DCL 1 ATTO,                                                       GNP00300
           2 SOURCE        CHAR(14),                                    GNP00310
           2 TOTLEV        PIC 99 ,                                     GNP00320
           2 TYPE          CHAR(5),                                     GNP00330
           2 UNITO         CHAR(11),                                    GNP00340
           2 KEYSZ(6)      PIC 9 ,                                      GNP00350
           2 WGTP          PIC 9 ,                                      GNP00360
           2 DECP          PIC 9 ,                                      GNP00370
           2 DUMMY         CHAR(1)                                      GNP00380
      DCL PLITDLI EXTERNAL ENTRY ,                                      GNP00390
          SUMKEYSZ  FIXED INIT(0),                                      GNP00400
          TESTSTR   CHAR(9),                                            GNP00410
          ALLZERO   CHAR(9) INIT( 000000000 ),                          GNP00420
          PLEVEL    PIC 9  INIT(0)                                      GNP00430
      DCL SPECIO    CHAR(30)                                            GNP00440
      DCL 1 MQYIO   ALIGNED,                                            GNP00450
           2 INITY  PIC 99 ,                                            GNP00460
           2 INITM  PIC 99 ,                                            GNP00470
           2 ENDY   PIC 99 ,                                            GNP00480
           2 ENDM   PIC 99 ,                                            GNP00490
           2 DATA(360) FIXED BIN(31,0) ,                                GNP00500
           2 DUMMY  CHAR(100)                                           GNP00510
      DCL 1 FYEARIO  ALIGNED,                                           GNP00520
           2 FYR CHAR(2),                                               GNP00530
           2 FYD CHAR(4),                                               GNP00540
```

```
      2 FDATA FIXED BIN(31,0)                                          GNP00550
   DCL 1 ATTIO ,                                                       GNP00560
      2 WGT  FLOAT,                                                    GNP00570
      2 UNIT CHAR(11)                                                  GNP00580
   DCL 1 LEVELIO ,                                                     GNP00590
      2 KEY  CHAR(9),                                                  GNP00600
      2 TITLE CHAR(30)                                                 GNP00610
   DCL 1 LEVELO,                                                       GNP00620
      2 KEYO  CHAR(5),                                                 GNP00630
      2 TITLEO CHAR(30)                                                GNP00640
   DCL  SPECSSA   CHAR(9) INIT( SPEC      ),                           GNP00650
      SPECLEV   PIC 9   DEF SPECSSA POS(5)                             GNP00660
   DCL  QSSA0  CHAR(25) INIT( TOTAL   (KEY0     =00000) ) ,             GNP00670
      QSSA1 CHAR(29) INIT( LEVEL1  (KEY1     =000000000) ) ,           GNP00680
      QSSA2 CHAR(29) INIT( LEVEL2  (KEY2     =000000000) ) ,           GNP00690
      QSSA3 CHAR(29) INIT( LEVEL3  (KEY3     =000000000) ) ,           GNP00700
      QSSA4 CHAR(29) INIT( LEVEL4  (KEY4     =000000000) ) ,           GNP00710
      QSSA5 CHAR(29) INIT( LEVEL5  (KEY5     =000000000) ) ,           GNP00720
      QSSA6 CHAR(29) INIT( LEVEL6  (KEY6     =000000000) ) ,           GNP00730
      UQSSA1 CHAR(9) INIT( LEVEL1    ),                                GNP00740
      UQSSA2 CHAR(9) INIT( LEVEL2    ),                                GNP00750
      UQSSA3 CHAR(9) INIT( LEVEL3    ),                                GNP00760
      UQSSA4 CHAR(9) INIT( LEVEL4    ),                                GNP00770
      UQSSA5 CHAR(9) INIT( LEVEL5    ),                                GNP00780
      UQSSA6 CHAR(9) INIT( LEVEL6    ),                                GNP00790
      SSAKEY0 CHAR(5)  DEF QSSA0  POS(20)                              GNP00800
      SSAKEY1 CHAR(9) DEF QSSA1  POS(20),                              GNP00810
      SSAKEY2 CHAR(9) DEF QSSA2  POS(20),                              GNP00820
      SSAKEY3 CHAR(9) DEF QSSA3  POS(20),                              GNP00830
      SSAKEY4 CHAR(9) DEF QSSA4  POS(20),                              GNP00840
      SSAKEY5 CHAR(9)  DEF QSSA5 POS(20) ,                             GNP00850
      SSAKEY6 CHAR(9)  DEF QSSA6 POS(20)                               GNP00860
   DCL  ATTSSA  CHAR(9) INIT( ATTRIT    ) STATIC                      GNP00870
      ATTLEV  PIC 9  DEF ATTSSA POS(7)                                 GNP00880
   DCL  MQYSSA  CHAR(9),                                               GNP00890
      MQYLEV PIC 9   DEF MQYSSA POS(2)                                 GNP00900
   DCL THREE FIXED BIN(31,0) INIT(3),                                  GNP00910
      FOUR  FIXED BIN(31,0) INIT(4),                                   GNP00920
      FIVE  FIXED BIN(31,0) INIT(5),                                   GNP00930
      SIX   FIXED BIN(31,0) INIT(6),                                   GNP00940
      SEVEN FIXED BIN(31,0) INIT(7),                                   GNP00950
      EIGHT FIXED BIN(31,0) INIT(8),                                   GNP00960
      NINE  FIXED BIN(31,0) INIT(9),                                   GNP00970
      TEN   FIXED BIN(31,0) INIT(10),                                  GNP00980
      P#    FIXED BIN(31,0)                                            GNP00990
   DCL GU  CHAR(4)  INIT( GU   ),                                      GNP01000
      GNP CHAR(4)  INIT( GNP  ),                                       GNP01010
      GN  CHAR(4)  INIT( GN   )                                        GNP01020
    /*******************************************/                      DB000980
    DCL   YEAR PIC 99 ,
          MONTH PIC 99 ,
          FLAG PIC 9 ,
          HEADED CHAR(30),
          VALUE PIC 9 ,
          LEVELIN PIC 9 ,
```

```
            ORIGN CHAR(5),
            ADJUST CHAR(5),
            CHOOSE PIC 9
         OPEN FILE(SYSPRINT) LINESIZE(132)
         ON ENDFILE(SYSIN) GO TO LAST
   JUMPING
      GET SKIP EDIT(YEAR,MONTH,FLAG,LEVELIN,ORIGN,ADJUST,CHOOSE,HEADED)
       (X(2),2 F(2),F(1),F(1),A(5),A(5),F(1),X(4),A(30))
      CALL RATIO
      GOTO JUMPING
     RATIO  PROC
  /****** THIS PROGRAM WAS DEVELOPED FOR COMPUTING THE RATE OF *******/ INC00080
  /****** VARIBLE INDEXES IN ORDER TO ANALIZE                 *******/ INC00090
           CALL PYOPUT                                                 INC00180
           PUT EDIT( JOB NORMAL END )(SKIP(2),X(5),A)                  INC00190
     PYOPUT  PROC                                                      INC00200
        SELECT(CHOOSE)
         WHEN(0)
         DO  CALL FIRST  CALL SECOND   CALL JISUALL  END
         WHEN(1)                                                       INC00210
         DO  CALL FIRST  END                                           INC00220
         WHEN(2)                                                       INC00230
         DO  CALL SECOND  END
         WHEN(3)                                                       INC00240
         DO  CALL FIRST  CALL SECOND  END
         OTHERWISE                                                     INC00240
         DO  CALL JISUALL  END
       END  /******* SELECT END *******/
     JISUALL  PROC                                                     INC00250
      /***** TABLE BY MONTHLY PROC *****/                               INC00260
        DCL I      FLOAT,                                              INC00270
              J      FLOAT,                                            INC00280
            CODE   CHAR(8),                                            INC00290
            ODATA(360)   FLOAT,                                        INC00300
            ST     FLOAT                                               INC00310
       IF ORIGN  = DONOT  THEN CALL STEP1                              INC00320
       IF ADJUST = DONOT  THEN CALL STEP2                              INC00330
       STEP1  PROC                                                     INC00340
           KEY=ALLZERO                                                 INC00350
           LOWLEV=LEVELIN                                              INC00360
           PLEVEL=0                                                    INC00370
           DBNAME=ORIGN                                                INC00380
           DBKEY=ALLZERO                                               INC00390
           FREQ= M                                                     INC00400
           PUT PAGE                                                    INC00410
           PUT LINE(30)                                                INC00420
           PUT SKIP EDIT( ***** ORIGINAL INDEX  TABLE ALL LIST         INC00430
                      ***** )(X(5),A)                                  INC00440
           CALL HEAD                                                   INC00450
           CALL LOOPED                                                 INC00460
       END  /****** STEP1 END *********/                               INC00470
       STEP2  PROC                                                     INC00480
           KEY=ALLZERO                                                 INC00490
           LOWLEV=LEVELIN                                              INC00500
           PLEVEL=0                                                    INC00510
```

```
          DBNAME=ADJUST                                                 INC00520
          DBKEY=ALLZERO                                                 INC00530
          FREQ= M                                                       INC00540
          PUT PAGE                                                      INC00550
          PUT LINE(30)                                                  INC00560
          PUT SKIP EDIT( ***** SEASONAL ADJUSTED INDEX  TABLE ALL       INC00570
           LIST ***** )(X(5),A)
          CALL HEAD                                                     INC00580
          CALL LOOPED                                                   INC00590
      END  /****** STEP2 END ********/                                   INC00600
    LOOPED  PROC                                                        INC00610
          /* READ  DB  AGAIN */                                         INC00620
  LOOP                                                                  INC00630
          CALL DBGET                                                    INC00640
          IF PLEVEL 0 THEN                                              INC00650
          DO  CALL HITTING                                              INC00660
              GOTO LOOP
          END                                                           INC00670
    BENCH                                                               INC00680
          END    /***** LOOPED PROC END *****/                          INC00690
  HITTING  PROC                                                         INC00700
          /* DB READ AND WRITE ON THE PLACE */                          INC00710
          CODE=SUBSTR(KEY,1,8)                                          INC00720
          IF ONCODE= 51 THEN                                            INC00730
            DO                                                          INC00740
               IF CODE =ALLZERO                                         INC00750
                  THEN PUT EDIT(CODE, NAME NOT FOUND )                  INC00760
                       (SKIP,A(10),A)                                   INC00770
            END                                                         INC00780
          IF ONCODE= 52 THEN PUT EDIT(CODE, DUPLICATE NAME )            INC00790
                       (SKIP,A(10),A)                                   INC00800
          I=(YEAR-INITY)*12+(MONTH-INITM)+1                             INC00810
          ODATA=0                                                       INC00820
          DO K=1 TO I                                                   INC00830
          ODATA(K)=DATA(K)/10**DECP                                     INC00840
          ODATA(K)=ODATA(K) /10
          END                                                           INC00850
          KK=K  CALL PRINTER                                            INC00860
  PRINTER  PROC                                                         INC00870
       PUT SKIP EDIT(DBNAME,CODE,YEAR,MONTH)(X(5),A(6),A(6),2 F(2))     INC00880
       PUT SKIP EDIT((ODATA(J) DO J=1 TO KK))                           INC00890
                       (SKIP,X(5),12 F(10,1))                           INC00900
      END  /****** PRINTER END ********/                                INC00910
    OUT                                                                 INC00920
          END  /* HITTING PROG END */                                  INC00930
    HEAD  PROC                                                          INC00940
           /* HEADS ARE PRINTING */                                     INC00950
          DCL  PYEAR   PIC 99 ,                                         INC00960
               PMONTH  PIC 99                                           INC00970
          PUT PAGE                                                      INC00980
          PUT LINE(3)                                                   INC00990
          IF DBNAME=ORIGN THEN                                          INC01000
           PUT SKIP EDIT(HEADED, * INDEX TABLE  BY MONTHLY****** )      INC01010
                  (X(30),A(30),A)                                       INC01020
          ELSE                                                          INC01030
```

```
          PUT SKIP EDIT(HEADED, * SEASONAL ADJUSTED INDEX TABLE BY     INC01040
                  MONTHLY *** )(X(30),A(30),A)                          INC01050
         PUT SKIP EDIT(YEAR, . ,MONTH)(X(100),F(2),A(1),F(2))          INC01060
         PUT SKIP EDIT( JAN , FEB , MAR , APR , MAY , JUN , JUL ,      INC01070
              AUG , SEP , OCT , NOV , DEC )(X(5),12 (X(7),A(3)))       INC01080
         END  /* HEAD PROC END */                                      INC01090
     END  /*** JISU-ALL   MONTHLY INDEX ***/                           INC01100
  FIRST  PROC                                                          INC01110
     /***** TABLE BY MONTHLY PROC *****/                                INC01120
       DCL I      FLOAT,                                               INC01130
              J     FLOAT,                                              INC01140
          CODE   CHAR(8),                                              INC01150
          ODATA(8)  FLOAT,                                             INC01160
          ST     FLOAT,                                                INC01170
          HYON   FLOAT,                                                INC01180
          JEON  FLOAT                                                  INC01190
       IF ORIGN  = DONOT  THEN CALL STEP3
       IF ADJUST = DONOT  THEN CALL STEP4
     STEP3  PROC
         KEY=ALLZERO                                                   INC01200
         LOWLEV=LEVELIN                                                INC01210
         PLEVEL=0                                                      INC01220
         DBNAME=ORIGN                                                  INC01230
         DBKEY=ALLZERO                                                 INC01240
         FREQ= M                                                       INC01250
         PUT PAGE                                                      INC01260
         PUT LINE(30)                                                  INC01270
         PUT SKIP EDIT( ***** ORIGINAL INDEX *******  ,HEADED)         INC01280
                      (X(5),2 A(30))                                   INC01290
         CALL HEAD                                                     INC01300
         CALL LOOPED                                                   INC01310
       END  /******* STEP3 END ********/
     STEP4  PROC
       KEY=ALLZERO                                                     INC01320
       LOWLEV=LEVELIN                                                  INC01330
       PLEVEL=0                                                        INC01340
       DBNAME=ADJUST                                                   INC01350
       DBKEY=ALLZERO                                                   INC01360
       FREQ= M                                                         INC01370
       PUT PAGE                                                        INC01380
       PUT LINE(30)                                                    INC01390
       PUT SKIP EDIT( ** SEASONAL ADJUSTED INDEX ** ,HEADED)           INC01400
                                  (X(5),2 A(30))
       CALL HEAD                                                       INC01410
       CALL LOOPED                                                     INC01420
       END  /******* STEP4 END ********/
   LOOPED  PROC                                                        INC01430
         /* READ  DB  AGAIN */                                         INC01440
  LOOP                                                                 INC01450
         CALL DBGET                                                    INC01460
         IF PLEVEL 0 THEN                                              INC01470
         DO  CALL HITTING                                              INC01480
             GOTO LOOP
         END                                                           INC01490
   BENCH                                                               INC01500
```

```
        END      /****** LOOPED PROC END *******/                    INC01510
  HITTING  PROC                                                      INC01520
        /* DB READ AND WRITE ON THE PLACE */                          INC01530
     DCL INO FIXED
        CODE=SUBSTR(KEY,1,8)                                         INC01540
        IF ONCODE= 51 THEN                                           INC01550
          DO                                                         INC01560
             IF CODE =ALLZERO                                        INC01570
                THEN PUT EDIT(CODE, NAME NOT FOUND )                 INC01580
                     (SKIP,A(10),A)                                  INC01590
          END                                                        INC01600
        IF ONCODE= 52 THEN PUT EDIT(CODE, DUPLICATE NAME )           INC01610
                     (SKIP,A(10),A)                                  INC01620
        I=(YEAR-INITY)*12+(MONTH-INITM)+1                            INC01630
        ODATA=0                                                      INC01640
        ODATA(1)=DATA(I-36)/10**DECP                                 INC01650
        ODATA(2)=DATA(I-24)/10**DECP                                 INC01660
        ODATA(3)=DATA(I-12)/10**DECP                                 INC01670
        ODATA(4)=DATA(I-1)/10**DECP                                  INC01680
        ODATA(5)=DATA(I )/10**DECP                                   INC01690
        DO INO=1 TO 5
        ODATA(INO)=ODATA(INO) / 10
        END
        IF ODATA(4)=0     ODATA(5)=0                                 INC01700
        THEN ODATA(6)=0                                              INC01710
        ELSE ODATA(6)=(ODATA(5)-ODATA(4))*100/ODATA(4)               INC01720
        IF ODATA(5)=0    ODATA(3)=0                                  INC01730
        THEN ODATA(7)=0                                              INC01740
        ELSE ODATA(7)=(ODATA(5)-ODATA(3))*100/ODATA(3)               INC01750
        ST=I-MONTH+1                                                 INC01760
        HYON=0                                                       INC01770
        JEON=0                                                       INC01780
        DO J=ST TO I                                                 INC01790
        HYON=HYON+(DATA(J)/10**DECP)                                 INC01800
        JEON=JEON+(DATA(J-12)/10**DECP)                              INC01810
        END                                                          INC01820
        IF JEON=0   HYON=0 THEN ODATA(8)=0                           INC01830
        ELSE ODATA(8)=(HYON-JEON)*100/JEON                           INC01840
  PRINTER                                                            INC01850
        PUT SKIP(2) EDIT(CODE,(ODATA(I) DO I=1 TO 8))                INC01860
                    (X(5),A(5),X(6),8 F(10,1))                       INC01870
    OUT                                                              INC01880
        END  /* HITTING PROG END */                                  INC01890
    HEAD  PROC                                                       INC01900
          /* HEADS ARE PRINTING */                                   INC01910
        DCL  PYEAR(5)    PIC 99 ,                                    INC01920
             PMONTH(3)   PIC 99                                      INC01930
        PYEAR(1)=YEAR-3                                              INC01940
        PYEAR(2)=YEAR-2                                              INC01950
        PYEAR(3)=YEAR-1                                              INC01960
        IF MONTH=1 THEN                                              INC01970
        DO  PYEAR(4)=YEAR-1                                          INC01980
            PMONTH(2)=12                                             INC01990
        END                                                          INC02000
        ELSE DO                                                      INC02010
```

```
            PYEAR(4)=YEAR                                                INC02020
            PMONTH(2)=MONTH-1                                            INC02030
        END                                                              INC02040
        PYEAR(5)=YEAR                                                    INC02050
        PMONTH(1)=MONTH                                                  INC02060
        PMONTH(3)=MONTH                                                  INC02070
        PUT PAGE                                                         INC02080
        PUT LINE(3)                                                      INC02090
        IF DBNAME=ORIGN THEN                                             INC02100
         PUT SKIP EDIT(HEADED, * ORIGINAL INDEX TABLE BY MONTHLY )       INC02110
                          (X(30),A(30),A)                                INC02120
        ELSE                                                             INC02130
         PUT SKIP EDIT(HEADED, * SEASONAL ADJUSTED INDEX TABLE BY        INC02140
                 MONTHLY *** )(X(30),A(30),A)                            INC02150
        PUT SKIP EDIT(YEAR, . ,MONTH)(X(100),F(2),A(1),F(2))             INC02160
        PUT SKIP EDIT( DIVISON ,PYEAR(1), - ,PMONTH(1),PYEAR(2), - ,     INC02170
            PMONTH(1),PYEAR(3), - ,PMONTH(1),PYEAR(4), . ,PMONTH(2),     INC02180
            PYEAR(5), . ,PMONTH(3),   VJW , VJNDW , VJNDG ,              INC02190
             ********** NOTE ********** )(X(5),A(11),5 (X(5),F(2),      INC02200
            A(1),F(2)),3 (X(5),A(5)),X(5),A(26))                         INC02210
        END  /* HEAD PROC END */                                         INC02220
    END  /*** FIRST PROC   MONTHLY INDEX ***/                            INC02230
  SECOND  PROC                                                           INC02240
     /***** OPERATION RATE INDEX LIST *****/                              INC02250
      DCL I       FLOAT,                                                 INC02260
          CODE    CHAR(8),                                               INC02270
          OWGT    FLOAT,                                                 INC02280
            SORTS   CHAR(5),                                             INC02290
        (ST,LT,J,K,G,DE,L)    FLOAT                                      INC02300
      DCL ODATA(13)  FLOAT,                                              INC02310
          TOTAL     FLOAT,                                               INC02320
          HYON(12)  FLOAT,                                               INC02330
          JEON(12)  FLOAT,                                               INC02340
          QUATER(4)   FLOAT,                                             INC02350
          QUATERB(4)  FLOAT,                                             INC02360
          QUATERK(4)  FLOAT,                                             INC02370
          AVE         FLOAT,                                             INC02380
          TOTALB      FLOAT,                                             INC02390
        TOTALK        FLOAT,                                             INC02400
          ORDER       FLOAT                                              INC02410
       DCL  ROLL FLOAT INIT(0),                                          INC02420
            CYCLE FLOAT INIT(1),                                         INC02430
            R       FLOAT INIT(1)                                        INC02440
        IF ORIGN  = DONOT  THEN CALL STEP5                               INC02490
        IF ADJUST = DONOT  THEN CALL STEP6                               INC02500
      STEP5  PROC                                                        INC02510
        ROLL=0
        PUT PAGE                                                         INC02450
        PUT LINE(30)                                                     INC02460
        PUT SKIP EDIT( ***** ORIGINAL INDEX LIST BY MONTHLY ***** )      INC02470
                      (X(5),A)                                           INC02480
        KEY=ALLZERO                                                      INC02520
        LOWLEV=LEVELIN                                                   INC02530
        PLEVEL=0                                                         INC02540
        DBNAME=ORIGN                                                     INC02550
```

```
          DBKEY=ALLZERO                                                 INC02560
          FREQ= M                                                       INC02570
          CALL HEAD                                                     INC02580
     LOOP                                                               INC02590
          CALL DBGET                                                    INC02600
          IF PLEVEL 0 THEN  DO                                          INC02610
             CALL INNING1                                               INC02620
             CALL INNING2                                               INC02630
             CALL INNING3                                               INC02640
             CALL INNING4                                               INC02650
          ROLL=ROLL+1                                                   INC02660
          IF ROLL=5 THEN                                                INC02670
          DO                                                            INC02680
               CALL HEAD                                                INC02690
               ROLL=0                                                   INC02700
           END                                                          INC02710
          IF CYCLE=(4*R)+1 THEN                                         INC02720
            DO  R=R+1                                                   INC02730
            END                                                         INC02740
          CYCLE=CYCLE+1                                                 INC02750
            GOTO LOOP
          END       /******** PLEVEL END *******/                       INC02760
   NEXTA                                                                INC02770
          END  /******** STEP5 END*******/                              INC02780
     STEP6  PROC                                                        INC02790
          ROLL=0
          PUT PAGE                                                      INC02800
          PUT LINE(30)                                                  INC02810
          PUT SKIP EDIT( **** SEASONAL INDEX LIST BY MONTHLY ***** )    INC02820
                        (X(5),A)                                        INC02830
          R=1                                                           INC02840
        KEY=ALLZERO                                                     INC02850
          LOWLEV=LEVELIN                                                INC02860
            PLEVEL=0                                                    INC02870
          DBNAME=ADJUST                                                 INC02880
          DBKEY=ALLZERO                                                 INC02890
          FREQ= M                                                       INC02900
          CALL HEAD                                                     INC02910
     LOOP2                                                              INC02920
          CALL DBGET                                                    INC02940
          IF PLEVEL 0 THEN  DO                                          INC02950
             CALL INNING1                                               INC02960
             CALL INNING2                                               INC02970
             CALL INNING3                                               INC02980
             CALL INNING4                                               INC02990
          ROLL=ROLL+1                                                   INC03000
          IF ROLL=5 THEN                                                INC03010
          DO                                                            INC03020
               CALL HEAD                                                INC03030
               ROLL=0                                                   INC03040
           END                                                          INC03050
          IF CYCLE=(4*R)+1 THEN                                         INC03060
            DO  R=R+1                                                   INC03070
            END                                                         INC03080
          CYCLE=CYCLE+1                                                 INC03090
```

```
          GOTO LOOP2
        END      /******* PLEVEL END ********/                          INC03100
     NEXTB  END  /******* STEP6 END ********/                           INC03110
    INNING1  PROC                                                       INC03120
        /* MONTHLY INDEX LIST */                                        INC03130
        ORDER=1                                                         INC03140
           CODE=SUBSTR(KEY,1,8)                                         INC03150
        IF ONCODE= 51 THEN                                              INC03160
          DO                                                            INC03170
             IF CODE =ALLZERO                                           INC03180
                THEN PUT EDIT(CODE, NAME NOT FOUND )                    INC03190
                     (SKIP,A(10),A)                                     INC03200
           END                                                          INC03210
        IF ONCODE= 52 THEN PUT EDIT(CODE, DUPLICATE NAME )              INC03220
                     (SKIP,A(10),A)                                     INC03230
           OWGT=WGT                                                     INC03240
           I=(YEAR-INITY)*12+(MONTH-INITM)+1                            INC03250
        ST=I-MONTH+1                                                    INC03260
        LT=I-MONTH                                                      INC03270
        ODATA=0                                                         INC03280
        TOTAL=0                                                         INC03290
        DO J=ST TO I                                                    INC03300
        ODATA(J-LT)=DATA(J)/10**DECP                                    INC03310
        ODATA(J-LT)=ODATA(J-LT) / 10
        TOTAL=TOTAL+ODATA(J-LT)                                         INC03320
        END                                                             INC03330
        ODATA(13)=TOTAL/MONTH                                           INC03340
        CALL PRTA                                                       INC03350
    OUT                                                                 INC03360
    END  /* FIRST INNING */                                             INC03370
    INNING2  PROC                                                       INC03380
        /* CHANGE FROM PREVIOUS MONTH */                                INC03390
        ORDER=2                                                         INC03400
           CODE=KEY                                                     INC03410
        IF ONCODE= 51 THEN                                              INC03420
          DO                                                            INC03430
             IF CODE =ALLZERO                                           INC03440
                THEN PUT EDIT(CODE, NAME NOT FOUND )                    INC03450
                     (SKIP,A(10),A)                                     INC03460
           END                                                          INC03470
        IF ONCODE= 52 THEN PUT EDIT(CODE, DUPLICATE NAME )              INC03480
                     (SKIP,A(10),A)                                     INC03490
           I=(YEAR-INITY)*12+(MONTH-INITM)+1                            INC03500
           OWGT=WGT                                                     INC03510
        ST=I-MONTH+1                                                    INC03520
        LT=I-MONTH                                                      INC03530
        ODATA=0                                                         INC03540
        TOTAL=0                                                         INC03550
        DO J=ST TO I                                                    INC03560
        IF DATA(J-1)=0   DATA(J)=0                                      INC03570
        THEN ODATA=0                                                    INC03580
        ELSE DO                                                         INC03590
      ODATA(J-LT)=((DATA(J)/10**DECP)-(DATA(J-1)/10**DECP))*100         INC03600
                     /(DATA(J-1)/10**DECP)                              INC03610
       END
```

```
            TOTAL=TOTAL+ODATA(J-LT)                                    INC03620
         END                                                           INC03630
         IF TOTAL=0 THEN ODATA(13)=0                                   INC03640
          ELSE ODATA(13)=TOTAL/MONTH                                   INC03650
         CALL PRTA                                                     INC03660
      OUT                                                              INC03670
         END  /* SECOND INNING END */                                  INC03680
      INNING3  PROC                                                    INC03690
         /* CHANGE FROM SAME MONTH OF PREVIOUS YEAR */                 INC03700
         ORDER=3                                                       INC03710
            CODE=KEY                                                   INC03720
         IF ONCODE= 51 THEN                                            INC03730
           DO                                                          INC03740
              IF CODE =ALLZERO                                         INC03750
                 THEN PUT EDIT(CODE, NAME NOT FOUND )                  INC03760
                      (SKIP,A(10),A)                                   INC03770
            END                                                        INC03780
         IF ONCODE= 52 THEN PUT EDIT(CODE, DUPLICATE NAME )            INC03790
                      (SKIP,A(10),A)                                   INC03800
            OWGT=WGT                                                   INC03810
            I=(YEAR-INITY)*12+(MONTH-INITM)+1                          INC03820
         ST=I-MONTH+1                                                  INC03830
         LT=I-MONTH                                                    INC03840
         ODATA=0                                                       INC03850
         TOTAL=0                                                       INC03860
         DO J=ST TO I                                                  INC03870
         IF DATA(J-12)=0  DATA(J)=0                                    INC03880
         THEN ODATA=0                                                  INC03890
         ELSE DO                                                       INC03900
       ODATA(J-LT)=((DATA(J)/10**DECP)-(DATA(J-12)/10**DECP))*100      INC03910
                    /(DATA(J-12)/10**DECP)                             INC03920
        END
            TOTAL=TOTAL+ODATA(J-LT)                                    INC03930
         END                                                           INC03940
         IF TOTAL=0 THEN ODATA(13)=0                                   INC03950
         ELSE  ODATA(13)=TOTAL/MONTH                                   INC03960
         CALL PRTA                                                     INC03970
      OUT                                                              INC03980
         END  /* THIRD INNING END */                                   INC03990
      INNING4  PROC                                                    INC04000
         /* CHANGE FROM SAME PERIOD OF PREVIOUS YEAR */                INC04010
         ORDER=4                                                       INC04020
            CODE=KEY                                                   INC04030
         IF ONCODE= 51 THEN                                            INC04040
           DO                                                          INC04050
              IF CODE =ALLZERO                                         INC04060
                 THEN PUT EDIT(CODE, NAME NOT FOUND )                  INC04070
                      (SKIP,A(10),A)                                   INC04080
            END                                                        INC04090
         IF ONCODE= 52 THEN PUT EDIT(CODE, DUPLICATE NAME )            INC04100
                      (SKIP,A(10),A)                                   INC04110
            I=(YEAR-INITY)*12+(MONTH-INITM)+1                          INC04120
            OWGT=WGT                                                   INC04130
         ST=I-MONTH+1                                                  INC04140
         HYON=0                                                        INC04150
```

```
        JEON=0                                                          INC04160
        ODATA=0                                                         INC04170
        TOTAL=0                                                         INC04180
        HYON(1)=DATA(ST)/10**DECP                                       INC04190
        JEON(1)=DATA(ST-12)/10**DECP                                    INC04200
        LT=0                                                            INC04210
        DO J=2 TO MONTH                                                 INC04220
        LT=LT+1                                                         INC04230
        HYON(J)=HYON(J-1)+DATA(ST+LT)/10**DECP                          INC04240
        JEON(J)=JEON(J-1)+DATA(ST-12+LT)/10**DECP                       INC04250
        END                                                             INC04260
        DO J=1 TO MONTH                                                 INC04270
        IF HYON(J)=0   JEON(J)=0                                        INC04280
        THEN ODATA(J)=0                                                 INC04290
        ELSE DO
        ODATA(J)=(HYON(J)-JEON(J))*100/JEON(J)                          INC04300
        END
        TOTAL=TOTAL+ODATA(J)                                            INC04310
        END                                                             INC04320
        IF TOTAL=0 THEN ODATA(13)=0                                     INC04330
        ELSE ODATA(13)=TOTAL/MONTH                                      INC04340
        CALL PRTA                                                       INC04350
    OUT                                                                 INC04360
        END  /* FOURTH INNING END */                                    INC04370
    PRTA  PROC                                                          INC04380
        IF ORDER=1 THEN SORTS= INDEX                                    INC04390
        ELSE IF ORDER=2 THEN SORTS=   VJW                               INC04400
        ELSE IF ORDER=3 THEN SORTS= VJNDW                               INC04410
        ELSE IF ORDER=4 THEN SORTS= VJNDG                               INC04420
        IF CYCLE=(4*R)+1                                                INC04430
         THEN DO                                                        INC04440
                IF ORDER=1 THEN                                         INC04450
                PUT SKIP(3) EDIT(CODE,DBNAME,OWGT,SORTS,                INC04460
                (ODATA(J) DO J=1 TO 13))(X(3),A(5),X(3),A(5),X(2),      INC04470
                 F(8,1),X(2),A(8),13 F(7,1))                            INC04480
                ELSE  PUT SKIP(2) EDIT(SORTS,(ODATA(J)                  INC04490
                  DO J=1 TO 13))(X(28),A(8),13 F(7,1))                  INC04500
             END                                                        INC04510
         ELSE DO                                                        INC04520
                IF ORDER=1 THEN                                         INC04530
                PUT SKIP(2) EDIT(CODE,DBNAME,OWGT,SORTS,                INC04540
                (ODATA(J) DO J=1 TO 13))(X(3),A(5),X(3),A(5),X(2),      INC04550
                F(8,1),X(2),A(8),13 F(7,1))                             INC04560
                ELSE PUT SKIP(2) EDIT(SORTS,(ODATA(J)                   INC04570
                 DO J=1 TO 13))(X(28),A(8),13 F(7,1))                   INC04580
             END                                                        INC04590
        END  /* PRINTER END */                                          INC04600
    HEAD  PROC                                                          INC04610
        /* TITLE AND HEAD ARE SELECTED BY ORDER */                      INC04620
        PUT PAGE                                                        INC04630
        PUT LINE(3)                                                     INC04640
        PUT SKIP EDIT( ***** YEARLY,MONTHLY INDEX   INCRE DECRE( )      INC04650
              TABLE ***** )(X(30),A)                                    INC04660
        PUT SKIP EDIT(YEAR, - ,MONTH)(X(100),F(2),A(1),F(2))            INC04670
  PUT SKIP EDIT( CODE , DNAME , WEIGHT , SORTS , JAN , FEB , MAR ,      INC04680
```

```
                APR , MAY , JUN , JUL , AUG , SEP , OCT , NOV , DEC ,    INC04690
     AVE )(X(3),A(8),A(5),X(4),A(6),X(2),A(8),13 (X(4),A(3)))            INC04700
          END  /* HEAD PROC END */                                       INC04710
    BENCH                                                                INC04720
     END  /* SECOND PROC END ****** */                                   INC04730
     END  /* PYOOUT PROC END *****/                                      INC04740
    INCLUDE DBGETLEE                                                     INC04750
   END  /****** ALLRATIO PROC END ******/
  LAST
   /******** ALL PROC END **********/
    END   /***** THE END OF THE DANCING WITH MISS JOE UEON OAK*****/
/*
```

프로그램 : COMINDEX

기능 : X-11 작업준비 과정에 대비 즉,

X-Ⅱ을 이용키 위한 Inputformat

으로 DB자료 DISK에 배열한다.

```
//ZDBCOM71 JOB CLASS=V,PRTY=13                                          COM00010
//         EXEC IMSPLI,MBR=DCOMIDEX                                     COM00020
//SYSIN    DD *                                                         COM00030
* PROCESS GS,NEST,INCLUDE                                               COM00040
    COMPIN  PROC(PCBP) OPTIONS(MAIN)                                    COM00050
      /***DBDCL***************************/                             COM00060
      DCL PCBP POINTER                                                  COM00070
      DCL 1 PCBNAME BASED(PCBP),                                        COM00080
            2 NAME     CHAR(8),                                        COM00090
            2 SEGLEV   CHAR(2),                                        COM00100
            2 STATUS   CHAR(2),                                        COM00110
           2 PROCOPT CHAR(4),                                          COM00120
            2 FILLER   FIXED BIN(31,0),                                COM00130
           2 SEGNAME CHAR(8),                                          COM00140
           2 LEN KFB FIXED BIN(31,0),                                  COM00150
           2 NOSENSG FIXED BIN(31,0),                                  COM00160
            2 KEYFB    CHAR(45)                                        COM00170
      DCL 1 DIREC,                                                      COM00180
            2 DBNAME CHAR(5),                                          COM00190
            2 DBKEY  CHAR(9),                                          COM00200
            2  FROMY PIC 99 ,                                          COM00210
           2  FROMM PIC 99 ,                                           COM00220
           2  TOY   PIC 99 ,                                           COM00230
           2  TOM  PIC 99 ,                                            COM00240
           2 UPLEV  PIC 9 ,                                            COM00250
           2 LOWLEV PIC 9  INIT(9),                                    COM00260
            2 FREQ    CHAR(1),                                         COM00270
            2 DUMMY   CHAR(1)                                          COM00280
      DCL 1 ATTO,                                                       COM00290
            2 AOURCE    CHAR(10),                                      COM00300
            2 NOITEM    PIC 9999 ,                                     COM00310
            2 TOTLEV    PIC 99 ,                                       COM00320
            2 BASEY     PIC 99 ,                                       COM00330
            2 BASEM     PIC 99 ,                                       COM00340
            2 TYPE      CHAR(1),                                       COM00350
            2 UNITO     CHAR(11),                                      COM00360
            2 KEYSZ(6) PIC 9 ,                                         COM00370
           2 WGTP      PIC 9 ,                                         COM00380
            2 DECP      PIC 9                                          COM00390
      DCL PLITDLI ENTRY,                                                COM00400
          GU  CHAR(4) INIT( GU   ),                                    COM00410
          GN  CHAR(4) INIT( GN   ),                                    COM00420
          GNP CHAR(4) INIT( GNP  ),                                    COM00430
          FOUR FIXED BIN(31,0) INIT(4),                                COM00440
          FIVE FIXED BIN(31,0) INIT(5),                                COM00450
          SIX  FIXED BIN(31,0) INIT(6),                                COM00460
          SEVEN FIXED BIN(31,0) INIT(7),                               COM00470
          EIGHT FIXED BIN(31,0) INIT(8),                               COM00480
          NINE  FIXED BIN(31,0) INIT(9),                               COM00490
          P#    FIXED BIN(31,0)                                        COM00500
      DCL 1 ATTIO,                                                      COM00510
            2 WGT FLOAT,                                               COM00520
            2 UNIT CHAR(11),                                           COM00530
          1 LEVELIO,                                                   COM00540
            2 KEY   CHAR(9),                                           COM00550
```

```
        2 TITLE CHAR(30),                                          COM00560
      1 LEVEL0,                                                    COM00570
        2 KEY0   CHAR(5),                                          COM00580
        2 TITLE0 CHAR(30),                                         COM00590
      1 MQYID ALIGNED,                                             COM00600
        2 INITY PIC 99 ,                                           COM00610
       2 INITM PIC 99 ,                                            COM00620
       2 ENDY  PIC 99 ,                                            COM00630
        2 ENDM  PIC 99 ,                                           COM00640
        2 DATA(360)  FIXED BIN(31,0),                              COM00650
        2  DUM10     CHAR(100),                                    COM00660
     SPECID CHAR(30)                                               COM00670
   DCL QSSA0 CHAR(25) INIT( TOTAL   (KEY0     =00000) ),           COM00680
     QSSA1 CHAR(29) INIT( LEVEL1  (KEY1     =000000000) ),          COM00690
     QSSA2 CHAR(29) INIT( LEVEL2  (KEY2     =000000000) ),          COM00700
     QSSA3 CHAR(29) INIT( LEVEL3  (KEY3     =000000000) ),          COM00710
     QSSA4 CHAR(29) INIT( LEVEL4  (KEY4     =000000000) ),          COM00720
     QSSA5 CHAR(29) INIT( LEVEL5  (KEY5     =000000000) ),          COM00730
     QSSA6 CHAR(29) INIT( LEVEL6  (KEY6     =000000000) ),          COM00740
     UQSSA1 CHAR(9) INIT( LEVEL1 ),                                COM00750
     UQSSA2 CHAR(9) INIT( LEVEL2 ),                                COM00760
     UQSSA3 CHAR(9) INIT( LEVEL3 ),                                COM00770
     UQSSA4 CHAR(9) INIT( LEVEL4 ),                                COM00780
     UQSSA5 CHAR(9) INIT( LEVEL5 ),                                COM00790
     UQSSA6 CHAR(9) INIT( LEVEL6 ),                                COM00800
     SSAKEY0 CHAR(5) DEF QSSA0 POS(20),                            COM00810
     SSAKEY1 CHAR(9) DEF QSSA1 POS(20),                            COM00820
     SSAKEY2 CHAR(9) DEF QSSA2 POS(20),                            COM00830
     SSAKEY3 CHAR(9) DEF QSSA3 POS(20),                            COM00840
     SSAKEY4 CHAR(9) DEF QSSA4 POS(20),                            COM00850
     SSAKEY5 CHAR(9) DEF QSSA5 POS(20),                            COM00860
     SSAKEY6 CHAR(9) DEF QSSA6 POS(20),                            COM00870
     ATTSSA CHAR(9) INIT( ATTRIT ),                                COM00880
     ATTLEV PIC 9  DEF ATTSSA POS(7),                              COM00890
     SPECSSA CHAR(9) INIT( SPEC ),                                 COM00900
     SPECLEV PIC 9  DEF SPECSSA POS(5),                            COM00910
     MQYSSA CHAR(9),                                               COM00920
     MQYLEV PIC 9  DEF MQYSSA POS(2)                               COM00930
    DCL  TESTSTR CHAR(9),                                          COM00940
     ALLZERO CHAR(9) INIT((9) 0 ),                                 COM00950
     SUMKEYSZ  PIC 99 ,                                            COM00960
     PLEVEL  PIC 9  INIT(0)                                        COM00970
    /*******************************************/                  COM00980
   DCL REGION FILE RECORD OUTPUT SEQUENTIAL                        COM00990
   DCL CARDP CHAR(260)                                             COM01000
   DCL CARDIN FILE RECORD INPUT                                    COM01010
   DCL CARDC CHAR(80)                                              COM01020
   DCL DATA1(360) FLOAT, YRNO PIC 99 ,                             COM01030
       CARD CHAR(80),                                              COM01040
       CDWGT PIC 99999V99  DEF CARD POS(74),                       COM01050
       DATA2(360) FLOAT, DATA3(360) FLOAT                          COM01060
   DCL BLANK CHAR(3) INIT(    )                                    COM01070
    /**********************/                                        COM01080
    /*COMP CONTROL DESIGN */                                        COM01090
    /**********************/                                        COM01100
```

```
DCL 1 CARD1 DEF CARDC,                                          COM01110
      2 CCOMP  CHAR(6),                                         COM01120
      2 CKNO   CHAR(2),                                         COM01130
      2 CKEYR  CHAR(2),                                         COM01140
      2 CWGT   CHAR(4),                                         COM01150
      2 CCONST CHAR(5),                                         COM01160
      2 CJFM   CHAR(2),                                         COM01170
      2 CJFY   CHAR(2),                                         COM01180
      2 CJEM   CHAR(2),                                         COM01190
      2 CJEY   CHAR(2),                                         COM01200
      2 CGNPGR CHAR(10),                                        COM01210
      2 DUMMY CHAR,                                             COM01220
      2 CGNPMQ CHAR(1),                                         COM01230
      2 CINV   CHAR,                                            COM01240
      2 CIPR   CHAR,                                            COM01250
      2 CNCHRT CHAR,                                            COM01260
      2 CIRD   CHAR,                                            COM01270
      2 CNDTR  CHAR,                                            COM01280
      2 CLOG   CHAR,                                            COM01290
      2 CMCD   CHAR,                                            COM01300
      2 CAMUL  CHAR(4),                                         COM01310
      2 CITERM CHAR(2),                                         COM01320
      2 CNDIFF CHAR(2)                                          COM01330
 /*       *****                    */                          COM01340
 /* ARIMA MAIN CONTROL DESIGN */                                COM01350
 /*       *****                    */                          COM01360
DCL CARDA CHAR(80),                                             COM01370
    1 RCARDA DEF CARDA,                                         COM01380
      2 AH1    CHAR,                                            COM01390
      2 AIFMT  CHAR,                                            COM01400
      2 ASERNO CHAR(8),                                         COM01410
      2 ALFDA  PIC ZZ ,                                         COM01420
      2 ALYR   PIC ZZ ,                                         COM01430
      2 ALSTMO PIC ZZ ,                                         COM01440
      2 ALSTYR PIC ZZ ,                                         COM01450
      2 AIDEC  CHAR,                                            COM01460
      2 AIPLUS CHAR,                                            COM01470
      2 AICARD CHAR,                                            COM01480
      2 AKDEC  CHAR,                                            COM01490
      2 AMULAD CHAR,                                            COM01500
      2 AIYRT  CHAR,                                            COM01510
      2 AFULSM CHAR,                                            COM01520
      2 APROPT CHAR,                                            COM01530
      2 ACHOPT CHAR,                                            COM01540
      2 ASIGML CHAR(2),                                         COM01550
      2 ASIGMU CHAR(2),                                         COM01560
      2 ALTERM CHAR,                                            COM01570
      2 ATCOPT CHAR,                                            COM01580
      2 AIFORC CHAR,                                            COM01590
      2 AEXOPT CHAR,                                            COM01600
      2 AKFMT CHAR,                                             COM01610
      2 AKSER  CHAR(8),                                         COM01620
      2 ADWT(7) CHAR(4),                                        COM01630
      2 ALOPT  CHAR,                                            COM01640
      2 ADWOPT CHAR,                                            COM01650
```

```
        2 ALCYR  CHAR(2),                                          COM01660
        2 ALAYR  CHAR(2),                                          COM01670
        2 ASIGM  CHAR(2)                                           COM01680
    /*                          */                                 COM01690
    /* DATA CONTROL DESIGN */                                      COM01700
     /*                          */                                COM01710
  DCL CARDD CHAR(80),                                              COM01720
      1 RCARDD DEF CARDD,                                          COM01730
        2 DNAME  CHAR(5),  2 DCODE   CHAR(7),                      COM01740
        2 DFREQ  CHAR,     2 DFY     PIC ZZ ,                      COM01750
        2 DFM    PIC ZZ , 2  DEY     PIC ZZ ,                      COM01760
        2 DEM    PIC ZZ ,  2 DPRWGT  CHAR(2),                      COM01770
        2 DBASEY CHAR(2),  2 DPUT    CHAR,                         COM01780
        2 DNAMER CHAR(5),                                          COM01790
        2 DCODER CHAR(7),                                          COM01800
        2 DNTNG CHAR,                                              COM01810
        2 DINV   CHAR,     2 DIPR    CHAR,                         COM01820
        2 DNCHRT CHAR,     2 DIRD    CHAR,                         COM01830
        2 DNDTR  CHAR,     2 DLOG    CHAR,                         COM01840
        2 DMCD   CHAR,     2 DAMUL   CHAR(4),                      COM01850
        2 DITERM CHAR(2),  2 DNDIFF  CHAR(2),                      COM01860
        2 DLLPCFD CHAR,     2 DWGT    CHAR(4),                     COM01870
        2 DNMCD  CHAR,     2 DNMOV   CHAR(2),                      COM01880
        2 DNRAD  CHAR,                                             COM01890
        2 DOOST  CHAR(8),  2 DSPAN  CHAR(1),                       COM01900
        2 DFFSTQ CHAR(6),                                          COM01910
        2 DPRFMT  CHAR,                                            COM01920
        2 DLEVEL CHAR(1)                                           COM01930
  DCL SERI# FIXED(5),                                              COM01940
      ERR PIC 9  INIT(0),                                          COM01950
      OFSET FIXED(5)                                               COM01960
  DCL COD PIC 9999  INIT(0)                                        COM01970
  DCL FIRST PIC 9  INIT(1)                                         COM01980
  /*                                  */                           COM01990
  /* T/P CONROL RECORD FOR IND. ITEM */                            COM02000
  /*                                  */                           COM02010
  DCL CARDT CHAR(80) INIT(    ),                                   COM02020
      1 TCARD DEF CARDT,                                           COM02030
        2 TNTNG   CHAR,                                            COM02040
        2 TINV    CHAR,                                            COM02050
        2 TIPR    CHAR,                                            COM02060
        2 TNCHRT  CHAR,                                            COM02070
        2 TIRD    CHAR,                                            COM02080
        2 TNDTR   CHAR,                                            COM02090
        2 TLOG    CHAR,                                            COM02100
        2 TMCD    CHAR,                                            COM02110
        2 TAMUL   CHAR(4),                                         COM02120
        2 TITERM  CHAR(2),                                         COM02130
        2 TNDIFF  CHAR(2),                                         COM02140
        2 TLLPCFD CHAR,                                            COM02150
        2 TWGT    CHAR(4),                                         COM02160
        2 TNMCD   CHAR,                                            COM02170
        2 TNMOV   CHAR(2),                                         COM02180
        2 TNRAD   CHAR,                                            COM02190
        2 TOOST   CHAR(8),                                         COM02200
```

```
          2 TSPAN   CHAR,                                          COM02210
          2 TFFSTQ  CHAR(6),                                       COM02220
     ON ENDFILE(CARDIN) GOTO LAST                                  COM02230
     OPEN FILE(CARDIN),FILE(REGION)                                COM02240
COMPR                                                              COM02250
     /*                                    */                      COM02260
     /* COMPSITE INDEX CONTROL CARD READ */                        COM02270
     /*                                    */                      COM02280
     READ FILE(CARDIN) INTO(CARDC)                                 COM02290
     CALL COMPRTN                                                  COM02300
     IF CCOMP= COMP  THEN                                          COM02310
     DO  READ FILE(CARDIN) INTO(CARDD)       /* COMP TITLE READ */ COM02320
         CARDP=CARDD                                               COM02330
         WRITE FILE(REGION) FROM(CARDP)                            COM02340
         PUT SKIP EDIT(CARDD)(A)                                   COM02350
     END                                                           COM02360
     /*                                  */                        COM02370
     /* X-11 MAIN CONTROL CARD READ */                             COM02380
     /*                                  */                        COM02390
     READ FILE(CARDIN) INTO(CARDA)                                 COM02400
LOOP                                                               COM02410
     /*                          */                                COM02420
     /* DATA CONTROL CARD READ */                                   COM02430
     /*                          */                                COM02440
     READ FILE(CARDIN) INTO(CARDD)                                 COM02450
     IF SUBSTR(CARDD,1,1)= Z  THEN                                 COM02460
     DO  COD=COD+1                                                 COM02470
         CARDP=CARDD                                               COM02480
         WRITE FILE(REGION) FROM(CARDP)                            COM02490
         PUT SKIP EDIT(CARDD)(A(80))                               COM02500
         GOTO COMPR                                                COM02510
     END                                                           COM02520
     CALL DATAR                                                    COM02530
     IF CCOMP= NOCOMP    DNTNG=    THEN GOTO LOOP                  COM02540
     CALL ARIMA                                                    COM02550
     CALL TPCOMP                                                   COM02560
     IF AIFMT= 1  THEN                                             COM02570
     DO  CARD= (8X,12F6.1)                                         COM02580
        IF DFREQ= Q  THEN                                          COM02590
     DO     CARD= (8X,4(12X,F6.1))                                 COM02600
            IF AIDEC =BLANK THEN SUBSTR(CARD,14,1)=AIDEC           COM02610
        END  ELSE                                                  COM02620
        IF AIDEC =BLANK THEN SUBSTR(CARD,10,1)=AIDEC               COM02630
        PUT SKIP EDIT(CARD)(A)                                     COM02640
         CARDP=CARD                                                COM02650
         WRITE FILE(REGION) FROM(CARDP)                            COM02660
     END                                                           COM02670
     IF DPRFMT= 2  THEN                                            COM02680
     DO  CARD= (8X,12F6.1)                                         COM02690
        IF DFREQ= Q  THEN CARD= (8X,4(12X,F6.1))                   COM02700
        PUT SKIP EDIT(CARD)(A)                                     COM02710
         CARDP=CARD                                                COM02720
         WRITE FILE(REGION) FROM(CARDP)                            COM02730
     END                                                           COM02740
     CARD= T                                                       COM02750
```

```
      IF  DBKEY=ALLZERO  THEN                                          COM02760
      DO   SUBSTR(CARD,3,78) = TITLEO                                  COM02770
          CDWGT=10**WGTP                                               COM02780
      END   ELSE                                                       COM02790
      DO   SUBSTR(CARD,3,78)=TITLE                                     COM02800
          CDWGT=WGT                                                    COM02810
      END                                                              COM02820
      PUT SKIP EDIT(CARD)(A)                                           COM02830
      CARDP=CARD                                                       COM02840
      WRITE FILE(REGION) FROM(CARDP)                                   COM02850
      CALL PUTDATA(ASERNO)                                             COM02860
      IF DPRFMT  =BLANK THEN                                           COM02870
      DO   CALL PRFCTOR                                                COM02880
      KEY=AKSER                                                        COM02890
         CALL PUTDATA(KEY)                                             COM02900
      END                                                              COM02910
      GOTO LOOP                                                        COM02920
 LAST  CLOSE FILE(CARDIN),FILE(REGION)                                 COM02930
      /*                                                  */           COM02940
      /* COMPSITE CONTROL RECORD CREATION ROUTINE */                   COM02950
      /*                                                  */           COM02960
 COMPRTN  PROC                                                         COM02970
      DCL 1 COMPREC,                                                   COM02980
            2 OCOMP CHAR(6) INIT( COMP ),                              COM02990
            2 OKNO  CHAR(2),                                           COM03000
            2 OKEYR CHAR(2) INIT( 60 ),                                COM03010
            2 OWGT  CHAR(4) INIT( 1000 ),                              COM03020
            2 OCONST CHAR(5) INIT( 01000 ),                            COM03030
            2 OJFM  CHAR(2) INIT( 01 ),                                COM03040
            2 OJFY  CHAR(2) INIT( 60 ),                                COM03050
            2 OJEM  CHAR(2) INIT( 12 ),                                COM03060
            2 OJEY  CHAR(2) INIT( 79 ),                                COM03070
            2 OGNPGR CHAR(10),                                         COM03080
            2 DUMM  CHAR,                                              COM03090
            2 OGNPMQ CHAR INIT( 1 ),                                   COM03100
            2 OINV  CHAR,                                              COM03110
            2 OIPR  CHAR INIT( 1 ),                                    COM03120
            2 ONCHRT CHAR INIT( 2 ),                                   COM03130
            2 OIRD  CHAR,                                              COM03140
            2 ONDTR CHAR,                                              COM03150
            2 OLOG  CHAR,                                              COM03160
            2 OMCD  CHAR,                                              COM03170
            2 OAMUL  CHAR(4),                                          COM03180
            2 OITERM CHAR(2),                                          COM03190
            2 ONDIFF CHAR(2),                                          COM03200
            2 DUMMY CHAR(30)                                           COM03210
      DCL CARDR CHAR(80) DEF COMPREC                                   COM03220
      IF CKNO=    THEN                                                 COM03230
          PUT SKIP EDIT( NO. OF SERIES ERROR(COL 7-8) )(X(10),A)       COM03240
      IF CGNPGR=    THEN                                               COM03250
          PUT SKIP EDIT( GROWTH RATE OF G.N.P ERROR(COL 28 37) )       COM03260
                        (X(10),A)                                      COM03270
      IF CKEYR=BLANK THEN CKEYR=OKEYR                                  COM03280
      IF CWGT=BLANK THEN  CWGT=OWGT                                    COM03290
      IF CCONST=BLANK THEN CCONST=OCONST                               COM03300
```

```
      IF CJFM=BLANK THEN  CJFM=OJFM                                     COM03310
            IF CJFY=BLANK THEN  CJFY=OJFY                               COM03320
            IF CJEM=BLANK THEN  CJEM=OJEM                               COM03330
            IF CJEY=BLANK THEN  CJEY=OJEY                               COM03340
      IF CGNPMQ=BLANK  THEN CGNPMQ=OGNPMQ                               COM03350
      IF CIPR=BLANK THEN CIPR=OIPR                                      COM03360
      IF CNCHRT=BLANK THEN CNCHRT=ONCHRT                                COM03370
      CARDP=CARDC                                                       COM03380
      WRITE FILE(REGION) FROM(CARDP)                                    COM03390
      PUT SKIP EDIT(CARDC)(A)                                           COM03400
      SUBSTR(CARDT,2,79)=SUBSTR(CARDC,40,41)                            COM03410
      END                                                               COM03420
      /*                                                        */      COM03430
      /* T/P CONTROL RECORD CREATION FOR INDIVISUAL ITEM */              COM03440
      /*                                                        */      COM03450
 TPCOMP  PROC                                                           COM03460
      SUBSTR(CARDT,2,79)=SUBSTR(CARDC,40,41)                            COM03470
      TNTNG=DNTNG                                                       COM03480
      IF DNTNG =BLANK THEN DO                                           COM03490
      IF DIPR =BLANK THEN TIPR=DIPR                                     COM03500
      IF DNCHRT =BLANK THEN TNCHRT=DNCHRT                               COM03510
      IF DIRD =BLANK THEN TIRD=DIRD                                     COM03520
      IF DNDTR =BLANK THEN TNDTR=DNDTR                                  COM03530
      IF DLOG =BLANK THEN TLOG=DLOG                                     COM03540
      IF DMCD =BLANK THEN TMCD=DMCD                                     COM03550
      IF DAMUL =BLANK THEN TAMUL=DAMUL                                  COM03560
      IF DITERM =BLANK THEN TITERM=DITERM                               COM03570
      IF DNDIFF =BLANK THEN TNDIFF=DNDIFF                               COM03580
      IF DLLPCFD =BLANK THEN TLLPCFD=DLLPCFD                            COM03590
      IF DWGT =BLANK THEN TWGT=DWGT                                     COM03600
      IF DNMCD= 9  THEN TNMCD=BLANK                                     COM03610
            ELSE TNMCD=DNMCD                                            COM03620
      IF DNRAD =BLANK THEN TNRAD=DNRAD                                  COM03630
      IF DNMOV =BLANK THEN TNMOV=DNMOV                                  COM03640
      IF DOOST =BLANK THEN TOOST=DOOST                                  COM03650
      IF DSPAN =BLANK THEN TSPAN=DSPAN                                  COM03660
      IF DFFSTQ =BLANK THEN TFFSTQ=DFFSTQ                               COM03670
      END                                                               COM03680
      CARDP=CARDT                                                       COM03690
      WRITE FILE(REGION) FROM(CARDP)                                    COM03700
      PUT SKIP EDIT(CARDT)(A)                                           COM03710
      END                                                               COM03720
      /*                                 */                             COM03730
      /* X-11 MAIN CONTROL RECORD */                                     COM03740
      /*                                 */                             COM03750
 ARIMA  PROC                                                            COM03760
      DCL OREC  CHAR(80),                                               COM03770
          1 ROREC DEF OREC,                                             COM03780
            2 OH1    CHAR,                                              COM03790
            2 OIFMT  CHAR,                                              COM03800
            2 OSERNO CHAR(8),                                           COM03810
            2 OLFDA  CHAR(2),                                           COM03820
            2 OLYR   CHAR(2),                                           COM03830
            2 OLSTMO CHAR(2),                                           COM03840
            2 OLSTYR CHAR(2),                                           COM03850
```

```
          2 OIDEC  CHAR,                                          COM03860
          2 OIPLUS CHAR,                                          COM03870
          2 OICARD CHAR,                                          COM03880
          2 OKDEC  CHAR,                                          COM03890
          2 OMULAD CHAR,                                          COM03900
          2 OIYRT  CHAR,                                          COM03910
          2 OFULSM CHAR,                                          COM03920
          2 OPROPT CHAR,                                          COM03930
          2 OCHOPT CHAR,                                          COM03940
          2 OSIGML CHAR(2),                                       COM03950
          2 OSIGMU CHAR(2),                                       COM03960
          2 OLTERM CHAR,                                          COM03970
          2 OTCOPT CHAR,                                          COM03980
          2 OIFORC CHAR,                                          COM03990
          2 OEXOPT CHAR,                                          COM04000
          2 OKFMT CHAR,                                           COM04010
          2 OKSER  CHAR(8),                                       COM04020
          2 ODWT(7) CHAR(4),                                      COM04030
          2 OLOPT  CHAR,                                          COM04040
          2 ODWOPT CHAR,                                          COM04050
          2 OLCYR  CHAR(2),                                       COM04060
          2 OLAYR  CHAR(2),                                       COM04070
          2 OSIGM  CHAR(2)                                        COM04080
     OREC= CARDA                                                  COM04090
     OIDEC= BLANK                                                 COM04100
     OHI=DFREQ                                                    COM04110
     OSERNO=TRANSLATE(DCODE, 0 ,    )                             COM04120
     IF DCODE=BLANK THEN OSERNO=DBNAME                            COM04130
     IF DPRWGT =BLANK THEN OKSER=DPRWGT                           COM04140
                      ELSE OKSER=DNAMER                           COM04150
     AKSER=OKSER                                                  COM04160
     IF DPRFMT =BLANK THEN OKFMT=DPRFMT                           COM04170
     IF AIFMT = 4  THEN SUBSTR(OSERNO,7,2)=                       COM04180
     ASERNO=OSERNO                                                COM04190
     PUT SKIP EDIT(OREC )(A)                                      COM04200
     CARDP=OREC                                                   COM04210
     WRITE FILE(REGION) FROM(CARDP)                               COM04220
     IF CCOMP= NOCOMP    DNING=    THEN CARDA=OREC                COM04230
     END                                                          COM04240
     /*                        */                                 COM04250
     /* DATA RECORD CREATION */                                   COM04260
     /*                        */                                 COM04270
DATAR  PROC                                                       COM04280
     DCL BLANK PIC ZZ  INIT(0)                                    COM04290
     PLEVEL=0                                                     COM04300
     IF CCOMP= NOCOMP    DNING=    THEN                           COM04310
     DO  CALL X11PGM                                              COM04320
     RETURN                                                       COM04330
     END                                                          COM04340
     IF DNAME= ULCST  THEN                                        COM04350
     DO  FREQ=DFREQ                                               COM04360
         DBNAME= EAGIN                                            COM04370
         DBKEY= 3                                                 COM04380
         CALL DBGET                                               COM04390
         IF PLEVEL   0 THEN                                       COM04400
```

```
      DO  IF DFY=BLANK THEN DFY=INITY                                   COM04410
       IF DFM=BLANK THEN DFM=INITM                                      COM04420
       IF DEY=BLANK THEN DEY=ENDY                                       COM04430
       IF DEM=BLANK THEN DEM=ENDM                                       COM04440
      ALFDA=DFM                                                         COM04450
      ALSTYR=DEY                                                        COM04460
      ALYR=DFY                                                          COM04470
      ALSTMO=DEM                                                        COM04480
      END                                                               COM04490
      CALL EDIT(DATA1)                                                  COM04500
      DNAMER= CPIAC                                                     COM04510
      CALL REAL                                                         COM04520
      DATA3=DATA1                                                       COM04530
      DBNAME= EMPEI                                                     COM04540
      DBKEY= 3                                                          COM04550
      PLEVEL=0                                                          COM04560
      CALL DBGET                                                        COM04570
      CALL EDIT(DATA1)                                                  COM04580
      CALL INDEX(DBASEY)                                                COM04590
      DATA2=DATA1                                                       COM04600
      DBNAME= PRODI                                                     COM04610
      PLEVEL=0                                                          COM04620
      CALL DBGET                                                        COM04630
      CALL EDIT(DATA1)                                                  COM04640
      DO I=1 TO SERI#                                                   COM04650
         IF DATA1(I) =0 THEN DATA1(I)=DATA3(I)*100                      COM04660
               /DATA1(I)*DATA2(I)                                       COM04670
      END                                                               COM04680
     IF DBASEY =       THEN                                             COM04690
      CALL INDEX(DBASEY)                                                COM04700
      TITLEO= UNIT LABOR COST                                           COM04710
      TITLE=TITLEO                                                      COM04720
      KEYO= ULCST                                                       COM04730
      DBNAME=KEYO                                                       COM04740
      DBKEY=ALLZERO                                                     COM04750
      RETURN                                                            COM04760
  END                                                                   COM04770
  DBNAME=DNAME                                                          COM04780
  DBKEY=DCODE                                                           COM04790
  FREQ=DFREQ                                                            COM04800
  CALL DBGET                                                            COM04810
  IF PLEVEL 0 THEN                                                      COM04820
  DO                                                                    COM04830
      IF DFY=BLANK THEN ALYR=INITY                                      COM04840
                   ELSE ALYR=DFY                                        COM04850
      IF DFM=BLANK THEN ALFDA=INITM                                     COM04860
                   ELSE ALFDA=DFM                                       COM04870
      IF DEY=BLANK THEN ALSTYR=ENDY                                     COM04880
                   ELSE ALSTYR=DEY                                      COM04890
      IF DEM=BLANK THEN ALSTMO=ENDM                                     COM04900
                   ELSE ALSTMO=DEM                                      COM04910
      CALL EDIT(DATA1)                                                  COM04920
      IF DNAMER =       THEN                                            COM04930
         CALL REAL                                                      COM04940
       IF DBASEY=       THEN  RETURN                                    COM04950
```

```
        CALL INDEX(DBASEY)                                          COM04960
    END                                                             COM04970
    RETURN                                                          COM04980
    END                                                             COM04990
    /*                          */                                  COM05000
    /* PRIOR FACTOR RECORD */                                       COM05010
    /*                          */                                  COM05020
PRFCTOR  PROC                                                       COM05030
    DBNAME= PRFCT                                                   COM05040
    DBKEY=TRANSLATE(DPRWGT, 0 ,    )                                COM05050
    FREQ=DFREQ                                                      COM05060
    PLEVEL=0                                                        COM05070
    CALL DBGET                                                      COM05080
    IF PLEVEL=0 THEN                                                COM05090
    DO  PUT SKIP EDIT(DBNAME,DBKEY, NOT FOUND FROM PRFCT PROC )     COM05100
                 (A(8),A(10),A)                                     COM05110
        RETURN                                                      COM05120
    END                                                             COM05130
    IF DBKEY=ALLZERO THEN AKSER=DBNAME                              COM05140
            ELSE AKSER=KEY                                          COM05150
    CALL EDIT(DATA1)                                                COM05160
    END                                                             COM05170
    /*                        */                                    COM05180
    /* DATA EDIT ROUTINE */                                         COM05190
    /*                        */                                    COM05200
EDIT  PROC(DATAC)                                                   COM05210
    DCL DATAC(360) FLOAT                                            COM05220
    IF FREQ= M  THEN PNO=12                                         COM05230
                ELSE PNO=4                                          COM05240
    SERI#=(ALSTYR-ALYR)*PNO+(ALSTMO-ALFDA)+1                        COM05250
    OFSET=(ALYR-INITY)*PNO+(ALFDA-INITM)                            COM05260
    IF ALFDA 1     THEN OFSET=OFSET+ALFDA-1                         COM05270
    IF OFSET 0 THEN                                                 COM05280
    DO  DO I=1 TO -OFSET                                            COM05290
            DATAC(I)=0                                              COM05300
        END                                                         COM05310
        DO I=1 TO SERI#                                             COM05320
            DATAC(I-OFSET)=DATA(I)/10**DECP                         COM05330
        END                                                         COM05340
        ST=-OFSET+SERI#+1                                           COM05350
    END  ELSE                                                       COM05360
    DO  DO I=1 TO SERI#                                             COM05370
            DATAC(I)=DATA(I+OFSET)/10**DECP                         COM05380
        END                                                         COM05390
        ST=SERI#+1                                                  COM05400
    END                                                             COM05410
    IF ST 361 THEN                                                  COM05420
    DO  DO I=ST TO 360                                              COM05430
            DATAC(I)=0                                              COM05440
        END                                                         COM05450
    END                                                             COM05460
    END                                                             COM05470
    /*                    */                                        COM05480
    /* PUT DATA RECORD */                                           COM05490
    /*                    */                                        COM05500
```

```
PUTDATA  PROC(ASERNO)                                                   COM05510
     DCL ASERNO CHAR(8)                                                 COM05520
     DCL  PRECD CHAR(80),                                               COM05530
        1 PRECO DEF PRECD,                                              COM05540
          2 DTID       CHAR(6),                                         COM05550
          2 YEAR       PIC ZZ ,                                         COM05560
          2 DATAC(12) CHAR(6),                                          COM05570
        1 PRECD1 DEF PRECD,                                             COM05580
          2 FILLO CHAR(8),                                              COM05590
          2 DATAP(12) PIC (6)Z ,                                        COM05600
        1 PREC1 DEF PRECD,                                              COM05610
          2 FILL1 CHAR(8),                                              COM05620
          2 DATAP1(12) PIC (5)ZVZ ,                                     COM05630
        1 PREC2 DEF PRECD,                                              COM05640
          2 FILL2 CHAR(8),                                              COM05650
          2 DATAP2(12) PIC (4)ZVZZ ,                                    COM05660
        1 PREC3 DEF PRECD,                                              COM05670
          2 FILL3 CHAR(8),                                              COM05680
          2 DATAP3(12) PIC ZZZVZZZ                                      COM05690
     IF ERR=1 THEN                                                      COM05700
     DO  ERR=0                                                          COM05710
        RETURN                                                          COM05720
     END                                                                COM05730
     SU=ALSTYR-ALYR+1                                                   COM05740
     ST=-11  LT=0  DTID=ASERNO                                          COM05750
     IF DFREQ= M  THEN                                                  COM05760
     DO  NO=1                                                           COM05770
         PNO=12                                                         COM05780
         ST=-11                                                         COM05790
     END  ELSE                                                          COM05800
     DO  NO=3                                                           COM05810
         ST=-3                                                          COM05820
         PNO=4                                                          COM05830
     END                                                                COM05840
     DO I=1 TO SU                                                       COM05850
        ST=ST+PNO                                                       COM05860
        LT=LT+PNO                                                       COM05870
        DATAC= 0                                                        COM05880
        K=0                                                             COM05890
           IF AIDEC  = BLANK THEN   DECB=3                              COM05900
                             ELSE DECB=DECP                             COM05910
        DO J=ST TO LT                                                   COM05920
           K=K+NO                                                       COM05930
           IF DECB=0 THEN DATAP1(K)=DATA1(J)+0.51                       COM05940
           ELSE IF DECB=1 THEN DATAP1(K)=DATA1(J)+0.051                 COM05950
           ELSE IF DECB=2 THEN DATAP1(K)=DATA1(J)+0.051                 COM05960
         ELSE DATAP2(K)=DATA1(J)+0.0051                                 COM05970
        END                                                             COM05980
        YEAR=ALYR+I-1                                                   COM05990
         CARDP=PRECD                                                    COM06000
         WRITE FILE(REGION) FROM(CARDP)                                 COM06010
        IF DPUT= 1  THEN PUT SKIP EDIT(PRECD)(A)                        COM06020
     END                                                                COM06030
     END                                                                COM06040
     /*                                          */                     COM06050
```

```
     /* INDEX CREATION BASED ON CONSTANT VALUE */
     /*                                        */
INDEX  PROC(BASEY)
     DCL BASE FLOAT(16) INIT(0)
     DCL BASEY  CHAR(2)
     DCL BASE9 PIC 99  DEF BASEY
         IF DFREQ= M  THEN PNO=12
                      ELSE PNO=4
         OFSET=(BASEY-ALYR)*PNO-ALFDA+1
         DO I=1 TO PNO
              BASE=BASE+DATA1(I+OFSET)
         END
         IF BASE=0 THEN RETURN
         BASE=BASE/PNO
     DO I=1 TO SERI#+ALFDA-1
        DATA1(I)=DATA1(I)*100/BASE
     END
     IF AIDEC=BLANK THEN DECP=1
     END
     /*                              */
     /* CONSTANT VALUE COMPUTATION */
     /*                              */
REAL  PROC
         IF DCODE=BLANK THEN CARD=TITLE0
                        ELSE CARD=TITLE
     DBNAME=DNAMER
     DBKEY=DCODER
     PLEVEL=0
     CALL DBGET
     IF PLEVEL=0 THEN
     DO  PUT SKIP EDIT(DBNAME,DBKEY, NOT FOUND FROM REAL PROC )
                  (A(8),A(10),A)
         RETURN
     END
     CALL EDIT(DATA2)
     DO I=1 TO SERI#+ALFDA-1
        IF DATA2(I) 0 THEN
           DATA1(I)=DATA1(I)*100/DATA2(I)
           ELSE DATA1(I)=0
     END
     DBNAME=DNAME
     DBKEY=DCODE
     DBKEY=TRANSLATE(DBKEY, 0 ,   )
     IF DCODE=BLANK THEN TITLE0=CARD
                    ELSE TITLE=CARD
     DECP=2
     END
     /*****************************************/
     /* SEASONAL ADJUSTMENT ONLY FOR ALL SERIES */
     /*****************************************/
X11PGM  PROC
     DCL CARD1 CHAR(80),
         CARD2 CHAR(80)
  DCL  FINITY PIC 99 ,
       FINITM  PIC 99 ,
```

```
      FENDY  PIC 99 ,                                                   COM06610
      FENDM  PIC 99                                                     COM06620
   DCL OFIRSTY PIC 99 ,                                                 COM06630
       OFIRSTM PIC 99 ,                                                 COM06640
       OENDY   PIC 99 ,                                                 COM06650
       OENDM   PIC 99                                                   COM06660
   DCL DECPO PIC 9                                                      COM06670
   DCL DECPF PIC 9                                                      COM06680
   PLEVEL=0                                                             COM06690
   DBNAME=DNAME                                                         COM06700
   DBKEY=TRANSLATE(DCODE, 0 ,    )                                      COM06710
   FREQ=DFREQ                                                           COM06720
   IF DLEVEL =BLANK THEN LOWLEV=DLEVEL                                  COM06730
   CALL DBGET                                                           COM06740
   IF PLEVEL=0 THEN RETURN                                              COM06750
   DECPO=DECP                                                           COM06760
   IF DFY=0 THEN ALYR =INITY                                            COM06770
            ELSE ALYR=DFY                                               COM06780
   IF DFM=0 THEN ALFDA=INITM                                            COM06790
            ELSE ALFDA=DFM                                              COM06800
   IF DEY=0 THEN ALSTYR=ENDY                                            COM06810
            ELSE ALSTYR=DEY                                             COM06820
   IF DEM=0 THEN ALSTMO=ENDM                                            COM06830
            ELSE ALSTMO=DEM                                             COM06840
   OFIRSTY=ALYR                                                         COM06850
   OFIRSTM=ALFDA                                                        COM06860
   OENDY=ALSTYR                                                         COM06870
   OENDM=ALSTMO                                                         COM06880
        PLEVEL=0                                                        COM06890
  IF DPRFMT =BLANK  THEN                                                COM06900
   DO  CALL PRFCTOR                                                     COM06910
       DECPF=DECP                                                       COM06920
       DATA3=DATA                                                       COM06930
      FINITY=INITY                                                      COM06940
      FINITM=INITM                                                      COM06950
      FENDY=ENDY                                                        COM06960
      FENDM=ENDM                                                        COM06970
   END                                                                  COM06980
   PLEVEL=0                                                             COM06990
   DBNAME=DNAME                                                         COM07000
   DBKEY=DCODE                                                          COM07010
   CALL DBGET                                                           COM07020
   CALL ARIMA                                                           COM07030
   CALL TPCOMP                                                          COM07040
   IF AIFMT= 1  THEN                                                    COM07050
   DO  CARD1= (8X,12F6.1)                                               COM07060
       IF DFREQ= Q  THEN                                                COM07070
       DO  CARD1= (8X,4(12X,F6.1))                                      COM07080
         IF AIDEC =BLANK THEN SUBSTR(CARD1,14,1)=AIDEC                  COM07090
       END  ELSE                                                        COM07100
       IF AIDEC =BLANK  THEN SUBSTR(CARD1,10,1)=AIDEC                   COM07110
       PUT SKIP EDIT(CARD1)(A)                                          COM07120
       CARDP=CARD1                                                      COM07130
       WRITE FILE(REGION) FROM(CARDP)                                   COM07140
   END                                                                  COM07150
```

```
      IF DPRFMT= 2  THEN                                                COM07160
      DO  CARD2= (8X,12F6.1)                                            COM07170
          IF DFREQ= Q  THEN                                             COM07180
                   CARD2= (8X,4(12X,F6.1))                              COM07190
          PUT SKIP EDIT(CARD2)(A)                                       COM07200
          CARDP=CARD2                                                   COM07210
          WRITE FILE(REGION) FROM(CARDP)                                COM07220
      END                                                               COM07230
      SUBSTR(CARD,1,2)= T                                               COM07240
      IF DBKEY=ALLZERO THEN                                             COM07250
      SUBSTR(CARD,3,78)=TITLEO                                          COM07260
                        ELSE SUBSTR(CARD,3,78)=TITLE                    COM07270
      CARDP=CARD                                                        COM07280
      WRITE FILE(REGION) FROM(CARDP)                                    COM07290
      PUT SKIP EDIT(CARD)(A)                                            COM07300
      CALL EDIT(DATA1)                                                  COM07310
      CALL PUTDATA(ASERNO)                                              COM07320
      IF DPRFMT  =BLANK THEN                                            COM07330
      DO    IF DCODER=BLANK THEN ASERNO=DNAMER                          COM07340
                                   ELSE ASERNO=DCODER                   COM07350
          DATA=DATA3                                                    COM07360
          DECP=DECPF                                                    COM07370
      INITY=FINITY                                                      COM07380
      INITM=FINITM                                                      COM07390
      ENDY=FENDY                                                        COM07400
      ENDM=FENDM                                                        COM07410
          CALL EDIT(DATA1)                                              COM07420
         CALL PUTDATA(ASERNO)                                           COM07430
      END                                                               COM07440
 LOOP                                                                   COM07450
      DECP=DECPO                                                        COM07460
      CALL DBGET                                                        COM07470
      IF PLEVEL=0 THEN RETURN                                           COM07480
      IF ERR=1 THEN                                                     COM07490
      DO  ERR=0                                                         COM07500
          GOTO LOOP                                                     COM07510
      END                                                               COM07520
      ALYR=OFIRSTY                                                      COM07530
      ALFDA=OFIRSTM                                                     COM07540
      ALSTYR=OENDY                                                      COM07550
      ALSTMO=OENDM                                                      COM07560
      IF INITY OFIRSTY THEN                                             COM07570
      DO  ALYR=INITY                                                    COM07580
          ALFDA=INITM                                                   COM07590
      END                                                               COM07600
      IF ENDY  OENDY THEN ALSTYR=ENDY                                   COM07610
      IF ENDM  OENDM THEN ALSTMO=ENDM                                   COM07620
      ASERNO=KEY                                                        COM07630
      CARDP=CARDA                                                       COM07640
      PUT SKIP EDIT(CARDA)(A)                                           COM07650
      WRITE FILE(REGION) FROM(CARDP)                                    COM07660
      CALL TPCOMP                                                       COM07670
      IF AIFMT= 1  THEN                                                 COM07680
      DO  COD=COD+1                                                     COM07690
          CARDP=CARD1                                                   COM07700
```

```
      PUT SKIP EDIT(CARD1)(A)                                          COM07710
          WRITE FILE(REGION) FROM(CARDP)                               COM07720
      END                                                              COM07730
      IF DPRFMT= 2  THEN                                               COM07740
      DO  COD=COD+1                                                    COM07750
          CARDP=CARD2                                                  COM07760
      PUT SKIP EDIT(CARD2)(A)                                          COM07770
          WRITE FILE(REGION) FROM(CARDP)                               COM07780
      END                                                              COM07790
      SUBSTR(CARD,1,2)= T                                              COM07800
      SUBSTR(CARD,3,78)=TITLE                                          COM07810
      CARDP=CARD                                                       COM07820
      WRITE FILE(REGION) FROM(CARDP)                                   COM07830
      PUT SKIP EDIT(CARD)(A)                                           COM07840
       CALL EDIT(DATA1)                                                COM07850
      CALL PUTDATA(KEY)                                                COM07860
      IF DPRFMT  =BLANK THEN                                           COM07870
 DO   DECP=DECPF                                                       COM07880
          DATA=DATA3                                                   COM07890
      INITY=FINITY                                                     COM07900
      INITM=FINITM                                                     COM07910
      ENDY=FENDY                                                       COM07920
      ENDM=FENDM                                                       COM07930
      CALL EDIT(DATA1)                                                 COM07940
     CALL PUTDATA(ASERNO)                                              COM07950
      END                                                              COM07960
      GOTO LOOP                                                        COM07970
      END                                                              COM07980
                                                  /******************/ COM07990
    DBGET   PROC                                                       COM08000
       SUBSTR(QSSAO,1,18)= TOTAL   (KEYO                               COM08010
       ALLZERO= 000000000                                              COM08020
       IF PLEVEL   LOWLEV THEN                                         COM08030
       DO   PLEVEL = 0                                                 COM08040
            RETURN                                                     COM08050
       END                                                             COM08060
       IF PLEVEL=0 THEN                                                COM08070
       DO   SSAKEYO=DBNAME                                             COM08080
           CALL PLITDLI(FOUR,GU,PCBP,LEVELO,QSSAO)                     COM08090
           IF STATUS  =     THEN                                       COM08100
           DO  CALL ERROR   RETURN    END                              COM08110
           MQYSSA=FREQ                                                 COM08120
           ATTLEV=PLEVEL                                               COM08130
           CALL PLITDLI(FOUR,GNP,PCBP,ATTO,ATTSSA)                     COM08140
           IF STATUS  =     THEN                                       COM08150
           DO  CALL ERROR   RETURN   END                               COM08160
           IF LOWLEV = 9 THEN LOWLEV = TOTLEV                          COM08170
           DBKEY=TRANSLATE(DBKEY, 0 ,   )                              COM08180
           IF DBKEY=ALLZERO THEN                                       COM08190
           DO  UPLEV=0                                                 COM08200
               UNIT=UNITO                                              COM08210
               WGT=10**WGTP                                            COM08220
               GO TO RDDATA                                            COM08230
           END                                                         COM08240
           TESTSTR=DBKEY                                               COM08250
```

```
          SUMKEYSZ=0                                                    COM08260
          SSAKEY1=ALLZERO                                               COM08270
          CALL LEVTEST(QSSA1)                                           COM08280
          SSAKEY2=ALLZERO                                               COM08290
          CALL LEVTEST(QSSA2)                                           COM08300
          SSAKEY3=ALLZERO                                               COM08310
          CALL LEVTEST(QSSA3)                                           COM08320
          SSAKEY4=ALLZERO                                               COM08330
          CALL LEVTEST(QSSA4)                                           COM08340
          SSAKEY5=ALLZERO                                               COM08350
          CALL LEVTEST(QSSA5)                                           COM08360
          SSAKEY6=ALLZERO                                               COM08370
          CALL LEVTEST(QSSA6)                                           COM08380
    LEVTEST  PROC(QSSA)                                                 COM08390
      DCL QSSA CHAR(29),                                                COM08400
          QSSAKEY CHAR(9) DEF QSSA POS(20)                              COM08410
          PLEVEL=PLEVEL+1                                               COM08420
          SUMKEYSZ=SUMKEYSZ+KEYSZ(PLEVEL)                               COM08430
          SUBSTR(TESTSTR,1,SUMKEYSZ)=ALLZERO                            COM08440
          SUBSTR(QSSAKEY,1,SUMKEYSZ)=SUBSTR(DBKEY,1,SUMKEYSZ)           COM08450
          IF TESTSTR=ALLZERO THEN GOTO FINDLEV                          COM08460
       END LEVTEST                                                      COM08470
          PUT SKIP EDIT(DBNAME,DBKEY, NOT FOUND )(A(6),A(10),A)         COM08480
          PLEVEL=0  RETURN                                              COM08490
  FINDLEV                                                               COM08500
          UPLEV=PLEVEL                                                  COM08510
          P#=PLEVEL+4                                                   COM08520
          CALL PLITDLI(P#,GU,PCBP,LEVELIO,QSSA0,QSSA1,QSSA2,QSSA3,      COM08530
                       QSSA4,QSSA5,QSSA6)                               COM08540
          IF STATUS  =     THEN                                         COM08550
          DO  CALL ERROR                                                COM08560
              PLEVEL=0   RETURN                                         COM08570
          END                                                           COM08580
          GOTO RDATT                                                    COM08590
     END                                                                COM08600
  RDLEVEL                                                               COM08610
     CALL DLI                                                           COM08620
     IF STATUS  =     THEN                                              COM08630
     DO  IF STATUS= GE  THEN                                            COM08640
         DO  K=UPLEV+1                                                  COM08650
             IF PLEVEL=K THEN                                           COM08660
             DO  PLEVEL=0                                               COM08670
                 RETURN                                                 COM08680
             END                                                        COM08690
             PLEVEL=PLEVEL-1  GOTO RDLEVEL                              COM08700
         END  ELSE                                                      COM08710
         DO  CALL ERROR  PLEVEL=0  RETURN  END                          COM08720
     END                                                                COM08730
  RDATT                                                                 COM08740
     ATTLEV=PLEVEL                                                      COM08750
     CALL PLITDLI(FOUR,GNP,PCBP,ATTIO,ATTSSA)                           COM08760
     IF STATUS  =     THEN                                              COM08770
     DO  IF STATUS = GE  THEN CALL ERROR                                COM08780
         WGT=0  UNIT=                                                   COM08790
     END                                                                COM08800
```

```
RDDATA                                                                  COM08810
    MQYLEV=PLEVEL                                                       COM08820
    CALL PLITDLI(FOUR,GNP,PCBP,MQYIO,MQYSSA)                            COM08830
    IF STATUS  =     THEN                                               COM08840
    DO  IF STATUS  = GE  THEN                                           COM08850
        DO  CALL ERROR                                                  COM08860
            PLEVEL=0                                                    COM08870
            RETURN                                                      COM08880
        END                                                             COM08890
        ERR=1                                                           COM08900
        DATA=0                                                          COM08910
    END                                                                 COM08920
    IF PLEVEL  =LOWLEV THEN                                             COM08930
    DO  PLEVEL=PLEVEL+1                                                 COM08940
        RETURN                                                          COM08950
    END                                                                 COM08960
    IF UPLEV=LOWLEV THEN                                                COM08970
    DO  PLEVEL=PLEVEL+1                                                 COM08980
        RETURN                                                          COM08990
    END                                                                 COM09000
DLI   PROC                                                              COM09010
      DCL LEV(6) LABEL                                                  COM09020
          GO TO LEV(PLEVEL)                                             COM09030
  LEV(1)  CALL PLITDLI(FIVE,GN,PCBP,LEVELIO,QSSA0,UQSSA1)               COM09040
          SSAKEY1=KEY  RETURN                                           COM09050
  LEV(2)  CALL PLITDLI(SIX,GN,PCBP,LEVELIO,QSSA0,QSSA1,UQSSA2)          COM09060
          SSAKEY2=KEY  RETURN                                           COM09070
  LEV(3)  CALL PLITDLI(SEVEN,GN,PCBP,LEVELIO,QSSA0,QSSA1,QSSA2,         COM09080
            UQSSA3)                                                     COM09090
          SSAKEY3=KEY  RETURN                                           COM09100
  LEV(4)  CALL PLITDLI(EIGHT,GN,PCBP,LEVELIO,QSSA0,QSSA1,QSSA2,         COM09110
            QSSA3,UQSSA4)                                               COM09120
          SSAKEY4=KEY  RETURN                                           COM09130
  LEV(5)  CALL PLITDLI(NINE,GN,PCBP,LEVELIO,QSSA0,QSSA1,QSSA2,          COM09140
            QSSA3,QSSA4,UQSSA5)                                         COM09150
          SSAKEY5=KEY  RETURN                                           COM09160
  LEV(6)  CALL PLITDLI(TEN,GN,PCBP,LEVELIO,QSSA0,QSSA1,QSSA2,           COM09170
            QSSA3,QSSA4,QSSA5,UQSSA6)                                   COM09180
          SSAKEY6=KEY  RETURN                                           COM09190
          END                                                           COM09200
  ERROR  PROC                                                           COM09210
         PUT EDIT( DBNAME ,DBNAME, DBKEY ,DBKEY, NOT FOUND )            COM09220
                 (SKIP,3 A(7),A(9),A)                                   COM09230
         PUT SKIP                                                       COM09240
         ERR=1   END   END                                              COM09250
        END     /**** COMPIN END *****/                                 COM09260
                                                                        COM09270
                                                                        COM09280
```

프로그램 : DLIX11

기능 : 조정지수 Updating 과정

```
//SDBAIL71  JOB   CLASS=S,PRTY=13                                    BOX00010
//          EXEC  IMSPLI,MBR=DLIX11                                  BOX00020
//SYSLIB    DD DSN=IMSVS.SSSLIB,DISP=SHR
//SYSPRINT  DD SYSOUT=A
//SYSIN   DD    *                                                   BOX00030
* PROCESS GS,A,XREF,NEST,INCLUDE                                    BOX00040
      BOXING  PROC(PCBP) OPTIONS(MAIN)                              BOX00050
          DCL PCBP    POINTER                                       BOX00060
          DCL 1 PCBNAME  BASED(PCBP),                               BOX00070
                2 M NAME          CHAR(8),                          BOX00080
                2 M SEF LEVEL     CHAR(2),                          BOX00090
                2 M STAT CODE     CHAR(2),                          BOX00100
                2 M PROC OPT      CHAR(4),                          BOX00110
                2 FILLER          FIXED BIN(31,0),                  BOX00120
                2 M SEG NAME      CHAR(8),                          BOX00130
                2 M LEV KFB       FIXED BIN(31,0),                  BOX00140
                2 M NO SENSEG     FIXED BIN(31,0),                  BOX00150
                2 M KEY KFB       CHAR(50)                          BOX00160
          DCL LEVEL    PIC 99 ,                                     BOX00170
              (FSTY,FSTM,ENDM,ENDY)  PIC  99 ,                      BOX00180
              FUNCT       CHAR(6),                                  BOX00190
              DBNAME      CHAR(5),                                  BOX00200
              DBKEY       CHAR(9),                                  BOX00210
              MQY         CHAR(1),                                  BOX00220
              ZEROCHAR    CHAR(9) INITI 000000000 ),                BOX00230
              INKEY(464) CHAR(9),                                   BOX00240
              INDATA(360) FLOAT(16),                                BOX00250
              HTITLE      CHAR(30) INIT(    ),                      BOX00260
              HUNIT       CHAR(11) INIT(    ),                      BOX00270
              HWGT        FLOAT    INIT(0),                         BOX00280
              1 GIGAN,                                              BOX00290
                2 FROMY  PIC 99 ,                                   BOX00300
                2 FROMM  PIC 99 ,                                   BOX00310
                2 TOY    PIC 99 ,                                   BOX00320
                2 TOM    PIC 99                                     BOX00330
          DCL  ADJFILE FILE RECORD INPUT SEQUENTIAL                 BOX00340
          DCL  1 INREC,                                             BOX00360
                 2 FILLER1      CHAR(6),                            BOX00370
                 2 IFROMY       PIC 99 ,                            BOX00380
                 2 FILLER2      CHAR(2),                            BOX00390
                 2 MDATA(12)    PIC 9999999V.9                      BOX00400
                 2 FILLER3      CHAR(142)                           BOX00410
          DCL  1 QINREC,                                            BOX00420
                 2 FIL1         CHAR(6),                            BOX00430
                 2 QFROMY       PIC 99 ,                            BOX00440
                 2 FIL2         CHAR(9),                            BOX00450
                 2 QDATA1       PIC 9999999V.9 ,                    BOX00460
                 2 FIL3         CHAR(15),                           BOX00470
                 2 QDATA2       PIC 9999999V.9 ,                    BOX00480
                 2 FIL4         CHAR(15),                           BOX00490
                 2 QDATA3       PIC 9999999V.9 ,                    BOX00500
                 2 FIL5         CHAR(15),                           BOX00510
                 2 QDATA4       PIC 9999999V.9 ,                    BOX00520
                 2 FIL6         CHAR(162)                           BOX00530
          DCL  YEAR     PIC 99 ,                                    BOX00540
```

```
            MONTH     PIC 99 ,
            WHAT      CHAR(6),                                          BOX00550
            COUNT     FIXED,                                            BOX00560
            SP        FIXED,                                            BOX00570
            PNO       FIXED,                                            BOX00580
            I         FIXED,                                            BOX00590
            J         FIXED,                                            BOX00600
            PSW       PIC 9  INIT(1),
            K         FIXED,                                            BOX00610
            SELECT    CHAR(6)                                           BOX00620
      ON ENDFILE(ADJFILE)  GO TO FINAL                                  BOX00630
      ON ENDFILE(SYSIN) GOTO PLAY                                       BOX00640
      OPEN FILE(SYSPRINT) LINESIZE(132)                                 BOX00650
      OPEN FILE(ADJFILE)                                                BOX00660
 GAME                                                                   BOX00670
      GET LIST(YEAR,MONTH,WHAT,SELECT)                                  BOX00680
      COUNT=1                                                           BOX00690
      IF SELECT= INSERT  THEN FUNCT= INSERT                             BOX00700
                         ELSE FUNCT= MODIFY                             BOX00710
      INKEY=                                                            BOX00720
      INDATA=0                                                          BOX00730
      GET LIST(DBNAME,MQY,SP)                                           BOX00740
      FSTM=1                                                            BOX00750
      IF MQY = Q  THEN DO                                               BOX00760
      PNO=12                                                            BOX00770
      MQY= M                                                            BOX00780
      END                                                               BOX00790
      ELSE PNO=4                                                        BOX00800
      ENDM=PNO                                                          BOX00810
      GET SKIP LIST((INKEY(I) DO I=1 TO SP))                            BOX00820
      I=0                                                               BOX00830
      J=0                                                               BOX00840
 AREAD                                                                  BOX00850
      READ FILE(ADJFILE)INTO(INREC)                                     BOX00860
      IF IFROMY =YEAR  THEN GOTO AREAD                                  BOX00870
      ELSE DO                                                           BOX00880
            COUNT=COUNT+1                                               BOX00890
            IF WHAT= ADJUST  THEN                                       BOX00900
            DO  IF COUNT=4 THEN DO  CALL KICKIN                         BOX00910
                                    COUNT=1                             BOX00920
                                    GOTO AREAD                          BOX00930
                                END                                     BOX00940
                            ELSE GOTO AREAD                             BOX00950
            END                                                         BOX00960
            ELSE DO  IF COUNT=2 THEN DO  CALL KICKIN                    BOX00970
                                         COUNT=1                        BOX00980
                                         GOTO AREAD                     BOX00990
                                     END                                BOX01000
                            ELSE GOTO AREAD                             BOX01010
            END                                                         BOX01020
      END                                                               BOX01030
 KICKIN  PROC                                                           BOX01040
       /* THIS IS A SUBPROG FOR DATA LOAD INTO DATA BASE */             BOX01050
       I=I+1
       DBKEY=INKEY(I)
```

```
     REREAD                                                          BOX01060
          IF MQY= M  THEN DO                                         BOX01070
                               READ FILE(ADJFILE) INTO(INREC)        BOX01080
                               IF IFROMY 30 THEN GOTO REREAD
                               IF IFROMY YEAR THEN GOTO REREAD
                               FROMY=IFROMY                          BOX01090
                               FROMM=1                               BOX01100
                               TOY=IFROMY                            BOX01110
                               IF IFROMY=YEAR THEN TOM=MONTH
                               ELSE  TOM=12
                               END                                   BOX01130
                          ELSE DO                                    BOX01140
                               READ FILE(ADJFILE) INTO(QINREC)       BOX01150
                               IF QFROMY 30 THEN GOTO REREAD
                               IF QFROMY YEAR THEN GOTO REREAD
                               FROMM=1                               BOX01170
                               TOY=QFROMY                            BOX01180
                               IF QFROMY=YEAR THEN TOM=MONTH/3
                               ELSE TOM=4
                               MDATA(1)=QDATA1                       BOX01200
                               MDATA(2)=QDATA2                       BOX01210
                               MDATA(3)=QDATA3                       BOX01220
                               MDATA(4)=QDATA4                       BOX01230
                               END                                   BOX01240
          DO K=1 TO PNO                                              BOX01250
           J=J+1                                                     BOX01260
           INDATA(J)=MDATA(K)*10                                     BOX01270
          END                                                        BOX01280
          J=0                                                        BOX01290
          PSW=1
          CALL LOAD(PCBP,FUNCT,DBNAME,DBKEY,MQY,GIGAN,INDATA,        BOX01320
                  HTITLE,HUNIT,HWGT)                                 BOX01330
          IF IFROMY=YEAR THEN                                        BOX01340
          DO   READ FILE(ADJFILE) INTO(INREC)
               GOTO DEAD
          END
          ELSE GOTO REREAD                                           BOX01350
      DEAD                                                           BOX01360
     END  /* KIKIN PROC END */                                       BOX01370
  PLAY  CLOSE FILE(SYSIN)                                            BOX01380
  LOAD   PROC(PCBP,FUNCT,DBNAME,DBKEY,MQY,GIGAN,INDATA,HTITLE,HUNIT, LOA00090
                 HWGT)                                               LOA00100
     DCL FUNCT  CHAR(6),                                             LOA00110
         HTITLE CHAR(30),                                            LOA00120
         HUNIT  CHAR(11),                                            LOA00130
         HWGT   FLOAT,                                               LOA00140
         INDATA(*) FLOAT(16)                                         LOA00150
   INCLUDE SDBDCL                                                    LOA00160
     DCL (SFYM,SEYM,RSFYM,RSEYM) FIXED,                              LOA00170
         RDNOSR FIXED,                                               LOA00180
         SDATA(360) FLOAT(16)                                        LOA00190
     IF MQY= M  THEN PNO=12                                          LOA00200
     ELSE                                                            LOA00210
     IF MQY= Q  THEN PNO=4                                           LOA00220
                ELSE PNO=1                                           LOA00230
```

```
     INDATA=ROUND(INDATA,0)                                              LOA00240
     DNOSR=(PNO-FROMM+1)+PNO*(TOY-FROMY-1)+TOM                           LOA00250
     DATA=0                                                              LOA00260
     PUT EDIT(DBNAME,DBKEY,MQY,FROMY,FROMM,TOY,TOM,                      LOA00270
             (INDATA(I) DO I=1 TO DNOSR),HTITLE,HUNIT,HWGT)              LOA00280
             (SKIP,X(5),A,X(1),A,X(1),A,X(2),4 F(2),X(5),(DNOSR) F(10),LOA00290
                   X(5),A(20),X(3),A(11),X(3),F(6,1))                    LOA00300
     PLEVEL=0                                                            LOA00310
     DBKEY=TRANSLATE(DBKEY, 0 ,    )                                     LOA00320
     SSAKEYO=DBNAME                                                      LOA00330
     CALL PLITDLI(FOUR,GU,PCBP,LEVELO,QSSAO)                             LOA00340
     IF M STAT CODE =      THEN DO                                       LOA00350
          ERRMSG= DBNAME NOT FOUND                                       LOA00360
          CALL ERRTN                                                     LOA00370
     END                                                                 LOA00380
     CALL PLITDLI(FOUR,GNP,PCBP,ATTRITO,ATTSSAO)                         LOA00390
     IF M STAT CODE =      THEN DO                                       LOA00400
          ERRMSG= ATTRIBUTE  0  NOT FOUND                                LOA00410
          CALL ERRTN                                                     LOA00420
          RETURN                                                         LOA00430
     END                                                                 LOA00440
     IF DBKEY = 000000000  THEN GOTO FINDLEV                             LOA00450
          TESTSTR=DBKEY                                                  LOA00460
          SUMKEYSZ=0                                                     LOA00470
          CALL LEVTEST(QSSA1)                                            LOA00480
          CALL LEVTEST(QSSA2)                                            LOA00490
          CALL LEVTEST(QSSA3)                                            LOA00500
          CALL LEVTEST(QSSA4)                                            LOA00510
          CALL LEVTEST(QSSA5)                                            LOA00520
          CALL LEVTEST(QSSA6)                                            LOA00520
  FINDLEV                                                                LOA00530
          P#=PLEVEL+4                                                    LOA00540
          CALL PLITDLI(P#,GU,PCBP,LEVELREC,                              LOA00550
                   QSSAO,QSSA1,QSSA2,QSSA3,QSSA4,QSSA5,QSSA6)            LOA00560
          YQMS=MQY                                                       LOA00570
          YQMLEV=PLEVEL                                                  LOA00580
          IF FUNCT= MODIFY  THEN CALL MODIFY                             LOA00590
                   ELSE CALL INSERT                                      LOA00600
          RETURN                                                         LOA00610
LEVTEST    PROC(QSSA)                                                    LOA00620
     DCL  QSSA CHAR(29),                                                 LOA00630
          QSSAKEY CHAR(9) DEFINED QSSA POSITION(20)                      LOA00640
          PLEVEL=PLEVEL+1                                                LOA00650
          SUMKEYSZ=SUMKEYSZ+KEYSZ(PLEVEL)                                LOA00660
          SUBSTR(TESTSTR,1,SUMKEYSZ)= 000000000                          LOA00670
          SUBSTR(QSSAKEY,1,SUMKEYSZ)=SUBSTR(DBKEY,1,SUMKEYSZ)            LOA00680
          SUBSTR(QSSAKEY,SUMKEYSZ+1,9-SUMKEYSZ)= 000000000               LOA00690
          IF TESTSTR= 000000000  THEN GOTO FINDLEV                       LOA00700
     END LEVTEST                                                         LOA00710
     INSERT   PROC                                                       LOA00720
          IF PLEVEL=0 THEN GOTO ISRTDATA                                 LOA00730
          IF PLEVEL=2 THEN QSSA=QSSA1                                    LOA00740
          IF PLEVEL=3 THEN QSSA=QSSA2                                    LOA00750
          IF PLEVEL=4 THEN QSSA=QSSA3                                    LOA00760
          IF PLEVEL=5 THEN QSSA=QSSA4                                    LOA00770
```

```
          IF PLEVEL=6 THEN QSSA=QSSA5                                   LOA00770
          UNQSSA= LEVEL     PLEVEL                                      LOA00780
          LEVELREC.KEY=DBKEY                                            LOA00790
          LEVELREC.TITLE=HTITLE                                         LOA00800
          IF PLEVEL=1 THEN                                              LOA00810
          CALL PLITDLI(FIVE,ISRT,PCBP,LEVELREC,QSSA0,UNQSSA)            LOA00820
              ELSE                                                      LOA00830
          CALL PLITDLI(FIVE,ISRT,PCBP,LEVELREC,QSSA,UNQSSA)             LOA00840
          ERRMSG= INSERT KEY  STAT CODE                                 LOA00850
          CALL ERRTN                                                    LOA00860
ISRTATT                                                                 LOA00870
          IF PLEVEL=1 THEN QSSA=QSSA1                                   LOA00880
          IF PLEVEL=2 THEN QSSA=QSSA2                                   LOA00890
          IF PLEVEL=3 THEN QSSA=QSSA3                                   LOA00900
          IF PLEVEL=4 THEN QSSA=QSSA4                                   LOA00910
          IF PLEVEL=5 THEN QSSA=QSSA5                                   LOA00920
          IF PLEVEL=6 THEN QSSA=QSSA6                                   LOA00920
          UNQSSA= ATTRIT     PLEVEL                                     LOA00930
          CALL PLITDLI(FIVE,GNP,PCBP,ATTRIT,QSSA,UNQSSA)                LOA00940
          IF M STAT CODE=     THEN DO                                   LOA00950
          ERRMSG= ALREADY EXIT UNIT WEIGHT                              LOA00960
          CALL ERRTN                                                    LOA00970
          GOTO ISRTDATA                                                 LOA00980
          END                                                           LOA00990
          ATTRIT.WGT=HWGT                                               LOA01000
          ATTRIT.UNIT=HUNIT                                             LOA01010
          CALL PLITDLI(FIVE,ISRT,PCBP,ATTRIT,QSSA,UNQSSA)               LOA01020
          ERRMSG= INSERT ATTRIBUTE   STAT CODE                          LOA01030
          CALL ERRTN                                                    LOA01040
ISRTDATA                                                                LOA01050
          IF MQY= A     MQY= B     MQY= C  THEN DO                      LOA01060
            FYEARIO.YR=FROMY    FROMM    TOY                            LOA01070
            FYEARIO.FDATA=INDATA(1)                                     LOA01080
            IF PLEVEL=0 THEN CALL PLITDLI(FIVE,ISRT,PCBP,FYEARIO,QSSA0,LOA01090
                        YQMSSA)                                         LOA01100
                    ELSE CALL PLITDLI(FIVE,ISRT,PCBP,FYEARIO,QSSA,      LOA01110
                        YQMSSA)                                         LOA01120
            ERRMSG= INSERT F DATA STAT CODE                             LOA01130
            CALL ERRTN                                                  LOA01140
            RETURN                                                      LOA01150
          END                                                           LOA01160
          CALL PLITDLI(FOUR,GNP,PCBP,DATAREC,YQMSSA)                    LOA01170
          IF M STAT CODE=     THEN DO                                   LOA01180
            ERRMSG= ALREADY EXIT DATA                                   LOA01190
            CALL ERRTN                                                  LOA01200
            RETURN                                                      LOA01210
          END                                                           LOA01220
          DO I=1 TO DNOSR                                               LOA01230
              DATA(I)=INDATA(I)                                         LOA01240
          END                                                           LOA01250
          INITY=FROMY                                                   LOA01260
          INITM=FROMM                                                   LOA01270
          ENDY=TOY                                                      LOA01280
          ENDM=TOM                                                      LOA01290
          IF PLEVEL=0 THEN CALL PLITDLI(FIVE,ISRT,PCBP,DATAREC,QSSA0,   LOA01300
```

```
                    YQMSSA)                                          LOA01310
            ELSE CALL PLITDLI(FIVE,ISRT,PCBP,DATAREC,QSSA,YQMSSA)    LOA01320
        ERRMSG= INSERT DATA  STAT CODE                               LOA01330
        CALL ERRTN                                                   LOA01340
        RETURN                                                       LOA01350
    END INSERT                                                       LOA01360
    MODIFY   PROC                                                    LOA01370
        IF M STAT CODE =      THEN DO                                LOA01380
            ERRMSG= DBKEY NOT FOUND                                  LOA01390
            CALL ERRTN                                               LOA01400
            RETURN                                                   LOA01410
        END                                                          LOA01420
      IF MQY= F  THEN DO                                             LOA01430
        FQSSALEV=PLEVEL                                              LOA01440
        FQSSALEK=PLEVEL                                              LOA01450
        FQSSAKEY=FROMY                                               LOA01460
        IF PLEVEL=0 THEN CALL PLITDLI(FIVE,GHU,PCBP,FYEARIO,QSSAO,   LOA01470
                    FQSSA)                                           LOA01480
        IF PLEVEL=1 THEN CALL PLITDLI(SIX,GHU,PCBP,FYEARIO,QSSAO,    LOA01490
                QSSA1,FQSSA)                                         LOA01500
        IF PLEVEL=2 THEN CALL PLITDLI(SEVEN,GHU,PCBP,FYEARIO,QSSAO,  LOA01510
                    QSSA1,QSSA2,FQSSA)                               LOA01520
        IF PLEVEL=3 THEN CALL PLITDLI(EIGHT,GHU,PCBP,FYEARIO,QSSAO,  LOA01530
                    QSSA1,QSSA2,QSSA3,FQSSA)                         LOA01540
        IF PLEVEL=4 THEN CALL PLITDLI(NINE,GHU,PCBP,FYEARIO,QSSAO,   LOA01550
                    QSSA1,QSSA2,QSSA3,QSSA4,FQSSA)                   LOA01560
        IF PLEVEL=5 THEN CALL PLITDLI(TEN,GHU,PCBP,FYEARIO,QSSAO,    LOA01570
                QSSA1,QSSA2,QSSA3,QSSA4,QSSA5,FQSSA)                 LOA01580
        IF PLEVEL=6 THEN CALL PLITDLI(ELEVEN,GHU,PCBP,FYEARIO,QSSAO,
                QSSA1,QSSA2,QSSA3,QSSA4,QSSA5,QSSA6,FQSSA)           LOA01580
        IF M STAT CODE =      THEN DO                                LOA01590
          ERRMSG= F RECORD NOT FOUND                                 LOA01600
          CALL ERRTN                                                 LOA01610
          RETURN                                                     LOA01620
        END                                                          LOA01630
        FYEARIO.FDATA=INDATA(1)                                      LOA01640
        CALL PLITDLI(THREE,REPL,PCBP,FYEARIO)                        LOA01650
        ERRMSG= REPLACE STAT CODE                                    LOA01660
        CALL ERRTN                                                   LOA01670
        RETURN                                                       LOA01680

    END                                                              LOA01690
    CALL PLITDLI(FOUR,GHNP,PCBP,DATAREC,YQMSSA)                      LOA01700
        IF M STAT CODE =      THEN DO                                LOA01710
          ERRMSG= ERROR MQY DATA NOT EXIT                            LOA01720
          CALL ERRTN                                                 LOA01730
          RETURN                                                     LOA01740
        END                                                          LOA01750
        SFYM=FROMY*PNO+FROMM                                         LOA01760
        SEYM=TOY*PNO+TOM                                             LOA01770
        RSFYM=INITY*PNO+INITM                                        LOA01780
        RSEYM=ENDY*PNO+ENDM                                          LOA01790
        IF SFYM RSEYM+1 THEN DO                                      LOA01800
           ERRMSG= DATA DISCONTINUE        FROMY    FROMM    TOY     LOA01810
                 TOM     -     INITY    INITM    ENDY    ENDM        LOA01820
           CALL ERRTN                                                LOA01830
```

```
            RETURN                                                      LOA01840
          END                                                           LOA01850
        IF SFYM RSFYM THEN DO                                           LOA01860
          IF SFYM+DNOSR RSFYM THEN DO                                   LOA01870
          ERRMSG= INCORRECT START     FROMY    FROMM    TOY    TOM      LOA01880
                  -     INITY    INITM    ENDY    ENDM                 LOA01890
           CALL ERRTN                                                   LOA01900
            RETURN                                                      LOA01910
          END                                                           LOA01920
          RDNOSR=RSEYM-RSFYM+1                                          LOA01930
          DO I=1 TO RDNOSR                                              LOA01940
              SDATA(I-SFYM+RSFYM)=DATA(I)                               LOA01950
          END                                                           LOA01960
          DO I=1 TO DNOSR                                               LOA01970
            SDATA(I)=INDATA(I)                                          LOA01980
          END                                                           LOA01990
          DATA=SDATA                                                    LOA02000
          INITY=FROMY                                                   LOA02010
          INITM=FROMM                                                   LOA02020
       END                                                              LOA02030
       ELSE DO                                                          LOA02040
          DO I = 1 TO DNOSR                                             LOA02050
            DATA(SFYM-RSFYM+I)=INDATA(I)                                LOA02060
          END                                                           LOA02070
       END                                                              LOA02080
          IF RSEYM SEYM THEN DO                                         LOA02090
            ENDY=TOY                                                    LOA02100
            ENDM=TOM                                                    LOA02110
       END                                                              LOA02120
            CALL PLIFDLI(THREE,REPL,PCBP,DATAREC)                       LOA02130
          ERRMSG= REPL STAT CODE                                        LOA02140
          CALL ERRTN                                                    LOA02150
          RETURN                                                        LOA02160
    END MODIFY                                                          LOA02170
    ERRTN   PROC                                                        LOA02180
          IF PSW=1 THEN
          PUT SKIP EDIT(ERRMSG,     ,M STAT CODE)(3 A)                  LOA02190
          M STAT CODE=
          RETURN                                                        LOA02200
    END                                                                 LOA02210
    END LOAD                                                            LOA02220
    FINAL  CLOSE FILE(ADJFILE),FILE(SYSPRINT)                           LOA02230
    END  /***** MAIN PROC END ******/                                   LOA02240
/*
//
```

프로그램 : READJUST

기능 : Forecasting 된 Data 의

DB Load 과정

```
//ZDBAIL71  JOB   CLASS=V,PRTY=13                                    REA00010
//          EXEC  IMSDLI,MBR=READJUST                                REA00020
//SYSLIB    DD DSN=IMSVS.SSSLIB,DISP=SHR                             REA00030
//SYSPRINT  DD SYSOUT=A                                              REA00040
//SYSIN   DD    *                                                    REA00050
* PROCESS GS,A,XREF,NEST,INCLUDE                                     REA00060
      BOXING  PROC(PCBP) OPTIONS(MAIN)                               REA00070
           DCL PCBP     POINTER                                      REA00080
           DCL 1 PCBNAME  BASED(PCBP),                               REA00090
                 2 M NAME           CHAR(8),                         REA00100
                 2 M SEF LEVEL      CHAR(2),                         REA00110
                 2 M STAT CODE      CHAR(2),                         REA00120
                 2 M PROC OPT       CHAR(4),                         REA00130
                 2 FILLER           FIXED BIN(31,0),                 REA00140
                 2 M SEG NAME       CHAR(8),                         REA00150
                 2 M LEV KFB        FIXED BIN(31,0),                 REA00160
                 2 M NO SENSEG      FIXED BIN(31,0),                 REA00170
                 2 M KEY KFB        CHAR(50)                         REA00180
           DCL LEVEL     PIC 99 ,                                   REA00190
               (FSTY,FSTM,ENDM,ENDY)  PIC  99 ,                     REA00200
              FUNCT       CHAR(6),                                  REA00210
              DBNAME      CHAR(5),                                  REA00220
              DBKEY       CHAR(9),                                  REA00230
              MQY         CHAR(1),                                  REA00240
              ZEROCHAR    CHAR(9) INIT( C00000000 ),                REA00250
              INKEY(50) CHAR(9),                                    REA00260
              D10(12)  PIC 999999V.99 ,                             REA00270
              INDATA(12) FLOAT(16),                                 REA00280
              HTITLE      CHAR(30) INIT(    ),                      REA00290
              HUNIT       CHAR(11) INIT(    ),                      REA00300
              HWGT        FLOAT    INIT(0),                         REA00310
              1 GIGAN,                                              REA00320
                2 FROMY  PIC 99 ,                                   REA00330
                2 FROMM  PIC 99 ,                                   REA00340
                2 TOY    PIC 99 ,                                   REA00350
                2 TOM    PIC 99                                     REA00360
           DCL  ADJFILE FILE RECORD INPUT SEQUENTIAL                REA00370
           DCL 1 SINREC,                                            REA00380
                 2 SFILLER1      CHAR(2),                           REA00390
                 2 SIFROMY       PIC 99 ,                           REA00400
                 2 SFILLER2      CHAR(4),                           REA00410
                 2 SMDATA(12)    PIC 999999V.99 ,                   REA00420
                 2 SFILLER3      CHAR(144)                          REA00430
           DCL 1 SQINREC,                                           REA00440
                 2 SFIL1         CHAR(2),                           REA00450
                 2 SQFROMY       PIC 99 ,                           REA00460
                 2 SFIL2         CHAR(13),                          REA00470
                 2 SQDATA1       PIC 999999V.99 ,                   REA00480
                 2 SFIL3         CHAR(15),                          REA00490
                 2 SQDATA2       PIC 999999V.99 ,                   REA00500
                 2 SFIL4         CHAR(15),                          REA00510
                 2 SQDATA3       PIC 999999V.99 ,                   REA00520
                 2 SFIL5         CHAR(15),                          REA00530
                 2 SQDATA4       PIC 999999V.99 ,                   REA00540
                 2 SFIL6         CHAR(162)                          REA00550
```

```
      DCL  YEAR      PIC 99 ,                                          REA00560
           MONTH     PIC 99 ,                                          REA00570
           STARTY    PIC 99 ,                                          REA00580
           STARTM    PIC 99 ,                                          REA00590
           LOADED    CHAR(4),                                          REA00600
           COUNT     FIXED,                                            REA00610
           SP        FIXED,                                            REA00620
           PNO       FIXED,                                            REA00630
           I         FIXED,                                            REA00640
           J         FIXED,                                            REA00650
           PSW       PIC 9  INIT(1),                                   REA00660
           K         FIXED,                                            REA00670
           FACTOR    CHAR(6),                                          REA00680
           SELECT    CHAR(6)                                           REA00690
      DCL ONSOURCE BUILTIN                                             REA00700
      ON ENDFILE(ADJFILE)  GO TO FINAL                                 REA00710
      ON ENDFILE(SYSIN) GOTO FINAL                                     REA00720
      ON CONVERSION BEGIN                                              REA00730
       PUT SKIP LIST( SOURCE =      SOURCE)                            REA00740
       ONSOURCE=(16) 0                                                 REA00750
      END                                                              REA00760
      OPEN FILE(SYSPRINT) LINESIZE(132)                                REA00770
      OPEN FILE(ADJFILE)                                               REA00780
GAME                                                                   REA00790
      GET LIST(YEAR,MONTH,SELECT,LOADED)                               REA00800
      IF SELECT= INSERT  THEN FUNCT= INSERT                            REA00810
                         ELSE FUNCT= MODIFY                            REA00820
      INKEY=                                                           REA00830
      GET LIST(DBNAME,MQY,SP)                                          REA00840
      FSTM=1                                                           REA00850
      IF MQY = Q  THEN DO                                              REA00860
      PNO=12                                                           REA00870
      MQY= M                                                           REA00880
      END                                                              REA00890
      ELSE PNO=4                                                       REA00900
      ENDM=PNO                                                         REA00910
      GET SKIP LIST((INKEY(I) DO I=1 TO SP))                           REA00920
      I=0                                                              REA00930
      J=0                                                              REA00940
AREAD                                                                  REA00950
     /*** THIS IS A PROCESSING FOR DATA LOAD INTO DATA BASE ***/       REA00960
       I=I+1                                                           REA00970
       DBKEY=INKEY(I)                                                  REA00980
INREAD                                                                 REA00990
      READ FILE(ADJFILE) INTO(SINREC)                                  REA01000
      IF MQY= M  THEN DO                                               REA01010
                      FROMY=SIFROMY                                    REA01020
                      FROMM=1                                          REA01030
                      TOY=SIFROMY                                      REA01040
                     IF SIFROMY=YEAR THEN TOM=MONTH                    REA01050
                      ELSE  TOM=12                                     REA01060
                      END                                              REA01070
                 ELSE DO                                               REA01080
                     IF SQFROMY YEAR THEN GOTO INREAD                  REA01090
                      FROMY=SQFROMY                                    REA01100
```

```
                          FROMM=1                                          REA01110
                          TOY=SQFROMY                                      REA01120
                         IF SQFROMY=YEAR THEN TOM=MONTH/3                  REA01130
                          ELSE TOM=4                                       REA01140
                         SMDATA(1)=SQDATA1                                 REA01150
                         SMDATA(2)=SQDATA2                                 REA01160
                         SMDATA(3)=SQDATA3                                 REA01170
                         SMDATA(4)=SQDATA4                                 REA01180
                          END                                              REA01190
        INDATA=0                                                           REA01200
        DO K=1 TO PNO                                                      REA01210
        INDATA(K)=SMDATA(K)*100                                            REA01220
        END                                                                REA01230
        PSW=1                                                              REA01240
        IF LOADED= PRTS  THEN DO                                           REA01250
        PUT SKIP EDIT(DBNAME,DBKEY,(INDATA(I) DO I=1 TO 12))               REA01260
            (X(5),A(5),X(3),A(5),12 F(8,2))                                REA01270
        END                                                                REA01280
        ELSE                                                               REA01290
        CALL LOAD(PCBP,FUNCT,DBNAME,DBKEY,MQY,GIGAN,INDATA,                REA01300
                HTITLE,HUNIT,HWGT)                                         REA01310
       IF SIFROMY=YEAR THEN                                                REA01320
          GOTO AREAD                                                       REA01330
        ELSE GOTO INREAD                                                   REA01340
       INCLUDE LOADS                                                       REA01350
     FINAL  CLOSE FILE(ADJFILE),FILE(SYSPRINT),FILE(SYSIN)                 REA01360
     END  /***** MAIN PROC END ******/                                     REA01370
/*                                                                         REA01380
//                                                                         REA01390
```

프로그램 : X－11, ARIMA SOURCE LIST (NEW VERSION)

```
      CALL GENFOR
C ---- THIS SUBROUTINE READS CONTROL CARDS AND EDITS THEM.
    1 CALL EDITOR(IFATAL)
      IF (IFATAL.NE.0) GO TO 2
C ---- CALL FORECASTING AND SEASONAL ADJUSTMENT SUBROUTINES
      CALL X11ARI
      GO TO 1
    2 STOP
      END
      SUBROUTINE GENFOR
C ---- THIS SUBROUTINE PRINTS THE HEADINGS FOR THE VARIOUS FILES AND
C ---- INITIALIZES VALUES.
C ---- THE UNIT MT IS THE CONTROL CARD INPUT FILE
C ----          MT1 IS THE MAIN PRINTOUT
C ----          MT2 IS THE LOG
C ----          MP IS CARD OUTPUT FILE
C ----          NG IS THE FILE CONTAINING ALL THE Q STATISTICS
C ----          NFORM CONTAINS ALL THE F TABLES FOR THE RUN
C ----          NR IS THE TAPE INPUT FILE AND IS DEFINED IN THE ROUTINE
C ----                  INPUT
      REAL*8 TITLE,SERNO
      COMMON /UNITS/ MT,MT2,MT1,MP,NG,NFORM
      COMMON /ROTE / TITLE(10),SERNO,KPAGE,IRUN,NEWPAG
      COMMON /LARGE/ BIG
      COMMON /MQ11 / IMESSG,IAGR
      DATA NEWPG/1H1/
      IMESSG = 0
      IAGR = 0
      MT = 5
      MT1 = 6
      MT2 = 6
      MP = 6
      NG= 6
      NFORM= 6
      IRUN=0
      NEWPAG = NEWPG
      BIG = 10.0 E 16
      WRITE(MT2,150)
      WRITE(MT1,160)
  150 FORMAT(1H1,34X,68HLOG FOR X-11-ARIMA MONTHLY AND QUARTERLY SEASONA
     1L ADJUSTMENT PROGRAM,//,32X,17(4H*-*-),1H*,// )
  160 FORMAT(1H )
    1 FORMAT(1H1)
      WRITE(NG,1)
      WRITE(NG,140)
  140 FORMAT(/,16H TYPE OF  SERIES,6X,10HADDITIONAL,19X,12HSERIES TITLE,
     1        15X,36HQUALITY CONTROL STATISTICS  CHECK IF,/,
     2        16H ADJUST.  IDENT.,6X,11HIDENTIFIERS,48X,6HFAILED,7X,
     3        6HPASSED,5X,9HPUBLISHED,//)
      RETURN
      END
      SUBROUTINE EDITOR(IFATAL)
      INTEGER*2 KCOL(80),BLK,ST,T1,ZM,H1,Q,Z,XOPT,IEF4ST(80),ALPHA(9),
     1          BETA(9),TYPE(6),T,X,ARCAIM(80),TRAN,LOG,POW,XCDIM(80),I
     2OPT,LABL
```

```
      INTEGER*4 CHK(10)                                                 00000560
      REAL*8    PRTS(6),CHTS(4),SERNO,TITLE,TIT(10),KSER,IDENT,MQT(2),  00000570
     1          AVG(5),PCDIF,RAD,MQ,R(4),PCD(6),X11(4)                  00000580
      DIMENSION QUMO(6),MALO(12),RUNS(10),NUM(12),KTC(5),NO(11),MQCDS(2)00000590
      COMMON /UNITS/ MT,MT2,MT1,MP,NG,NFORM                             00000600
      COMMON /ROTE / TITLE(10),SERNO,KPAGE,IRUN,NEWPG                   00000610
      COMMON /WORK / TEMP(372)                                          00000620
      COMMON /F4STC/ IEF4ST,ARCAIM,JIFORC,XCDIM,LFMT,LFORM(20),XOPT     00000630
      COMMON /ARIM1/ PV(10),PW,ADD,IPDQ(7),IORDER(10),NLOG,MIT,NDP,NSEA 00000640
      COMMON /PRIOR/ KSER,KFORM(20),KFMT                                00000650
      COMMON /TRADE/ DWT(7),D(7),SIGM,LOPT,KDWOPT,LCYR,LAYR,LFDC,LFDR   00000660
      COMMON /INPT1/ SERIES(372),ORIG(372)                              00000670
      COMMON /INPT2/ SPRIOR(372)                                        00000680
      COMMON /OPT1 / IFORM(20),IPLUS,IFMT                               00000690
      COMMON /OPT2 / KDEC,ICARD,KHCNOP,KHCARO(20),KHCARC(20)            00000700
      COMMON /OPT3 / IDENT(9),LDEC(9),JFORM(20),JFMT                    00000710
      COMMON /OPT4 / NUMB(9),IHOLD(9),NOP                               00000720
      COMMON /OPT5 / MULADD,IYRT,MA,IWT,W                               00000730
      COMMON /OPT6 / SIGML,SIGMU                                        00000740
      COMMON /OPT7 / RATI(36),RATIS,LTERM,MTYPE,LTER(12)                00000750
      COMMON /OPT8 / RATIC,KTCOPT,NTERM                                 00000760
      COMMON /OPT9 / LFDA,LYR,LSTMO,LSTYR,LLDA,KFULSM,KPROPT,KCHOPT,    00000770
     1               KEXOPT,NY,IFORC,LLDAF,LLAF,LFD1,LLD1,NSOPT,IDEC,IAG00000780
     2               ,JAREX,KEBACK                                      00000790
      COMMON /MQ2  / KPART,KSWV,KSECT,KPRNT,MCD                         00000800
      COMMON /MQ3  / PCDIF(3),RAD(2),MQ,QM(3),OUT,MQCD                  00000810
      COMMON /MQ11 / IMESSG,IAGR                                        00000820
      COMMON /KCJH/ LENGTH,IFTNY                                        00000830
      DATA QUMO /4H MON,4HTHLY,1H ,4HQUAR,4HTERL,1HY/                   00000840
      DATA MALO /4HMULT,4HIPLI,4HCATI,2HVE,4H    ,4H  AD,4HDITI,2HVE,    00000850
     1           4H   L,4HOGAR,4HITHM,2HIC/                             00000860
      DATA PRTS /8HSTANDARD,8H    FULL,8H    LONG,8H   SHORT,8HANALYSIS,00000870
     1           8H   BRIEF/                                            00000880
      DATA CHTS /8HSTANDARD,8H      NO,8H     ALL,8H     ALL/           00000890
      DATA BLK,ST /2H  ,2H* /                                           00000900
      DATA T1,ZM,Z,Q,X /1HT,1HM,1HZ,1HQ,1HX/                            00000910
      DATA TYPE /2HA ,2HB ,2HC ,2HD ,2HE ,2HF /                         00000920
      DATA RUNS /4HSEAS,4HONAL,4H ADJ,4HUSTM,4HENT ,4H   S,4HUMMA,      00000930
     1           4HRY M,4HEASU,4HRES /                                  00000940
      DATA NUM  /3HST/,3HND/,3HRD/,3HTH/,3HTH/,3HTH/,3HTH/,3HTH/,3HTH/, 00000950
     1           3HTH/,3HTH/,3HTH//                                     00000960
      DATA MQT  /7H  MONTH,7HQUARTER/                                   00000970
      DATA AVG  /6H  3X3 ,6H  3X5 ,6H  3X9 ,8HSTABLE  ,                 00000980
     1            8HVARIABLE/                                           00000990
      DATA KTC  / 9,13,23,5,7/                                          00001000
      DATA BK,OT/3H   ,3HOUT/                                           00001010
      DATA R/6HRATIOS,5H     ,6HDIFFER,5HENCES/                         00001020
      DATA PCD/6H PER C,6HENT CH,5HANGE ,6H   DIF,6HFERENC,5HES   /     00001030
      DATA ND/204,209,214,215,314,315,409,504,1316,1318,1104/           00001040
      DATA X11/6H     X,5H-11  ,6HX-11 A,5HRIMA /                       00001050
      DATA MQCDS/3HMCD,3HQCD/                                           00001060
      DATA LOG,POW/1HL,1HP/                                             00001070
      DATA FSTR/4H****/                                                 00001080
      EQUIVALENCE (TEMP(1),ALPHA(1)),(TEMP(6),BETA(1)),                 00001090
     1            (TEMP(11),TIT(1)),(TEMP(31),KCOL(1))                  00001100
```

```
      DO 86 I=1,10
      CHK(I)=FSTR
   86 CONTINUE
      ICHK=1
      KPAGE = 0
        KHCNOP=0
C --- READ CONTROL CARD (AUGUST 1976).
    1 READ (MT,1000,ERR=2000,END=2001) IEF4ST,
     1  H1,IFMT,SERNO,LFDA,LYR,LSTMO,LSTYR,IDEC,IPLUS,ICARD,KDEC,MULADD,
     2  IYRT,KFULSM,KPROPT,KCHOPT,SIGML,SIGMU,LTERM,KTCOPT,IFORC,KEXOPT,
     3  KFMT,KSER,DWT,XOPT,KDWOPT,LCYR,LAYR,SIGM
C --- FORMAT FOR CONTROL CARD.
 1000 FORMAT( 80A1,
     1  T1,A1,I1,A8,4I2,9I1,2F2.1,5I1,A8,7F4.3,A1,I1,2I2,F2.1)
      CHK(ICHK)=H1
      ICHK=ICHK+1
      LFMT=IFMT
      IXOPT=XOPT
C --- CHANGE TO KPROPT, THE PRINT OPTION VARIABLE
      KPROPT=6-KPROPT
      IF(KPROPT.EQ.6)KPROPT=0
C --- DEFINE INDICATOR FOR ELIMINATION OF BACKCASTS
      KEBACK=0
      IF(IFORC.LE.4) GO TO 92
      KEBACK=1
      IFORC=IFORC-4
C --- DEFINE INDICATOR FOR ARIMA REPLACEMENT OF EXTREMES
   92 JAREX=0
      IF(IFORC.LE.2) GO TO 93
      JAREX=1
      IFORC=IFORC-2
   93 JIFORC=IFORC
  101 NOAG = 0
C --- TEST FOR AGGREGATE RUN.
      IF (H1.NE.TYPE(3)) GO TO 2
      CALL AGR1(LFDA,LLDA,IAGR)
      IF (IAGR.EQ.1) GO TO 1
      IF (IAGR.EQ.3) GO TO 1200
C --- ERROR AGGREGATE NOT PERFORMED.
      NOAG = 1
      GO TO 1200
C --- TEST FOR AN END-OF-RUN CARD.
    2 IF (H1.EQ.Z) GO TO 2004
      MQ = MQT(1)
      DO 102 I = 1,3
  102 QM(I) = QUMO(I)
      NY = 12
      MQCD = MQCDS(1)
      IF (H1.EQ.ZM) GO TO 1200
      MQCD = MQCDS(2)
      KDWOPT = 0
      LOPT = 0
      NY = 4
      MQ = MQT(2)
      DO 202 I = 4,6
```

```
  202 QM(I-3) = QUMO(I)                                                 00001660
      IF (H1.NE.Q) GO TO 2002                                           00001670
C  READ IN EXTRA CONTROL CARD IF                                        00001680
C             A-CENTERED 24(8)-TERM MOVING AVERAGE FOR MONTHLY(QUARTERLY)00001690
C               SERIES IS WANTED                                        00001700
C             B-LENGTH-OF-MONTH VARIATION IS TO BE INCLUDED IN          00001710
C               TRADING-DAY FACTORS INSTEAD OF IN SEASONAL FACTORS      00001720
C             C-COMPOSITING USING SUBTRACTION, MULTIPLICATION, OR       00001730
C               DIVISION IS WANTED                                      00001740
C             D-A CONSTANT IS TO BE USED WHEN COMPOSITING               00001750
C             E-IF TYPE OF MOVING AVERAGE IS TO BE SELECTED INDIVIDUALLY 00001760
C               FOR SOME OR ALL OF THE MONTHS                           00001770
 1200 IFATAL=0                                                          00001780
      IWT=0                                                             00001790
      LOPT=0                                                            00001800
      IAG=0                                                             00001810
      W=0.                                                              00001820
      DO 84 I=1,12                                                      00001830
   84 LTER(I)=0                                                         00001840
      IF(XOPT.EQ.BLK)GO TO 3                                            00001850
      IF(XOPT.EQ.X)GO TO 19                                             00001860
      IFATAL=IFATAL+1                                                   00001870
      WRITE(MT2,38)                                                     00001880
   38 FORMAT('0ERROR IN THE MAIN CONTROL CARD. COLUMN 73 MUST CONTAIN EI00001890
     1THER A BLANK OR AN X.')                                           00001900
      GO TO 3                                                           00001910
   19 READ(MT,6666)XCDIM,H2,IWT,LOPT,IAG,W,(LTER(I),I=1,12)             00001920
 6666 FORMAT (80A1,T1,A1,3I1,F5.3,12I1)                                 00001930
      CHK(ICHK)=H2                                                      00001940
      ICHK=ICHK+1                                                       00001950
    3 DO 4 I = 1,80                                                     00001960
    4 KCOL(I) = BLK                                                     00001970
      IERROR = 0                                                        00001980
      IF (NOAG.EQ.1) IFATAL = 1                                         00001990
      ICOUNT = 0                                                        00002000
      LPRIOR = 0                                                        00002010
C --- TEST FOR ARIMA OPTION.                                            00002020
      IF (IFORC-2) 204,104,2035                                         00002030
C --- READ ARIMA MODEL SELECTION CARD.                                  00002040
  104 READ(MT,1006,ERR=2057,END=2052)ARCAIM,TRAN,IPDQ,PW,ADD,MIT,PV,IORD00002050
     1ER                                                                00002060
 1006 FORMAT(80A1,T1,A1,7I1,F3.2,F7.2,I2,10F4.2,10I2)                   00002070
      CHK(ICHK)=TRAN                                                    00002080
      ICHK=ICHK+1                                                       00002090
      NLOG=4                                                            00002100
      IF(TRAN.EQ.LOG)NLOG=1                                             00002110
      IF(TRAN.EQ.BLK)NLOG=0                                             00002120
      IF(TRAN.EQ.POW)NLOG=2                                             00002130
      IF(NLOG.LE.3)GO TO 204                                            00002140
      WRITE(MT2,18)                                                     00002150
   18 FORMAT('0ERROR IN THE ARIMA CARD. THE FIRST COLUMN MUST CONTAIN EI00002160
     1THER A BLANK, AN L, OR A P.')                                     00002170
      IFATAL=IFATAL+1                                                   00002180
  204 IF (H1.EQ.TYPE(3)) GO TO 5                                        00002190
C --- TEST FOR FORMAT OPTIONS.                                          00002200
```

```
      IF (IFMT.GT.7) GO TO 2005
      IF (IFMT.NE.1.AND.IFMT.LT.6) GO TO 5
      READ(MT,1001,ERR=2006,END=2007) IFORM,T
 1001 FORMAT(20A4,T1,A1)
      CHK(ICHK)=T
      ICHK=ICHK+1
      DO 85 J=1,20
      LFORM(J)=IFORM(J)
   85 CONTINUE
      IF (T.EQ.T1) GO TO 2059
C --- CHECK PRIOR ADJUSTMENT
    5 IF (KFMT.GT.8) GO TO 2047
      IF (IFATAL.NE.0) GO TO 105
      KKFMT = KFMT - 1
      IF (KFMT.NE.2.AND.KFMT.LT.7) GO TO 105
      READ(MT,1001,ERR=2049,END=2051) KFORM,T
      CHK(ICHK)=T
      ICHK=ICHK+1
      IF (T.EQ.T1) GO TO 2060
C --- TEST FOR CARD PUNCH OPTION.
  105 IF (ICARD.EQ.0) GO TO 16
      IF (ICARD.NE.1) GO TO 2008
      IF (IFATAL.NE.0) GO TO 16
      READ(MT,1002,ERR=2009,END=2052)KHCARD,NOP,JFMT,(ALPHA(I),NUMB(I),
     1 BETA(I),IDENT(I),LDEC(I),I=1,6)
        KHCNOP=NOP
 1002 FORMAT(20A4,T1,2I1,6(A1,I2,A1,A8,I1))
      CHK(ICHK)=NOP
      ICHK=ICHK+1
      IF (NOP.LT.7) GO TO 7
      IF (NOP.GE.10) GO TO 2010
    6 READ(MT,1003,ERR=2009,END=2052)KHCARC,(ALPHA(I),NUMB(I),BETA(I),
     1 IDENT(I),LDEC(I),I=7,NOP)
 1003 FORMAT(20A4,T1,3(A1,I2,A1,A8,I1))
      CHK(ICHK)=ALPHA(1)
      ICHK=ICHK+1
    7 IF (JFMT.GT.7) GO TO 2011
      IF (JFMT.NE.1.AND.JFMT.LT.6) GO TO 8
      READ (MT,1001,ERR=2012,END=2052) JFORM,T
      CHK(ICHK)=T
      ICHK=ICHK+1
      IF (T.EQ.T1) GO TO 2061
    8 IF (NOP.EQ.0) GO TO 2013
      DO 15 I = 1,NOP
      IF (NUMB(I).GT.20.OR.NUMB(I).LE.0) GO TO 10
      IHOLD(I) = 0
      DO 9 J = 1,6
      IF (ALPHA(I).EQ.TYPE(J)) GO TO 11
    9 CONTINUE
   10 NUMS = NUMB(I)-NUMB(I)/100*100
      WRITE(MT2,3014) ALPHA(I),NUMS,BETA(I)
 3014 FORMAT(1H0,'ERROR IN TABLE NO. ',A1,I2,A1,' TABLE CAN NOT BE PUNCH
     1ED.')
      IHOLD(I) = 1
      GO TO 15
```

```
   11 NUMB(I) = J*100+NUMB(I)                                           00002760
      IF (BETA(I).EQ.BLK) GO TO 14                                      00002770
      DO 12 J = 1,3                                                     00002780
      IF (BETA(I).EQ.TYPE(J)) GO TO 13                                  00002790
   12 CONTINUE                                                          00002800
      GO TO 10                                                          00002810
   13 NUMB(I) = NUMB(I)+1000*J                                          00002820
   14 DO 114 J = 1,11                                                   00002830
      IF (NUMB(I).EQ.NO(J)) GO TO 10                                    00002840
  114 CONTINUE                                                          00002850
      IF (LDEC(I).LT.6) GO TO 15                                        00002860
      LDEC(I) = 2                                                       00002870
      WRITE(MT2,3015)                                                   00002880
 3015 FORMAT(1H0,'DECIMALS ON CARD OUTPUT MUST NOT EXCEED 5. REDEFINED T00002890
     10 BE 2.')                                                         00002900
   15 CONTINUE                                                          00002910
C --- GENERATE FORMAT FOR THE CARD OUTPUT                               00002920
      CALL FMTS(JFORM,JFMT,2,NY)                                        00002930
C --- CHECK DATES AND LENGTH OF THE SERIES.                             00002940
   16 IF (LFDA.EQ.0) GO TO 2016                                         00002950
      IF (LFDA.GT.NY) GO TO 2017                                        00002960
   17 IF (LSTMO.EQ.0) GO TO 2018                                        00002970
      IF (LSTMO.GT.NY) GO TO 2019                                       00002980
      LLDA = (LSTYR - LYR) * NY + LSTMO                                 00002990
      IF (IFORC.NE.0) LLAF=LLDA+12                                      00003000
      IF (IFORC.EQ.0) LLAF=LLDA                                         00003010
      LENGTH = LLDA - LFDA + 1                                          00003020
      ITEST = 3 * NY                                                    00003030
      IF (LENGTH.LT.ITEST) GO TO 2020                                   00003040
      ITEST = 30 * NY                                                   00003050
      IF (LLDA.GT.ITEST) GO TO 2021                                     00003060
C --- CHECK DECIMALS ON INPUT CARDS.                                   00003070
   20 IF (IDEC.GT.5) GO TO 2023                                         00003080
      IF (IDEC.EQ.0) IDEC = 6                                           00003090
C --- GENERATE FORMAT FOR INPUT OF SERIES TO BE ADJUSTED                00003100
      CALL FMTS (IFORM,IFMT,IDEC,NY)                                    00003110
C --- GENERATE FORMAT FOR READING PRIOR ADJUSTMENT SERIES               00003120
      JDEC = IDEC                                                       00003130
      IF (MULADD.NE.1) JDEC = 3                                         00003140
      IF (KFMT.NE.0) CALL FMTS(KFORM,KKFMT,JDEC,NY)                     00003150
C --- CHECK TAPE INPUT.                                                 00003160
   21 IF (IPLUS.GT.3) GO TO 2024                                        00003170
C --- CHECK DECIMALS ON OUTPUT TABLES.                                  00003180
   22 IF (KDEC.GT.5) GO TO 2025                                         00003190
C --- CHECK TYPE OF ADJUSTMENT.                                         00003200
   23 IF (MULADD-2) 24,27,2026                                          00003210
   24 IF (MULADD) 25,25,26                                              00003220
   25 IT2 = 1                                                           00003230
      GO TO 28                                                          00003240
   26 IT2 = 5                                                           00003250
      GO TO 28                                                          00003260
   27 IT2 = 9                                                           00003270
C --- CHECK TYPE OF PRINTOUT.                                           00003280
   28 IF (KPROPT.GT.5) GO TO 2027                                       00003290
      IF (KPROPT - 2) 128,328,428                                       00003300
```

```
  128 IF (KPROPT.EQ.1) GO TO 228
      KPRNT = 2
      GO TO 29
  228 KPRNT = 8
      GO TO 29
  328 KPRNT = 5
      GO TO 29
  428 KPRNT = 0
C --- CHECK CHARTS OPTION.
   29 IF (KCHOPT.GT.3) GO TO 2028
C --- CHECK ADJUSTMENT OF YEARLY TOTALS OPTION.
   30 IF (IYRT.GT.1) GO TO 2029
      IF (IYRT.EQ.0) GO TO 31
      IYRS = LSTYR - LYR + 1
      IF (LLDA.NE.1) IYRS = IYRS - 1
      IF (LSTMO.NE.NY) IYRS = IYRS - 1
      IF (IYRS.LT.5) GO TO 2050
C --- CHECK TYPE OF PROGRAM OPTION.
   31 IF (KFULSM.GT.1) GO TO 2030
C --- CHECK SIGMA LIMITS FOR GRADUATING EXTREMES.
   32 IF (SIGML) 2031,33,34
   33 SIGML = 1.5
   34 IF (SIGMU) 2032,35,36
   35 SIGMU = 2.5
   36 IF (SIGMU-SIGML) 2033,37,37
C --- CHECK VARIABLE SEASONAL SECTION OPTION.
   37 IF (LTERM.GT.5) GO TO 2034
C --- CHECK STRIKE OPTION.
   39 IF (KEXOPT.GT.1) GO TO 2036
   40 IF (NY.EQ.12) GO TO 41
      IF (KTCOPT.GT.2) GO TO 2037
      GO TO 57
   41 IF (KTCOPT.GT.3) GO TO 2037
C --- CHECK LENGTH-OF-MONTH OPTION
      IF (LOPT-1) 43,42,2038
   42 LOPT = 4
C --- CHECK PRIOR WEIGHTS.
   43 DO 44 I = 1,7
      IF (DWT(I).LT.0.0) GO TO 2039
   44 CONTINUE
  144 IF (KDWOPT.GT.3) GO TO 2040
   45 IF (MULADD.NE.1) GO TO 47
      DO 46 I = 1,7
      IF (DWT(I).GT.0.0) GO TO 2041
   46 CONTINUE
      IF (LOPT.EQ.4) GO TO 2042
   47 IF (KDWOPT.EQ.0) GO TO 58
C --- CHECK TRADING DAY OPTIONS.
      IF (SIGM) 2043,48,49
   48 SIGM = 2.5
   49 LLYR=LYR+(LLDA-1)/12
      IF (LCYR) 2044,52,50
   50 IF (LYR-LCYR) 51,52,2044
   51 IF (LLYR-LCYR) 2044,53,53
   52 LCYR=LYR
```

```
      LFDC=LFDA
      GO TO 153
   53 LFDC=(LCYR-LYR)*12+1
  153 IF (LAYR) 2045,56,54
   54 IF (LYR-LAYR) 55,56,2045
   55 IF (LLYR-LAYR) 2045,57,57
   56 LAYR=LYR
      LFDR=LFDA
      GO TO 58
   57 LFDR=(LAYR-LYR)*NY+1
C --- READ TITLE CARD.
   58 IF (IFATAL.NE.0) GO TO 59
      READ(MT,1004,ERR=2053,END=2054) T,NOTC,TITLE
 1004 FORMAT( A1,I1,9A8,A6)
      CHK(ICHK)=T
      IF (T1.NE.T) GO TO 2046
   59 IFE = IFATAL+IERROR
      IF (IFE.EQ.0) GO TO 60
      WRITE(MT2,4000) SERNO,IERROR,IFATAL
 4000 FORMAT(1H0,'IN SERIES NO. ',A8,' THERE WERE ',I2,' NON-FATAL ERROR
     1S AND ',I2,' FATAL ERRORS.')
      WRITE(MT2,4001) IEF4ST
 4001 FORMAT(1H0,6X,'CARD IMAGE: ',80A1)
      WRITE(MT2,4002) KCOL
 4002 FORMAT(     7X,'     ERRORS: ',80A1)
      WRITE(MT2,87)CHK
   87 FORMAT(1H0,'HERE IS A LIST OF FIRST COLUMNS OF CONTROL CARDS THAT
     1WERE READ.',/,10A4)
      IF (IFATAL.NE.0) GO TO 2048
      IF (IERROR.GE.5) GO TO 2003
   60 I1 = 1
      IF (IFORC.NE.0) I1 = 3
      LLDAF = LLDA
      IRUN = IRUN+1
      IF (IRUN.EQ.1) GO TO 61
      WRITE(MT1,5000)
 5000 FORMAT(1H1)
   61 IT3 = 1
      IF (KFULSM.EQ.0) GO TO 62
      IT3 = 6
   62 JT2 = IT2 + 3
      JT3 = IT3 + 4
      NP1 = KPROPT + 1
      NC1 = KCHOPT + 1
      MA = MULADD
C --- WRITE LOG ENTRY.
      WRITE(MT2,5001) SERNO,LFDA,NUM(LFDA),LYR,LSTMO,NUM(LSTMO),LSTYR,
     1                (MALO(I),I=IT2,JT2),(RUNS(J),J=IT3,JT3),PRTS(NP1),
     2                QM
 5001 FORMAT(1H0,5X,A8,2X,I2,A3,I2,3H - ,I2,A3,I2,2X,3A4,A2,1X,5A4,2H, ,
     1  A8,11H PRINTOUT, ,2A4,A1, 9H PROGRAM.)
C --- WRITE TITLE PAGE.
      WRITE(MT1,5021)
 5021 FORMAT(/,53X,17HSTATISTICS CANADA)
      WRITE(MT1,5002) X11(I1),X11(I1+1),QM,(RUNS(J),J=1,5)
```

```
 5002 FORMAT(/,38X,A6,A5,2A4,A1,1X,5A4,6HMETHOD,////,                    00004410
     1  35X,'THIS METHOD MODIFIES THE X-11 VARIANT ',                    00004420
     2  'OF CENSUS METHOD II',/,34X,' BY J. SHISKIN,',                   00004430
     3  ' A.H. YOUNG AND J.C. MUSGRAVE OF FEBRUARY, 1967.',/,            00004440
     4  34X,'THE MODIFICATIONS MADE ARE ',                               00004450
     5  'BASED ON THE METHODOLOGICAL ',                                  00004460
     6  ' RESEARCH',/,34X,' DEVELOPED BY',                               00004470
     7  ' ESTELA BEE DAGUM, CHIEF OF ',                                  00004480
     8  'THE SEASONAL ADJUSTMENT',/,                                     00004490
     9  35X,' AND TIME SERIES STAFF',                                    00004500
     A  ' OF STATISTICS CANADA. SEPTEMBER, 1979.'////)                   00004510
      WRITE(MT1,5003) TITLE,SERNO                                        00004520
 5003 FORMAT(10X,14HSERIES TITLE- ,10A8,10X,11HSERIES NO. ,A8)           00004530
      IF (NOTC.LE.1) GO TO 64                                            00004540
      DO 63 I = 2,NOTC                                                   00004550
      READ(MT,1005,END=2054) TIT                                         00004560
 1005 FORMAT(10A8)                                                       00004570
      WRITE(MT1,5004) TIT                                                00004580
 5004 FORMAT(24X,10A8)                                                   00004590
   63 CONTINUE                                                           00004600
   64 WRITE(MT1,5006) LFDA,NUM(LFDA),MQ,LYR,LSTMO,NUM(LSTMO),            00004610
     1MQ,LSTYR                                                           00004620
 5006 FORMAT(1H0,12X,17H-PERIOD COVERED- ,I2,A2,1X,A7,3H,19,I2,4H TO ,I200004630
     1, A2,1X,A7,3H,19,I2)                                               00004640
      WRITE(MT1,5007) (MALO(I),I=IT2,JT2),(RUNS(J),J=IT3,JT3),PRTS(NP1),00004650
     1                CHTS(NC1)                                          00004660
 5007 FORMAT(1H ,12X,15H-TYPE OF RUN - ,3A4,A2,1X,5A4,/,                 00004670
     1  13X,1H-,A8,11H PRINTOUT. ,A8,8H CHARTS.)                         00004680
      WRITE(MT1,5008) SIGML,SIGMU                                        00004690
 5008 FORMAT(1H ,12X,48H-SIGMA LIMITS FOR GRADUATING EXTREME VALUES ARE 00004700
     1,F3.1,5H AND ,F3.1,2H .)                                           00004710
      IF (IAGR.NE.3) CALL INPUT(SERIES,IPLUS,SERNO,IFORM,IFMT,&2004)     00004720
      IF (KFMT.EQ.0) GO TO 164                                           00004730
      IF (IAGR.NE.3) CALL INPUT(SPRIOR,0,KSER,KFORM,KKFMT,&2058)         00004740
      IF (MULADD.EQ.1) GO TO 364                                         00004750
      DO 464 I = LFDA,LLDA                                               00004760
      IF (SPRIOR(I).LT.0.0) GO TO 2056                                   00004770
      IF (SPRIOR(I).NE.0.0) GO TO 464                                    00004780
      SPRIOR(I) = 100.0                                                  00004790
  464 SPRIOR(I) = SPRIOR(I)/100.0                                        00004800
  164 IF (MULADD.EQ.1) GO TO 364                                         00004810
      DO 264 I = LFDA,LLDA                                               00004820
      IF (SERIES(I).LE.0.0) GO TO 2055                                   00004830
  264 CONTINUE                                                           00004840
      DO 864 I = 1,3                                                     00004850
  864 PCDIF(I) = PCD(I)                                                  00004860
      RAD(1) = R(1)                                                      00004870
      RAD(2) = R(2)                                                      00004880
      GO TO 764                                                          00004890
  364 DO 564 I = 1,3                                                     00004900
  564 PCDIF(I) = PCD(I + 3)                                              00004910
      RAD(1) = R(3)                                                      00004920
      RAD(2) = R(4)                                                      00004930
C --- GENERATE FORMATS FOR THE TABLES ROUTINE.                           00004940
  764 CALL TFMTS(NY,IRUN)                                                00004950
```

```
      IF (KEXOPT.NE.1) GO TO 65                                         00004960
      WRITE(MT1,5009)                                                   00004970
 5009 FORMAT(1H ,12X,'-MODIFY EXTREME VALUES BEFORE COMPUTING THE B7 T  00004980
     1REND CYCLE CURVE.')                                               00004990
   65 CONTINUE                                                          00005000
      IF (IYRT.EQ.0) GO TO 66                                           00005010
      WRITE(MT1,5010)                                                   00005020
 5010 FORMAT(1H ,12X,67H-MODIFY THE D11. SERIES TO MAKE THE YEARLY TOTAL00005030
     1S OF THE SEASONALLY,/,                                            00005040
     2 14X,47HADJUSTED SERIES AGREE WITH THE ORIGINAL SERIES.)          00005050
   66 CONTINUE                                                          00005060
      IF (LTERM.EQ.0) GO TO 67                                          00005070
      IF (LTERM.NE.5) GO TO 671                                         00005080
      WRITE(MT1,5022)                                                   00005090
 5022 FORMAT(1H ,12X,'-3X3 MOVING AVERAGE USED IN SECTION 1 OF EACH ITER00005100
     1ATION, THE PROGRAM',/,28X,'WILL SELECT THE APPROPRIATE AVERAGE FOR00005110
     2SECTION 2 OF EACH ITERATION')                                     00005120
      GO TO 67                                                          00005130
  671 WRITE(MT1,5011) AVG(LTERM)                                        00005140
 5011 FORMAT(1H ,12X,3H-A ,A8,56H MOVING AVERAGE SELECTED FOR THE SEASON00005150
     1AL FACTOR CURVES.)                                                00005160
   67    LABL=ST                                                        00005180
      IFTNY=15*NY                                                       00005190
      IF (LENGTH .LE. IFTNY .AND. KEBACK .EQ. 0) GO TO 94               00005200
            LABL=BLK                                                    00005210
   94 IF (IFORC-1) 70,68,69                                             00005220
   68 IF (LABL .EQ. BLK)WRITE(MT1,5012)                                 00005230
      IF (LABL .NE. BLK) WRITE(MT1,5088)                                00005240
 5012 FORMAT(1H ,12X,'-ONE YEAR OF FORECASTS FROM ARIMA',               00005250
     1' MODEL SELECTED BY THE PROGRAM.')                                00005260
 5088 FORMAT(1H ,12X,'-ONE YEAR OF FORECASTS AND BACKCASTS FROM ARIMA', 00005270
     1' MODEL SELECTED BY THE PROGRAM.')                                00005280
      IF(JAREX.EQ.0) GO TO 70                                           00005290
      WRITE(MT1,90)                                                     00005300
   90 FORMAT(1H ,12X,'-IF AN ARIMA MODEL IS ACCEPTED, VALUES OF THE ORIG00005310
     1INAL OR PRIOR-ADJUSTED SERIES FARTHER',/,14X,'THAN 2.5 STANDARD DE00005320
     2VIATIONS FROM THE FITTED MODEL VALUES WILL BE REPLACED BY THE MODE00005330
     3L VALUES.')                                                       00005340
      GO TO 70                                                          00005350
   69 IF (LABL .EQ. BLK) WRITE(MT1,5013)                                00005360
      IF (LABL .NE. BLK) WRITE(MT1,5099)                                00005370
 5013 FORMAT(1H ,12X,'-ONE YEAR OF FORECASTS FROM AN ARI',              00005380
     1'MA MODEL SELECTED BY THE USER.')                                 00005390
 5099 FORMAT(1H ,12X,'-ONE YEAR OF FORECASTS AND BACKCASTS FROM AN ARI',00005400
     1'MA MODEL SELECTED BY THE USER.')                                 00005410
      IF(JAREX.EQ.0) GO TO 70                                           00005420
      WRITE(MT1,91)                                                     00005430
   91 FORMAT(1H ,12X,'-VALUES OF THE ORIGINAL OR PRIOR-ADJUSTED ORIGINAL00005440
     1 SERIES FARTHER THAN',/,14X,'2.5 STANDARD DEVIATIONS FROM THE FITT00005450
     2ED MODEL VALUES WILL BE REPLACED BY THE MODEL VALUES.')           00005460
   70 CONTINUE                                                          00005470
      IF (KTCOPT.EQ.0) GO TO 72                                         00005480
      KT = KTCOPT                                                       00005490
      IF (NY.EQ.12) GO TO 71                                            00005500
      KT = KT+3                                                         00005510
```

```
   71 WRITE(MT1,5014) KTC(KT)
 5014 FORMAT(24X,56HMOVING AVERAGE FOR THE VARIABLE TREND CYCLE ROUTINE
     1IS A,I3,15H-TERM HENDERSON)
   72 KTCOPT = KTCOPT + 1
      IF (NY.EQ.4) KTCOPT = KTCOPT+4
      IF (KFMT.EQ.0) GO TO 73
      WRITE(MT1,5015) QM
 5015 FORMAT(47X,6HPRIOR ,2A4,A1,12H ADJUSTMENT.)
   73 KSWV = 0
      OUT = OT
      IF (LOPT.EQ.0) GO TO 74
      OUT = BK
   74 I = 1
   75 IF (DWT(I).EQ.0.0) GO TO 76
      KSWV = 1
   76 I = I+1
      IF (I.LE.7) GO TO 75
      IF (KSWV.NE.0) GO TO 78
      IF (MULADD.EQ.1) GO TO 79
      DO 77 I = 1,7
   77 DWT(I) = 1.0
      GO TO 79
   78 CONTINUE
      DO 178 I = 1,7
      IF (DWT(I).EQ.0.AND.LPRIOR.EQ.1) GO TO 2039
  178 CONTINUE
      WRITE(MT1,5016) OUT,MQ
 5016 FORMAT(1H ,12X,34H-PRIOR TRADING DAY ADJUSTMENT WITH,A3,10H LENGTH
     1 OF,A7,12H ADJUSTMENT.)
   79 CONTINUE
      IF (KDWOPT.EQ.0) GO TO 83
      WRITE(MT1,5017) LCYR,SIGM
 5017 FORMAT(1H ,12X,44H-TRADING DAY REGRESSION COMPUTED STARTING 19,I2,
     1 36H EXCLUDING IRREGULAR VALUES OUTSIDE ,F3.1,14H-SIGMA LIMITS.)
      IF (KDWOPT-2) 80,81,82
   80 WRITE(MT1,5018)
 5018 FORMAT(1H ,12X,46H-TRADING DAY REGRESSION ESTIMATES NOT APPLIED.)
      GO TO 83
   81 WRITE(MT1,5019) LAYR
 5019 FORMAT(1H ,12X,53H-TRADING DAY REGRESSION ESTIMATES APPLIED STARTI
     1NG 19 ,I2)
      GO TO 83
   82 WRITE(MT1,5020) LAYR
 5020 FORMAT(1H ,12X,53H-TRADING DAY REGRESSION ESTIMATES APPLIED STARTI
     1NG 19 ,I2,16H IF SIGNIFICANT.)
C --- PRINT THE MAIN CONTROL CARD IMAGES
   83 CALL F4GEN(MT1)
      IFATAL = 0
      RETURN
 2000 WRITE(MT2,3000)
 3000 FORMAT(1H0,'READ ERROR ON CONTROL CARD. THE CARD WILL BE SCANNED.'
     1)
      GO TO 101
 2001 WRITE(MT2,3001)
 3001 FORMAT(1H0,' THERE IS NO Z CARD PRESENT. END OF INPUT FILE DETECTE
```

```
     1D.')
      GO TO 2004
 2002 ICOUNT = ICOUNT+1
      WRITE(MT2,3002)
 3002 FORMAT(1H0,' FIRST CARD IS NOT AN M OR Q CARD.')
      IF (ICOUNT.LT.35) GO TO 1
 2003 WRITE(MT2,3003)
 3003 FORMAT(1H0,' JOB TERMINATING DUE TO ERROR COUNT.')
 2004 WRITE(MT2,3004)
      WRITE(MT1,3004)
 3004 FORMAT(1H0,' END OF X-11 ARIMA')
      IFATAL = 1
      IF (IMESSG.EQ.0) RETURN
      WRITE(NG,4004)
 4004 FORMAT(1H0,66X,'*IDENTIFIABLE SEASONALITY NOT PRESENT')
      RETURN
 2005 IFATAL = IFATAL+1
      KCOL(2) = ST
      WRITE(MT2,3005)
 3005 FORMAT(1H0,'ERROR IN THE INPUT FORMAT SPECIFICATION. THE VALUE CAN
     1 NOT BE GREATER THAN FIVE.')
      GO TO 5
 2006 IFATAL = IFATAL+1
      WRITE(MT2,3006)
 3006 FORMAT(1H0,'READ ERROR ON USER SUPPLIED FORMAT CARD.')
      GO TO 5
 2007 WRITE(MT2,3007)
 3007 FORMAT(1H0,'UNEXPECTED END OF FILE ON INPUT UNIT.')
      IFATAL = IFATAL + 1
      GO TO 5
 2008 IFATAL = IFATAL+1
      WRITE(MT2,3008)
 3008 FORMAT(1H0,'ERROR IN PUNCHED CARD OPTION. VALUE MUST BE ZERO OR ON
     1E')
      KCOL(21) = ST
      GO TO 16
 2009 IFATAL = IFATAL+1
      WRITE(MT2,3009)
 3009 FORMAT(1H0,'READ ERROR ON CARD SPECIFYING PUNCH OUTPUT.')
      GO TO 16
 2010 IERROR = IERROR+1
      WRITE(MT2,3010)
 3010 FORMAT(1H0,'ONLY 9 TABLES MAY BE PUNCHED. FIRST 9 WILL BE PUNCHED.
     1')
      IF (NOP.EQ.0) GO TO 2013
      NOP = 9
      GO TO 6
 2011 JFMT = 3
      IERROR = IERROR+1
      WRITE(MT2,3011)
 3011 FORMAT(1H0,'ERROR IN THE CARD OUTPUT FORMAT SPECIFICATION. STC. ST
     1ANDARD FORMAT WILL BE USED.')
      GO TO 8
 2012 IFATAL = IFATAL+1
      WRITE(MT2,3012)
```

```
 3012 FORMAT(1H0,'READ ERROR ON USER SUPPLIED FORMAT FOR CARD OUTPUT.') 00006620
      GO TO 16                                                          00006630
 2013 IFATAL = IFATAL+1                                                 00006640
      WRITE(MT2,3013)                                                   00006650
 3013 FORMAT(1H0,'NO. OF SERIES SPECIFIED FOR CARD OUTPUT IS ZERO. THERE00006660
     1 MUST BE AT LEAST ONE.')                                          00006670
      KCOL(21) = ST                                                     00006680
      GO TO 16                                                          00006690
 2016 IFATAL = IFATAL+1                                                 00006700
      WRITE(MT2,3016)                                                   00006710
 3016 FORMAT(1H0,'FIRST MONTH OR QUARTER IS ZERO.')                     00006720
      KCOL(11) = ST                                                     00006730
      KCOL(12) = ST                                                     00006740
      GO TO 17                                                          00006750
 2017 IFATAL = IFATAL+1                                                 00006760
      KCOL(11) = ST                                                     00006770
      KCOL(12) = ST                                                     00006780
      WRITE(MT2,3017)                                                   00006790
 3017 FORMAT(1H0,'FIRST MONTH OR QUARTER IS TOO LARGE.')                00006800
      GO TO 17                                                          00006810
 2018 IFATAL = IFATAL+1                                                 00006820
      KCOL(15) = ST                                                     00006830
      KCOL(16) = ST                                                     00006840
      WRITE(MT2,3018)                                                   00006850
 3018 FORMAT(1H0,'LAST MONTH OR QUARTER IS ZERO.')                      00006860
      GO TO 20                                                          00006870
 2019 IFATAL = IFATAL+1                                                 00006880
      KCOL(15) = ST                                                     00006890
      KCOL(16) = ST                                                     00006900
      WRITE(MT2,3019)                                                   00006910
 3019 FORMAT(1H0,'LAST MONTH OR QUARTER IS TOO LARGE.')                 00006920
      GO TO 20                                                          00006930
 2020 IFATAL = IFATAL+1                                                 00006940
      WRITE(MT2,3020)                                                   00006950
 3020 FORMAT(1H0,'SERIES MUST HAVE 3 COMPLETE YEARS.')                  00006960
      DO 4020 I=11,18                                                   00006970
 4020 KCOL(I) = ST                                                      00006980
      GO TO 20                                                          00006990
 2021 IFATAL = IFATAL+1                                                 00007000
      WRITE(MT2,3021)                                                   00007010
 3021 FORMAT(1H0,'SERIES MUST NOT EXCEED 30 YEARS.')                    00007020
      DO 4021 I=11,18                                                   00007030
 4021 KCOL(I) = ST                                                      00007040
      GO TO 20                                                          00007050
 2023 KCOL(19) = ST                                                     00007060
      IDEC = 5                                                          00007070
      WRITE(MT2,3023)                                                   00007080
 3023 FORMAT(1H0,'NO. OF DECIMALS ON INPUT CARDS MUST NOT EXCEED 5 FOR T00007090
     1HE SUPPLIED FORMATS. FIVE DECIMALS ASSUMED.')                     00007100
      IERROR = IERROR+1                                                 00007110
      GO TO 21                                                          00007120
 2024 IFATAL = IFATAL+1                                                 00007130
      KCOL(20) = ST                                                     00007140
      WRITE(MT2,3024)                                                   00007150
 3024 FORMAT(1H0,'INPUT ON TAPE OR DISK MUST BE SPECIFIED BY A 1 IN COLU00007160
```

```
     1MN 20.')                                                          00007170
      GO TO 22                                                          00007180
 2025 IERROR = IERROR+1                                                 00007190
      KCOL(22) = ST                                                     00007200
      KDEC = 0                                                          00007210
      WRITE(MT2,3025)                                                   00007220
 3025 FORMAT(1H0,'NO. OF DECIMALS ON OUTPUT TABLES MUST NOT EXCEED 5. N00007230
     1 DECIMALS ASSUMED.')                                              00007240
      GO TO 23                                                          00007250
 2026 IERROR = IERROR+1                                                 00007260
      MULADD = 0                                                        00007270
      KCOL(23) = ST                                                     00007280
      WRITE(MT2,3026)                                                   00007290
 3026 FORMAT(1H0,'ERROR IN THE SPECIFICATION OF THE TYPE OF SERIES. THE 00007300
     1SERIES IS ASSUMED TO BE MULTIPLICATIVE.')                         00007310
      GO TO 25                                                          00007320
 2027 IERROR = IERROR+1                                                 00007330
      KCOL(26) = ST                                                     00007340
      KPROPT = 0                                                        00007350
      WRITE(MT2,3027)                                                   00007360
 3027 FORMAT(1H0,'ERROR IN THE PRINTOUT SPECIFICATION. ASSUMED TO BE A S00007370
     1TANDARD PRINTOUT.')                                               00007380
      GO TO 29                                                          00007390
 2028 IERROR = IERROR+1                                                 00007400
      KCOL(27) = ST                                                     00007410
      KCHOPT = 0                                                        00007420
      WRITE(MT2,3028)                                                   00007430
 3028 FORMAT(1H0,'ERROR IN THE CHARTS OPTION. STANDARD CHARTS ASSUMED.')00007440
      GO TO 30                                                          00007450
 2029 IERROR = IERROR+1                                                 00007460
      IYRT = 0                                                          00007470
      KCOL(24) = ST                                                     00007480
      WRITE(MT2,3029)                                                   00007490
 3029 FORMAT(1H0,'ERROR IN COLUMN 24 OF CONTROL CARD. NO ADJUSTMENT OF Y00007500
     1EARLY TOTALS ASSUMED.')                                           00007510
      GO TO 31                                                          00007520
 2030 KCOL(25) = ST                                                     00007530
      IERROR = IERROR+1                                                 00007540
      KFULSM = 0                                                        00007550
      WRITE(MT2,3030)                                                   00007560
 3030 FORMAT(1H0,'ERROR IN SUMMARY MEASURES SPECIFICATION. SEASONAL ADJU00007570
     1STMENT PROGRAM ASSUMED.')                                         00007580
      GO TO 32                                                          00007590
 2031 IERROR = IERROR+1                                                 00007600
      KCOL(28) = ST                                                     00007610
      KCOL(29) = ST                                                     00007620
      WRITE(MT2,3031)                                                   00007630
 3031 FORMAT(1H0,'ERROR IN SELECTION OF LOWER SIGMA LIMIT. DEFAULT VALUE00007640
     1 ASSUMED.')                                                       00007650
      GO TO 33                                                          00007660
 2032 WRITE(MT2,3032)                                                   00007670
 3032 FORMAT(1H0,'ERROR IN SELECTION OF UPPER SIGMA LIMIT. DEFAULT VALUE00007680
     1 ASSUMED.')                                                       00007690
      IERROR = IERROR+1                                                 00007700
      KCOL(30) = ST                                                     00007710
```

```
      KCOL(31) = ST                                                     00007720
      GO TO 35                                                          00007730
 2033 SIGML = 1.5                                                       00007740
      IERROR = IERROR+1                                                 00007750
      WRITE(MT2,3033)                                                   00007760
 3033 FORMAT(1H0,'ERROR IN SIGMA LIMITS. DEFAULT VALUES ASSUMED.')      00007770
      DO 4033 I = 28,31                                                 00007780
 4033 KCOL(I) = ST                                                      00007790
      GO TO 35                                                          00007800
 2034 KCOL(32) = ST                                                     00007810
      IERROR = IERROR+1                                                 00007820
      LTERM = 0                                                         00007830
      WRITE(MT2,3034)                                                   00007840
 3034 FORMAT(1H0,'ERROR IN THE SELECTION OF THE MOVING AVERAGE FOR THE S00007850
     1EASONAL FACTOR CURVES. THE PROGRAM WILL SELECT THE MOVING AVERAGE.00007860
     2')                                                                00007870
      GO TO 39                                                          00007880
 2035 IERROR = IERROR+1                                                 00007890
      KCOL(34) = ST                                                     00007900
      IFORC = 0                                                         00007910
      WRITE(MT2,3035)                                                   00007920
 3035 FORMAT(1H0,'ERROR IN THE ARIMA MODEL SELECTION OPTION. NO FORECAST00007930
     1 WILL BE DONE.')                                                  00007940
      GO TO 39                                                          00007950
 2036 KCOL(35) = ST                                                     00007960
      IF (KEXOPT.LE.4) GO TO 40                                         00007970
      KEXOPT = 0                                                        00007980
      IERROR = IERROR+1                                                 00007990
      WRITE(MT2,3036)                                                   00008000
 3036 FORMAT(1H0,'ERROR IN STRIKE OPTION. NO STRIKE OPTION ASSUMED.')   00008010
      GO TO 40                                                          00008020
 2037 KCOL(33) = ST                                                     00008030
      KTCOPT = 0                                                        00008040
      IERROR = IERROR+1                                                 00008050
      WRITE(MT2,3037)                                                   00008060
 3037 FORMAT(1H0,'ERROR IN VARIABLE-TREND SELECTION. PROGRAM WILL SELECT00008070
     1 MOVING AVERAGE.')                                                00008080
      GO TO 40                                                          00008090
 2038 KCOL(73) = ST                                                     00008100
      IERROR = IERROR+1                                                 00008110
      LOPT = 0                                                          00008120
      WRITE(MT2,3038)                                                   00008130
 3038 FORMAT(1H0,'ERROR IN LENGTH-OF-MONTH OPTION. ASSUMED NO ADJUSTMENT00008140
     1.')                                                               00008150
      GO TO 43                                                          00008160
 2039 IERROR = IERROR+1                                                 00008170
      DO 5039 J = 1,4                                                   00008180
 5039 KCOL(43+I*4-J+1) = ST                                             00008190
      KSWV = 0                                                          00008200
      DO 4039 I = 1,7                                                   00008210
 4039 DWT(I) = 0.0                                                      00008220
      WRITE(MT2,3039)                                                   00008230
 3039 FORMAT(1H0,'ERROR IN PRIOR DAILY WEIGHTS. NO PRIORS ASSUMED')     00008240
      IF (LPRIOR.EQ.0) GO TO 144                                        00008250
      LPRIOR = 0                                                        00008260
```

```
      DO 6039 I = 1,7                                                   00008270
      IF (DWT(I).EQ.0.0) GO TO 6039                                     00008280
      DO 7039 J = 1,4                                                   00008290
 7039 KCOL(I*4+43-J+1) = ST                                             00008300
 6039 DWT(I) = 0.0                                                      00008310
      MULADD = 1                                                        00008320
      WRITE(MT2,8039)                                                   00008330
 8039 FORMAT(1H0,'SWITCHING BACK TO ADDITIVE.')                         00008340
      GO TO 144                                                         00008350
 2040 KCOL(74) = ST                                                     00008360
      KDWOPT = 0                                                        00008370
      IERROR = IERROR+1                                                 00008380
      WRITE(MT2,3040)                                                   00008390
 3040 FORMAT(1H0,'ERROR IN TRADING-DAY SELECTION. NO TRADING-DAY ADJUSTM00008400
     1ENT ASSUMED.')                                                    00008410
      GO TO 45                                                          00008420
 2041 KCOL(23) = ST                                                     00008430
      LPRIOR = 1                                                        00008440
      MULADD = 0                                                        00008450
      IERROR = IERROR+1                                                 00008460
      WRITE(MT2,3041)                                                   00008470
 3041 FORMAT(1H0,'THE ADDITIVE OPTION CAN NOT BE USED WITH PRIOR DAILY W00008480
     1EIGHTS. MULTIPLICATIVE MODEL ASSUMED.')                           00008490
      GO TO 47                                                          00008500
 2042 KCOL(73) = ST                                                     00008510
      KCOL(23) = ST                                                     00008520
      IERROR = IERROR+1                                                 00008530
      LOPT = 0                                                          00008540
      WRITE(MT2,3042)                                                   00008550
 3042 FORMAT(1H0,'LENGTH-OF-MONTH ADJUSTMENT OPTION CAN NOT BE USED WITH00008560
     1 ADDITIVE OPTION. NO LENGTH-OF-MONTH ADJUSTMENT DONE.')           00008570
      GO TO 47                                                          00008580
 2043 KCOL(80) = ST                                                     00008590
      KCOL(79) = ST                                                     00008600
      IERROR = IERROR+1                                                 00008610
      WRITE(MT2,3043)                                                   00008620
 3043 FORMAT(1H0,'ERROR IN THE SIGMA LIMIT FOR TRADING-DAY ADJUSTMENT. D00008630
     1EFAULT VALUE ASSUMED.')                                           00008640
      GO TO 48                                                          00008650
 2044 KCOL(76) = ST                                                     00008660
      KCOL(75) = ST                                                     00008670
      IERROR = IERROR+1                                                 00008680
      WRITE(MT2,3044)                                                   00008690
 3044 FORMAT(1H0,'ERROR IN THE STARTING DATE FOR COMPUTING THE TRADING-D00008700
     1AY REGRESSION. IT IS ASSUMED TO START AT THE BEGINNING OF THE SERI00008710
     2ES.')                                                             00008720
      GO TO 52                                                          00008730
 2045 KCOL(78) = ST                                                     00008740
      KCOL(77) = ST                                                     00008750
      IERROR = IERROR+1                                                 00008760
      WRITE(MT2,3045)                                                   00008770
 3045 FORMAT(1H0,'ERROR IN THE STARTING DAY FOR APPLYING THE TRADING-DAY00008780
     1 REGRESSION. IT IS ASSUMED TO START FROM COMPUTING DATE.')        00008790
      GO TO 56                                                          00008800
 2046 IFATAL = IFATAL+1                                                 00008810
```

```
      IF(T1.EQ.X)GO TO 88
      WRITE(MT2,3046)
 3046 FORMAT(1H0,'NO TITLE CARD PRESENT.')
      GO TO 59
   88 WRITE(MT2,89)
   89 FORMAT(1H0,'UNEXPECTED EXTRA OPTION CARD READ. PLEASE PUT AN
     1 X IN',/,'COLUMN 73 OF THE MAIN OPTION CARD AND RESUBMIT THE JOB
     2.')
      GO TO 59
 2047 IFATAL = IFATAL+1
      KCOL(36) = ST
      WRITE(MT2,3047)
 3047 FORMAT(1H0,'ERROR IN THE FORMAT OPTION FOR THE PRIOR ADJUSTMENT FA
     1CTORS.')
      GO TO 105
 2048 WRITE(MT2,3048)
 3048 FORMAT(1H0,'JOB TERMINATING DUE TO FATAL ERROR.')
      GO TO 2004
 2049 WRITE(MT2,3006)
      IFATAL = IFATAL + 1
      GO TO 105
 2050 IYRT = 0
      WRITE(MT2,3050)
 3050 FORMAT(1H0,'ERROR - NOT ENOUGH DATA TO ADJUST THE YEARLY TOTALS OF
     1 THE SEASONALLY ADJUSTED SERIES.',/,1H0,'THE SERIES MUST HAVE AT L
     2EAST FIVE COMPLETE YEARS.')
      IERROR = IERROR + 1
      KCOL(24) = ST
      GO TO 31
 2051 IFATAL = IFATAL + 1
      WRITE(MT2,3007)
      GO TO 105
 2052 IFATAL = IFATAL + 1
      WRITE(MT2,3007)
      GO TO 16
 2053 WRITE(MT2,3053)
 3053 FORMAT(1H0,'READ ERROR ON THE TITLE CARD')
      NOP = 0
      IF (T1.NE.T) IFATAL = IFATAL+1
      IF (T1.NE.T) GO TO 59
      WRITE(MT2,4053)
 4053 FORMAT(1H0,'THE READ ERROR WILL BE IGNORED AND AN ATTEMPT WILL BE
     1MADE TO READ THE DATA.')
      GO TO 59
 2054 IFATAL = IFATAL + 1
      WRITE(MT2,3007)
      GO TO 59
 2055 IERROR = IERROR + 1
      KCOL(23) = ST
      WRITE(MT2,3055)
 3055 FORMAT(1H0,'MULTIPLICATIVE OR LOGARITHMIC MODELS CAN NOT BE RUN WI
     1TH NEGATIVE SERIES VALUES.',/,1X,'SWITCHING TO ADDITIVE MODEL.')
      WRITE(MT1,4055)
 4055 FORMAT(1H-,12X,82HADDITIVE SEASONAL ADJUSTMENT RUN DUE TO NEGATIVE
     1 OR ZERO VALUES IN THE INPUT DATA.,/)
```

```
      MULADD = 1                                                        00009370
      MA = 1                                                            00009380
      LOPT = 0                                                          00009390
      DO 5055 I = 1,7                                                   00009400
 5055 DWI(I) = 0.0                                                      00009410
      IF (KFMT.EQ.0) GO TO 364                                          00009420
      IFATAL = 1                                                        00009430
      WRITE(MT2,6055)                                                   00009440
 6055 FORMAT(1H0,'PRIOR ADJUSTMENT REQUESTED, SWITCH CAN NOT BE MADE. SE00009450
     1RIES CAN NOT BE ADJUSTED.')                                       00009460
      GO TO 2048                                                        00009470
 2056 IFATAL = 1                                                        00009480
      WRITE(MT2,3056)                                                   00009490
 3056 FORMAT(1H0,'MULTIPLICATIVE OR LOGARITHMIC MODELS CAN NOT BE RUN WI00009500
     1TH NEGATIVE PRIOR SERIES VALUES.')                                00009510
      GO TO 2048                                                        00009520
 2057 IFATAL=IFATAL+1                                                   00009530
      WRITE(MT2,3057)                                                   00009540
 3057 FORMAT(1H0,'READ ERROR ON THE ARIMA MODEL SELECTION CARD. EXECUTI00009550
     1N TERMINATED.')                                                   00009560
      GO TO 59                                                          00009570
 2058 WRITE(MT2,3058) QM                                                00009580
 3058 FORMAT(1H ,'ERROR OCCURRED DURING THE READING OF THE PRIOR ',2A4, 00009590
     1  A1,' FACTORS')                                                  00009600
      GO TO 2004                                                        00009610
 2059 IFATAL = IFATAL+1                                                 00009620
      WRITE(MT2,3059)                                                   00009630
 3059 FORMAT(1H0,'MISSING USERS-SUPPLIED FORMAT CARD. TITLE CARD READ.')00009640
      GO TO 5                                                           00009650
 2060 IFATAL = IFATAL+1                                                 00009660
      WRITE(MT2,3060)                                                   00009670
 3060 FORMAT(1H0,'MISSING USERS-SUPPLIED FORMAT CARD FOR THE PRIOR ADJUS00009680
     1TMENT CARDS. TITLE CARD READ.')                                   00009690
      GO TO 105                                                         00009700
 2061 IFATAL = IFATAL+1                                                 00009710
      WRITE(MT2,3061)                                                   00009720
 3061 FORMAT(1H0,'MISSING USERS-SUPPLIED FORMAT CARD FOR THE PUNCHED CAR00009730
     1DS OPTION. TITLE CARD READ.')                                     00009740
      GO TO 8                                                           00009750
      END                                                               00009760
      SUBROUTINE FMTS(IFORM,KFMT,IDEC,NY)                               00009770
C --- THIS SUBROUTINE GENERATES THE INPUT AND OUTPUT FORMATS.           00009780
      DIMENSION IFORM(20),IF1(14),IF2(14),IF3(14),IF4(14),IF5(14),      00009790
     1          IF6(14),IF7(14),KDEC(6),IF8(11)                         00009800
      DATA IF1 /4H (   ,4H     ,4H     ,4H    1,4H2F6.,4H0    ,4H,    ,4H     ,00009810
     1          4H     ,4H     ,4H     ,4HI2, ,4HA6  ,4H)    /          00009820
      DATA IF2 /4H (   ,4H     ,4H     ,4H    6,4HF12.,4H0    ,4H,/,  ,4H    6,00009830
     1          4HF12.,4H0    ,4H,    ,4HI2, ,4HA6  ,4H)    /          00009840
      DATA IF3 /4H (   ,4H A6,,4H I2,,4H    1,4H2F6.,4H0    ,4H     ,4H     ,00009850
     1          4H     ,4H     ,4H     ,4H     ,4H     ,4H)    /        00009860
      DATA IF4 /4H (   ,4H A8,,4H I2,,4H10X,,4H12E1,4H6.10,4H,18X,4H     ,00009870
     1          4H     ,4H     ,4H     ,4H     ,4H     ,4H)    /        00009880
      DATA IF5 /4H (   ,4H     ,4H     ,4H    4,4HF12.,4H0    ,4H,    ,4H24X,,00009890
     1          4H     ,4H     ,4H     ,4HI2, ,4HA6  ,4H)    /          00009900
      DATA IF6 /4H (   ,4H4(   ,4H12X,,4H     ,4H F6.,4H0    ,4H),   ,4H     ,00009910
```

```
     1          4H     ,4H     ,4H     ,4HI2, ,4HA6   ,4H)    /                 00009920
      DATA IF7 /4H (   ,4H A6,,4HI2,4,4H(12X,4H,F6.,4H0     ,4H)    ,4H     ,00009930
     1          4H     ,4H     ,4H     ,4H     ,4H     ,4H)    /                 00009940
      DATA IF8 /4H (   ,4H A6,,4H I2,,4H    6,4HF12.,4H0     ,4H,/,8,4HX, 6,00009950
     1          4HF12.,4H0     ,4H)    /                                        00009960
      DATA KDEC/4H1    ,4H2    ,4H3    ,4H4    ,4H5    ,4H0    /                00009970
      DATA IBLK/4H     /                                                        00009980
      IF (KFMT.EQ.1.OR.KFMT.GT.5) RETURN                                        00009990
      IFMT = KFMT                                                               00010000
      IF (NY.EQ.4) IFMT = IFMT+10                                               00010010
      J = IDEC                                                                  00010020
      IF (J.GT.5) J = 6                                                         00010030
      IF (J.EQ.0) J = 6                                                         00010040
      DO 1 I = 1,20                                                             00010050
    1 IFORM(I) = IBLK                                                           00010060
      IF (IFMT.NE.0) GO TO 3                                                    00010070
      DO 2 I = 1,14                                                             00010080
    2 IFORM(I) = IF1(I)                                                         00010090
      GO TO 20                                                                  00010100
    3 IF (IFMT.NE.2) GO TO 5                                                    00010110
      DO 4 I = 1,14                                                             00010120
    4 IFORM(I) = IF2(I)                                                         00010130
      IFORM(10) = KDEC(J)                                                       00010140
      GO TO 20                                                                  00010150
    5 IF (IFMT.NE.3) GO TO 7                                                    00010160
      DO 6 I = 1,14                                                             00010170
    6 IFORM(I) = IF3(I)                                                         00010180
      GO TO 20                                                                  00010190
    7 IF (KFMT.EQ.4) GO TO 8                                                    00010200
      GO TO 10                                                                  00010210
    8 DO 9 I = 1,14                                                             00010220
    9 IFORM(I) = IF4(I)                                                         00010230
      RETURN                                                                    00010240
   10 IF (KFMT.NE.5) GO TO 14                                                   00010250
      DO 11 I = 1,11                                                            00010260
   11 IFORM(I) = IF8(I)                                                         00010270
      IF (IFMT.EQ.5) GO TO 13                                                   00010280
      DO 12 I = 7,10                                                            00010290
   12 IFORM(I) = IBLK                                                           00010300
      GO TO 20                                                                  00010310
   13 IFORM(10) = KDEC(J)                                                       00010320
      GO TO 20                                                                  00010330
   14 IF (IFMT.NE.10) GO TO 16                                                  00010340
      DO 15 I = 1,14                                                            00010350
   15 IFORM(I) = IF6(I)                                                         00010360
      GO TO 20                                                                  00010370
   16 IF (IFMT.NE.12) GO TO 18                                                  00010380
      DO 17 I = 1,14                                                            00010390
   17 IFORM(I) = IF5(I)                                                         00010400
      GO TO 20                                                                  00010410
   18 DO 19 I = 1,14                                                            00010420
   19 IFORM(I) = IF7(I)                                                         00010430
   20 IFORM(6) = KDEC(J)                                                        00010440
      RETURN                                                                    00010450
      END                                                                       00010460
```

```
      SUBROUTINE IFMTS(IFORM,KFMT,IDEC,NY)
C --- THIS SUBROUTINE CONVERTS THE OUTPUT FORMATS FROM FLOATING POINT TO
C --- INTEGER IF NO DECIMALS ARE REQUESTED.
      DIMENSION IFORM(20),KDEC(6),IF(10)
      DATA IF   /4H    ,4H12I6,4HI12 ,4H I6 ,4H,I6 ,
     1           4H   1,4H2F6.,4HF12.,4H F6.,4H,F6./
      DATA KDEC /4H1   ,4H2   ,4H3   ,4H4   ,4H5   ,4H    /
      IF (KFMT.EQ.1.OR.KFMT.EQ.4) RETURN
      IF (KFMT.GT.5) RETURN
      L = 5
      LDEC = IDEC
      IF (IDEC.NE.0) GO TO 1
      L = 0
      LDEC = 6
    1 JFMT = KFMT
      IF (NY.EQ.4) JFMT = JFMT+10
      IFORM(6) = KDEC(LDEC)
      IF (JFMT.NE.0.AND.JFMT.NE.3) GO TO 2
      IFORM(4) = IF(L+1)
      IFORM(5) = IF(L+2)
      RETURN
    2 IF (JFMT.EQ.10.OR.JFMT.EQ.13) GO TO 3
      IFORM(5) = IF(L+3)
      IF (NY.EQ.4) RETURN
      IFORM(9) = IF(L+3)
      IFORM(10) = KDEC(LDEC)
      RETURN
    3 IF (KFMT.EQ.3) GO TO 4
      IFORM(5) = IF(L+4)
      RETURN
    4 IFORM(5) = IF(L+5)
      RETURN
      END
      SUBROUTINE INPUT(X,IPLUS,SERNO,IFORM,KFMT,*)
C --- THIS SUBROUTINE WILL READ SERIES X IN FORMAT IFORM.
C --- SERNO IS THE SERIES IDENTIFIER AND IS CHECKED IF KFMT = 0,2,3,4,5
C --- IF IPLUS NE. 0 THE INPUT SERIES IS READ ON UNIT 13. IF IPLUS GE.
C --- 1 AND KFMT NE. 1 UNIT 13 WILL BE SEARCHED FOR A SERIES WITH
C --- IDENTIFIER SERNO. KFMT REPRESENTS THE FORMAT OF THE INPUT DATA.
      REAL*8 SERNO,SER
      COMMON /UNITS/ MT,MT2,MT1,MP,NG,NF
      COMMON /OPT9 / LFDA,LYR,LSTMO,LSTYR,LLDA,IDUM(4),NY,IFORC,LLDAF,
     1               LLAF
      DIMENSION X(1),IFORM(20)
      NR = MT
      IF (IPLUS.GE.1) NR = 13
      IF (IPLUS.EQ.2) REWIND NR
      IFMT = KFMT+1
      NY1 = NY
      IC = 1
      ISER = 0
      IF (KFMT.NE.4) GO TO 1
      IC = 12/NY
      NY1 = 12
    1 NYR = LYR
```

```
      NNYR = LSTYR-IC/3*2
      N = 1
      M = NY1
    2 GO TO (3,4,3,5,5,5,3,5),IFMT
    3 READ(NR,IFORM,ERR=1007,END=1001) (X(I),I=N,M),KYR,SER
      GO TO 6
    4 READ(NR,IFORM,ERR=1000,END=1001) (X(I),I=1,LLDA)
      GO TO 9
    5 IF (M.GT.LLDA) M = LLDA
      READ(NR,IFORM,ERR=1007,END=1001) SER,KYR,(X(I),I=N,M)
    6 IF (SER.NE.SERNO) GO TO 15
      ISER = 0
      IF (KYR-NYR) 2,7,1002
    7 IF (KYR-NNYR) 8,9,9
    8 N = N+NY1
      M = M+NY1
      NYR = NYR+IC
      GO TO 2
    9 IF (LFDA.EQ.1) GO TO 11
      K = LFDA-1
      DO 10 I = 1,K
   10 X(I) = 0.0
   11 IF (LSTMO.EQ.NY) GO TO 13
      K = LLDA+1
      L = (LSTYR-LYR+1)*NY
      DO 12 I = K,L
   12 X(I) = 0.0
   13 IF (IPLUS.NE.2) GO TO 14
      WRITE(MT2,2006)
 2006 FORMAT(1H0,'THE AUTOMATIC REWIND OPTION IS IN EFFECT. THE INPUT FI
     1LE IS REWOUND.')
   14 RETURN
   15 IF (NYR.NE.LYR) GO TO 1005
      IF (IPLUS.LT.1) GO TO 1003
      IF (ISER.NE.0) GO TO 1
      WRITE(MT2,2004) SERNO
 2004 FORMAT(1H0,
     1 'THE FILE WILL BE SEARCHED FOR SERIES ',A8,1H.)
      ISER = 1
      GO TO 1
 1000 WRITE(MT2,2000) NR,SERNO
 2000 FORMAT(1H0,'READ ERROR ON FILE ',I3,'. WHILE READING SERIES ',A8,
     11H.)
      RETURN 1
 1001 IF (IPLUS.EQ.3) GO TO 1008
      WRITE(MT2,2001) NR,SERNO
 2001 FORMAT(1H0,'END OF FILE ENCOUNTER ON FILE ',I3,' WHILE READING SER
     1IES ',A8,1H.)
      RETURN 1
 1002 WRITE(MT2,2002) SERNO,NYR,KYR
 2002 FORMAT(1H0,A8,25H REJECTED, YEAR SHOULD BE,I3,8H ,BUT IS,I3,1H.)
      RETURN 1
 1003 WRITE(MT2,2003) SERNO,SER
 2003 FORMAT(1H0,A8,11H REJECTED, ,A8,28H IS THE WRONG SERIES NUMBER.)
      RETURN 1
```

```
 1005 NYR = NYR-IC
      WRITE(MT2,2005) SERNO,NYR
 2005 FORMAT(1H0,'ALL THE DATA FOR SERIES ',A8,' WAS NOT AVAILABLE. THE
     1LAST YEAR READ WAS ',I3,1H.)
      RETURN 1
 1007 IF (IPLUS.LT.1) GO TO 1000
      IF (SER.EQ.SERNO) GO TO 1000
      IF (KFMT.EQ.1) GO TO 1000
      IF (ISER.NE.0) GO TO 1
      WRITE(MT2,2007) SERNO
 2007 FORMAT(1H0,'READ ERROR ENCOUNTERED ON INPUT FILE WHILE ATTEMPTING
     1TO READ SERIES ',A8,/,' THE FILE WILL BE SEARCHED.')
      ISER = 1
      GO TO 1
 1008 WRITE(MT2,2008) NR
 2008 FORMAT(1H0,'END OF FILE ENCOUNTERED ON UNIT ',I3,' CONCATENATED DA
     1TA SETS ASSUMED. THE FORTRAN SEQUENCE NUMBER IS INCREASED BY 1.')
      GO TO 1
      END
      SUBROUTINE TFMTS(NY,IRUN)
C --- THIS SUBROUTINE GENERATES THE FORMATS FOR SUBROUTINE TABLES.
      DIMENSION IF9(22),IF1(3),IF2(3),IF3(2),IF4(13),IF5(13)
      COMMON /TFMT / IFMT1(18),IFMT2(14),IFMT3(22)
      DATA IF1/4H(4X,,4H2H19,4H,I2,/
      DATA IF2/4H(3H0,4H  ,A,4H5,  /
      DATA IF3/4H    ,4H,/) /
      DATA IF4/4H1X, ,4H    ,4HF10.,4H    ,4H,11(,4H1X, ,4H    ,4H1F8.,
     1         4H    ,4H)   ,4H,5X,,4H    ,4HF10./
      DATA IF5/4H3X, ,4H    ,4HF15.,4H    ,4H,3(9,4HX,  ,4H    ,4HF15.,
     1         4H    ,4H)   ,4H,7X,,4H    ,4HF15./
      DATA IF9/4H(15X,4H,13H,4HTABL,4HE TO,4HTAL-,4H ,  ,4HF12.,4H    ,
     1         4H,8X,,4H6HME,4HAN- ,4H,F8.,4H    ,4H,8X,,4H16HS,4HTD. ,
     2         4HDEVI,4HATIO,4HN- ,,4H F8.,4H    ,4H)   /
      IF (NY.EQ.4) GO TO 2
      DO 1 I = 4,16
      IFMT1(I) = IF4(I-3)
      IF (I.GE.14) GO TO 1
      IFMT2(I) = IFMT1(I)
    1 CONTINUE
      GO TO 4
    2 DO 3 I = 4,16
      IFMT1(I) = IF5(I-3)
      IF (I.GE.14) GO TO 3
      IFMT2(I) = IFMT1(I)
    3 CONTINUE
    4 IFMT2(14) = IF3(2)
      IF (IRUN.GT.1) GO TO 8
      IFMT2(14) = IF3(2)
      DO 5 I = 1,3
      IFMT1(I) = IF1(I)
    5 IFMT2(I) = IF2(I)
      DO 6 I = 17,18
    6 IFMT1(I) = IF3(I-16)
      DO 7 I = 1,22
    7 IFMT3(I) = IF9(I)
```

```
    8 RETURN                                                            00012120
      END                                                               00012130
      SUBROUTINE X11ARI                                                 00012140
C --- THIS SUBROUTINE PARTITIONS X-11 AND ARIMA ROUTINES TO             00012150
C --- IMPROVE THE OVERLAY STRUCTURE.                                    00012160
      INTEGER*2 ALPHA(7)                                                00012180
      COMMON /UNITS/ MT,MT2,MT1,MP,NG,NFORM                             00012190
      COMMON /OPT2 / KDEC,ICARD                                         00012200
      COMMON /OPT4 / NUMB(9),IHOLD(9),NOP                               00012210
      COMMON /OPT9 / LFDA,LYR,LSTMO,LSTYR,LLDA,KFULSM,KPROPT,KCHOPT,    00012220
     1               KEXOPT,NY,IFORC,LLDAF,LLAF,LFD1,LLD1,NSOPT,IDEC,IAG00012230
      COMMON /MQ2  / KPART,KSWV,KSECT,KPRNT,MCD                         00012240
      COMMON /MQ11 / IMESSG,IAGR                                        00012250
      DATA ALPHA/1HA,1HB,1HC,1HD,1HE,1HF,1H /                           00012260
C --- PRIOR ADJUSTMENTS.                                                00012270
      CALL X11PT1                                                       00012280
C --- X-11 PARTS B1 TO D7.                                              00012290
    3 CALL X11PT2                                                       00012300
C --- X-11 PARTS D8 TO D16.                                             00012310
      CALL X11PT3                                                       00012320
C --- X-11 PARTS E1 TO F4.                                              00012330
      CALL X11PT4                                                       00012340
      IF (KCHOPT.EQ.1) GO TO 4                                          00012350
C --- X-11 PARTS G (CHARTS).                                            00012360
      CALL X11PT5                                                       00012370
    4 IF (ICARD.EQ.0) GO TO 6                                           00012380
      DO 5 I = 1,NOP                                                    00012390
      IF (IHOLD(I).EQ.1) GO TO 5                                        00012400
      I1 = NUMB(I)/1000                                                 00012410
      I2 = (NUMB(I)-1000*I1)/100                                        00012420
      I3 = NUMB(I)-1000*I1-100*I2                                       00012430
      IF (I1.EQ.0) I1 = 7                                               00012440
      WRITE(MT2,100) ALPHA(I2),I3,ALPHA(I1)                             00012450
  100 FORMAT(1H0,'TABLE ',A1,I2,A1,' WAS NOT AVAILABLE FOR PUNCHING. USE00012460
     1 A PRINTOUT OPTION CONTAINING THIS TABLE.')                       00012470
    5 CONTINUE                                                          00012480
    6 IF (IAGR.LE.0) GO TO 8                                            00012490
C --- INDIRECT AGGREGATE SEASONAL ADJUSTMENT.                           00012500
      IF (IAGR.NE.3) GO TO 7                                            00012510
      CALL AGR3(IAGR)                                                   00012520
      CALL X11PT4                                                       00012530
      IF (KCHOPT.EQ.1) GO TO 7                                          00012540
      CALL X11PT5                                                       00012550
    7 CALL AGR2(IAGR)                                                   00012560
    8 RETURN                                                            00012570
      END                                                               00012580
      SUBROUTINE X11PT1                                                 00012590
      REAL*8 KSER                                                       00012600
      COMMON /PRIOR/ KSER,KFORM(20),KFMT                                00012610
      COMMON /TRADE/ DWT(7),D(7),SIGM,LOPT,KDWOPT,LCYR,LAYR,LFDC,LFDR   00012620
      COMMON /INPT1/ SERIES(372),ORIG(372)                              00012630
      COMMON /INPT2/ SPRIOR(372)                                        00012640
      COMMON /OPT5 / MULADD,IYRT,MA,IWT                                 00012650
      COMMON /OPT9 / LFDA,LYR,LSTMO,LSTYR,LLDA,KFULSM,KPROPT,KCHOPT,    00012660
     1               KEXOPT,NY,IFORC,LLDAF,LLAF                         00012670
```

```
      COMMON /MQ2  / KPART,KSWV,KSECT,KPRNT,MCD
      COMMON /MQ5  / STCSI(372),STO(372)
      COMMON /MQ8  / STPTD(372),STTD(372)
      DIMENSION STD(372)
      IF (MULADD.EQ.2) MULADD = 0
C --- PART A.
      BAR = 1.0-MULADD
      KLDA = LLDA+NY
      DO 10 I = LFDA,KLDA
   10 STPTD(I) = BAR
      KPART = 1
C --- SET STO EQUAL TO INPUT SERIES.
      DO 1 I = LFDA,LLDA
      ORIG(I)=SERIES(I)
    1 STO(I) = SERIES(I)
C --- WRITE UNADJUSTED ORIGINAL SERIES A1.
      IF (KFMT+KSWV+IFORC) 4,4,2
    2 CALL TABLE(SERIES,LFDA,LLDA,1,1,2,0)
C --- TEST FOR PRIOR ADJUSTMENT.
      IF (KFMT.EQ.0) GO TO 3
C --- WRITE PRIOR FACTORS A2.
      CALL TABLE(SPRIOR,LFDA,LLDA,2,1,1,0)
C --- DIVIDE (SUBTRACT) ORIGINAL SERIES BY THE PRIOR FACTORS IN THE
C --- MULTIPLICATIVE AND LOGARITHMIC (ADDITIVE) MODELS.
      CALL DIVSUB(STO,SERIES,SPRIOR,LFDA,LLDA)
C --- TEST FOR PRIOR TRADING DAY ADJUSTMENT.
    3 IF (KSWV.EQ.0) GO TO 4
      IF(KFMT.EQ.0) GO TO 5
C --- IF THERE WAS ANY PRIOR ADJUSTMENT,
C --- WRITE PRIOR ADJUSTED SERIES BEFORE T.D. ADJUSTMENT A3.
      CALL TABLE(STO,LFDA,LLDA,3,1,2,0)
C --- COMPUTE MONTHLY T.D. FACTORS FROM PRIOR DAILY WEIGHTS.
    5 LIR=LYR-1
      LNDA=LLDA+NY
      CALL DOW(DWT,STPTD,LIR,LFDA,LNDA,LOPT)
C --- WRITE PRIOR TRADING FACTORS A4.
      LYR=LYR-1
      LFD1=LFDA+NY
      CALL TABLE(STPTD,LFD1,LNDA,4,1,1,DWT)
      LYR=LYR+1
C --- DIVIDE (SUBTRACT) PRIOR ADJUSTED OR ORIGINAL BY PRIOR T.D. FACTORS
      DO 100 I=LFDA,LLDA
      STD(I)=STPTD(I+NY)
  100 CONTINUE
      CALL DIVSUB(STO,STO,STD,LFDA,LLDA)
    4 RETURN
      END
      SUBROUTINE X11PT2
      REAL*8 KSER
      DIMENSION TEMPO(372)
      COMMON /WORK / TEMP(372)
      COMMON /PRIOR/ KSER,KFORM(20),KFMT
      COMMON /TRADE/ DWT(7),D(7),SIGM,LOPT,KDWOPT,LCYR,LAYR,LFDC,LFDR
      COMMON /INPT1/ SERIES(372),ORIG(372)
      COMMON /INPT2/ SPRIOR(372)
```

```
      COMMON /OPT5 / MULADD,IYRT,MA,IWT                                 00013230
      COMMON /OPT9 / LFDA,LYR,LSTMO,LSTYR,LLDA,KFULSM,KPROPT,KCHOPT,    00013240
     1               KEXOPT,NY,IFORC,LLDAF,LLAF,LFD1,LLD1,NSOPT,IDEC,IAG00013250
     2               ,JAREX,KEBACK                                      00013260
      COMMON /MQ1  / STS(372),STSI(372)                                 00013270
      COMMON /MQ2  / KPART,KSWV,KSECT,KPRNT,MCD                         00013280
      COMMON /MQ4  / STC(372),STCI(372)                                 00013290
      COMMON /MQ5  / STCSI(372),STO(372)                                00013300
      COMMON /MQ6  / STWT(372),STDEV(35)                                00013310
      COMMON /MQ7  / STI(372)                                           00013320
      COMMON /MQ8  / STPTD(372),STTD(372)                               00013330
      COMMON /MQ10 / STEX(372)                                          00013340
      COMMON /KCJH/LENGTH,IFTNY                                         00013350
      LENGTH=LLDA-LFDA+1                                                00013360
      IFTNY=15*NY                                                       00013370
      IF (IFORC.EQ.0) GO TO 55                                          00013380
      ITB=0                                                             00013390
      CALL ARIMA(1,ITB)                                                 00013400
      IF (LENGTH.GT.IFTNY.OR.IFORC.EQ.0.OR.KEBACK.EQ.1) GO TO 55        00013410
      DO 500 I=LFDA,LLDAF                                               00013420
      TEMPO(I)=STO(I)                                                   00013430
  500 CONTINUE                                                          00013440
      DO 501 I=LFDA,LLDA                                                00013450
      STO(I)=TEMPO(LLDA-I+LFDA)                                         00013460
  501 CONTINUE                                                          00013470
      CALL ARIMA(2,ITB)                                                 00013480
      IF (IFORC.EQ.0) LLDAF=LLDA                                        00013490
      IF (IFORC.EQ.0) GO TO 5499                                        00013500
      IF (KFMT.EQ.0) GO TO 5555                                         00013510
      DO 5554 I=1,LLDA                                                  00013520
 5554 SPRIOR(LLDA+NY+1-I)=SPRIOR(LLDA+1-I)                              00013530
 5555 DO 502 I=1,NY                                                     00013540
      STCSI(NY+LFDA-I)=STO(LLDA+I)                                      00013550
      STCSI(LLDA+I+NY)=TEMPO(LLDA+I)                                    00013560
  502 CONTINUE                                                          00013570
      DO 503 I=LFDA,LLDA                                                00013580
      STCSI(I+NY)=TEMPO(I)                                              00013590
  503 CONTINUE                                                          00013600
      NY2 = NY/2                                                        00013610
      GO TO 56                                                          00013620
 5499 IF(JAREX.GT.0)GO TO 55                                            00013630
      DO 5500 I=LFDA,LLDA                                               00013640
 5500 STO(I)=TEMPO(I)                                                   00013650
C --- SET STCSI EQUAL TO STO.                                           00013660
   55 DO 5 I = LFDA,LLDAF                                               00013670
    5 STCSI(I) = STO(I)                                                 00013680
      NY2 = NY/2                                                        00013690
C --- PART B                                                            00013700
   56 KPART = 2                                                         00013710
      KSECT = 1                                                         00013720
C --- WRITE ORIGINAL OR PRIOR ADJUSTED B1.                              00013730
      IF (IFORC.NE.0.AND.LENGTH.LE.IFTNY.AND.KEBACK.EQ.0) LYR=LYR-1     00013740
        KHCLDA=LLDAF-NY                                                 00013750
        IF (IFORC .EQ. 0) KHCLDA=LLDA                                   00013760
      CALL TABLE(STCSI,LFDA,KHCLDA,1,1,2,0)                             00013770
```

```
      LFD1=LFDA+NY
      LLD1=LLDA+NY
      IF (IFORC.EQ.0.OR.LENGTH.GT.IFTNY.OR.KEBACK.EQ.1) LFD1=LFDA
      IF (IFORC.EQ.0.OR.LENGTH.GT.IFTNY.OR.KEBACK.EQ.1) LLD1=LLDA
      IF (IFORC.EQ.0) GO TO 888
      IF (LENGTH.GT.IFTNY.OR.KEBACK.EQ.1) GO TO 603
      LFDD=LFDA+NY-1
      DO 601 I=LFDA,LFDD
  601 SERIES(I)=STCSI(I)
  603 LLAF=LLDAF-NY+1
      DO 602 I=LLAF,LLDAF
  602 SERIES(I)=STCSI(I)
C --- TAKE LOGARITHM OF B1 IF THE LOGARITHMIC MODEL IS SELECTED.
  888 MULADD = MA
      IF (MULADD.NE.2) GO TO 8
      CALL LOGAR(STCSI,LFDA,LLDAF)
    8 DO 600 I=LFD1,LLD1
      SERIES(I)=ORIG(LFDA+I-LFD1)
  600 CONTINUE
      DO 6 I = LFDA,LLDAF
    6 STO(I) = STCSI(I)
      IF (KSWV.EQ.0.OR.IFORC.EQ.0) GO TO 7778
      DO 7777 I=1,NY
      SERIES(LLD1+I)=SERIES(LLD1+I)*STPTD(LLD1+I)
      IF (LENGTH.GT.IFTNY.OR.KEBACK.EQ.1) GO TO 7777
      SERIES(LFDA+I-1)=SERIES(LFDA+I-1)*STPTD(LFDA+I-1)
 7777 CONTINUE
C --- APPLY A CENTERED NY-TERM MOVING AVERAGE.
 7778 IF (NY.EQ.12) MQU=1
      IF (NY.EQ.4) MQU=2
   80 IF (IWT.EQ.1) CALL WEIGHT(STCSI,STC,LFDA,LLDAF,MQU)
      IF (IWT.EQ.0) CALL AVERAG(STCSI,STC,LFDA,LLDAF,2,NY)
      MFDA = LFDA+NY2
      MFD1=LFD1+NY2
      MLDA = LLDAF-NY2
      KLDA = LLD1-NY2
      CALL DIVSUB(STSI,STCSI,STC,MFDA,MLDA)
C --- SECTION 1. APPLY AN F-TEST TO THE SI TO TEST FOR THE PRESENCE
C --- OF SEASONALITY.
      IF (KSECT.EQ.1.AND.KPART.EQ.2) CALL FTEST(STSI,MFD1,KLDA,NY,2,
     1   KPROPT)
      IF (KPRNT.LT.5) GO TO 12
C --- WRITE TREND CYCLE B2,C2,D2.
      LFD=LFD1+(NY/2)
      LLD=LLD1-(NY/2)
      IF(IFORC.NE.0.AND.LENGTH.GT.IFTNY)CALL TABLE(STC,LFD,LLD1,2,1,2,0)
      IF(IFORC.NE.0.AND.LENGTH.LE.IFTNY.AND.KEBACK.EQ.1)CALL TABLE(STC,L
     1FD,LLD1,2,1,2,0)
      IF(IFORC.NE.0.AND.LENGTH.LE.IFTNY.AND.KEBACK.EQ.0)CALL TABLE(STC,L
     1FD1,LLD1,2,1,2,0)
      IF (IFORC.EQ.0) CALL TABLE(STC,LFD,LLD,2,1,2,0)
C --- DIVIDE (SUBTRACT) ORIGINAL BY TREND CYCLE FOR SI RATIOS.
   12 IF (KPART.GT.2) GO TO 13
C --- PART B. REPLACE EXTREME SI RATIOS (DIFFERENCES)
      CALL SI(KSECT,MFDA,MLDA,NY,KPROPT,IFORC,LFD1,LLD1)
```

```
      GO TO 14                                                          00014330
C --- PART C AND D. COMPUTE A 5-TERM MOVING AVERAGE OF EACH MONTH       00014340
C --- (QUARTER) FOR ESTIMATE OF SEASONAL FACTORS.                       00014350
   13 CALL VSFB(MFDA,MLDA,NY)                                           00014360
      IF (KPROPT.NE.1) GO TO 14                                         00014370
C --- WRITE MODIFIED SI RATIOS (DIFFERENCES) C4 AND D4.                 00014380
      IF (IFORC.NE.0) CALL TABLE(STSI,LFD1,LLD1,4,1,1,0)                00014390
      IF (IFORC.EQ.0) CALL TABLE(STSI,LFD,LLD,4,1,1,0)                  00014400
C --- REPLACE MISSING VALUES AT EACH END DUE TO CENTERED NY-TERM MOVING 00014410
C --- AVERAGE.                                                          00014420
   14 CALL FORCST(STS,MFDA,MLDA,LLDAF,NY,1,0.0,1.0)                     00014430
C --- DIVIDE (SUBTRACT) THE ORIGINAL SERIES BY THE SEASONAL FACTORS TO  00014440
C --- GET A PRELIMINARY ESTIMATE OF THE SEASONALLY ADJUSTED SERIES.     00014450
      CALL DIVSUB(STCI,STCSI,STS,LFDA,LLDAF)                            00014460
      IF (KPROPT.NE.1) GO TO 15                                         00014470
C --- WRITE SEASONAL FACTORS B5,C5, AND D5.                             00014480
      CALL TABLE (STS,LFD1,LLD1,5,1,1,0)                                00014490
C --- WRITE SEASONALLY ADJUSTED SERIES B6,C6, AND D6.                   00014500
      CALL TABLE(STCI,LFD1,LLD1,6,1,2,0)                                00014510
C --- SECTION 2                                                         00014520
   15 KSECT = 2                                                         00014530
C --- APPLY VARIABLE TREND-CYCLE ROUTINE TO SEASONALLY ADJUSTED SERIES. 00014540
      CALL VTC(KPART)                                                   00014550
      IF (KEXOPT.NE.1) GO TO 16                                         00014560
C --- MODIFY CI VALUES BEFORE TREND-CYCLE SELECTED IF THE STRIKE OPTION 00014570
C --- IS SELECTED.                                                      00014580
      IF (KPART.NE.2) GO TO 16                                          00014590
C --- DIVIDE SEASONALLY ADJUSTED SERIES BY THE TREND-CYCLE FOR AN       00014600
C --- ESTIMATE OF THE IRREGULAR.                                        00014610
      CALL DIVSUB(STI,STCI,STC,LFDA,LLDAF)                              00014620
C --- IDENTIFY EXTREME IRREGULAR VALUES.                                00014630
      CALL XTRM(STI,LFDA,LLDAF,NY)                                      00014640
C --- REPLACE EXTREME VALUES.                                           00014650
      CALL REPLAC(STCI,TEMP,STWT,LFDA,LLDA,1)                           00014660
C --- APPLY THE VARIABLE TREND-CYCLE ROUTINE TO THE MODIFIED CI VALUES. 00014670
      CALL VTC(KPART)                                                   00014680
   16 IF (KPRNT.LE.4) GO TO 17                                          00014690
C --- WRITE THE TREND-CYCLE B7,C7, AND D7.                              00014700
      CALL TABLE(STC,LFD1,LLD1,7,1,2,0)                                 00014710
C --- DIVIDE B1 BY B7,CI BY C7, AND D1 BY D7 TO OBTAIN SI RATIOS.       00014720
   17 CALL DIVSUB(STSI,STCSI,STC,LFDA,LLDAF)                            00014730
      IF (KPART.NE.2) GO TO 18                                          00014740
C --- PART B. REPLACE EXTREME SI RATIOS (DIFFERENCES).                 00014750
      CALL SI(KSECT,LFDA,LLDAF,NY,KPROPT,IFORC,LFD1,LLD1)               00014760
      GO TO 19                                                          00014770
C --- PART C AND D. APPLY THE VARIABLE SEASONAL FACTOR ROUTINE FOR AN   00014780
C --- ESTIMATE OF THE SEASONAL FACTORS.                                 00014790
   18 CALL VSFA(LFDA,LLDA,NY)                                           00014800
      CALL VSFB(LFDA,LLDAF,NY)                                          00014810
      IF (KPART.EQ.4) GO TO 100                                         00014820
C --- PART C.                                                           00014830
      IF (KPROPT.NE.1) GO TO 19                                         00014840
C --- WRITE THE MODIFIED SI C9.                                         00014850
      CALL TABLE(STSI,LFD1,LLD1,9,1,1,0)                                00014860
C --- OBTAIN A PRELIMINARY SEASONALLY ADJUSTED SERIES.                  00014870
```

```
   19 CALL DIVSUB(STCI,STO,STS,LFDA,LLDAF)                              00014880
C --- DIVIDE SEASONALLY ADJUSTED SERIES BY THE TREND CYCLE TO OBTAIN A  00014890
C --- PRELIMINARY IRREGULAR SERIES.                                     00014900
      CALL DIVSUB(STI,STCI,STC,LFDA,LLDAF)                              00014910
      IF (KPRNT.LE.4) GO TO 21                                          00014920
C --- WRITE SEASONAL FACTORS B10 AND C10.                               00014930
      CALL TABLE(STS,LFD1,LLD1,10,1,1,0)                                00014940
      IF (KPROPT.EQ.2) GO TO 20                                         00014950
C --- WRITE SEASONALLY ADJUSTED SERIES B11 AND C11.                     00014960
      CALL TABLE(STCI,LFD1,LLD1,11,1,2,0)                               00014970
C --- WRITE IRREGULAR SERIES B13 AND C13.                               00014980
   20 CALL TABLE(STI,LFD1,LLD1,13,1,3,STDEV)                            00014990
   21 IF (KDWOPT.EQ.0) GO TO 29                                         00015000
C --- TRADING DAY REGRESSION OPTION.                                    00015010
      IF (MULADD.NE.2) GO TO 22                                         00015020
C --- IF THE LOGARITHMIC MODEL IS SELECTED TAKE ANTILOG OF THE IRREGULAR00015030
C --- AND TREAT SERIES MULTIPLICATIVELY.                                00015040
      MULADD = 0                                                        00015050
      CALL ANTILG(STI,LFDA,LLDAF)                                       00015060
C --- COMPUTE TRADING DAY REGRESSION FACTORS IF TRADING DAY OPTION      00015070
C --- SELECTED.                                                         00015080
   22 CALL TRADAY                                                       00015090
      IF (MULADD.EQ.1) GO TO 24                                         00015100
C --- IF MULTIPLICATIVE ADD ONE TO EACH DAILY FACTOR.                   00015110
      DO 23 I = 1,7                                                     00015120
   23 D(I) = D(I) + 1.0                                                 00015130
C --- COMPUTE MONTHLY FACTORS FROM 7 T.D. REGRESSION FACTORS.           00015140
      MOPT = 0                                                          00015150
      IF (KSWV.EQ.0) GO TO 24                                           00015160
      MOPT = 8                                                          00015170
   24 CALL DOW(D,STTD,LYR,LFDA,LLDAF,MOPT)                              00015180
      IF (KPRNT.EQ.0.OR.KPRNT.EQ.2) GO TO 25                            00015190
C --- WRITE TRADING DAY REGRESSION FACTORS B16 OR C16.                  00015200
      CALL TABLE(STTD,LFD1,LLD1,16,1,1,D)                               00015210
C --- SKIP BALANCE OF TRADING DAY REGR. IF * COMPUTE BUT DO NOT APPLY * 00015220
C --- OPTION IS SELECTED.                                               00015230
   25 IF (KDWOPT - 2) 28,27,27                                          00015240
C --- DISREGARD RESULTS OF THE F-TEST IN PART B. IF * APPLY ON BASIS OF 00015250
C --- F* OPTION SELECTED AND APPLY T.D. REGRESSION.                     00015260
C --- IN PART C APPLY T.D. REGR. OF NOT ON BASIS OF THE F-TEST.         00015270
C --- DIVIDE (SUBTRACT) IRREGULAR BY THE T.D. REGR. FACTORS TO OBTAIN AN00015280
C --- IRREGULAR FREE OF T.D. VARIATIONS.                               00015290
   27 CALL DIVSUB(STI,STI,STTD,LFDR,LLDAF)                              00015300
   28 IF (MA.NE.2) GO TO 29                                             00015310
C --- TAKE LOG OF IRREGULAR IF LOGARITHMIC MODEL IS SELECTED.           00015320
      MULADD = 2                                                        00015330
      CALL LOGAR(STI,LFDA,LLDAF)                                        00015340
C --- COMPUTE WEIGHTS FOR IRREGULAR COMPONENT.                          00015350
   29 CALL XTRM(STI,LFDA,LLDAF,NY)                                      00015360
      IF (KPRNT.LE.2) GO TO 30                                          00015370
C --- WRITE WEIGHTS FOR IRREGULAR COMPONENT B17 OR C17.                 00015380
      CALL TABLE (STWT,LFD1,LLD1,17,1,4,STDEV)                          00015390
C --- COMPUTE EXTREME COMPONENT.                                        00015400
   30 IF (MULADD.EQ.0) GO TO 32                                         00015410
C --- ADDITIVE AND LOGARITHMIC MODEL.                                   00015420
```

```
      DO 31 I = LFDA,LLDAF
   31 STEX(I) = STI(I) * (1.0-STWT(I))
      GO TO 34
C --- MULTIPLICATIVE MODEL.
   32 DO 33 I = LFDA,LLDAF
   33 STEX(I) = STI(I)/(1.0+STWT(I)*(STI(I)-1.0))
C --- IF THERE ARE PRIOR T.D. WEIGHTS AND REGR.T.D. WTS. B18,C18,B19,
C --- AND C19 ARE PRINTED. IF THERE ARE NO T.D. WEIGHTS THESE TABLES ARE
C --- SUPPRESSED. IF THERE ARE PRIOR T.D. WTS. BUT NO REG. WTS. THESE
C --- TABLES ARE SUPPRESSED SINCE B18 AND C18 ARE THE SAME AS A4, AND
C --- B19 AND C19 ARE THE SAME AS B1. IF THERE ARE REGR. WTS. BUT NO
C --- PRIOR T.D. WTS. B18 AND C18 ARE SUPPRESSED SINCE THEY ARE THE
C --- SAME AS B16 AND C16 RESPECTIVELY. B19 AND C19 ARE PRINTED.
   34 IF (KDWOPT.LE.1) GO TO 40
      IF (KSWV.NE.0) GO TO 35
      IF (KPART.EQ.2) GO TO 37
      IF (LOPT.EQ.0) GO TO 37
C --- COMBINE PRIOR AND REGRESSION WEIGHTS.
   35 DO 36 I = 1,7
   36 D(I) = D(I) + DWT(I) - 1.0
C --- COMPUTE MONTHLY FACTORS FROM THE COMBINED WEIGHTS.
      CALL DOW(D,STTD,LYR,LFDA,LLDAF,LOPT)
      IF (KPRNT.EQ.0.OR.KPRNT.EQ.2) GO TO 37
C --- WRITE TRADING DAY FACTORS FROM COMBINED WEIGHTS B18 OR C18.
      CALL TABLE(STTD,LFD1,LLD1,18,1,1,D)
C --- SET STCSI EQUAL TO THE RAW SERIES.
   37 DO 38 I = LFDA,LLDAF
   38 STCSI(I) = SERIES(I)
      IF (MULADD.EQ.2) MULADD = 0
      IF (KFMT.EQ.0) GO TO 39
C --- DIVIDE (SUBTRACT) BY THE PRIOR ADJUSTMENT SERIES.
      CALL DIVSUB(STCSI,STCSI,SPRIOR,LFD1,LLD1)
C --- DIVIDE (SUBTRACT) THE PRIOR MONTHLY ADJUSTED SERIES A3 OR A1 BY
C --- THE COMBINED TRADING DAY FACTORS TO OBTAIN THE CALENDAR ADJUSTED
C --- SERIES.
   39 CALL DIVSUB(STCSI,STCSI,STTD,LFDA,LLDAF)
      IF (MA.NE.2) GO TO 139
      MULADD = 2
      CALL LOGAR(STCSI,LFDA,LLDAF)
  139 IF (KPRNT.LE.2) GO TO 42
C --- WRITE THE ORIGINAL SERIES ADJUSTED FOR CALENDAR VARIATION B19 OR
C --- C19.
      CALL TABLE(STCSI,LFD1,LLD1,19,1,2,0)
      GO TO 42
C --- SET STCSI EQUAL TO STO.
   40 DO 41 I = LFDA,LLDAF
   41 STCSI(I) = STO(I)
C --- MODIFY THE ORIGINAL SERIES ADJUSTED FOR CALENDAR VARIATION TO
C --- ELIMINATE THE EXTREMES.
   42 CALL DIVSUB(STCSI,STCSI,STEX,LFDA,LLDAF)
      IF (KPRNT.LE.4) GO TO 43
C --- WRITE EXTREME SERIES B20 AND C20.
      CALL TABLE(STEX,LFD1,LLD1,20,1,3,0)
C --- INCREASE ITERATION STEP BY 1.
   43 KPART = KPART + 1
```

```
      KSECT = 1
      IF (KPRNT.LE.4) GO TO 44
C --- WRITE THE MODIFIED ORIGINAL SERIES C1 OR D1.
      CALL TABLE(STCSI,LFD1,LLD1,1,1,2,0)
   44 KPRNT = KPRNT + 1
      GO TO 80
  100 RETURN
      END
      SUBROUTINE X11PT3
      REAL*8 KSER
      COMMON /WORK / TEMP(372)
      COMMON /PRIOR/ KSER,KFORM(20),KFMT
      COMMON /TRADE/ DWT(7),D(7),SIGM,LOPT,KDWOPT,LCYR,LAYR,LFDC,LFDR
      COMMON /INPT1/ SERIES(372),ORIG(372)
      COMMON /INPT2/ SPRIOR(372)
      COMMON /WORK3/ STSIE(372)
      COMMON /OPT5 / MULADD,IYRT,MA,IWT
      COMMON /OPT9 / LFDA,LYR,LSTMO,LSTYR,LLDA,KFULSM,KPROPT,KCHOPT,
     1               KEXOPT,NY,IFORC,LLDAF,LLAF,LFD1,LLD1,NSOPT,IDEC,IAG
     2               ,JAREX,KEBACK
      COMMON /MQ1  / STS(372),STSI(372)
      COMMON /MQ2  / KPART,KSWV,KSECT,KPRNT,MCD
      COMMON /MQ4  / STC(372),STCI(372)
      COMMON /MQ5  / STCSI(372),STIME(372)
      COMMON /MQ6  / STWT(395)
      COMMON /MQ7  / STI(372)
      COMMON /MQ8  / STPTD(372),STTD(372)
      COMMON /MQ10 / STEX(372)
      COMMON /LARGE/ BIG
      COMMON /KCJH /LENGTH,IFTNY
      COMMON /KCSER/ CKHS(372)
      DIMENSION STSTD(372),STCIME(372),STOME(372),STCI2(372)
      EQUIVALENCE (STWT(1),STCIME(1)),(STCSI(1),STSTD(1)),
     2            (STCI2(1),STEX(1)),(STOME(1),STPTD(1))
      IFTNY=15*NY
C --- MULTIPLY (ADD) STSI BY STEX TO GET THE UNMODIFIED SI.
      CALL ADDMUL(STSIE,STSI,STEX,LFDA,LLDAF)
      IF (KPROPT.GT.3) GO TO 1
C --- WRITE UNMODIFIED SI RATIOS (DIFFERENCES) D8.
      CALL TABLE(STSIE,LFD1,LLD1,8,1,1,0)
C --- PERFORM ANALYSIS OF VARIANCE ON THE UNMODIFIED SI RATIOS.
      CALL TABLE(0,0,0,8,2,0,0)
    1 CALL FTEST(STSIE,LFD1,LLD1,NY,0,KPROPT)
      CALL KWTEST(STSIE,LFDA,LLDA,NY,KPROPT)
      CALL ADDMUL(STSIE,STSI,STEX,LFDA,LLDAF)
C --- PERFORM F-TEST FOR MOVING SEASONALITY.
      CALL MSTEST(STSIE,LFDA,LLDA,KPROPT,NY)
C --- PERFORM TEST FOR THE PRESENCE OF IDENTIFIABLE SEASONALITY.
      CALL COMBFT(KPROPT)
      EBAR = 0.0
      IF (MULADD.EQ.0) EBAR = 1.0
      IF (KPROPT.GT.3) GO TO 5
C --- IDENTIFY SI RATIOS (DIFFERENCES) WHICH ARE MODIFIED.
      DO 4 I = LFDA,LLDAF
      IF (STEX(I) - EBAR) 2,3,2
```

```
    2 TEMP(I) = STSI(I)
      GO TO 4
    3 TEMP(I) = BIG
    4 CONTINUE
C --- WRITE FINAL REPLACEMENT VALUES FOR EXTREME SI RATIOS (DIFFERENCES)
C --- D9.
      CALL TABLE(TEMP,LFD1,LLD1,9,1,5,0)
C --- COMPUTE YEAR AHEAD SEASONAL FACTORS.
    5 KLDA = LLDAF + NY
      CALL FORCST(STS,0,LLDAF,KLDA,NY,1,0.5,1.0)
C --- DIVIDE (SUBTRACT) D1 BY THE FINAL SEASONAL FACTORS TO OBTAIN A
C --- MODIFIED SEASONALLY ADJUSTED SERIES.
      CALL DIVSUB(STCI,STCSI,STS,LFDA,LLDAF)
      IF (MULADD.NE.2) GO TO 6
      MULADD = 0
C --- IF LOGARITHMIC MODEL SELECTED TAKE THE ANTILOG OF THE SEASONAL
C --- FACTORS.
      CALL ANTILG(STS,LFDA,KLDA)
      CALL ANTILG(STSI,LFDA,LLDAF)
      CALL ANTILG(STSIE,LFDA,LLDAF)
C --- WRITE THE FINAL SEASONAL FACTORS D10.
    6 CALL TABLE(STS,LFD1,LLD1,10,1,1,0)
      IJK=IDEC
      IF (MULADD.NE.1) IJK=3
      MULADD = MA
        DO 808 I=LFDA,LLDAF
  808      CKHS(I)=STCI(I)
      IF (KFULSM.EQ.0) GO TO 9
C --- SUMMARY MEASURES VERSION ONLY. REPLACE THE FINAL SEASONALLY
C --- ADJUSTED SERIES WITH D1.
      DO 7 I = LFDA,LLDAF
    7 STCI(I) = STCSI(I)
C --- APPLY THE VARIABLE TREND CYCLE ROUTINE TO D1.
      CALL VTC(KPART)
C --- REPLACE THE FINAL SEASONALLY ADJUSTED SERIES WITH A1.
      DO 8 I = LFDA,LLDAF
         CKHS(I)=SERIES(I)
    8 STCI(I) = SERIES(I)
      IF(MULADD.NE.2)GO TO 16
C --- IF THE LOGARITHMIC OPTION WAS SELECTED, TAKE THE ANTILOG
C --- OF THE COMPONENTS.
      CALL ANTILG(STC,LFDA,LLDAF)
      EBAR=1.0
      MULADD=0
C --- SET THE COMBINED FACTORS EQUAL TO THE SEASONAL FACTORS AND IF
C --- TRADING DAY IS REQUESTED MULTIPLY BY THE TRADING-DAY FACTORS.
   16 KLDA=LLD1+NY
      DO 19 I=LFDA,KLDA
      STSTD(I)=STS(I)
   19 CONTINUE
      IF(KDWOPT.LE.1)GO TO 20
      CALL ADDMUL(STSTD,STSTD,STTD,LFDA,KLDA)
      GO TO 13
   20 IF(KSWV.EQ.0)GO TO 13
      DO 21 I=LFDA,KLDA
```

```
      STTD(I)=STPTD(I)
   21 CONTINUE
      IF(IFORC.NE.0.AND.KEBACK.EQ.0.AND.LENGTH.LE.IFTNY)GO TO 28
      DO 29 I=LFDA,KLDA
      STPTD(I)=STPTD(I+NY)
   29 CONTINUE
   28 CALL ADDMUL(STSTD,STS,STPTD,LFDA,KLDA)
      GO TO 13
C --- FULL ADJUSTMENT VERSION ONLY. APPLY THE VARIABLE TREND CYCLE TO
C --- THE MODIFIED SEASONALLY ADJUSTED SERIES TO OBTAIN THE FINAL TREND
C --- CYCLE.
    9 CALL VTC(KPART)
      IF(MULADD.NE.2) GO TO 10
C --- IF THE LOGARITHMIC MODEL WAS SELECTED TAKE THE ANTILOG OF THE
C --- COMPONENTS.
      CALL ANTILG(STC,LFDA,LLDAF)
      EBAR = 1.0
      MULADD = 0
C --- DIVIDE (SUBTRACT) THE ORIGINAL SERIES BY THE SEASONAL AND TRADING
C --- DAY FACTORS TO OBTAIN THE FINAL SEASONALLY ADJUSTED SERIES.
   10 CALL DIVSUB(STCI,SERIES,STS,LFDA,LLDAF)
      KLDA = LLD1+NY
C --- SET THE COMBINED FACTORS EQUAL TO THE SEASONAL FACTORS AND IF
C --- TRADING DAY IS REQUESTED MULTIPLY BY THE TRADING-DAY FACTORS.
      DO 11 I = LFDA,KLDA
   11 STSTD(I) = STS(I)
      IF (KDWOPT.LE.1) GO TO 12
      CALL DIVSUB(STCI,STCI,STTD,LFDA,LLDAF)
      CALL ADDMUL(STSTD,STSTD,STTD,LFDA,KLDA)
      GO TO 13
   12 IF (KSWV.EQ.0) GO TO 13
      DO 112 I = LFDA,KLDA
  112 STTD(I) = STPTD(I)
      IF (IFORC.NE.0.AND.KEBACK.EQ.0.AND.LENGTH.LE.IFTNY) GO TO 114
      DO 113 I=LFDA,KLDA
  113 STPTD(I)=STPTD(I+NY)
  114 CALL DIVSUB(STCI,STCI,STPTD,LFDA,LLDAF)
      CALL ADDMUL(STSTD,STS,STPTD,LFDA,KLDA)
C --- DIVIDE (SUBTRACT) THE FINAL SEASONALLY ADJUSTED SERIES BY THE
C --- FINAL TREND-CYCLE TO OBTAIN THE FINAL IRREGULAR.
   13 CALL DIVSUB(STI,STCI,STC,LFDA,LLDAF)
      IF (KFMT.EQ.0) GO TO 14
C --- IF PRIOR ADJUSTMENT REQUESTED REMOVE IT FROM THE IRREGULAR.
      CALL DIVSUB(STI,STI,SPRIOR,LFD1,LLD1)
C --- WRITE THE FINAL SEASONALLY ADJUSTED SERIES D11.
   14 CALL TABLE(STCI,LFD1,LLD1,11,1,2,0)
      IF (KPROPT.GE.3) GO TO 15
C --- DO TEST FOR RESIDUAL SEASONALITY.
      CALL FTEST(STCI,LFD1,LLD1,NY,1,KPROPT)
   15 IF (IYRT.EQ.0) GO TO 17
C --- IF OPTION SELECTED ADJUST YEARLY TOTALS OF D11 TO EQUAL THE YEARLY
C --- TOTALS OF THE ORIGINAL SERIES.
      CALL QMAP(SERIES,STCI,STCI2,LFD1,LLD1,NY,IB,IE)
C --- WRITE SEASONALLY ADJUSTED SERIES WITH REVISED YEARLY TOTALS D11A.
      CALL TABLE(STCI2,IB,IE,11,2,2,0)
```

```
      IF (KPROPT.GE.3) GO TO 18
C --- APPLY THE TEST FOR RESIDUAL SEASONALITY
      CALL FTEST(STCI2,LFD1,LLD1,NY,1,KPROPT)
   17 IF (KPROPT.GE.3) GO TO 18
C --- WRITE THE FINAL TREND CYCLE D12.
      CALL TABLE(STC,LFD1,LLD1,12,1,2,0)
C --- WRITE THE FINAL IRREGULAR D13.
      CALL TABLE(STI,LFD1,LLD1,13,1,3,0)
C --- IF TRADING DAY IS APPLIED WRITE COMBINED SEASONAL AND TRADING
C --- DAY FACTORS D16.
   18 IF (KDWOPT.LE.1.AND.KSWV.EQ.0) GO TO 22
      CALL TABLE(STSTD,LFD1,LLD1,16,1,1,0)
C --- BEGIN PART E.
   22 KPART = 5
C --- COMPUTE THE MODIFIED ORIGINAL, SEASONALLY ADJUSTED, AND IRREGULAR
C --- SERIES BY REPLACING THOSE VALUES WHICH WERE ASSIGNED A WEIGHT OF
C --- ZERO IN TABLE C17.
      DO 27 I = LFDA,LLDAF
      IF (STWT(I).EQ.0.0) GO TO 23
      STOME(I) = SERIES(I)
      STIME(I) = STI(I)
      STCIME(I) = STCI(I)
      GO TO 27
C --- REPLACE THE SEASONALLY ADJUSTED SERIES WITH THE FINAL TREND CYCLE
   23 STCIME(I) = STC(I)
C --- REPLACE THE IRREGULAR BY ITS EXPECTED VALUE.
      STIME(I) = EBAR
C --- REPLACE THE ORIGINAL SERIES BY COMBINING THE TREND CYCLE, SEASONAL
C --- TRADING DAY AND PRIOR COMPONENTS.
      IF (MULADD.NE.1) GO TO 24
      STOME(I) = SERIES(I)-STI(I)
      GO TO 25
   24 STOME(I) = SERIES(I)/STI(I)
   25 IF (KFMT.EQ.0) GO TO 27
      IF (MULADD.NE.1) GO TO 26
      STCIME(I) = STC(I)+SPRIOR(I)
      GO TO 27
   26 STCIME(I) = STC(I)*SPRIOR(I)
   27 CONTINUE
      RETURN
      END
      SUBROUTINE X11PT4
      REAL*8 KSER
      REAL*4 IBAR(12),IBAR2(12),ISQ(12),ISD(12),IMBAR(12)
      COMMON /WORK / TEMP(372)
      COMMON /PRIOR/ KSER,KFORM(20),KFMT
      COMMON /TRADE/ DWT(7),D(7),SIGM,LOPT,KDWOPT,LCYR,LAYR,LFDC,LFDR
      COMMON /INPT1/ SERIES(372),ORIG(372)
      COMMON /INPT2/ TDBAR(12),TDSQ(12),SBAR(12),SBAR2(12),SSQ(12),
     1               SSD(12),IBAR,IBAR2,ISQ,ISD,IMBAR,CBAR(12),
     2               CBAR2(12),CSQ(12),CSD(12),OBAR(12),OBAR2(12),
     3               OSQ(12),OSD(12),OSQ2(12),SMBAR(12),SMBAR2(12),
     4               SMSD(12),OMBAR(12),CIMBAR(12),CIBAR(12),CIBAR2(12),
     5               CISD(12),SMIC(12),VI,VC,VS,VP,VTD,RV,ADRCI,ADRI,
     6               ADRC,ADRMCD,TMP1,TMP2
```

```
      COMMON /WORK2/ DUM(70),PBAR(12),PSQ(12),AUTOC(14)
      COMMON /WORK4/ RD1(30),RD2(81)
      COMMON /OPT5 / MULADD,IYRT,MA,IWT
      COMMON /OPT9 / LFDA,LYR,LSTMO,LSTYR,LLDA,KFULSM,KPROPT,KCHOPT,
     1               KEXOPT,NY,IFORC,LLDAF,LLAF,LFD1,LLD1,NSOPT,IDEC,IAG
      COMMON /MQ1  / STS(372),STSI(372)
      COMMON /MQ2  / KPART,KSWV,KSECT,KPRNT,MCD
      COMMON /MQ4  / STC(372),STCI(372)
      COMMON /MQ5  / STMCD(372),STIME(372)
      COMMON /MQ6  / STCIME(395)
      COMMON /MQ7  / STI(372)
      COMMON /MQ8  / STOME(372),STTD(372)
      COMMON /MQ9  / MFDA,MLDAF
      COMMON /MQ10 / STCI2(372)
      DIMENSION SPRIOR(372),TREND(372)
      EQUIVALENCE (SPRIOR(1),TDBAR(1)),(TREND(1),STTD(1))
      IF (KPROPT.GE.3) GO TO 28
      IF (KFULSM.NE.0) GO TO 29
C --- WRITE ORIGINAL SERIES MODIFIED FOR EXTREMES AND PRIORS E1.
      CALL TABLE(STOME,LFD1,LLD1,1,1,2,0)
C --- WRITE MODIFIED SEASONALLY ADJUSTED SERIES E2.
      CALL TABLE(STCIME,LFD1,LLD1,2,1,2,0)
C --- WRITE THE MODIFIED IRREGULAR SERIES E3.
      CALL TABLE(STIME,LFD1,LLD1,3,1,3,0)
C --- COMPUTE THE RATIOS (DIFFERENCES) OF THE ANNUAL TOTALS OF B1(A1)
C --- TO D11 AND E1 TO E2.
      J = 1
      JYR = LSTYR-LYR+1
      IF (LFDA.EQ.1) GO TO 24
      J = NY+1
      JYR = JYR-1
   24 IF (LSTMO.NE.NY) JYR = JYR-1
      IF (IFORC.NE.0) JYR=JYR+1
      DO 27 JJ = 1,JYR
      TOTO = 0.0
      TOTCI = 0.0
      TOTOM = 0.0
      TOTCIM = 0.0
      K = J+NY-1
      DO 25 L = J,K
      TOTO = TOTO+SERIES(L)
      TOTOM = TOTOM+ STOME(L)
      TOTCI = TOTCI+ STCI(L)
   25 TOTCIM = TOTCIM+STCIME(L)
      IF (MULADD.EQ.1) GO TO 26
      RD1(JJ) = TOTO/TOTCI*100.0
      RD2(JJ) = TOTOM/TOTCIM*100.0
      GO TO 27
   26 RD1(JJ) = TOTO-TOTCI
      RD2(JJ) = TOTOM-TOTCIM
   27 J = J+NY
C --- WRITE RATIOS (DIFFERENCES)OF ANNUAL TOTALS E4.
      CALL TABLE(RD1,1,JYR,4,1,1,RD2)
      GO TO 29
   28 IF (KPROPT.NE.4) GO TO 30
```

```
C --- COMPUTE THE AVERAGE PERCENT CHANGES (DIFFERENCES) IN THE ORIGINAL 00018610
C --- SERIES.                                                           00018620
   29 MFDA = LFD1+1                                                      00018630
      CALL CHANGE(SERIES,TEMP,MFDA,LLDAF)                                00018640
C --- WRITE THE CHANGES IN THE ORIGINAL SERIES E5.                      00018650
      CALL TABLE(TEMP,MFDA,LLD1,5,1,1,0)                                 00018660
      IF (KFULSM.NE.0) GO TO 30                                          00018670
C --- CALCULATE THE CHANGES IN THE SEASONALLY ADJUSTED SERIES.           00018680
      CALL CHANGE(STCI,TEMP,MFDA,LLDAF)                                  00018690
C --- WRITE THE CHANGES IN THE SEASONALLY ADJUSTED SERIES E6.            00018700
      CALL TABLE(TEMP,MFDA,LLD1,6,1,1,0)                                 00018710
      IF (IYRT.EQ.0) GO TO 30                                            00018720
C --- IF THE YEARLY TOTALS OF THE SEASONALLY ADJUSTED SERIES ARE         00018730
C --- MODIFIED CALCULATE THE CHANGES IN THE MODIFIED SERIES.             00018740
      IFY=2                                                              00018750
      IF(LFD1.GT.1)IFY=2+NY                                              00018760
      LFY=LLD1/NY*NY                                                     00018770
      CALL CHANGE(STCI2,TEMP,IFY,LFY)                                    00018780
C --- WRITE THE CHANGES E6.A.                                            00018790
      CALL TABLE(TEMP,IFY,LFY,6,2,1,0)                                   00018800
   30 CONTINUE                                                           00018810
C --- PART F.                                                            00018820
      KPART = 6                                                          00018830
C --- COMPUTE MEASURES FOR THE SUMMARY MEASURES TABLE.                   00018840
      IF (KFMT.NE.0) GO TO 32                                            00018850
      DO 31 I = 1,NY                                                     00018860
      PBAR(I) = 0.0                                                      00018870
   31 PSQ(I) = 0.0                                                       00018880
      VP = 0.0                                                           00018890
      GO TO 33                                                           00018900
   32 CALL SUMRY(SPRIOR,PBAR,0,PSQ,0,2,LFD1,LLD1)                        00018910
      VPP = VARS(SPRIOR,LFD1,LLD1,0)                                     00018920
      VP = VPP                                                           00018930
   33 IF (KDWOPT.LE.1.AND.KSWV.EQ.0) GO TO 34                            00018940
      CALL SUMRY(STTD,TDBAR,0,TDSQ,0,2,LFD1,LLD1)                        00018950
      VTD = VARS(STTD,LFD1,LLD1,1)                                       00018960
      GO TO 36                                                           00018970
   34 VTD = 0.0                                                          00018980
      DO 35 I = 1,NY                                                     00018990
      TDBAR(I) = 0.0                                                     00019000
   35 TDSQ(I) = 0.0                                                      00019010
   36 CALL SUMRY(STI,IBAR,IBAR2,ISQ,ISD,0,LFD1,LLD1)                     00019020
      CALL AVEDUR(STI,LFD1,LLD1,ADRI)                                    00019030
      VI = VARS(STI,LFD1,LLD1,1)                                         00019040
      CALL SUMRY(STS,SBAR,SBAR2,SSQ,SSD,0,LFD1,LLD1)                     00019050
      VS = VARS(STS,LFD1,LLD1,1)                                         00019060
      CALL SUMRY(STC,CBAR,CBAR2,CSQ,CSD,0,LFD1,LLD1)                     00019070
C --- REMOVE LINEAR TREND (OR LINEAR PERCENTAGE GROWTH IF THE SERIES IS 00019080
C --- MULTIPLICATIVE) FROM TREND CYCLE.                                  00019090
      IF (MULADD.NE.1) CALL LOGAR(STC,LFD1,LLD1)                         00019100
      TMP1 = -LFD1-LLD1                                                  00019110
      TMP2 = 0.0                                                         00019120
      DO 37 I = LFD1,LLD1                                                00019130
   37 TMP2 = TMP2+STC(I)*(2.0*I+TMP1)                                    00019140
      FN = LLD1-LFD1+1                                                   00019150
```

```
      TMP1 = 6.0*TMP2/(FN*(FN*FN-1.0))                                  00019160
      DO 38 I = LFD1,LLDAF                                              00019170
   38 TREND(I) = TMP1*(I-LFDA+1.0)                                      00019180
      IF (MULADD.EQ.1) GO TO 39                                         00019190
      CALL ANTILG(STC,LFD1,LLD1)                                        00019200
      CALL ANTILG(TREND,LFD1,LLD1)                                      00019210
   39 CALL DIVSUB(TEMP,STC,TREND,LFD1,LLD1)                             00019220
      CALL AVEDUR(STC,LFD1,LLD1,ADRC)                                   00019230
      VC = VARS(TEMP,LFD1,LLD1,0)                                       00019240
      CALL SUMRY(SERIES,OBAR,OBAR2,OSQ,OSD,0,LFD1,LLD1)                 00019250
      CALL DIVSUB(TEMP,SERIES,TREND,LFD1,LLD1)                          00019260
      VO = VARS(TEMP,LFD1,LLD1,0)/100.0                                 00019270
      VP = VP/VO                                                        00019280
      VTD = VTD/VO                                                      00019290
      VC  = VC/VO                                                       00019300
      VS  = VS/VO                                                       00019310
      VI  = VI/VO                                                       00019320
      RV = VP+VTD+VC+VS+VI                                              00019330
      DO 40 I = 1,NY                                                    00019340
      OSQ2(I) = ISQ(I)+CSQ(I)+SSQ(I)+PSQ(I)+TDSQ(I)                     00019350
      ISQ(I) = ISQ(I)/OSQ2(I)                                           00019360
      CSQ(I) = CSQ(I)/OSQ2(I)                                           00019370
      SSQ(I) = SSQ(I)/OSQ2(I)                                           00019380
      PSQ(I) = PSQ(I)/OSQ2(I)                                           00019390
      TDSQ(I) = TDSQ(I)/OSQ2(I)                                         00019400
      OSQ2(I) = OSQ2(I)/OSQ(I)                                          00019410
C --- COMPUTE I/C RATIOS.                                               00019420
   40 SMIC(I) = IBAR(I)/CBAR(I)                                         00019430
      CALL SUMRY(STCI,CIBAR,CIBAR2,0,CISD,1,LFD1,LLD1)                  00019440
      CALL AVEDUR(STCI,LFD1,LLD1,ADRCI)                                 00019450
      CALL SUMRY(STOME,OMBAR,0,0,0,3,LFD1,LLD1)                         00019460
      CALL SUMRY(STIME,IMBAR,0,0,0,3,LFD1,LLD1)                         00019470
      CALL SUMRY(STCIME,CIMBAR,0,0,0,3,LFD1,LLD1)                       00019480
C --- COMPUTE MCD.                                                      00019490
      MCD = NY                                                          00019500
   41 IF (SMIC(MCD).GE.1.0) GO TO 141                                   00019510
      IF (MCD.EQ.1) GO TO 42                                            00019520
      MCD = MCD-1                                                       00019530
      GO TO 41                                                          00019540
  141 MCD = MCD+1                                                       00019550
      IF (MCD.GT.NY) MCD = NY                                           00019560
   42 N = MCD                                                           00019570
      IF (N.GT.6) N =6                                                  00019580
      M = 2-N+N/2*2                                                     00019590
C --- APPLY THE MCD MOVING AVERAGE.                                     00019600
      CALL AVERAG(STCI,STMCD,LFDA,LLDAF,M,N)                            00019610
      MFDA = LFD1+N/2                                                   00019620
      MLDA = LLD1-N/2                                                   00019630
      MLDAF = LLDAF-N/2                                                 00019640
      IF (MLDAF.GT.LLD1) MLDAF = LLD1                                   00019650
      CALL SUMRY(STMCD,SMBAR,SMBAR2,0,SMSD,1,MFDA,MLDA)                 00019660
      CALL AVEDUR(STMCD,MFDA,MLDA,ADRMCD)                               00019670
      IF (KPROPT.GE.3) GO TO 43                                         00019680
      KPART = 6                                                         00019690
C --- WRITE THE MCD MOVING AVERAGE F1.                                  00019700
```

```
      CALL TABLE(STMCD,MFDA,MLDA,1,1,2,0)
   43 DO 143 I = MFDA,MLDAF
  143 STMCD(I) = TEMP(I)
C --- CALCULATE THE AUTOCORRELATION FUNCTION OF THE IRREGULARS FOR SPANS
C --- 1 TO NY+2.
      NN = 2-MULADD
      EBAR = 1-MULADD
      TMP1 = VARIAN(STI,LFD1,LLD1,NN)/FN
      N = NY+2
      DO 45 I = 1,N
      AUTOC(I) = 0.0
      IJ = LFD1+I
      DO 44 J = IJ,LLD1
   44 AUTOC(I) = AUTOC(I)+(STI(J)-EBAR)*(STI(J-I)-EBAR)
   45 AUTOC(I) = AUTOC(I)/((FN-I)*TMP1)
      CALL F3CAL(IFAIL)
      IF (KPROPT.EQ.5) GO TO 46
C --- WRITE TABLES F2 AND F3.
      CALL FGEN(0)
   46 IF (KPROPT.EQ.5.AND.IFAIL.EQ.0) GO TO 47
      CALL FGEN(1)
   47 KPART = 7
      CALL QCONTR(MA,NY)
      RETURN
      END
      SUBROUTINE X11PT5
C --- THIS SUBROUTINE PRINTS THE CHARTS.
      DIMENSION TMP1(30),TMP2(30),TMP3(30)
      COMMON /WORK / TEMP(372)
      COMMON /INPT1/ SERIES(372),ORIG(372)
      COMMON /WORK3/ STSIE(372)
      COMMON /SCALE/ BIG,SMALL
      COMMON /OPT5 / MULADD,IYRT,MA,IWT
      COMMON /OPT9 / LFDA,LYR,LSTMO,LSTYR,LLDA,KFULSM,KPROPT,KCHOPT,
     1               KEXOPT,NY,IFORC,LLDAF,LLAF,LFD1,LLD1,NSOPT,IDEC,IAG
     2               ,JAREX,KEBACK
      COMMON /MQ1  / STS(372),STSI(372)
      COMMON /MQ4  / STC(372),STCI(372)
      COMMON /MQ5  / STMCD(372),STIME(372)
      COMMON /MQ7  / STI(372)
      COMMON /MQ8  / STOME(372),CYCLE(372)
      COMMON /MQ9  / MFDA,MLDA,MMLDA
      EQUIVALENCE (TMP1(1),TEMP(1)),(TMP2(1),TEMP(31)),
     1            (TMP3(1),TEMP(61))
      LENGTH=LLDA-LFDA+1
C --- PRINT PLOT OF THE ORIGINAL SERIES AND THE MODIFIED ORIGINAL
C --- SERIES G1.
      BIG = SERIES(LFDA)
      SMALL = BIG
      CALL TABLE(0,0,0,1,0,0,0)
      LFD1=LFDA+NY
      IFTNY=15*NY
      IF (LENGTH.LE.IFTNY.AND.IFORC.NE.0.AND.KEBACK.EQ.0) LLD1=LLDA+NY
      IF (IFORC.EQ.0.OR.LENGTH.GT.IFTNY.OR.KEBACK.EQ.1) LFD1=LFDA
      IF (IFORC.EQ.0.OR.LENGTH.GT.IFTNY.OR.KEBACK.EQ.1) LLD1=LLDA
```

```
      CALL CHART(SERIES,STOME,0,LFD1,LLD1,LFDA,LLDAF,2,MULADD,LYR,NY,0,20
     1)
C --- PRINT THE PLOT OF THE SEASONALLY ADJUSTED SERIES AND THE  TREND
C --- CYCLE G2.
    1 CALL TABLE(0,0,0,2,0,0,0)
      BIG = STCI(LFDA)
      SMALL = BIG
      CALL CHART(STCI,STC,0,LFD1,LLD1,LFD1,LLD1,2,MULADD,LYR,NY,0,0)
      IF (KFULSM.EQ.1.AND.KCHOPT.EQ.0) RETURN
C --- PRINT NY SEASONAL CHARTS G3.
      EBAR = 0.0
      IF (MULADD.NE.1) EBAR = 1.0
      BIG = STSIE(LFDA)
      SMALL = BIG
      B = ABS(STI(LFDA)-EBAR)
      DO 2 I = LFD1,LLD1
      BB = ABS(STI(I)-EBAR)
      B = AMAX1(B,BB)
      BIG = AMAX1(BIG,STSIE(I))
    2 SMALL = AMIN1(SMALL,STSIE(I))
      IF (KFULSM.EQ.1) GO TO 5
      DO 4 I = 1,NY
      CALL TABLE(0,0,0,3,0,0,0)
      K = 0
      L = I
      IF (L.LT.LFDA) L = L+NY
      DO 3 J = L,LLD1,NY
      N = J
      K = K+1
      TMP1(K) = STS(J)
      TMP3(K) = STSI(J)
    3 TMP2(K) = STSIE(J)
      M = K+1
      TMP1(M) = STS(N+NY)
      KYR = LYR
      IF (L.NE.I) KYR = KYR+1
      IJ=1
      IF(IFORC.NE.0.AND.LENGTH.LE.IFTNY.AND.KEBACK.EQ.0)IJ=2
    4 CALL CHART(TMP1,TMP2,TMP3,IJ,K,IJ,M,3,1,KYR,NY,I,1)
      IF (KCHOPT.EQ.0) RETURN
C --- PRINT SEASONAL CHART G4.
      CALL TABLE(0,0,0,4,0,0,0)
      KLDA = LLDA+NY
      CALL CHART(STS,STSIE,STSI,LFD1,LLD1,LFD1,KLDA,3,1,LYR,NY,0,1)
C --- PLOT IRREGULARS G5.
    5 CALL TABLE(0,0,0,5,0,0,0)
      BIG = EBAR+B
      SMALL = EBAR-B
      IF (SMALL.LT.0.0.AND.MULADD.EQ.0) SMALL = 0.0
      CALL CHART(STI,STIME,0,LFD1,LLD1,LFD1,LLD1,2,1,LYR,NY,0,1)
      IF (KCHOPT.LT. 2) RETURN
C --- CALCULATE AND PLOT THE KOLMOGOROV-SMIRNOV SIGNIFICANCE TESTS G6.
      CALL TABLE(0,0,0,6,0,0,0)
      CALL PERIOD
      RETURN
```

```
      END
      SUBROUTINE PUNCH(X,MFDA,MLDA,J,IPOW)
C --- THIS SUBROUTINE OUTPUTS SERIES ON FILE MP.
      REAL*8 IDNT
      COMMON /UNITS/ MT,MT2,MT1,MP,NG,NF
      COMMON /WORK2/ TMP(12),ITMP(12),TMP2(24),TEMP(60)
      COMMON /OPT3 / IDNT(9),LDEC(9),JFORM(20),JFMT
      COMMON /OPT4 / NUMB(9),IHOLD(9),NOP
      COMMON /OPT9 / LFDA,LYR,LSTMO,LSTYR,LLDA,IDUM(4),NY,IFORC,LLDAF,
     1               LLAF
      DIMENSION X(1)
      IHOLD(J) = 1
      CALL IFMTS(JFORM,JFMT,LDEC(J),NY)
      KFMT = JFMT+1
      IF (LDEC(J).EQ.0.AND.JFMT.NE.1) KFMT = KFMT+10
      IC = 1
      NY1 = NY
      IF (JFMT.NE.4.OR.NY.EQ.12) GO TO 1
      IC = 12/NY
      NY1 = 12
    1 NYR = LYR
      IF (MFDA.GT.LFDA) NYR=NYR+1
      KLDA = MLDA
      IF (JFMT.GT.2) GO TO 3
      IF (JFMT.EQ.1) GO TO 3
      KLDA = ((MLDA-1)/NY+1)*NY
      IF (KLDA.EQ.MLDA) GO TO 3
      KFDA = MLDA+1
      DO 2 I = KFDA,KLDA
    2 X(I) = 0.0
    3 M = 1
      IF (MFDA.GT.NY) M = (MFDA-1)/NY*NY+1
        KHCM=M
      N = M+NY1-1
      L = NY1
      K = MFDA-1
      KK = MFDA-M
      IF (KK.EQ.0) GO TO  5
      DO 4 I = 1,K
      TMP2(I) = X(I)
    4 X(I) = 0.0
    5 IF (N.LT.KLDA) GO TO 6
      N = KLDA
      L = N-M+1
    6 DO 7 I = M,N
    7 TMP(I-M+1) = X(I)
      IF (IPOW.EQ.0) GO TO 9
      DO 8 I = 1,L
    8 TMP(I) = 100.0*TMP(I)
    9 IF (LDEC(J).NE.0) GO TO 11
      DO 10 I = 1,L
      TP = TMP(I)
   10 ITMP(I) = TP+SIGN(0.5,TP)
   11 GO TO (12,13,12,17,17,17,12,17,23,23,18,13,18,19,17,19,12,17,23,
     1 23),KFMT
```

```
   12 WRITE(MP,JFORM) (TMP(I),I = 1,L),NYR,IDNT(J)
      GO TO 20
   13 IF (IPOW.EQ.0) GO TO 15
      DO 14 I = 1,KLDA
   14 X(I) = 100.0*X(I)
   15 WRITE(MP,JFORM) (X(I),I = KHCM,KLDA)
      IF (IPOW.EQ.0) GO TO 21
      DO 16 I = 1,KLDA
   16 X(I) = X(I)/100.0
      GO TO 21
   17 WRITE(MP,JFORM) IDNT(J),NYR,(TMP(I),I = 1,L)
      GO TO 20
   18 WRITE(MP,JFORM) (ITMP(I),I = 1,L),NYR,IDNT(J)
      GO TO 20
   19 WRITE(MP,JFORM) IDNT(J),NYR,(ITMP(I),I = 1,L)
   20 IF (N.EQ.KLDA) GO TO 21
      M = M+NY1
      N = N+NY1
      NYR = NYR+IC
      GO TO 5
   21 IF (KK.EQ.0) GO TO 23
      DO 22 I = 1,K
   22 X(I) = TMP2(I)
   23 RETURN
      END
      SUBROUTINE TRADAY
C --- THIS ROUTINE COMPUTES SEVEN DAILY WEIGHTS FROM THE DATA BY
C --- REGRESSION METHODS.
      REAL*4 MATRX(6,7)
      COMMON /UNITS/ MT,NP,NW,MP,NG,NFORM
      COMMON /WORK / STMP(372)
      COMMON /TRADE/ DP(7),DR(7),SIGM,LOPT,KDWOPT,LYR,LAYR,KFDA,LFDR
      COMMON /TESTS/ F1,F2,F3,F,CHI,P1,P2,P3,PROB,P5,TEST1,TEST2,IQFAIL
      COMMON /OPT5 / MULADD,IYRT,MA,IWT
      COMMON /OPT9 / ID(4),KLDA,KFULSM,KPROPT,IE(2),NY,IFORC,IDUM(8),KEB
     1ACK
      COMMON /MQ8  / SPTD(372),STDR(372)
      COMMON /MQ7  / ARRAY(372)
      COMMON /WORK3/ KARRAY(372)
      COMMON /MQ2  / KPART,KSWV,KSECT,KPRNT,MCD
      COMMON /WORK4/ TMEAN(22),TC(22),X(7),C(6),S(7),V(6),TP(7),DC(7),
     1               T1(7),SETDAF(3),WT(3),STAR1(7),STAR2(7)
      COMMON /LARGE/ BIG
      COMMON /KCJH /LENGTH,IFTNY
      DIMENSION KDAYS(12),MDAYS(12),ADJ(8)
      DATA  ADJ /31.,30.,29.,0.,30.4375,30.4375,30.4375,0./
      DATA  BLK,ST1,ST2,ST3/3H   ,3H*  ,3H** ,3H***/
      DATA  KDAYS /3,0,3,2,3,2,3,3,2,3,2,3/
      DATA  MDAYS /0,3,3,6,1,4,6,2,5,0,3,5/
 8200 FORMAT (22X,65HCOMBINED       PRIOR     REGRESSION    ST.ERROR
     1  T          T/
     223X,52HWEIGHT       WEIGHT       COEFF.     (COMB.WT.)        (,I1,
     317H)      (PRIOR WT.)/
     411X,9H   MONDAY,F9.3,7X,F5.3,3F12.3,A3,2X,F7.3,A3/
     511X,9H  TUESDAY,F9.3,7X,F5.3,3F12.3,A3,2X,F7.3,A3/
```

```
     611X,9HWEDNESDAY,F9.3,7X,F5.3,3F12.3,A3,2X,F7.3,A3/
     711X,9H THURSDAY,F9.3,7X,F5.3,3F12.3,A3,2X,F7.3,A3/
     811X,9H   FRIDAY,F9.3,7X,F5.3,3F12.3,A3,2X,F7.3,A3/
     911X,9H SATURDAY,F9.3,7X,F5.3,3F12.3,A3,2X,F7.3,A3/
     911X,9H   SUNDAY,F9.3,7X,F5.3,3F12.3,A3,2X,F7.3,A3)
 8201 FORMAT(/,14X,'THE STARS INDICATE THE COMBINED WT. IS SIGNIFICANTLY
     1 DIFFERENT',/,14X,'FROM ',I1,' OR THE PRIOR WT. THE SIGNIFICANCE L
     2EVELS ARE',/,14X,'3 STARS (0.1 PERCENT), 2 STARS (1 PERCENT), 1 ST
     3AR (5 PERCENT),',/,14X,'AND NO STARS INDICATES NOT SIGNIFICANT AT
     4THE 5 PERCENT LEVEL')
 8203 FORMAT (//21X,9HSOURCE OF, 9X, 6HSUM OF,5X,7HDGRS.OF,10X,4HMEAN/
     122X,8HVARIANCE,9X,7HSQUARES,4X,7HFREEDOM,9X,6HSQUARE,9X,1HF//
     220X,10HREGRESSION,F17.3,7X,2H 6,F17.3,F12.3,A3/
     325X,5HERROR,F17.3,I9,F17.3/
     425X,5HTOTAL,F17.3,I9,/)
 8204 FORMAT (26X,A3,63H RESIDUAL TRADING DAY VARIATION PRESENT AT THE 1
     1 PER CENT LEVEL/)
 8205 FORMAT (11X,86HSTANDARD ERRORS OF TRADING DAY ADJUSTMENT FACTORS D
     1ERIVED FROM REGRESSION COEFFICIENTS/15X,15H31-DAY MONTHS- ,F8.2/
     215X,15H30-DAY MONTHS- ,F8.2/15X,15H29-DAY MONTHS- ,F8.2/15X,23H28-
     3DAY MONTHS-       .00)
 8206 FORMAT(30X,'RESIDUAL TRADING DAY VARIATION NOT PRESENT AT THE 1 PE
     1RCENT LEVEL',/)
      LFDA = KFDA
      LLDA = KLDA
      IF (IFORC.NE.0) LFDA=LFDA+NY
      IF (IFORC.NE.0) LLDA=LLDA+NY
      IF(KEBACK.EQ.1.OR.LENGTH.GT.IFTNY)LFDA=KFDA
      IF(KEBACK.EQ.1.OR.LENGTH.GT.IFTNY)LLDA=KLDA
      TKON = 100.
      Z1 = 28.25
      Z2 = 28.25
      IF (KSWV.EQ.0) GO TO 81
      Z1 = 28.
      Z2 = 29.
   81 KON = 1
      IF (MULADD.NE.1) GO TO 199
      TKON = 0.
      KON = 0
  199 IF (KPART.EQ.3) GO TO 300
C --- DETERMINE DAY-OF-WEEK OF FIRST DAY OF FIRST MONTH OF SERIES
C ---     DETERMINE D-O-W OF JAN.1 OF LYR. 1=MON., 7=SUN.
      N = LYR + 12 - ((LYR + 11)/28) * 28
      N = N + (N-1)/4
      N = N - ((N-1)/7) * 7
C ---    D-O-W FOR ACTUAL FIRST MONTH OF SERIES
      J = LFDA - (LFDA-1)/12*12
      N = N + MDAYS(J)
C ---    ADD A DAY IF FIRST MONTH IS IN LEAP YEAR AND AFTER FEB.
      L = LYR - 1
      IF (J-2) 11,11,12
   12 L = LYR
      IF (LYR-LYR/4*4) 11,13,11
   13 N = N + 1
   11 IF (N - 7) 10, 10, 9
```

```
    9 N = N - 7                                                         00022450
   10 M = N                                                             00022460
C --- STORE TYPE CODE FOR EACH MONTH IN KARRAY.ALSO ACCUMULATE IRREGULAR00022470
C ---   AND COUNT BY TYPE TO GET TYPE MEAN                              00022480
      DO 8 I = LFDA, LLDA                                               00022490
      STMP(I) = BIG                                                     00022500
      K = KDAYS(J)                                                      00022510
C --- TEST FOR FEB.                                                     00022520
      IF (K) 4, 1, 4                                                    00022530
C --- TEST FOR LEAP YEAR FEB.                                           00022540
    1 L = L + 1                                                         00022550
      IF (L - (L/4) * 4) 2, 3, 2                                        00022560
C --- NON-LEAP YEAR FEB. TYPE CODE IS 15                                00022570
    2 MTYPE = 15                                                        00022580
      GO TO 14                                                          00022590
C --- LEAP YEAR FEB. TYPE CODE IS 16-22 FOR 1ST DAY MON. THRU SUN.      00022600
    3 MTYPE = M + 15                                                    00022610
      K = 1                                                             00022620
      GO TO 14                                                          00022630
C --- TYPE CODES FOR NON-FEB ARE 1-7  AND  8-14 FOR 30 AND 31 DAY MOS.  00022640
    4 MTYPE = (K - 2) * 7 + M                                           00022650
C --- STORE TYPE CODE IN KARRAY                                         00022660
   14 KARRAY(I) = MTYPE                                                 00022670
C --- FIND FIRST DAY FOR NEXT MONTH IN SERIES                           00022680
      M = M + K                                                         00022690
      IF (M - 7) 6, 6, 5                                                00022700
    5 M = M - 7                                                         00022710
    6 J = J + 1                                                         00022720
      IF (J - 12) 8, 8, 7                                               00022730
    7 J = 1                                                             00022740
    8 CONTINUE                                                          00022750
C --- COMPUTE SIGMA LIMITS AND IDENTIFY EXTREME IRREGULARS              00022760
      KSTD = 1                                                          00022770
   21 DO 22 I = 1, 22                                                   00022780
      TC(I) = 0.                                                        00022790
   22 TMEAN(I) = 0.                                                     00022800
      TK = 0.                                                           00022810
      TSD = 0.                                                          00022820
      DO 24 I = LFDA, LLDA                                              00022830
      M = KARRAY(I)                                                     00022840
      IF (M - 15) 23, 23, 24                                            00022850
   23 TMEAN(M) = TMEAN(M) + ARRAY(I)                                    00022860
      TC(M) = TC(M) + 1.                                                00022870
      TK = TK + 1.                                                      00022880
   24 CONTINUE                                                          00022890
      DO 25 I = 1, 15                                                   00022900
   25 TMEAN(I) = TMEAN(I)/TC(I)                                         00022910
C --- COMPUTE SQ.DEV. OF IRR.FM.TYPE-MEANS FOR NON-LEAP YEAR FEBS.      00022920
      DO 28 I = LFDA, LLDA                                              00022930
      M = KARRAY(I)                                                     00022940
      IF (M - 15) 27, 27, 28                                            00022950
   27 TSD = TSD + (ARRAY(I) - TMEAN(M))*(ARRAY(I)-TMEAN(M))             00022960
   28 CONTINUE                                                          00022970
      TSD = SQRT(TSD/TK) * SIGM                                         00022980
C --- IDENTIFY EXTREME IRREGULARS BY ADDING 22 TO TYPE CODE             00022990
```

```
      DO 35 I = LFDA, LLDA
      M = KARRAY(I)
      IF (M - 22) 29, 29, 35
   29 IF (M - 15) 30, 30, 33
   30 IF (ABS(ARRAY(I) - TMEAN(M))-TSD) 35,35,32
   33 IF (ABS(ARRAY(I)-TKON)-TSD) 35,35,32
   32 KARRAY(I) = KARRAY(I) + 22
      STMP(I) = ARRAY(I)
   35 CONTINUE
      GO TO ( 36, 37), KSTD
   36 KSTD = 2
      GO TO 21
  300 DO 302 I = LFDA, LLDA
  333 IF (KARRAY(I)-22) 302,302,301
  301 KARRAY(I) = KARRAY(I) - 22
      GO TO 333
  302 CONTINUE
      KSTD = 1
  310 TSD = 0.
      TK = 0.
      DO 304 I = LFDA, LLDA
      IF (KARRAY(I) - 22) 303,303,304
  303 TSD = TSD + (ARRAY(I) - STDR(I))*(ARRAY(I)-STDR(I))
      TK = TK + 1.
  304 CONTINUE
      TSD = SQRT(TSD/TK) * SIGM
      DO 308 I = LFDA, LLDA
      STMP(I) = BIG
      IF (ABS(ARRAY(I)-STDR(I))-TSD) 308,308,307
  307 KARRAY(I) = KARRAY(I) + 22
      STMP(I) = ARRAY(I)
  308 CONTINUE
      GO TO (309, 37), KSTD
  309 KSTD = 2
      GO TO 310
C --- REGRESSION COMPUTATION
   37 IF (KPRNT.LT.3) GO TO 39
      CALL TABLE(STMP,LFDA,LLDA,14,1,5,0)
      CALL TABLE(STMP,LFDA,LLDA,15,1,5,0)
   39 SUMY = 0.
      SUMYSQ = 0.
      DO 40 I = 1, 6
      DO 40 J = 1, 7
   40 MATRX(I,J) = 0.
      DO 58 I = LFDA, LLDA
      DO 41 J = 1, 7
   41 X(J) = 0.
      M = KARRAY(I)
      Y = ARRAY(I)
      IF (MULADD.NE.1) Y = Y*100.0
       IF (M-7) 251,251,201
  201 M = M-7
      IF (M-8) 245,210,202
  202 M = M-8
      IF (M-7) 203,203,58
```

```
C---  28 DAY MONTH                                                  00023550
  210 IF (MULADD.EQ.1) GO TO 56                                      00023560
      IF (KSWV.EQ.1) GO TO 213                                       00023570
      Y = Y*.2825-28.                                                00023580
      GO TO 56                                                       00023590
  213 Y = Y*.28-28.                                                  00023600
      GO TO 56                                                       00023610
C---  29 DAY MONTH                                                  00023620
  203 N = 1                                                          00023630
      IF (MULADD.EQ.1) GO TO 50                                      00023640
      IF (KSWV.EQ.1) GO TO 222                                       00023650
      Y = Y*.2825 - 29.                                              00023660
      GO TO 50                                                       00023670
  222 Y = (Y/100.-1.)*(28.+DP(M))                                    00023680
      GO TO 50                                                       00023690
C---  30 DAY MONTH                                                  00023700
  251 N = 2                                                          00023710
      IF (MULADD.EQ.1) GO TO 50                                      00023720
      M1 = M + 1                                                     00023730
      IF (M1-8) 254,253,253                                          00023740
  253 M1 = 1                                                         00023750
  254 Y =(Y/100.-1.)*(28.+DP(M)+DP(M1))                              00023760
      GO TO 50                                                       00023770
C---  31 DAY MONTH                                                  00023780
  245 N = 3                                                          00023790
      IF (MULADD.NE.0) GO TO 50                                      00023800
      M1 = M+1                                                       00023810
      M2 = M+2                                                       00023820
      IF (M1-7) 249,248,247                                          00023830
  247 M1 = M1-7                                                      00023840
  248 M2 = M2-7                                                      00023850
  249 Y = (Y/100.-1.)*(28.+DP(M)+DP(M1)+DP(M2))                      00023860
   50 DO 53 J = 1, N                                                 00023870
      IF (M - 7) 52, 52, 51                                          00023880
   51 M = M - 7                                                      00023890
   52 X(M) = 1.                                                      00023900
   53 M = M + 1                                                      00023910
      IF (X(7)) 56, 56, 54                                           00023920
   54 DO 55 J = 1, 6                                                 00023930
   55 X(J) = X(J) - 1.                                               00023940
   56 SUMY = SUMY + Y                                                00023950
      SUMYSQ = SUMYSQ + Y * Y                                        00023960
      DO 57 M = 1, 6                                                 00023970
      MATRX(M,7) = MATRX(M,7) + Y * X(M)                             00023980
      DO 57 J = 1, 6                                                 00023990
   57 MATRX(M,J) = MATRX(M,J) + X(M) * X(J)                          00024000
   58 CONTINUE                                                       00024010
      DO 59 I = 1, 6                                                 00024020
   59 V(I) = MATRX(I,7)                                              00024030
      DO 65 I = 1, 6                                                 00024040
      R = 1.0/MATRX(I,1)                                             00024050
      DO 62 J = 1, 6                                                 00024060
   62 MATRX(I,J) = R * MATRX(I, J+1)                                 00024070
      MATRX(I,7) = R                                                 00024080
      DO 65 M = 1, 6                                                 00024090
```

```
      IF (M - I) 63, 65, 63
   63 R = -MATRX(M,1)
      DO 64 J = 1, 6
   64 MATRX(M,J) = MATRX(M,J+1) + R * MATRX(I,J)
      MATRX(M,7) = R * MATRX(I,7)
   65 CONTINUE
      S(7) = 0.
      DO 66 I = 1, 6
      DO 66 J = 2, 7
   66 S(7) = S(7) + MATRX(I,J)
      DO 67 I = 1, 6
   67 C(I) = MATRX(I,I+1)
      DSS = 0.
      XN = 0.
      DO 69 I = LFDA,LLDA
      IF (KARRAY(I)-22) 68,68,69
   68 XN = XN + 1.
   69 CONTINUE
      DR(7) = 0.
      DO 71 I = 1, 6
      DR(I) = MATRX(I,1)
      DR(7) = DR(7) - MATRX(I,1)
   71 DSS = DSS + DR(I) * V(I)
      RSS = SUMYSQ - DSS
      TMP = RSS/(XN - 6.)
      SE = SQRT(TMP)
      F = (DSS/6.)/TMP
      N1 = INT(XN + 0.1) - 6
      PROB = FVALUE(F,6,N1)*100.0
      N2 = N1 + 6
      IF (PROB.GT.0.1) GO TO 73
      FSTAR = ST3
      GO TO 76
   73 IF (PROB.GT.1.0) GO TO 74
      FSTAR = ST2
      GO TO 76
   74 IF (PROB.GT.5.0) GO TO 75
      FSTAR = ST1
      GO TO 76
   75 FSTAR = BLK
   76 IF (KPART.EQ.2) GO TO 77
      IF (PROB.LT.1.0) GO TO 77
      IF (KDWOPT.NE.3) GO TO 77
      KDWOPT = 1
   77 IF (KPRNT.LT.3) GO TO 107
      DO 72 I = 1,6
   72 S(I) = SE * SQRT(C(I))
      S(7) = SE * SQRT(S(7))
      DSSM = DSS/6.
      RSSM = RSS/(XN-6.)
      KSIGT = 0
      DO 99 I = 1,7
      STAR2(I) = BLK
      IF (MULADD.EQ.1) GO TO 93
      X(I) = DP(I)
```

```
      DC(I) = DP(I) + DR(I)
      TP(I) = DR(I) / S(I)
      T1(I) = (DC(I) - 1.0) / S(I)
      PT = TVALUE(N1,TP(I))*100.0
      IF (PT.GT.0.1) GO TO 90
      STAR2(I) = ST3
      GO TO 92
   90 IF (PT.GT.1.0) GO TO 91
      STAR2(I) = ST2
      GO TO 92
   91 IF (PT.GT.5.0) GO TO 94
      STAR2(I) = ST1
   92 KSIGT = 1
      GO TO 94
   93 X(I) = BIG
      TP(I) = BIG
      STAR2(I) = BLK
      DC(I) = DR(I)
      T1(I) = DC(I) / S(I)
   94 PT = TVALUE(N1,T1(I))*100.0
      IF (PT.GT.0.1) GO TO 95
      STAR1(I) = ST3
      GO TO 96
   95 IF (PT.GT.1.0) GO TO 97
      STAR1(I) = ST2
   96 KSIGT = 1
      GO TO 99
   97 IF (PT.GT.5.0) GO TO 93
      STAR1(I) = ST1
      GO TO 96
   98 STAR1(I) = BLK
   99 CONTINUE
      WRITE(NW,8200) KON,(DC(I),X(I),DR(I),S(I),T1(I),STAR1(I),TP(I),
     1               STAR2(I), I = 1,7)
      IF (KSIGT.EQ.0) GO TO 101
      WRITE(NW,8201) KON
  101 WRITE (NW,8203) DSS,DSSM,F,FSTAR,RSS,N1,RSSM,SUMYSQ,N2
      IF (PROB.GT.1.0) GO TO 102
      WRITE (NW, 8204) FSTAR
      GO TO 103
  102 WRITE(NW,8206)
  103 CONTINUE
      DO 106 K = 1,3
      M = K + LOPT
      WT(K) = ADJ(M)
      M = 4 - K
      SETDAF(M) = 0.
      L = K+1
      DO 106 I = 1,K
      DO 106 J = 2,L
  106 SETDAF(M) = SETDAF(M) + MATRX(I,J)
      DO 108 I = 1,3
  108 SETDAF(I) = SQRT(SETDAF(I))*100.*SE/WT(I)
      WRITE (NW,8205) SETDAF
  107 RETURN
```

```
      END
      SUBROUTINE XTRM(XI,KFDA,KLDA,NY)
C
C --- THIS ROUTINE COMPUTES WEIGHTS FOR THE IRREGULAR COMPONENT
C --- AND IDENTIFIES EXTREME IRREGULAR VALUES
C
      COMMON /OPT5 / MULADD,IYRT,MA,IWT
      COMMON /OPT6 / SIGML,SIGMU
      COMMON /MQ6  / STWT(372),STDEV(35)
      DIMENSION XI(1)
      N1 = 2 * NY
      N2 = N1 + NY - 1
      N3 = N2 + N1
      ISTEP = 1
      JFDA = (KFDA + NY - 2) / NY * NY + 1
      JLDA = KLDA / NY * NY - N3
      NFDA = KFDA/NY*NY+1
      NLDA = (KLDA+NY-2)/NY*NY-N3
      IF(JLDA-JFDA) 1,2,2
    1 NLDA = NFDA
C --- SET ALL WEIGHTS EQUAL TO 1.0 TO START
    2 DO 3 I = KFDA,KLDA
    3 STWT(I) = 1.0
      XBAR = 1.0
      IF (MULADD.EQ.0) GO TO 5
      XBAR = 0.0
    5 IN = 3
      DO 21 I = NFDA,NLDA,NY
      IF (NLDA-NFDA) 6,6,7
C --- LESS THAN 5 YEARS AVAILABLE.
    6 J = KFDA
      K = KLDA
      L = KFDA
      M = KLDA
      GO TO 12
    7 IF (I-NFDA) 9,9,8
    8 IF (I-NLDA) 11,10,10
C --- BEGINNING OF SERIES
    9 J = KFDA
      K = NFDA + N2
      L = KFDA
      M = JFDA + N3
      GO TO 12
C --- END OF SERIES
   10 J = NLDA + N1
      K = KLDA
      L = JLDA
      M = KLDA
      GO TO 12
C --- CENTRAL YEARS
   11 J = I + N1
      K = I + N2
      L = I
      M  = N3 + I
      GO TO 12
```

```
C --- COMPUTE FIVE YEAR STANDARD DEVIATION OF THE IRREGULARS.
   12 SDEV = 0.0
      XN = 0.0
      DO 15 N = L,M
      GO TO (14,13),ISTEP
C --- OMIT EXTREMES FROM THE CALCULATION OF THE FIVE YEAR STANDARD DEV.
   13 IF (STWT(N)) 14,15,14
   14 XN = XN+1.0
      SDEV = SDEV+(XI(N)-XBAR)*(XI(N)-XBAR)
   15 CONTINUE
      SDEV = SQRT(SDEV/XN)
C --- STORE STANDARD DEVIATIONS FOR PRINTING IN TABLE OF WEIGHTS.
      STDEV(IN) = SDEV
      IN = IN+1
      DO 21 N = J,K
      IF (SDEV) 21,21,16
C --- COMPUTE DEVIATION OF EACH IRREGULAR VALUE.
   16 TEMP = ABS(XI(N)-XBAR)/SDEV
      IF (TEMP-SIGMU) 17,17,19
   17 IF (TEMP.LE.SIGML) GO TO 21
      GO TO (21,18),ISTEP
C --- ASSIGN GRADUATED WEIGHT BETWEEN THE LIMITS.
   18 STWT(N) = (SIGMU-TEMP)/(SIGMU-SIGML)
      GO TO 21
   19 IF (ISTEP.EQ.1) GO TO 20
      IF (STWT(N).EQ.0.0) GO TO 21
C --- IN THE SECOND ITERATION ASSIGN A TEMPORARY WEIGHT OF -1.0 TO
C --- ALL EXTREMES.
      STWT(N) = -1.0
      GO TO 21
C --- IN THE FIRST ITERATION ASSIGN A WEIGHT OF 0.0 TO ALL EXTREMES.
   20 STWT(N) = 0.0
   21 CONTINUE
      GO TO (22,23),ISTEP
   22 ISTEP = 2
      GO TO 5
   23 CONTINUE
      DO 24 I = 1,3
      STDEV(I-1+IN) = SDEV
   24 STDEV(I) = STDEV(3)
      DO 26 I = KFDA,KLDA
      IF (STWT(I)+1.0) 25,25,26
   25 STWT(I) = 0.0
   26 CONTINUE
      RETURN
      END
      SUBROUTINE REPLAC(X,Y,STWT,LFDA,LLDA,NM)
C --- THIS SUBROUTINE REPLACES VALUES IN ARRAY X WHICH HAVE A WEIGHT
C --- LESS THAN 1.0. THE REPLACEMENT VALUES ARE STORED IN ARRAY Y.
C --- THE REPLACEMENT VALUES ARE COMPUTED USING AN AVERAGE
C ---  OF THE NON-FULL WEIGHT VALUE TIMES ITS WEIGHT AND THE NEAREST
C --- 4 FULL-WEIGHT VALUES.
      REAL*8 TOTALS
      COMMON /LARGE/ BIG
      DIMENSION X(LLDA),Y(LLDA),STWT(LLDA)
```

```
      IF (NM.EQ.1) GO TO 2
      DO 1 I = 1,LLDA
    1 Y(I) = BIG
    2 CONTINUE
      DO 21 I = 1,NM
      KFDA = LFDA+I-1
      KLDA = (LLDA-KFDA)/NM*NM+KFDA
      IF (NM.EQ.1) GO TO 222
      AVE = TOTALS(X,KFDA,KLDA,NM,1)
  222 CONTINUE
      DO 21 J = KFDA,LLDA,NM
C --- TEST FOR FULL WEIGHT (1.0).
      IF (STWT(J)-1.0) 3,21,3
    3 N = 0
      SUMX = STWT(J)*X(J)
      IF (J-NM-KFDA) 5,5,4
    4 IF (KLDA-NM-J) 6,6,11
C --- EXTREME VALUES IN THE FIRST 2 LOCATIONS AT EITHER END OF THE ARRAY
C --- ARE REPLACED USING THE THREE NEAREST NON-EXTREME VALUES.
    5 M = KFDA
      L = KLDA
      INC = NM
      GO TO 7
    6 M = KLDA
      L = KFDA
      INC = -NM
    7 IF (STWT(M)-1.0) 8,10,10
    8 IF (M-L) 9,16,9
    9 M = M+INC
      GO TO 7
   10 SUMX = SUMX+X(M)
      N = N+1
      IF (N-4) 8,19,19
C --- EXTREME CENTRAL VALUES ARE REPLACED BY THE 2 NEAREST NON-EXTREME
C --- VALUES ON EACH SIDE
   11 M = J
      L = KLDA
      INC = NM
   12 IF (M-L) 13,16,13
   13 M = M+INC
      IF (STWT(M)-1.0) 12,14,12
   14 SUMX = SUMX+X(M)
      N = N+1
      GO TO (12,15,12,19),N
   15 M = J
      L = KFDA
      INC = -NM
      GO TO 12
   16 IF (NM-1) 21,21,17
C --- IF FEWER THAN 4 FULL WEIGHT VALUES IN THE MONTH, REPLACE THE
C --- EXTREME VALUE WITH THE AVERAGE OF ALL THE SI VALUES.
   17 X(J) = AVE
      GO TO 20
   19 X(J) = SUMX/(N+STWT(J))
      IF (NM-1) 21,21,20
```

```
   20 Y(J) = X(J)
   21 CONTINUE
      RETURN
      END
      SUBROUTINE SI(KSECT,KFDA,KLDA,NYR,KPROPT,IFORC,LFD1,LLD1)
C --- THIS SUBROUTINE CALCULATES THE SEASONALS FROM THE SI ESTIMATES
C --- FOR PART B.
      COMMON /WORK / TEMP(372)
      COMMON /MQ1  / STS(372),STSI(372)
      COMMON /MQ6  / STWT(372),STDEV(35)
      COMMON /MQ7  / STI(372)
      COMMON /OPT9 / IDUM(19),KEBACK
      LFD=LFD1+(NYR/2)
      LLD=LLD1-(NYR/2)
      L=LLD1-LFD1+1
      LLDA = KLDA
      IF (IFORC.NE.0.AND.KSECT.EQ.1) LLDA = KLDA-NYR
      IF (KSECT.EQ.1) GO TO 1
      CALL VSFA(KFDA,LLDA,NYR)
    1 CALL VSFB(KFDA,KLDA,NYR)
      K = KSECT * 5 - 2
      IFTNY=15*NYR
      IF (KPROPT.EQ.1) GO TO 3
      IF (KPROPT.NE.2) GO TO 4
      IF (KSECT.EQ.2) GO TO 4
    3 IF (IFORC.NE.0.AND.L.LE.IFTNY.AND.KEBACK.EQ.0) CALL TABLE(STSI,LFD
     11,LLD1,K,1,1,0)
      IF(IFORC.NE.0.AND.L.LE.IFTNY.AND.KEBACK.EQ.1)CALL TABLE(STSI,LFD,
     1LLD1,K,1,1,0)
      IF (IFORC.EQ.0) CALL TABLE(STSI,LFD,LLD,K,1,1,0)
      IF (IFORC.NE.0.AND.L.GT.IFTNY) CALL TABLE(STSI,LFD,LLD1,K,1,1,0)
    4 CALL DIVSUB(STI,STSI,STS,KFDA,KLDA)
      CALL XTRM(STI,KFDA,KLDA,NYR)
      CALL REPLAC(STSI,TEMP,STWT,KFDA,KLDA,NYR)
      IF (KPROPT.NE.1) GO TO 5
      K = K + 1
      IF(IFORC.NE.0.AND.L.LE.IFTNY.AND.KEBACK.EQ.0)CALL TABLE(TEMP,LFD1,
     1LLD1,K,1,4,STDEV)
      IF(IFORC.NE.0.AND.L.LE.IFTNY.AND.KEBACK.EQ.1)CALL TABLE(TEMP,LFD,L
     1LD1,K,1,4,STDEV)
      IF (IFORC.EQ.0) CALL TABLE(TEMP,LFD,LLD,K,1,4,STDEV)
      IF(IFORC.NE.0.AND.L.GT.IFTNY)CALL TABLE(TEMP,LFD,LLD1,K,1,4,STDEV)
    5 CALL VSFB(KFDA,KLDA,NYR)
      RETURN
      END
      SUBROUTINE TABLE(Z,IB,IE,KTABL,ITYPE,NOP,Y)
C --- THIS SUBROUTINE WRITES TABLE OUTPUT.
C --- Z IS THE ARRAY TO BE PRINTED FROM IB TO IE.
C --- Y IS AN ADDITIONAL ARRAY TO BE PRINTED ON TABLE.
C --- KTABL IS THE TABLE NUMBER.
C --- NOP = 1,AVERAGE.
C ---     = 2,TOTAL.
C ---     = 3,STANDARD DEVIATION.
C ---     = 4,MOVING 5-YEAR STD. DEV.
C ---     = 5,NONE.
```

```
C --- ITYPE SPECIFIES IF THIS IS ADDITIONAL KTABL.                          00027400
      REAL*8 TITLE,SERNO,LSER,KSER,PCDIF,RAD,FI(4),MQ,AVGS(6),TYRLY(5),T00027410
     1OTALS                                                                 00027420
      DIMENSION IP(20),Z(1),Y(1),ALPH(7),TYPE(4),UM(2),ITAB(5)               00027430
      COMMON /UNITS/ MT,MT2,MT1,MP,NG,NFORM                                  00027440
      COMMON /ROTE / TITLE(10),SERNO,KPAGE,IRUN,NEWPG                        00027450
      COMMON /PRIOR/ KSER,KFORM(20),KFMT                                     00027460
      COMMON /TRADE/ DWT(7),D(7),SIGM,LOPT,KDWOPT,LCYR,LAYR,LFDC,LFDR        00027470
      COMMON /LARGE/ BIG                                                     00027480
      COMMON /TFMT / IF1(18),IF2(14),IF3(22)                                 00027490
      COMMON /WORK2/ TMP(108)                                                00027500
      COMMON /WORK5/ X(372)                                                  00027510
      COMMON /OPT1 / IFORM(20),IPLUS,IFMT                                    00027520
      COMMON /OPT2 / KDEC,ICARD                                              00027530
      COMMON /OPT4 / NUMB(9),IHOLD(9),NOCD                                   00027540
      COMMON /OPT5 / MULADD,IYRT,MA,IWT                                      00027550
      COMMON /OPT6 / SIGML,SIGMU                                             00027560
      COMMON /OPT7 / RATI(36),RATIS,LTERM,MTYPE,LTER(12)                     00027570
      COMMON /OPT8 / RATIC,KTCOPT,NTERM                                      00027580
      COMMON /OPT9 / LFDA,LYR,LSTMO,LSTYR,LLDA,KFULSM,KPROPT,KCHOPT,         00027590
     1               KEXOPT,NY,IFORC,LLDAF,LLAF,LFD1,LLD1,NSOPT,IDEC,IAG00027600
     2               ,JAREX,KEBACK                                           00027610
      COMMON /MQ2  / KPART,KSWV,KSECT,KPRNT,MCD                              00027620
      COMMON /MQ3  / PCDIF(3),RAD(2),MQ,QM(3),OUT,MQCD                       00027630
      COMMON /KCJH / LENGTH,IFTNY                                            00027640
      DATA  ALPH,TYPE/1HA,1HB,1HC,1HD,1HE,1HF,1HG,2H. ,2HA.,2HB.,2HC./       00027650
      DATA TYRLY/5H AVGE,5HTOTAL,5H S.D.,5H S.D.,5H     /                    00027660
      DATA  AVGS/8H  3X1   ,8H  3X3   ,8H  3X5   ,8H  3X9   ,8H STABLE ,00027670
     1           8HVARIABLE/                                                 00027680
      DATA    FI/8H        ,8H FINAL  ,8H PRELIM ,8HMODIFIED/                00027690
      DATA    IP/0,0,3*1,0,0,3*1,0,0,1,1,0,1,1,1,0,1/                        00027700
      DATA  UM,BLK,YEAR/2H  ,2HUN,4H    ,4HYEAR/                             00027710
      DATA ITAB/316,318,410,416,104/                                         00027720
C --- INITIALIZATION                                                         00027730
      IF (KPART.EQ.1) IP(3)=0                                                00027740
      IF (KPART.NE.1) IP(3)=1                                                00027750
      NOPP = NOP                                                             00027760
      KE = IE                                                                00027770
      IN = 1                                                                 00027780
      IPOW = 0                                                               00027790
      LDEC = KDEC                                                            00027800
      IF (IB.EQ.0) GO TO 315                                                 00027810
      DO 90 I = IB,IE                                                        00027820
   90 X(I) = Z(I)                                                            00027830
      IF (MULADD.EQ.1) GO TO 300                                             00027840
      IF (MULADD.EQ.0) GO TO 2                                               00027850
      IF (KTABL.EQ.17) GO TO 2                                               00027860
      DO 92 I = IB,IE                                                        00027870
      IF (Z(I).EQ.BIG) GO TO 92                                              00027880
      X(I) = EXP(Z(I))                                                       00027890
   92 CONTINUE                                                               00027900
    2 IPOW = IP(KTABL)                                                       00027910
      IF (KTABL.EQ.2.AND.KPART.EQ.1) IPOW = 1                                00027920
      IF (KTABL.EQ.6.AND.KPART.EQ.5) IPOW = 1                                00027930
      IF (KPART.EQ.5.AND.KTABL.EQ.8) IPOW = 0                                00027940
```

```
      IF (KDEC.NE.0) GO TO 1
      LDEC = IDEC
      IF (IDEC.GE.6) LDEC = 0
      IF (IPOW.EQ.1) LDEC = 1
    1 CONTINUE
      GO TO (300,300,200,201,300,300,300),KPART
  200 IF (KTABL.NE.16.AND.KTABL.NE.18) GO TO 300
      GO TO 3
  201 IF (KTABL.NE.10.AND.KTABL.NE.16) GO TO 300
    3 LDEC = 2
  300 ISTOP = 0
      IF (ICARD.EQ.0) GO TO 314
      ITEST = 1000*(ITYPE-1)+100*KPART+KTABL
      DO 301 I = 1,NOCD
      IF (IHOLD(I).EQ.1) GO TO 301
      JTEST = 100*KPART+KTABL+NUMB(I)/1000*1000
      IF (ITEST.EQ.NUMB(I)) GO TO 307
      IF (JTEST.EQ.NUMB(I)) GO TO 302
  301 CONTINUE
      GO TO 314
  302 DO 303 J = 1,5
      IF (ITEST.EQ.ITAB(J)) GO TO 304
  303 CONTINUE
      GO TO 314
  304 IF (JTEST.EQ.2104) GO TO 307
      IF (JTEST.EQ.2316) GO TO 307
      IF (JTEST.EQ.2318) GO TO 307
      JB = IE+1
      JE = IE+NY
      K = I+1
      IF (K.GT.NOCD) GO TO 311
      DO 305 J = K,NOCD
      IF (ITEST.EQ.NUMB(J)) GO TO 306
      IF (JTEST-2000-ITEST) 305,306,305
  305 CONTINUE
      GO TO 311
  306 IHOLD(J) = 1
      JB = IB
      GO TO 311
  307 JB = IB
      JE = IE
      DO 308 J = 1,5
      IF (ITEST.EQ.ITAB(J)) GO TO 309
  308 CONTINUE
      GO TO 313
  309 JTEST = JTEST+1000
      K = I+1
      IF (K.GT.NOCD) GO TO 311
      DO 310 J = K,NOCD
      IF (JTEST.NE.NUMB(J)) GO TO 310
      JE = IE+NY
      GO TO 306
  310 CONTINUE
  311 IF (JE.EQ.IE) GO TO 313
      IE1 = IE+1
```

```
      DO 312 J = IE1,JE
  312 X(J) = Z(J)
  313 CALL PUNCH(X,JB,JE,I,IPOW)
  314 IOPT = IP(KTABL)+IPOW
      IF (KPART.EQ.5.AND.KTABL.EQ.8) IOPT = 0
      LSER = SERNO
      IF (KPART.NE.1) GO TO 315
      IF (KTABL.NE.2) GO TO 315
      LSER = KSER
  315 KPAGE = KPAGE+1
      WRITE(MT1,1000) NEWPG,TITLE,KPAGE,LSER
 1000 FORMAT(A1,10X,9A8,A6,12X,4HPAGE,I3,9H, SERIES ,A8,//)
      GO TO (4,5,7,9,10,67,71),KPART
    4 GO TO (13,14,15,16),KTABL
    5 IF (KTABL.GT.13) GO TO 6
      GO TO (17,18,19,21,23,25,27,19,21,23,25,27,29),KTABL
    6 MTABL = KTABL-13
      GO TO (31,32,34,35,37,39,64),MTABL
    7 IF (KTABL.GT.13) GO TO 8
      GO TO (51,18,19,40,23,25,27,19,40,23,25,27,29),KTABL
    8 MTABL = KTABL-13
      GO TO (31,41,34,42,37,39,64),MTABL
    9 IF (KTABL.GT.13) GO TO 91
      GO TO (56,18,19,40,23,25,27,43,44,45,46,47,48),KTABL
   91 MTABL = KTABL-13
      GO TO (9999,9999,61),MTABL
   10 GO TO (57,49,50,58,59,60,54,55),KTABL
C --- TITLES FOR THE TABLES.
   13 WRITE(MT1,1001) ALPH(KPART),KTABL,TYPE(ITYPE)
 1001 FORMAT(1X,A1,I2,A2,16H ORIGINAL SERIES )
      GO TO 100
   14 WRITE(MT1,1002) QM
 1002 FORMAT(' A 2. PRIOR ',2A4,A1,' ADJUSTMENT FACTORS')
      GO TO 100
   15 WRITE(MT1,1003) QM
 1003 FORMAT(' A 3. ORIGINAL SERIES ADJUSTED BY PRIOR ',2A4,A1,' ADJUSTM
     1ENT FACTORS')
      GO TO 100
   16 WRITE(MT1,1004)
 1004 FORMAT(' A 4. PRIOR TRADING DAY ADJUSTMENT FACTORS')
      WRITE(MT1,2004) (Y(I),I = 1,7 )
 2004 FORMAT(' A4A. PRIOR DAILY WEIGHTS - MON     TUE     WED    THUR
     1  FRI     SAT     SUN',/,23X,7F8.3)
      WRITE(MT1,3004) OUT
 3004 FORMAT(' A4B. PRIOR TRADING DAY FACTORS WITH',A3,' LENGTH OF MONTH
     1 ADJUSTMENT')
      GO TO 100
   17 IF(KFMT.EQ.0.AND.KSWV.EQ.0.AND.IFORC.EQ.0) GO TO 13
      IF(KFMT.EQ.0.AND.KSWV.EQ.0.AND.IFORC.GE.1.AND.KEBACK.EQ.1.AND.
     1JAREX.EQ.0) GO TO 13
      IF(KFMT.EQ.0.AND.KSWV.EQ.0.AND.IFORC.GE.1.AND.LENGTH.GE.IFTNY
     1.AND.JAREX.EQ.0)GO TO 13
  417 WRITE(MT1,1005)
 1005 FORMAT(' B 1. PRIOR ADJUSTED ORIGINAL SERIES')
      GO TO 100
```

```
   18 NYY=(IWT+1)*NY
      WRITE(MT1,1006) ALPH(KPART),KTABL,NYY
 1006 FORMAT(1X,A1,I2,'. TREND CYCLE- CENTERED',I3,'-TERM MOVING AVERAGE
     1')
      GO TO 100
   19 IU = 2
      IF = 1
   20 WRITE(MT1,1007) ALPH(KPART),KTABL,TYPE(ITYPE),FI(IF),UM(IU),RAD
 1007 FORMAT(1X,A1,I2,A2,A8,A2,12HMODIFIED SI ,A6,A5)
      GO TO 100
   21 IF = 1
      ISTOP = 1
   22 WRITE(MT1,1008) ALPH(KPART),KTABL,TYPE(ITYPE),FI(IF),RAD
 1008 FORMAT(1X,A1,I2,A2,A8,'REPLACEMENT VALUES FOR EXTREME SI ',A6,A5)
      GO TO 100
   23 IF = 1
   24 WRITE(MT1,1009) ALPH(KPART),KTABL,TYPE(ITYPE),FI(IF)
 1009 FORMAT(1X,A1,I2,A2,A8,'SEASONAL FACTORS')
 2009 FORMAT(14X,A8,' MOVING AVERAGE SELECTED.')
        DO 11111 I=1,NY
      IF (LTER(I).GT.0) WRITE (MT1,3009) MQ
 3009 FORMAT (14X,'DIFFERENT MOVING AVERAGES SELECTED FOR DIFFERENT ',
     1A7,'S')
        IF (LTER(I) .GT. 0) GO TO 11112
11111   CONTINUE
        WRITE(MT1,2009) AVGS(MTYPE)
11112   CONTINUE
      GO TO 100
   25 IF = 1
   26 WRITE(MT1,1010) ALPH(KPART),KTABL,TYPE(ITYPE),FI(IF)
 1010 FORMAT(1X,A1,I2,A2,1X,A8,' SEASONALLY ADJUSTED SERIES')
      GO TO 100
   27 IF = 1
   28 WRITE(MT1,1011) ALPH(KPART),KTABL,TYPE(ITYPE),FI(IF)
 1011 FORMAT(1X,A1,I2,A2,A8,'TREND CYCLE - HENDERSON CURVE')
      WRITE(MT1,2011) NTERM,RATIC
 2011 FORMAT(19X,I2,43H-TERM MOVING AVERAGE SELECTED. I/C RATIO IS,F6.2)
      GO TO 100
   29 IF = 1
   30 WRITE(MT1,1012) ALPH(KPART),KTABL,TYPE(ITYPE),FI(IF)
 1012 FORMAT(1X,A1,I2,A2,1X,A8,' IRREGULAR SERIES')
      GO TO 100
   31 WRITE(MT1,1013) ALPH(KPART)
 1013 FORMAT(1X,A1,'14. EXTREME IRREGULAR VALUES EXCLUDED FROM TRADING D
     1AY REGRESSION')
      WRITE(MT1,2013) SIGM
 2013 FORMAT(9X,9H(OUTSIDE ,F3.1,13H-SIGMA LIMIT))
      GO TO 100
   32 IF = 3
   33 WRITE(MT1,1014) ALPH(KPART),FI(IF)
 1014 FORMAT(1X,A1,3H15.,A8,'TRADING DAY REGRESSION')
      GO TO 9999
   34 WRITE(MT1,1015) ALPH(KPART)
 1015 FORMAT(1X,A1,'16. TRADING DAY ADJUSTMENT FACTORS DERIVED FROM REGR
     1ESSION COEFFICIENTS')
```

```
      GO TO (100,100, 62),KPART
   35 IF = 3
   36 WRITE(MT1,1016) ALPH(KPART),KTABL,TYPE(ITYPE),FI(IF)
 1016 FORMAT(1X,A1,I2,A2,1X,A8,'WEIGHTS FOR IRREGULAR COMPONENT')
      WRITE(MT1,2016) SIGML,SIGMU
 2016 FORMAT(10X,'GRADUATION RANGE FROM',F4.1,3H TO,F4.1,6H SIGMA)
      IPOW = 1
      ISTOP = 1
      IF (MULADD.NE.1) GO TO 100
      KLYR = LYR+(IB-1)/NY
      KKYR = (IE+NY-1)/NY+LYR-1
      KK = 1
      DO 435 I = KLYR,KKYR
      Y(KK) = Y(KK)/100.0
  435 KK = KK+1
      GO TO 100
   37 WRITE(MT1,1017) ALPH(KPART)
 1017 FORMAT(1X,A1,'18. TRADING-DAY ADJUSTMENT FACTORS FROM COMBINED DAI
     1LY WEIGHTS')
      IF (KSWV.NE.0.OR.LOPT.NE.0) GO TO 38
      WRITE(MT1,2017) ALPH(KPART)
 2017 FORMAT(9X,15H(SAME AS TABLE ,A1,4H16.))
      GO TO 9999
   38 GO TO (100,100, 63),KPART
   39 WRITE(MT1,1018) ALPH(KPART)
 1018 FORMAT(1X,A1,'19.   ADJUSTED* ORIGINAL SERIES')
  239 WRITE(MT1,2018)
 2018 FORMAT(8X,'*ADJUSTED BY...TRADING DAY ADJUSTMENT FACTORS DERIVED F
     1ROM REGRESSION COEFFICIENTS')
      IF (KFMT.EQ.0) GO TO 100
      WRITE(MT1,3018)
 3018 FORMAT(21X,'...PRIOR ADJUSTMENT FACTORS')
      GO TO 100
   40 IF = 1
      IU = 1
      GO TO 20
   41 IF = 2
      GO TO 33
   42 IF = 2
      GO TO 36
   43 IF = 2
      IU = 2
      IF (ITYPE.NE.1) GO TO 9999
      GO TO 20
   44 IF = 2
      GO TO 22
   45 IF = 2
      GO TO 24
   46 IF = 2
      IF (ITYPE.EQ.1) GO TO 26
      WRITE(MT1,1025) ALPH(KPART),KTABL,TYPE(ITYPE),FI(IF)
 1025 FORMAT(1X,A1,I2,A2,1X,A8,' SEASONALLY ADJUSTED SERIES WITH REVISED
     1 YEARLY TOTALS')
      GO TO 100
   47 IF = 2
```

```
      GO TO 28                                                          00030150
   48 IF = 2                                                            00030160
      GO TO 30                                                          00030170
   49 WRITE(MT1,1054)                                                   00030180
 1054 FORMAT(' E 2.  FINAL SEASONALLY ADJUSTED SERIES MODIFIED FOR EXTRE00030190
     1MES WITH ZERO WEIGHTS')                                           00030200
      GO TO 100                                                         00030210
   50 IF = 4                                                            00030220
      GO TO 30                                                          00030230
   51 IF = 3                                                            00030240
   52 IF (KFMT.NE.0) GO TO 53                                           00030250
      IF (KDWOPT.GE.2) GO TO 53                                         00030260
      WRITE(MT1,1019) ALPH(KPART),FI(IF)                                00030270
 1019 FORMAT(1X,A1,' 1. ORIGINAL SERIES MODIFIED BY',A8,7HWEIGHTS)      00030280
      GO TO 100                                                         00030290
   53 WRITE(MT1,1020) ALPH(KPART),FI(IF)                                00030300
 1020 FORMAT(1X,A1,' 1.   ADJUSTED* ORIGINAL SERIES MODIFIED BY',A8,    00030310
     1 7HWEIGHTS)                                                       00030320
      IF (KDWOPT.GE.2) GO TO 239                                        00030330
      WRITE(MT1,2020)                                                   00030340
 2020 FORMAT(8X,'*ADJUSTED BY...PRIOR ADJUSTMENT FACTORS')              00030350
      GO TO 100                                                         00030360
   54 WRITE(MT1,1051)                                                   00030370
 1051 FORMAT(1X,1H )                                                    00030380
      GO TO 100                                                         00030390
   55 WRITE(MT1,1052)                                                   00030400
 1052 FORMAT(1X,1H )                                                    00030410
      IF (MCD.NE.1) GO TO 100                                           00030420
      WRITE(MT1,2052)                                                   00030430
 2052 FORMAT(/,7X,'THE SAME AS TABLE E 7.')                             00030440
      RETURN                                                            00030450
   56 IF = 2                                                            00030460
      GO TO 52                                                          00030470
   57 WRITE(MT1,1021)                                                   00030480
 1021 FORMAT(' E 1. ORIGINAL SERIES MODIFIED FOR EXTREMES WITH ZERO FINA00030490
     1L WEIGHTS')                                                       00030500
      GO TO 100                                                         00030510
   58 WRITE(MT1,1022) ALPH(KPART),KTABL,TYPE(ITYPE),RAD                 00030520
 1022 FORMAT(1X,A1,I2,A2,A6,A5,' OF ANNUAL TOTALS, ORIGINAL AND ADJUSTED00030530
     1 SERIES')                                                         00030540
      GO TO 102                                                         00030550
   59 WRITE(MT1,1023) MQ,MQ                                             00030560
 1023 FORMAT(' E 5. ',A7,4H-TO-,A7,' CHANGES IN THE ORIGINAL SERIES')   00030570
      IOPT = 0                                                          00030580
      GO TO 100                                                         00030590
   60 WRITE(MT1,1024) TYPE(ITYPE),MQ,MQ,TYPE(ITYPE)                     00030600
 1024 FORMAT(' E 6',A2,1X,A7,4H-TO-,A7,' CHANGES IN THE FINAL SEASONALLY00030610
     1 ADJUSTED SERIES (D11',A2,1H))                                    00030620
      IOPT = 0                                                          00030630
      GO TO 100                                                         00030640
   67 WRITE(MT1,1027) MQCD                                              00030650
 1027 FORMAT(6H F 1. ,A3,' MOVING AVERAGE')                             00030660
      WRITE(MT1,2027) MQCD,MCD                                          00030670
 2027 FORMAT(12X,A3,3H IS,I4)                                           00030680
      IF (MCD.NE.1) GO TO 100                                           00030690
```

```
      WRITE(MT1,3027)
 3027 FORMAT(/,7X,'THE SAME AS TABLE D11.')
      RETURN
   71 WRITE(MT1,1029) KTABL
 1029 FORMAT(2H G,I2,7H. CHART)
      IF (ITYPE.GT.1) GO TO 9999
      GO TO (72,73,74,74,75,76),KTABL
   72 WRITE(MT1,1030)
 1030 FORMAT(9X,26H(X) - B 1. ORIGINAL SERIES,/,
     1        9X,48H(O) - E 1. ORIGINAL SERIES MODIFIED FOR EXTREMES,
     2        24H WITH ZERO FINAL WEIGHTS,/,
     3        9X,27H(*) - COINCIDENCE OF POINTS)
      IF (IFORC.NE.0) WRITE(MT1,2031)
 2031 FORMAT(9X,25H(E) - ARIMA EXTRAPOLATION)
      GO TO 9999
   73 WRITE(MT1,1031)
 1031 FORMAT(9X,43H(X) - D11. FINAL SEASONALLY ADJUSTED SERIES,/,
     1        9X,28H(O) - D12. FINAL TREND CYCLE,/,
     2        9X,27H(*) - COINCIDENCE OF POINTS)
      GO TO 9999
   74 WRITE(MT1,1032) RAD,RAD
 1032 FORMAT(9X,33H(X) - D10. FINAL SEASONAL FACTORS,/,
     1        9X,31H(O) - D 8. FINAL UNMODIFIED SI ,A6,A5,/,
     2        9X,17H(+) - D 9. FINAL ,A6,A5,22H MODIFIED FOR EXTREMES,/,
     3        9X,27H(*) - COINCIDENCE OF POINTS,/,
     4        9X,35H(E) - EXTRAPOLATED SEASONAL FACTORS)
      GO TO 9999
   75 WRITE(MT1,1033)
 1033 FORMAT(9X,33H(X) - D13. FINAL IRREGULAR SERIES,/,
     1        9X,42H(O) - E 3. FINAL MODIFIED IRREGULAR SERIES,/,
     2        9X,27H(*) - COINCIDENCE OF POINTS)
      GO TO 9999
   76 WRITE(MT1,1035)
 1035 FORMAT(1H0,5X,' KOLMOGOROV-SMIRNOV SIGNIFICANCE TESTS FOR THE FINA
     1L IRREGULARS')
      GO TO 9999
   61 WRITE(MT1,1036)  ALPH(KPART),KTABL
 1036 FORMAT(1X,A1,I2,'. COMBINED SEASONAL AND TRADING DAY FACTORS')
      GO TO 100
   62 WRITE(MT1,1037)
 1037 FORMAT(96H C16A. REGRESSION COEFFICIENTS - MON        TUE        WED
     1      THUR        FRI        SAT        SUN)
      WRITE(MT1,2037) (Y(I), I = 1,7 )
 2037 FORMAT(27X,7F10.3)
      WRITE(MT1,3037)
 3037 FORMAT(48H C16B. REGRESSION TRADING DAY ADJUSTMENT FACTORS)
      GO TO 100
   63 WRITE(MT1,1038) (Y(I), I = 1,7 )
 1038 FORMAT(83H C18A. COMBINED DAILY WEIGHTS - MON      TUE      WED     T
     1HUR      FRI      SAT      SUN,/,
     2 27X,7F8.3)
      WRITE(MT1,2038) OUT
 2038 FORMAT(51H C18B. COMBINED TRADING-DAY ADJUSTMENT FACTORS WITH,A3,
     1        27H LENGTH OF MONTH ADJUSTMENT)
      GO TO 100
```

```
   64 WRITE(MT1,1050) ALPH(KPART)                                       00031250
 1050 FORMAT(1X,A1,'20. EXTREME VALUES')                                00031260
C --- WRITE COLUMN HEADINGS.                                            00031270
  100 IF (NY.EQ.4) GO TO 101                                            00031280
      L = 13                                                            00031290
      WRITE(MT1,1039) YEAR,TYRLY(NOPP)                                  00031300
 1039 FORMAT(1H0,3X,A4,'       JAN       FEB       MAR       APR       MAY  00031310
     1    JUN       JUL       AUG       SEP       OCT       NOV       DEC', 00031320
     210X,A5,/)                                                         00031330
      GO TO 103                                                         00031340
  101 L = 5                                                             00031350
      WRITE(MT1,1040) YEAR,TYRLY(NOPP)                                  00031360
 1040 FORMAT(1H0,3X,A4,10X,8H1ST QUAR,16X,8H2ND QUAR,16X,               00031370
     1 8H3RD QUAR,16X,8H4TH QUAR,17X,A5,/)                              00031380
      GO TO 103                                                         00031390
  102 WRITE(MT1,1041)                                                   00031400
 1041 FORMAT(1H0,3X,4HYEAR,6X,'UNMODIFIED   MODIFIED')                  00031410
      GO TO 130                                                         00031420
  103 IF (NOP.EQ.5) L = L-1                                             00031430
C --- GENERATE TABLE FORMAT.                                            00031440
      CALL TFMT2(IF1,LDEC,0,IPOW)                                       00031450
C --- WRITE TABLE.                                                      00031460
      JYR = LYR+(IB-1)/NY                                               00031470
      KYR = (IE+NY-1)/NY + LYR - 1                                      00031480
      DO 104 I = 1,13                                                   00031490
  104 TMP(I) = BIG                                                      00031500
      IB1 = IB                                                          00031510
      IE1 = (JYR-LYR+1)*NY                                              00031520
      IM = IB-(IB-1)/NY*NY                                              00031530
  105 IM1 = IM                                                          00031540
      DO 106 I = IB1,IE1                                                00031550
      TMP(IM) = X(I)                                                    00031560
  106 IM = IM+1                                                         00031570
      IM2 = IM-1                                                        00031580
      GO TO (107,108,109,110,111),NOPP                                  00031590
  107 TMP(L) = TOTALS(TMP,IM1,IM2,1,1)                                  00031600
      GO TO 111                                                         00031610
  108 TMP(L) = TOTALS(TMP,IM1,IM2,1,0)                                  00031620
      GO TO 111                                                         00031630
  109 TMP(L) = SDEV(TMP,IM1,IM2,1,IOPT)                                 00031640
      GO TO 111                                                         00031650
  110 TMP(L) = Y(IN)                                                    00031660
      IN = IN+1                                                         00031670
  111 WRITE(MT1,IF1) JYR,(TMP(I),I = 1,L)                               00031680
      IF (NOP.EQ.5) WRITE(MT1,1111)                                     00031690
 1111 FORMAT(1H )                                                       00031700
      JYR = JYR+1                                                       00031710
      IM = 1                                                            00031720
      IB1 = IE1+1                                                       00031730
      IE1 = IE1+NY                                                      00031740
      IF (KYR-JYR) 114,112,105                                          00031750
  112 DO 113 I = 1,NY                                                   00031760
  113 TMP(I) = BIG                                                      00031770
      IE1 = KE                                                          00031780
      GO TO 105                                                         00031790
```

```
  114 IF (KPART.EQ.4.AND.KTABL.EQ.9) GO TO 128
C --- CALCULATE AND WRITE COLUMN SUMMARIES.
      IF (NOP.EQ.5.OR.ISTOP.EQ.1) GO TO 9999
      IE1 = IB+NY-1
      IM = IB-(IB-1)/NY*NY
      DO 117 I = IB,IE1
      IF (NOP.GT.2) GO TO 115
      TMP(IM) = TOTALS(X,I,IE,NY,1)
      GO TO 116
  115 TMP(IM) = SDEV(X,I,IE,NY,IOPT)
  116 IF (IM.EQ.NY) IM = 0
  117 IM = IM+1
C --- GENERATE COLUMN SUMMARY FORMATS.
      CALL TFMT2(IF2,LDEC,1,IPOW)
      NOP1 = (NOP - 1) / 2 * 2 + 1
      WRITE(MT1,IF2) TYRLY(NOP1),(TMP(I),I = 1,NY)
      TMP(1) = TOTALS(X,IB,IE,1,0)
      TMP(2) = TMP(1)/(IE-IB+1)
      TMP(3) = SDEV(X,IB,IE,1,IOPT)
C --- GENERATE  FORMAT FOR THE TABLE SUMMARY
      CALL TFMT2(IF3,LDEC,2,IPOW)
      IF (IPOW.EQ.0) GO TO 120
      DO 119 I = 1,3
  119 TMP(I) = TMP(I)*100.0
C --- WRITE TABLE SUMMARY.
  120 WRITE(MT1,IF3) (TMP(I),I = 1,3)
      GO TO (121,11,122,1121,9999,9999),KPART
C --- WRITE ADDITIONAL TABLES.
  121 IF(KTABL.NE.4) GO TO 9999
      WRITE(MT1,1049)
 1049 FORMAT(' A4C. PRIOR TRADING DAY ADJUSTMENT FACTORS, ONE YEAR AHEAD
     1')
      IFACT=1
      GO TO 124
   11 IF(KTABL.NE.1)GO TO 9999
      IF(IFORC.EQ.0)GO TO 9999
      IF(KFMT+KSWV+JAREX)12,12,65
   12 WRITE(MT1,66)
   66 FORMAT('0 ORIGINAL SERIES EXTRAPOLATED ONE YEAR AHEAD')
      IFACT=0
      GO TO 124
   65 WRITE(MT1,68)
   68 FORMAT('0 PRIOR-ADJUSTED ORIGINAL SERIES EXTRAPOLATED ONE YEAR AHE
     1AD')
      IFACT=0
      GO TO 124
 1121 IF (KTABL.NE.10.AND.KTABL.NE.16) GO TO 9999
      IF (KTABL.EQ.16) GO TO 1122
      WRITE(MT1,1042)
 1042 FORMAT('0D10A. SEASONAL FACTORS, ONE YEAR AHEAD')
      IFACT=1
      GO TO 124
 1122 WRITE(MT1,1053)
 1053 FORMAT('0D16A. COMBINED SEASONAL AND TRADING DAY FACTORS, ONE YEAR
     1 AHEAD')
```

```
      IFACT=1
      GO TO 124
  122 IF (KTABL.EQ.18) GO TO 123
      IF (KTABL.NE.16) GO TO 9999
      WRITE(MT1,1043)
 1043 FORMAT(64H C16C. REGRESSION TRADING DAY ADJUSTMENT FACTORS, ONE YE
     1AR AHEAD)
      IFACT=1
      GO TO 124
  123 WRITE(MT1,1044)
 1044 FORMAT(62H C18C. COMBINED TRADING-DAY ADJUSTMENT FACTORS, ONE YEAR
     1 AHEAD)
      IFACT=1
  124 IF (NY.EQ.4) GO TO 125
      WRITE(MT1,1039) YEAR,TYRLY(NOPP)
      GO TO 126
  125 WRITE(MT1,1040) YEAR,TYRLY(NOPP)
  126 IB1 = IE+1
      KE = IE + NY
      DO 1126 I = IB1,KE
 1126 X(I) = Z(I)
      IF (MULADD.NE.2.OR.IFACT.EQ.0) GO TO 2126
      CALL ANTILG(X,IB1,KE)
 2126 IE1 = (IB1 + NY - 1) / NY * NY
      JYR = KYR + 1
      KYR = JYR
      ITEST = NY - IE1 + IB1 - 1
      IF (ITEST.NE.0) JYR = JYR - 1
      DO 127 I = 1,L
  127 TMP(I) = BIG
      IM = IB1-(IB1-1)/NY*NY
      ISTOP = 1
      GO TO 105
  128 WRITE(MT1,1045)
 1045 FORMAT('0D 9A. YEAR TO YEAR CHANGE IN IRREGULAR AND SEASONAL COMP
     1NENTS AND MOVING SEASONALITY RATIO')
      IF (NY.EQ.4) GO TO 129
      WRITE(MT1,1039) BLK,TYRLY(5)
      WRITE(MT1,1046) (RATI(I), I = 1,36)
 1046 FORMAT(6X,1HI,3X,12(1X,F8.3),/,
     1        6X,1HS,3X,12(1X,F8.3),/,
     2        4X,5HRATIO,1X,12(1X,F8.2))
      GO TO 9999
  129 WRITE(MT1,1040) BLK,TYRLY(5)
      WRITE(MT1,1047) (RATI(I), I = 1,12)
 1047 FORMAT(6X,1HI,4X,F15.3,3(9X,F15.3),/,
     1        6X,1HS,4X,F15.3,3(9X,F15.3),/,
     2        4X,5HRATIO,2X,F15.2,3(9X,F15.2))
      GO TO 9999
  130 JYR = LYR
      IF (LFDA.NE.1) JYR = JYR+1
      DO 131 I = 1,IE
      WRITE(MT1,1048) JYR,X(I),Y(I)
  131 JYR = JYR+1
 1048 FORMAT(/,4X,2H19,I2,8X,F7.2,5X,F7.2)
```

```
 9999 RETURN
      END
      SUBROUTINE FORCST(STS,IB,IE,KE,NYR,IORDER,WT,R)
C --- THIS SUBROUTINE FORECASTS SEASONALS FROM IE+1 TO KE
C --- IORDER REPRESENTS THE ORDER OF DIFFERENCES USED.
C --- WT IS THE WEIGHT GIVEN TO THE FORECASTED DIFFERENCE.
C --- THERE IS A RATIO OF R BETEEN SUCCESSIVE DIFFERENCES.
      DIMENSION STS(1)
      IF (R.NE.1.0) GO TO 1
      W = WT
      GO TO 2
    1 W = WT*(R-1.0)/(R**IORDER-1.0)
    2 L = KE-IE
      DO 3 I = 1,L
      J = IE+I
      STS(J) = STS(J-NYR)
      DO 3 K = 1,IORDER
    3 STS(J) = STS(J)+W*(R**(IORDER-K))*(STS(J-K*NYR)-STS(J-(K+1)*NYR))
C --- THIS PART IS A BACKCAST FROM IB-1 TO IB-L.
      IF (IB.LE.1) GO TO 5
      DO 4 I = 1,L
      J = IB-I
      STS(J) = STS(J+NYR)
      DO 4 K = 1,IORDER
    4 STS(J) = STS(J)+W*(R**(IORDER-K))*(STS(J+K*NYR)-STS(J+(K+1)*NYR))
    5 RETURN
      END
      SUBROUTINE VTC(KPART)
C --- VARIABLE TREND CYCLE ROUTINE
      COMMON /WORK / TEMP(372)
      COMMON /OPT5 / MULADD,IYRT,MA,IWT
      COMMON /OPT8 / RATIC,KTCOPT,NTERM
      COMMON /OPT9 / LFDA,LYR,LSTMO,LSTYR,LLDA,ID(3),KEXOPT,NYR,IFORC,
     1               LLDAF
      COMMON /MQ4  / STC(372),STCI(372)
      COMMON /WORK2/ W(108)
C --- IF THE SERIES IS MONTHLY APPLY A 13-TERM HENDERSON. IF IT IS
C --- QUARTERLY APPLY A 5-TERM HENDERSON.
      IF (NYR.EQ.4) GO TO 1
      NTERM = 13
      GO TO 2
    1 NTERM = 5
    2 CALL HENDER(W,NTERM)
      IB = LFDA+NTERM/2
      IE = LLDAF-NTERM/2
      DO 3 I = IB,IE
    3 STC(I) = APPLY(STCI,I,W,NTERM)
C --- DROP END  TERMS AND CALCULATE IRREGULAR SERIES.
      CALL DIVSUB(TEMP,STCI,STC,IB,IE)
C --- CALCULATE IBAR/CBAR RATIO.
      IE1 = LLDA-NTERM/2-1
      APCC = 0.0
      APCI = 0.0
      IF (MULADD.EQ.0) GO TO 5
      DO 4 I = IB,IE1
```

```
      APCC = APCC+ABS(STC(I+1)-STC(I))
    4 APCI = APCI+ABS(TEMP(I+1)-TEMP(I))
      GO TO 7
    5 DO 6 I = IB,IE1
      APCC = APCC+ABS(STC(I+1)-STC(I))/STC(I)
    6 APCI = APCI+ABS(TEMP(I+1)-TEMP(I))/TEMP(I)
    7 RATIC = APCI/APCC
      R = RATIC*12/NYR
C --- CHECK IF TREND CYCLE MOVING AVERAGE PRESELECTED.
      GO TO (8,12,15,10,9,15,11),KTCOPT
    8 IF (KPART.EQ.2) GO TO 15
      IF (R.LT.1.0) GO TO 12
    9 IF (KPART.EQ.2) GO TO 15
      IF (R.LT.3.5) GO TO 15
      IF (NYR.EQ.4) GO TO 11
   10 NTERM = 23
      GO TO 13
   11 NTERM = 7
      GO TO 13
   12 NTERM = 9
   13 CALL HENDER(W,NTERM)
      IB = LFDA+NTERM/2
      IE = LLDAF-NTERM/2
      DO 14 I = IB,IE
   14 STC(I) = APPLY(STCI,I,W,NTERM)
   15 I = NTERM
   16 IF (I.EQ.7) GO TO 17
      CALL ENDS(STC,STCI,LFDA,LLDAF,I)
      RETURN
C --- REDUCE SPAN OF HENDERSON AT THE ENDS OF THE SERIES FOR 7 TERM.
   17 I = I-2
      CALL HENDER(W,I)
      IB = IB-1
      IE = IE+1
      STC(IB) = APPLY(STCI,IB,W,I)
      STC(IE) = APPLY(STCI,IE,W,I)
      GO TO 16
      END
      SUBROUTINE SUMRY(X,XBAR,XBAR2,XSQ,XSD,IOPT,I,J)
C --- SUBROUTINE TO CALCULATE SUMMARY MEASURES.
C --- X IS THE INPUT SERIES. I IS THE 1ST VALUE OF X AND J IS THE LAST
C --- VALUE OF X.
C --- XBAR  = AVERAGE CHANGE WITHOUT REGARD TO SIGN.
C --- XBAR2 = AVERAGE CHANGE WITH REGARD TO SIGN.
C --- XSQ   = AVERAGE CHANGE SQUARED WITHOUT REGARD TO SIGN.
C --- XSD   = STD. DEV. OF CHANGES WITH REGARD TO SIGN.
C --- NY    = NUMBER OF SPANS IN ONE YEAR.
C --- XBAR,XBAR2,XSQ, AND XSD ARE CALCULATED FOR SPANS 1 TO NY.
C --- IF THE ADJUSTMENT IS ADDITIVE, THE CHANGES ARE DIFFERENCES.
C --- IF THE ADJUSTMENT IS MULTIPLICATIVE, THEY ARE PERCENT CHANGES.
      DIMENSION X(1),XBAR(1),XBAR2(1),XSQ(1),XSD(1)
      COMMON /OPT5 / MULADD,IYRT,MA,IWT
      COMMON /OPT9 / L1(9),NY,IFORC,LLDAF,LLAF
      DO 9 K = 1,NY
      XBAR(K) = 0.0
```

```
      IF (IOPT.GT.1) GO TO 1
      XBAR2(K) = 0.0
      XSD(K) = 0.0
    1 KJ = J - K
      DO 3 L = I,KJ
      C = X(L+K) - X(L)
      IF (MULADD.NE.0) GO TO 2
      C = C * 100.0 / X(L)
    2 XBAR(K) = XBAR(K) + ABS(C)
      IF (IOPT.GT.1) GO TO 3
      XBAR2(K) = XBAR2(K) + C
    3 CONTINUE
      COUNT = FLOAT(KJ - I + 1)
      XBAR(K) = XBAR(K) / COUNT
      IF (IOPT.EQ.3) GO TO 9
      IF (IOPT.EQ.1) GO TO 4
      XSQ(K) = XBAR(K) * XBAR(K)
      IF (IOPT.EQ.2) GO TO 9
    4 XBAR2(K) = XBAR2(K) / COUNT
      XSD(K) = 0.0
      IF (MULADD.EQ.0) GO TO 6
      DO 5 L = I,KJ
    5 XSD(K)=XSD(K)+(X(L+K)-X(L)-XBAR2(K))**2
      GO TO 8
    6 DO 7 L = I,KJ
    7 XSD(K) = XSD(K)+((X(L+K)-X(L))/X(L)*100.0-XBAR2(K))**2
    8 IF (XSD(K).LE.0.0) GO TO 9
      XSD(K) = SQRT(XSD(K) / COUNT)
    9 CONTINUE
      RETURN
      END
      SUBROUTINE TFMT2(IFMT,KDEC,ITYPE,IPOW)
      DIMENSION IFMT(2),JDEC(6),IP(2)
      DATA JDEC/4H0    ,4H1    ,4H2    ,4H3    ,4H4    ,4H5    /
      DATA IP/4H     ,4H  2P/,IF1,IF2/4H,/) ,4H)    /
      LDEC = JDEC(KDEC + 1)
      IF (ITYPE - 2) 1,2,3
    1 IPW = IP(IPOW + 1)
      IFMT( 5) = IPW
      IFMT( 7) = LDEC
      IFMT(10) = IPW
      IFMT(12) = LDEC
      IF (ITYPE.EQ.1) RETURN
      IFMT(15) = IPW
      IFMT(17) = LDEC
      IFMT(18) = IF1
      RETURN
    2 IFMT( 8) = LDEC
      IFMT(13) = LDEC
      IFMT(21) = LDEC
      RETURN
    3 IFMT(18) = IF2
      RETURN
      END
      SUBROUTINE ADDMUL(Z,X,Y,IB,IE)
```

```
C --- THIS SUBROUTINE MULTIPLIES (ADDS) SERIES X TO Y AND STORES THE
C --- RESULT IN Z.
      COMMON /OPT5 / MULADD,IYRT,MA,IWT
      DIMENSION X(IE),Y(IE),Z(IE)
      IF (MULADD.EQ.0) GO TO 2
      DO 1 I= IB,IE
    1 Z(I) = X(I) + Y(I)
      RETURN
    2 DO 3 I = IB,IE
    3 Z(I) = X(I) * Y(I)
      RETURN
      END
      SUBROUTINE CHANGE(X,Y,IB,IE)
C --- THIS SUBROUTINE CALCULATES THE PERCENT CHANGES (DIFFERENCES) IN
C --- X AND STORES THEM IN Y.
      COMMON /OPT5 / MULADD,IYRT,MA,IWT
      DIMENSION X(1),Y(1)
      IF (MULADD.EQ.1) GO TO 2
      DO 1 I = IB,IE
    1 Y(I) = (X(I)-X(I-1))/X(I-1)
      RETURN
    2 DO 3 I = IB,IE
    3 Y(I) = X(I)-X(I-1)
      RETURN
      END
      SUBROUTINE VSFA(LFDA,LLDA,NYR)
C
C --- SEASONAL FACTOR CURVE ROUTINE.
C
      REAL*4 W9(40)
      REAL*8 TOTALS
      COMMON /WORK / TEMP(372)
      COMMON /WORK2/ SAVG(36),SIMON(36),STIMON(36)
      COMMON /OPT5 / MULADD,MODEL,MA,IWT
      COMMON /OPT7 / RATI(36),RATIS,LTERM,MTYPE,LTER(12)
      COMMON /MQ1  / STS(372),STSI(372)
      COMMON /MQ2  / KPART,KSWV,KSECT,KPRNT,MCD
      DATA W9/0.246,0.221,0.197,0.173,0.112,0.051,0.208,0.192,0.176,
     1        0.160,0.144,0.092,0.028,0.173,0.163,0.154,0.143,0.133,
     2        0.123,0.079,0.032,0.141,0.137,0.132,0.128,0.123,0.117,
     3        0.113,0.075,0.034,0.084,0.120,0.118,0.117,0.116,0.114,
     4        0.113,0.111,0.073,0.034/
      KFDA = LFDA + NYR - 1
      RATIS = 999.99
      R1 = 0.0
      R2 = 0.0
C --- PLACE MONTHLY SI IN SIMON.
      DO 100 J = LFDA,KFDA
      M = J-(J-1)/NYR*NYR
      K = 3
      DO 10 I = J,LLDA,NYR
      K = K+1
   10 SIMON(K) = STSI(I)
C --- COMPUTE A 7-TERM MOVING AVERAGE FOR AN ESTIMATE OF S.
      TMP1 = (SIMON(4)+SIMON(5)+SIMON(6))/3.0
```

```
      TMP2 = (SIMON(K)+SIMON(K-1)+SIMON(K-2))/3.0
      DO 20 I = 1,3
      KI = K+I
      SIMON(I) = TMP1
   20 SIMON(KI) = TMP2
      CALL AVERAG(SIMON,SAVG,1,KI,1,7)
      RATI(M) = 0.0
      RATI(M+NYR) = 0.0
      RATI(M+2*NYR) = 999.99
C --- DIVIDE SI/S FOR AN ESTIMATE OF I.
      CALL DIVSUB(STIMON,SIMON,SAVG,4,K)
C --- ADJUST FOR THE LENGTH OF THE SERIES.
      N = K-4
C --- COMPUTE IBAR,SBAR, AN RATIOS.
      IF (MULADD-1) 30,50,50
   30 DO 40 I = 5,K
      RATI(M) = RATI(M)+ABS(STIMON(I-1)-STIMON(I))/STIMON(I)
   40 RATI(M+NYR) = RATI(M+NYR)+ABS(SAVG(I-1)-SAVG(I))/SAVG(I)
      RATI(M) = RATI(M)*100.0*FIS(CS,N)
      RATI(M+NYR) = RATI(M+NYR)*100.0*CS
      GO TO 70
   50 DO 60 I = 5,K
      RATI(M) = RATI(M)+ABS(STIMON(I-1)-STIMON(I))*FIS(CS,N)
   60 RATI(M+NYR) = RATI(M+NYR)+ABS(SAVG(I-1)-SAVG(I))*CS
   70 R1 = R1+RATI(M)
      R2 = R2+RATI(M+NYR)
      IF (RATI(M)-999.*RATI(M+NYR)) 80,80,90
   80 RATI(M+NYR*2) = RATI(M)/RATI(M+NYR)
   90 FK = N
      RATI(M) = RATI(M)/FK
  100 RATI(M+NYR) = RATI(M+NYR)/FK
      IF (R1-999.*R2) 101,101,102
  101 RATIS = R1/R2
  102 IF (MULADD.NE.2) GO TO 104
      DO 103 I = 1,NYR
      RATI(I) = 100.0*RATI(I)
  103 RATI(I+NYR) = 100.0*RATI(I+NYR)
  104 RETURN
      ENTRY VSFB(LFDA,LLDA,NYR)
      KFDA = LFDA + NYR - 1
      MTYPE = LTERM+1
C --- CHECK IF MOVING AVERAGE IS PRESELECTED.
      GO TO (120,170,180,182,150,181),MTYPE
  120 IF (KSECT.EQ.1) GO TO 170
      IF (KPART.EQ.2) GO TO 180
      IF (LTERM.EQ.0) GO TO 180
      IF (RATIS-1.5) 160,130,130
  130 IF (RATIS-2.5) 170,140,140
  140 IF (RATIS-7.0) 180,150,150
  150 MTYPE = 5
      GO TO 200
  160 MTYPE = 1
      GO TO 200
  170 MTYPE = 2
      GO TO 190
```

```
  180 MTYPE = 3                                                         00035650
      GO TO 190                                                         00035660
  181 MTYPE = 6                                                         00035670
      IF (KSECT.EQ.1) GO TO 170                                         00035680
      GO TO 190                                                         00035690
  182 MTYPE = 4                                                         00035700
C --- IF LESS THAN 5 COMPLETE SWITCH TO STABLE SEASONALITY.             00035710
  190 IF (LLDA-LFDA-5*NYR+1) 150,200,200                                00035720
  200 DO 310 J = LFDA,KFDA                                              00035730
      IF (J.GT.NYR) JJJ=J-NYR                                           00035740
      IF (JJJ .GT. NYR) JJJ=JJJ-NYR                                     00035742
      IF (J.LE.NYR) JJJ=J                                               00035750
      IF (LTER(JJJ).GT.0) MTYPE=LTER(JJJ)+1                             00035760
      K = 0                                                             00035770
      DO 210 I = J,LLDA,NYR                                             00035780
      K = K+1                                                           00035790
  210 SIMON(K) = STSI(I)                                                00035800
      GO TO (250,260,270,291,220,211),MTYPE                             00035810
  211 M = J - (J-1) / NYR * NYR                                         00035820
      IF (RATI(M+2*NYR) - 1.5) 250,212,212                              00035830
  212 IF (RATI(M+2*NYR) - 2.5) 260,213,213                              00035840
  213 IF (RATI(M+2*NYR) - 7.0) 270,220,220                              00035850
C --- STABLE SEASONAL. AVERAGE OF ALL SI RATIOS FOR THIS MONTH.         00035860
  220 TMP1 = TOTALS(SIMON,1,K,1,1)                                      00035870
      DO 240 I = 1,K                                                    00035880
  240 SAVG(I) = TMP1                                                    00035890
      GO TO 300                                                         00035900
C --- COMPUTE A 3-TERM MOVING AVERAGE.                                  00035910
  250 CALL AVERAG(SIMON,SAVG,1,K,1,3)                                   00035920
      SAVG(1) = 0.61*SIMON(1)+0.39*SIMON(2)                             00035930
      SAVG(K) = 0.61*SIMON(K)+0.39*SIMON(K-1)                           00035940
      GO TO 300                                                         00035950
C --- COMPUTE A 3X3 MOVING AVERAGE.                                     00035960
  260 CALL AVERAG(SIMON,SAVG,1,K,3,3)                                   00035970
      SAVG(1) = (11.0*(SIMON(1)+SIMON(2))+5.0*SIMON(3))/27.0            00035980
      SAVG(K) = (11.0*(SIMON(K)+SIMON(K-1))+5.0*SIMON(K-2))/27.0        00035990
      SAVG(2) = (0.7*(SIMON(1)+SIMON(3))+SIMON(2)+0.3*SIMON(4))/2.7     00036000
      SAVG(K-1) = (0.7*(SIMON(K)+SIMON(K-2))+SIMON(K-1)+0.3*SIMON(K-3))/00036010
     1 2.7                                                              00036020
      GO TO 300                                                         00036030
C --- COMPUTE A 3X5 MOVING AVERAGE.                                     00036040
  270 CALL AVERAG(SIMON,SAVG,1,K,3,5)                                   00036050
      SAVG(1) = (17.0*(SIMON(1)+SIMON(2)+SIMON(3))+9.0*SIMON(4))/60.0   00036060
      SAVG(K) = (17.0*(SIMON(K)+SIMON(K-1)+SIMON(K-2))+9.0*SIMON(K-3))/ 00036070
     1 60.0                                                             00036080
      SAVG(2) = (15.0*(SIMON(1)+SIMON(2)+SIMON(3))+11.0*SIMON(4)+       00036090
     1 4.0*SIMON(5))/60.0                                               00036100
      SAVG(K-1) = (15.0*(SIMON(K)+SIMON(K-1)+SIMON(K-2))+11.0*SIMON(K-3)00036110
     1 +4.0*SIMON(K-4))/60.0                                            00036120
      IF (K.EQ.5) GO TO 290                                             00036130
      SAVG(3) = (9.0*SIMON(1)+13.0*(SIMON(2)+SIMON(3)+SIMON(4))+        00036140
     1 8.0*SIMON(5)+4.0*SIMON(6))/60.0                                  00036150
      SAVG(K-2) = (9.0*SIMON(K)+13.0*(SIMON(K-1)+SIMON(K-2)+SIMON(K-3))+00036160
     1 8.0*SIMON(K-4)+4.0*SIMON(K-5))/60.0                              00036170
      GO TO 300                                                         00036180
```

```
  290 SAVG(3) = (SIMON(1)+SIMON(2)+SIMON(3)+SIMON(4)+SIMON(5))/5.0
      GO TO 300
C --- COMPUTE A 3X9 MOVING AVERAGE
  291 CALL AVERAG(SIMON,SAVG,1,K,3,9)
C --- APPLY END WEIGHTS FOR THE 3X9
      KK = 0
      DO 293 JJ = 1,5
      J1 = JJ
      J2 = K-JJ+1
      JK = JJ+5
      IF (JK.GT.K) GO TO 294
      SAVG(J1) = 0.0
      SAVG(J2) = 0.0
      DO 292 L = 1,JK
      SAVG(J1) = SAVG(J1)+W9(KK+L)*SIMON(L)
      IF (J1.EQ.J2) GO TO 292
      SAVG(J2) = SAVG(J2)+W9(KK+L)*SIMON(K-L+1)
  292 CONTINUE
  293 KK = KK+JK
      GO TO 300
  294 SAVG(J1) = TOTALS(SIMON,1,K,1,1)
      SAVG(J2) = SAVG(J1)
  300 K = 0
      DO 310 I = J,LLDA,NYR
      K = K+1
  310 STS(I) = SAVG(K)
C --- APPLY A 2 X NYR MOVING AVERAGE TO THE SEASONALS.
      CALL AVERAG(STS,TEMP,LFDA,LLDA,2,NYR)
      K = NYR/2
      KFDA = LFDA+K
      KLDA = LLDA-K
C --- FILL IN THE MISSING END TERMS BY REPEATING FIRST AND LAST
C --- AVAILABLE MOVING AVERAGE VALUE.
      IF (MTYPE.EQ.5) GO TO 330
      DO 320 I = 1,K
      TEMP(KFDA-I) = TEMP(KFDA)
  320 TEMP(KLDA+I) = TEMP(KLDA)
      GO TO 350
  330 DO 340 I = 1,K
      TEMP(KFDA-I) = TEMP(KFDA-I+NYR)
  340 TEMP(KLDA+I) = TEMP(KLDA+I-NYR)
C --- DIVIDE SEASONALS BY THE 2 X NYR MOVING AVERAGE
  350 CALL DIVSUB(STS,STS,TEMP,LFDA,LLDA)
      RETURN
      END
      SUBROUTINE AVERAG(X,Y,IB,IE,M,N)
C
C --- THIS SUBROUTINE APPLIES AN M-OF-N MOVING AVERAGE TO THE SERIES
C ---  X AND STORES THE RESULTS IN Y.
C
      DIMENSION X(IE),Y(IE)
      KI = (M+N)/2-1
      KB = IB+KI
      KE = IE-KI
      IF (KE-KB) 4,1,1
```

```
    1 FMN = M*N
      DO 3 K = KB,KE
      TMP = 0.0
      I1 = K-KI
      I2 = I1+M-1
      DO 2 I = I1,I2
      JI = I+N-1
      DO 2 J = I,JI
    2 TMP = TMP+X(J)
    3 Y(K) = TMP/FMN
    4 RETURN
      END
      SUBROUTINE HENDER(W,N)
C
C     THIS SUBROUTINE GENERATES THE WEIGHTS FOR AN N-TERM HENDERSON
C     MOVING AVERAGE. THE WEIGHTS ARE STORED IN W. ONLY HALF THE WEIGHTS
C     ARE GENERATED SINCE W IS SYMMETRIC.
C
      DIMENSION W(1)
      Y = FLOAT((N+3)/2)
      M = (N+1)/2
      Y1 = (Y-1.0)*(Y-1.0)
      Y2 = Y*Y
      Y3 = (Y+1.0)*(Y+1.0)
      Y4 = 3.0*Y2-16.0
      Y5 = 4.0*Y2
      DENOMI = 8.0*Y*(Y2-1.0)*(Y5-1.0)*(Y5-9.0)*(Y5-25.0)/315.0
      DO 1 I = 1,M
      X = FLOAT((I-1)*(I-1))
    1 W(I) = (Y1-X)*(Y2-X)*(Y3-X)*(Y4-11.0*X)/DENOMI
      RETURN
      END
      FUNCTION APPLY(X,K,W,N)
C
C     THIS FUNCTION APPLIES SYMMETRIC WEIGHTS TO X(K)
C
      DIMENSION X(1),W(1)
      M = (N+1)/2
      APPLY = W(1)*X(K)
      DO 1 I = 2,M
      J = K-I+1
      L = K+I-1
    1 APPLY = APPLY + W(I) * (X(J) + X(L))
      RETURN
      END
      SUBROUTINE F4GEN(NW)
C --- THIS SUBROUTINE  PRINTS IMAGES OF THE MAIN CONTROL
C --- CARDS.
      INTEGER*2  IEF4ST(80),N(10),ARCAIM(80),XOPT,XCDIM(80),BLNK
      REAL*8  KSER,IDENT
      COMMON /PRIOR/ KSER,KFORM(20),KFMT
      COMMON /OPT2/ KDEC,ICARD,KHCNOP,KHCARD(20),KHCARC(20)
      COMMON /OPT3/ IDENT(9),LDFC(9),JFORM(20),JFMT
      COMMON /F4STC/ IEF4ST,ARCAIM,JIFORC,XCDIM,LFMT,LFORM(20),XOPT
      DATA N/1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1H0/,DASH/4H----/
```

```
      DATA BLNK/2H  /
      WRITE(NW,10) (DASH,I=1,30)
   10 FORMAT(////////,1X,30A4)
      WRITE(NW,30) (N(I),I=1,8),(N,I=1,8)
   30 FORMAT(//,T39,8(9X,A1),/,T11,'COLUMN NUMBER                    : ',80A1)
      WRITE(NW,40) IEF4ST
   40 FORMAT(/  T8, 'IMAGE OF THE MAIN OPTION CARD: ', 80A1)
      IF(XOPT.NE.BLNK)WRITE(NW,3)XCDIM
    3 FORMAT(1H ,T7,'IMAGE OF THE EXTRA OPTION CARD: ',80A1)
    2 IF(JIFORC.EQ.2)WRITE(NW,1)ARCAIM
    1 FORMAT(1H ,T7,'IMAGE OF THE ARIMA OPTION CARD: ',80A1)
    4 IF(LFMT.EQ.1)WRITE(NW,5)LFORM
    5 FORMAT(1H ,T7,'IMAGE OF THE INPUT FORMAT CARD: ',20A4)
   65 FORMAT(1H ,T7,'IMAGE OF THE PRIOR FORMAT CARD: ',20A4)
   70 FORMAT(1H ,T5,'IMAGE OF THE SPECIAL OUTPUT CARD: ',20A4)
   75 FORMAT(1H ,T2,'IMAGE OF SPECIAL OUTPUT FORMAT CARD: ',20A4)
      IF (KFMT .EQ. 2) WRITE(NW,65)KFORM
      IF (ICARD .EQ. 1) WRITE(NW,70)KHCARO
      IF (KHCNOP .GT. 7) WRITE(NW,70)KHCARC
      IF (JFMT .EQ. 1) WRITE(NW,75)JFORM
      RETURN
      END
      SUBROUTINE DIVSUB (RESULT,ARRAY1,ARRAY2,JFDA,JLDA)
C
C --- THIS ROUTINE DIVIDES ARRAY1 BY ARRAY2 OR SUBTRACTS ARRAY2
C --- FROM ARRAY1 DEPENDING ON WHETHER A MULTIPLICATIVE OR ADDITIVE
C --- ADJUSTMENT IS BEING MADE.
      DIMENSION RESULT(1),ARRAY1(1),ARRAY2(1)
      COMMON /OPT5 / MULADD,IYRT,MA,IWT
      IF (MULADD.NE.0) GO TO 2
      DO 3 I = JFDA,JLDA
    3 RESULT(I) = ARRAY1(I)/ARRAY2(I)
      RETURN
    2 DO 4 I = JFDA,JLDA
    4 RESULT(I) = ARRAY1(I)-ARRAY2(I)
      RETURN
      END
      SUBROUTINE DOW (DWT,ARRAY,JYR,JFDA,JLDA,JOPT)
C
C --- THIS ROUTINE COMBINES SEVEN DAILY WEIGHTS INTO MONTHLY FACTORS
      DIMENSION ARRAY(1),DWT(7),MDAYS(12),ADJ(12)
      COMMON /OPT5 / MULADD,IYRT,MA,IWT
      COMMON /WORK2/ WT(108)
      DATA  MDAYS /3,0,3,2,3,2,3,3,2,3,2,3/
      DATA  ADJ /28.25,30.,31.,28.25,30.4375,30.4375,30.4375,
     130.4375,29.,30.,31.,28./
      TKON = 0.
      IF (MULADD.EQ.1) GO TO 33
    2 TKON = 28.
      DO 1 I = 1,4
      M = I + JOPT
    1 WT(I) = ADJ(M)
C --- STANDARDIZE WEIGHTS TO TOTAL 7.0
      TMP=0
      DO 32 I = 1,7
```

```
   32 TMP = TMP + DWT(I)
      DO 31 I = 1, 7
   31 DWT(I) = (DWT(I)*7.)/TMP
C --- SET PERPETUAL CALENDAR FOR JANUARY THE FIRST OF YEAR JYR
   33 I = JYR + 12 -((JYR+11)/28) * 28
      I = I + (I-1)/4
      IF (JFDA.EQ.1) GO TO 40
C --- SET THE CALENDAR FOR MONTH JFDA
      NEXDAY=0
      JFDAM1=JFDA-1
      DO 3 J=1,JFDAM1
      NEXDAY=NEXDAY+MDAYS(J)
    3 CONTINUE
      IF (JYR.NE.(JYR/4)*4) GO TO 6
      IF (JFDA.LE.2) GO TO 6
      NEXDAY=NEXDAY+1
    6 I=I+NEXDAY
   40 I = I - ((I-1)/7) * 7
      M = JFDA
      IF (JFDA-12) 5,5,4
    4 M = M - 12
    5 JL = JLDA + 12
      L = JYR - 1
      IF (JFDA-2) 8,8,7
    7 L = JYR
    8 DO 30 J = JFDA,JL
      ARRAY(J) = TKON
      N = MDAYS(M)
      IF (N) 19, 13, 19
   13 L = L + 1
      N = 4
      IF (L - (L/4) * 4) 25, 9, 25
    9 N = 1
   19 K = 0
   20 ARRAY(J) = ARRAY(J) + DWT(I)
      I = I + 1
      IF (I - 7) 22, 22, 21
   21 I = 1
   22 K = K + 1
      IF (K-N) 20, 25, 25
   25 IF (MULADD.EQ.1) GO TO 28
      ARRAY(J) = ARRAY(J)/WT(N)
   28 M = M + 1
      IF (M-12) 30, 30, 29
   29 M = 1
   30 CONTINUE
      RETURN
      END
      SUBROUTINE AVEDUR(Y,L,M,ADR)
C --- AVERAGE DURATION OF RUN SUBROUTINE
      DIMENSION Y(1)
      I = L
      RUNS = 1.
      IF (Y(I)-Y(I+1)) 15,12,18
   12 I = I + 1
```

```
      IF (I-M) 13,21,21
   13 IF (Y(I)-Y(I+1)) 15,12,17
   15 I = I + 1
      IF (I-M) 16,21,21
   16 IF (Y(I)-Y(I+1)) 15,15,17
   17 RUNS = RUNS + 1.0
   18 I = I + 1
      IF (I-M) 19,21,21
   19 IF (Y(I)-Y(I+1)) 20,18,18
   20 RUNS = RUNS + 1.0
      GO TO 15
   21 COUNT = M - L
      ADR = COUNT / RUNS
      RETURN
      END
      SUBROUTINE QCONTR(MAL,MQ)
C --- THIS SUBROUTINE PRINTS THE QUALITY CONTROL STATISTICS IN A
C --- SUMMARIZED FORM AT THE END OF THE PRINTOUT.
      REAL*8 TITLE,SERNO,TYPE(6),AGGS(3)
      COMMON /UNITS/ MT,MT2,MT1,MP,NF
      COMMON /ROTE / TITLE(10),SERNO
      COMMON /WORK2/ TMP(12),Q
      COMMON /TESTS/ F(12),IQFAIL
      COMMON /MQ11 / IMESSG,IAGR
      DIMENSION IFMT(21),INDF(2)
      DATA IFMT/4H(2X,,4HA7,A,4H8,19,4HH --,4H----,4H-- -,4H----,4H--- ,
     1          4H,5A8,4H,    ,4H 5X,,4HF4.2,4H,A1 ,4H,T11,4H0,3H,4H---,,
     2          4H 1X,,4HA8, ,4H/,37,4HX,5A,4H8,/)/
      DATA TYPE/5HM-MLT,5HM-ADD,5HM-LOG,5HQ-MLT,5HQ-ADD,5HQ-LOG/
      DATA IF1,IF2/4H 6X,,4H19X,/,INDF/1H ,1H*/
      DATA AGGS/8H        ,8HDIRECT  ,8HINDIRECT/
      K = IAGR-1
      IF (K.LT.1) K = 1
      I = MAL+1
      IF (MQ.EQ.4) I = I+3
      IFMT(11) = IF2
      IF (Q.GT.1.0.OR.IQFAIL.EQ.2) IFMT(11) = IF1
      WRITE(NF,IFMT) TYPE(I),SERNO,(TITLE(J),J=1,5),Q,INDF(IQFAIL),
     1               AGGS(K),(TITLE(J),J=6,10)
      IF (IQFAIL.EQ.2) IMESSG = 1
      RETURN
      END
      SUBROUTINE PERIOD
C
      INTEGER*2 XX,BLK,S1,S2
      COMMON /UNITS/ MT,MT2,NW,MP,NG,NFORM
      COMMON /WORK / C(180),XN(180),Z(11),FN
      COMMON /WORK3/ CI(180),DUM(168),FMT,PRAVIN,PRAVIM,SUM1,X1,X2,
     1               VAR,FI,SUM2,T,T1,HIPS
      COMMON /OPT5 / MULADD,IYRT,MA,IWT
      COMMON /OPT9 / N1,I1(3),N2,I2(4),MFREQ
      COMMON /MQ7  / R(372)
      DATA PI2/6.2831853/
      DATA BLK,S1,S2/2H  ,2H* ,2H**/
      SCALE = 1.0
```

```
      IF (MULADD.NE.1) SCALE = 10000.
      N=N2-N1+1
      FN=FLOAT(N)
      IMP=N/2
      FMT=SQRT(FLOAT(IMP-1))
      PRAVIN=1.36/FMT
      PRAVIM=1.02/FMT
      VAR = VARIAN(R,N1,N2,0)
      DO 1 I=1,IMP
      FI=FLOAT(I)/FN
      XN(I) = FI*MFREQ
      FI=FI*PI2
      SUM1=0.0
      SUM2=0.0
      DO 2  J=N1,N2
      T=FLOAT(J-N1+1)*FI
      SUM1=SUM1+R(J)*COS(T)
    2 SUM2=SUM2+R(J)*SIN(T)
    1 C(I)=(2.0/FN)*(SUM1*SUM1+SUM2*SUM2)
      IN=2*IMP
      IF(IN.EQ.N)C(IMP)=C(IMP)/2.0
      CI(1)=C(1)/VAR
      DO 3 I=2,IMP
    3 CI(I)=CI(I-1)+C(I)/VAR
      WRITE(NW,4)
    4 FORMAT(1H0,9X,'FREQUENCY',21X,'  PERIOD  ',49X,'CUMULATIVE',
     2 4X,'CONFIDENCE')
      WRITE(NW,5)
    5 FORMAT(1H ,8X,'IN CYC/YEAR ',20X,'IN YEARS ',18X,'PERIODOGRAM',
     *   19X,'PERIODOGRAM',6X,'LEVELS',/)
      DO 6 I=1,IMP
      C(I) = C(I)*SCALE
      HIPS=1.0/XN(I)
      XX=BLK
      X1=XN(I)/XN(IMP)-PRAVIM
      X2=X1+2.0*PRAVIM
      IF(X1.LT.CI(I).AND.X2.GT.CI(I))GO TO 6
      XX=S1
      X1=X1+PRAVIM-PRAVIN
      X2=X1+2.0*PRAVIN
      IF(X1.LT.CI(I).AND.X2.GT.CI(I))GO TO 6
      XX=S2
    6 WRITE(NW,7) XN(I),HIPS,C(I),CI(I),XX
    7 FORMAT(1H ,6X,F12.6,17X,F12.4,15X,F15.4,14X,F15.4,8X,A2)
      WRITE(NW,13)
   13 FORMAT(1H0,7X, 'CONFIDENCE LEVELS (  )-0% TO 75% LEVEL',/,26X,'(*
     2)-75% TO 95% LEVEL',/,26X,'(**)- OVER 95% LEVEL')
      WRITE(NW,9) PRAVIN,PRAVIM
      WRITE(NW,8)
    8 FORMAT(1H0,5X,'KOLMOGOROV-SMIRNOV SIGNIFICANCE TESTS FOR THE FINAL
     1 IRREGULARS')
    9 FORMAT(1H0,10X,'95% LEVEL=',F10.4,10X,'75% LEVEL=',F10.4,//)
      WRITE(NW,10)
   10 FORMAT(1H0,10X,'GRAPH OF THE CUMULATIVE PERIODOGRAM')
      WRITE(NW,12)
```

```
   12 FORMAT(1H ,23X,'(.) REPRESENTS 95% CONFIDENCE LIMITS',/,
     2 24X,'(*) REPRESENTS CUMULATIVE PERIODOGRAM')
      CALL GRAPH(XN,CI,IMP,PRAVIN,Z,NW)
      RETURN
      END
      SUBROUTINE GRAPH(X,Y,NPOINT,SIG,Z,NW)
      INTEGER*2 A(102),SP,PA,BLANK,GRIDMK
      DIMENSION X(1),Y(1),Z(11)
      DATA GRIDMK/'*'/,BLANK/' '/,A/102*' '/,PA/'.'/,SP/'+'/
  702 FORMAT(/17X,11F10.1)
  706 FORMAT(25X,10('+---------'),'+')
  708 FORMAT(/11X,F13.2,1X,101A1)
      DO 150 I=1,11
  150 Z(I)=FLOAT(I-1)/10.0
      WRITE(NW,702)Z
      WRITE(NW,706)
      DO 220 I=1,NPOINT
      A(1)=SP
      A(101)=SP
      Y1=X(I)/X(NPOINT)-SIG
      N1=INT(Y1*100.0+1.5)
      IF(N1.LT.101.AND.N1.GT.0)A(N1)=PA
      Y2=Y1+2.0*SIG
      N2=INT(Y2*100.0+1.5)
      IF(N2.LT.101.AND.N2.GT.0)A(N2)=PA
      N=INT(Y(I)*100.0+1.5)
      A(N)=GRIDMK
      WRITE(NW,708)X(I),(A(K),K=1,101)
      IF(N1.LT.101.AND.N1.GT.0)A(N1)=BLANK
      IF(N2.LT.101.AND.N2.GT.0)A(N2)=BLANK
  220 A(N)=BLANK
      WRITE(NW,706)
      WRITE(NW,702)Z
      RETURN
      END
      SUBROUTINE QMAP (SERIES,STCI,STCI2,LFDA,LLDA,NY,NS,NE)
C
C --- THIS SUBROUTINE MODIFIES THE FINAL SEASONALLY ADJUSTED SERIES
C --- BY FORCING THE YEARLY TOTALS TO BE EQUAL TO THE YEARLY TOTALS OF
C --- OF THE ORIGINAL SERIES A1 (OR B1).
C
      REAL*8 W,WQ,WM1,WM2,TMP1,TMP2,R
      COMMON /WORK / R(186)
      DIMENSION W(200),WM1(75),WM2(75),WQ(50),SERIES(1),STCI(1),STCI2(1)
      EQUIVALENCE (W(1),WQ(1)),      (W(51),WM1(1)),      (W(126),WM2(1))
      DATA WQ /   0.3101014,-0.0745478, 0.0179083,-0.0042489, 0.0007868,
     *            0.2860609,-0.0447286, 0.0107450,-0.0025492, 0.0004721,
     *            0.2379797, 0.0149095,-0.0035817, 0.0008498,-0.0001575,
     *            0.1658580, 0.1043667,-0.0250716, 0.0059485,-0.0011016,
     *            0.0696957, 0.2236430,-0.0537249, 0.0127467,-0.0023605,
     *            0.0033526, 0.2818963,-0.0436963, 0.0103673,-0.0019199,
     *           -0.0331716, 0.2791267, 0.0050143,-0.0011897, 0.0002203,
     *           -0.0398767, 0.2153340, 0.0924069,-0.0219243, 0.0040601,
     *           -0.0167627, 0.0905184, 0.2184814,-0.0518365, 0.0095994,
     *           -0.0008120, 0.0043849, 0.2815186,-0.0430668, 0.0079753/
```

```
      DATA WM1 /  0.1053950,-0.0279005, 0.0073776,-0.0019194, 0.0003807,00040040
     *            0.1044693,-0.0267298, 0.0070680,-0.0018389, 0.0003647,00040050
     *            0.1026180,-0.0243885, 0.0064489,-0.0016778, 0.0003327,00040060
     *            0.0998410,-0.0208766, 0.0055203,-0.0014362, 0.0002849,00040070
     *            0.0961383,-0.0161940, 0.0042821,-0.0011141, 0.0002210,00040080
     *            0.0915100,-0.0103407, 0.0027344,-0.0007114, 0.0001410,00040090
     *            0.0859560,-0.0033168, 0.0008771,-0.0002282, 0.0000453,00040100
     *            0.0794764, 0.0048777,-0.0012898, 0.0003356,-0.0000666,00040110
     *            0.0720711, 0.0142429,-0.0037662, 0.0009799,-0.0001943,00040120
     *            0.0637402, 0.0247787,-0.0065521, 0.0017046,-0.0003381,00040130
     *            0.0544835, 0.0364852,-0.0096476, 0.0025100,-0.0004978,00040140
     *            0.0443012, 0.0493624,-0.0130527, 0.0033959,-0.0006735,00040150
     *            0.0331933, 0.0634102,-0.0167673, 0.0043624,-0.0008652,00040160
     *            0.0232560, 0.0750521,-0.0189211, 0.0049227,-0.0009764,00040170
     *            0.0144893, 0.0842882,-0.0195142, 0.0050770,-0.0010070/00040180
      DATA WM2 /  0.0068933, 0.0911184,-0.0185466, 0.0048253,-0.0009570,00040190
     *            0.0004679, 0.0955427,-0.0160183, 0.0041675,-0.0008267,00040200
     *           -0.0047868, 0.0975612,-0.0119292, 0.0031037,-0.0006156,00040210
     *           -0.0088708, 0.0971739,-0.0062794, 0.0016337,-0.0003240,00040220
     *           -0.0117842, 0.0943806, 0.0009312,-0.0002423, 0.0000480,00040230
     *           -0.0135270, 0.0891815, 0.0097025,-0.0025243, 0.0005007,00040240
     *           -0.0140991, 0.0815765, 0.0200345,-0.0052124, 0.0010338,00040250
     *           -0.0135006, 0.0715657, 0.0319272,-0.0083065, 0.0016475,00040260
     *           -0.0117315, 0.0591490, 0.0453807,-0.0118068, 0.0023417,00040270
     *           -0.0087915, 0.0443265, 0.0603949,-0.0157130, 0.0031165,00040280
     *           -0.0061612, 0.0310647, 0.0729068,-0.0180586, 0.0035816,00040290
     *           -0.0038405, 0.0193636, 0.0829163,-0.0188434, 0.0037373,00040300
     *           -0.0018293, 0.0092233, 0.0904234,-0.0180674, 0.0035834,00040310
     *           -0.0001277, 0.0006437, 0.0954281,-0.0157308, 0.0031200,00040320
     *            0.0012644,-0.0063752, 0.0979305,-0.0118334, 0.0023470/00040330
      NS = LFDA                                                           00040340
      NTEST = (LFDA-1)/NY*NY+1                                            00040350
      IF (NTEST.NE.LFDA) NS = NTEST+NY                                    00040360
      NE = LLDA/NY*NY                                                     00040370
      N1 = (NS-1)/NY+1                                                    00040380
      N2 = NE/NY                                                          00040390
      DO 1 I = N1,N2                                                      00040400
      N3 = (I-1)*NY+1                                                     00040410
      N4 = I*NY                                                           00040420
      R(I) = 0.0D0                                                        00040430
      DO 1 J = N3,N4                                                      00040440
    1 R(I) = R(I)+(DBLE(SERIES(J))-DBLE(STCI(J)))                         00040450
      JY = 0                                                              00040460
      IF (NY.EQ.12) JY = 50                                               00040470
      K1 = 2*NY                                                           00040480
      DO 3 I = 1,K1                                                       00040490
      I1 = NS+I-1                                                         00040500
      I2 = NE-I+1                                                         00040510
      TMP1 = DBLE(STCI(I1))                                               00040520
      TMP2 = DBLE(STCI(I2))                                               00040530
      II = (I-1)*5+JY                                                     00040540
      DO 2 J = 1,5                                                        00040550
      J1 = N1+J-1                                                         00040560
      J2 = N2-J+1                                                         00040570
      IJ = II+J                                                           00040580
```

```
      TMP1 = TMP1+R(J1)*W(IJ)
    2 TMP2 = TMP2+R(J2)*W(IJ)
      STCI2(I1) = SNGL(TMP1)
    3 STCI2(I2) = SNGL(TMP2)
      L1 = N1+2
      L2 = N2-2
      DO 5 L = L1,L2
      K2 = (L-1)*NY+1
      K3 = K2+NY/2-1
      J1 = L-2
      J2 = L+2
      DO 5 I = K2,K3
      I2 = 2*K2+NY-I-1
      TMP1 = DBLE(STCI(I))
      TMP2 = DBLE(STCI(I2))
      II = 5*(I+K1-K2)+1+JY
      DO 4 J = J1,J2
      IJ = II+J-J1
      TMP1 = TMP1+R(J)*W(IJ)
    4 TMP2 = TMP2+R(J2+J1-J)*W(IJ)
      STCI2(I) = SNGL(TMP1)
    5 STCI2(I2) = SNGL(TMP2)
      RETURN
      END
      SUBROUTINE FTEST(X,IB,IE,NYR,IND,KPROPT)
C --- THIS ROUTINE COMPUTES A ONE-WAY ANALYSIS OF VARIANCE ON SERIES
C --- X. IF THE TREND HAS NOT BEEN REMOVED FROM X, IT IS  ELIMINATED
C --- BY A FIRST DIFFERENCE.
      REAL*8 MQ,SUMT,SSQT,SSM,SUMM,ST,SSR,FMSM,FMSR,DFR,DFM,C
      DIMENSION X(IE)
      COMMON /UNITS/ MT,MT2,NW,MP,NG,NFORM
      COMMON /WORK / TEMP(372),SUMM,SUMT,SSQT,SSM,SSR,FMSR
      COMMON /TESTS/ FSTABL,FMOVE,FPRES,FTDAY,CHI,P1,P2,P3,P4,P5,TEST1,
     1               TEST2,IQFAIL
      COMMON /OPT5 / MULADD,IYRT,MA,IWT
      COMMON /MQ3  / PCD(10),MQ,QM(4),MQCD
      DATA BLANK,STAR/2H  ,2H* /,STAR2/2H**/
      C = 1.0D0
      IF (MULADD.EQ.0) C = 10000.0D0
      ITYPE = 0
      IF (IND.EQ.1) GO TO 2
      DO 1 I = IB,IE
    1 TEMP(I) = X(I)
      KB = IB
      GO TO 4
    2 L = NYR / 4
      KB = IB + L
      DO 3 I = KB,IE
    3 TEMP(I) = X(I)-X(I - L)
    4 NT = 0
      SUMT = 0.0D0
      SSQT = 0.0D0
      SSM = 0.0D0
      DO 6 I = 1,NYR
      NM = 0
```

```
      SUMM = 0.0D0                                                      00041140
      JI = I+KB-1                                                       00041150
      DO 5 J = JI,IE,NYR                                                00041160
      NM = NM+1                                                         00041170
      SUMM = SUMM+TEMP(J)                                               00041180
    5 SSQT = SSQT+TEMP(J)*TEMP(J)                                       00041190
      NT = NT+NM                                                        00041200
      SUMT = SUMT+SUMM                                                  00041210
    6 SSM = SSM+SUMM*SUMM/NM                                            00041220
      ST = NT                                                           00041230
      SSQT = (SSQT-SUMT*SUMT/ST)*C                                      00041240
      SSM = (SSM-SUMT*SUMT/ST)*C                                        00041250
      SSR = SSQT - SSM                                                  00041260
      KDFR = NT-NYR                                                     00041270
      KDFT = NT-1                                                       00041280
      KDFB = NYR-1                                                      00041290
      DFR = KDFR                                                        00041300
      DFB = KDFB                                                        00041310
      FMSM = SSM/DFB                                                    00041320
      FMSR = SSR/DFR                                                    00041330
      F = FMSM/FMSR                                                     00041340
      PROB = FVALUE(F,KDFB,KDFR)*100.0                                  00041350
      IF (IND.NE.0) GO TO 7                                             00041360
      FSTABL = F                                                        00041370
      P1 = PROB                                                         00041380
      GO TO 8                                                           00041390
    7 IF (IND.EQ.1) GO TO 8                                             00041400
      FPRES = F                                                         00041410
      P3 = PROB                                                         00041420
      IF (KPROPT.EQ.3) RETURN                                           00041430
    8 IF (KPROPT.GT.3) RETURN                                           00041440
      IF(PROB.GT.0.1) GO TO 9                                           00041450
      FSTAR = STAR2                                                     00041460
      GO TO 11                                                          00041470
    9 IF(PROB.GT.1.0) GO TO 10                                          00041480
      FSTAR = STAR                                                      00041490
      GO TO 11                                                          00041500
   10 FSTAR = BLANK                                                     00041510
   11 IF (IND.EQ.1) GO TO 13                                            00041520
      WRITE(NW,100)                                                     00041530
  100 FORMAT(/,20X,55HTEST FOR THE PRESENCE OF SEASONALITY ASSUMING STAB00041540
     1ILITY,/)                                                          00041550
      WRITE(NW,110) MQ,SSM,KDFB,FMSM,F,FSTAR,SSR,KDFR,FMSR,SSQT,KDFT    00041560
  110 FORMAT(43X,6HSUM OF,5X,7HDGRS.OF,9X,4HMEAN,/,                     00041570
     1 42X,7HSQUARES,5X,7HFREEDOM,8X,6HSQUARE,7X,7HF-VALUE,/,           00041580
     2 18X,8HBETWEEN ,A7,1HS,F17.4,I9,F17.5,F12.3,A2,/,                 00041590
     3 26X,8HRESIDUAL,F17.4,I9,F17.5,/,                                 00041600
     4 29X,5HTOTAL,F17.4,I9,/)                                          00041610
      IF (FSTAR.NE.STAR2) GO TO 12                                      00041620
      WRITE(NW,120) FSTAR                                               00041630
  120 FORMAT(28X,A2,45HSEASONALITY PRESENT AT THE 0.1 PER CENT LEVEL)   00041640
      GO TO 20                                                          00041650
   12 WRITE(NW,130) FSTAR                                               00041660
  130 FORMAT(28X,A2,59HNO EVIDENCE OF STABLE SEASONALITY AT THE 0.1 PER 00041670
     1CENT LEVEL)                                                       00041680
```

```
      GO TO 20                                                          00041690
   13 IF (ITYPE.EQ.1) GO TO 16                                          00041700
      WRITE(NW,140)                                                     00041710
  140 FORMAT(/,42X,'TEST FOR THE PRESENCE OF RESIDUAL SEASONALITY')     00041720
      IF (FSTAR.EQ.BLANK) GO TO 14                                      00041730
      WRITE(NW,150) FSTAR,F                                             00041740
  150 FORMAT(/,18X,A2,'RESIDUAL SEASONALITY PRESENT IN THE ENTIRE SERIES00041750
     1 AT THE 1 PER CENT LEVEL.          F =',F10.2)                    00041760
      GO TO 15                                                          00041770
   14 WRITE(NW,160) FSTAR,F                                             00041780
  160 FORMAT(/,18X,A2,'NO EVIDENCE OF RESIDUAL SEASONALITY IN THE ENTIRE00041790
     1 SERIES AT THE 1 PER CENT LEVEL. F =',F10.2)                      00041800
   15 KB = IE-3*NYR+1                                                   00041810
      IF (KB.LE.IB) RETURN                                              00041820
      ITYPE = 1                                                         00041830
      GO TO 4                                                           00041840
   16 IF (FSTAR.NE.BLANK) GO TO 18                                      00041850
      WRITE(NW,170) FSTAR,F                                             00041860
  170 FORMAT(/,18X,A2,'NO EVIDENCE OF RESIDUAL SEASONALITY IN THE LAST 300041870
     1 YEARS AT THE 1 PER CENT LEVEL.  F =',F10.2)                      00041880
      IF (PROB.GT.5.0) GO TO 17                                         00041890
      WRITE(NW,180) FSTAR                                               00041900
  180 FORMAT(/,18X,A2,'RESIDUAL SEASONALITY PRESENT IN THE LAST 3 YEARS 00041910
     1AT THE 5 PER CENT LEVEL.')                                        00041920
      GO TO 19                                                          00041930
   17 WRITE(NW,190) FSTAR                                               00041940
  190 FORMAT(/,18X,A2,'NO EVIDENCE OF RESIDUAL SEASONALITY IN THE LAST 300041950
     1 YEARS AT THE 5 PER CENT LEVEL.')                                 00041960
      GO TO 19                                                          00041970
   18 WRITE(NW,200) FSTAR,F                                             00041980
  200 FORMAT(/,18X,A2,'RESIDUAL SEASONALITY PRESENT IN THE LAST 3 YEARS 00041990
     1AT THE 1 PER CENT LEVEL.          F =',F10.2)                    00042000
   19 WRITE(NW,210)                                                     00042010
  210 FORMAT(/,1X,'NOTE: SUDDEN LARGE CHANGES IN THE LEVEL OF THE SEASON00042020
     1ALLY ADJUSTED SERIES WILL INVALIDATE THE RESULTS OF THIS TEST FOR 00042030
     2THE',/,50X,'LAST THREE YEAR PERIOD.')                             00042040
      RETURN                                                            00042050
   20 IF (IND.NE.0) RETURN                                              00042060
      IF (MULADD.NE.0) RETURN                                           00042070
      IF (FMSR.GT.0.01D0) RETURN                                        00042080
      WRITE(NW,220)                                                     00042090
  220 FORMAT(/,28X,'DUE TO THE SMALL RESIDUAL MEAN SQUARE ERROR ALL THE 00042100
     1ANALYSIS OF VARIANCE TESTS FOR THIS SERIES MAY BE INVALID')       00042110
      RETURN                                                            00042120
      END                                                               00042130
      SUBROUTINE MSTEST(ARRAY,JFDA,JLDA,KPROPT,NYR)                     00042140
C --- AN F TEST FOR MOVING SEASONALITY                                  00042150
C                                                                       00042160
      REAL*8 SUMA1,XBAR,COLSS,COLMN,ROWSS,ROWMN,FNYR,TOTSS,ERRSS,       00042170
     1       DEGFRE,FNOYRS,ROWSSN,ERRSSN,C                              00042180
      COMMON /UNITS/ MT,NP,NW,MP,NG,NFORM                               00042190
      COMMON /WORK / ARRAY1(372),SUMA1,XBAR,COLSS,COLMN,ROWSS,ROWMN,    00042200
     1               TOTSS,ERRSS,DEGFRE,ROWSSN,ERRSSN,DUM               00042210
      COMMON /TESTS/ FXX,FRATIO,FPRES,FTDAY,CHI,P1,P2,P3,P4,P5,TEST1,   00042220
     1               TEST2,IQFAIL                                       00042230
```

```
      COMMON /OPT5 / MULADD,IYRT,MA,IWT                                 00042240
      DATA S1,S2,BK /2H* ,2H**,2H  /                                    00042250
      DIMENSION ARRAY(JLDA)                                             00042260
      C = 1.0D0                                                         00042270
      IFMO = (JFDA + NYR - 2) / NYR * NYR + 1                           00042280
      LMO = JLDA / NYR * NYR                                            00042290
      NOYRS = (LMO - IFMO) / NYR + 1                                    00042300
      FNOYRS=NOYRS                                                      00042310
      IF (MULADD.EQ.0) GO TO 2                                          00042320
C --- ADDITIVE MODEL                                                    00042330
    1 DO 3 I=IFMO,LMO                                                   00042340
    3 ARRAY1(I)=ABS(ARRAY(I))                                           00042350
      GO TO 4                                                           00042360
C -- MULTIPLICATIVE MODEL                                               00042370
    2 C = 10000.0D0                                                     00042380
      DO 5 J=IFMO,LMO                                                   00042390
    5 ARRAY1(J)=ABS(ARRAY(J)-1.0)                                       00042400
C --- ANALYSIS OF VARIANCE TEST                                         00042410
    4 SUMA1=0.0D0                                                       00042420
      DO 6 K=IFMO,LMO                                                   00042430
    6 SUMA1=SUMA1+ARRAY1(K)                                             00042440
      FNYR = NYR                                                        00042450
      XBAR = SUMA1 / (FNYR * FNOYRS)                                    00042460
      COLSS=0.0D0                                                       00042470
      DO 7 L=1,NYR                                                      00042480
      COLMN=0.0D0                                                       00042490
      K1=IFMO+L-1                                                       00042500
      DO 8 M=K1,LMO,NYR                                                 00042510
    8 COLMN=COLMN+ARRAY1(M)                                             00042520
      COLMN=COLMN/FNOYRS                                                00042530
    7 COLSS=COLSS+(COLMN-XBAR)*(COLMN-XBAR)                             00042540
      COLSS=COLSS*FNOYRS*C                                              00042550
      ROWSS=0.0D0                                                       00042560
      DO 9 N=IFMO,LMO,NYR                                               00042570
      ROWMN=0.0D0                                                       00042580
      L1=N+NYR-1                                                        00042590
      DO 10 I1=N,L1                                                     00042600
   10 ROWMN=ROWMN+ARRAY1(I1)                                            00042610
      ROWMN=ROWMN/FNYR                                                  00042620
    9 ROWSS=ROWSS+(ROWMN-XBAR)*(ROWMN-XBAR)                             00042630
      ROWSS=ROWSS*FNYR*C                                                00042640
      TOTSS=0.0D0                                                       00042650
      DO 11 J1=IFMO,LMO                                                 00042660
   11 TOTSS=TOTSS+(ARRAY1(J1)-XBAR)*(ARRAY1(J1)-XBAR)                   00042670
      ERRSS=TOTSS*C-COLSS-ROWSS                                         00042680
      DEGFRE=FNOYRS-1.0D0                                               00042690
      ROWSSN=ROWSS/DEGFRE                                               00042700
      ERRSSN=ERRSS/((FNYR-1.0D0)*DEGFRE)                                00042710
      NDGFRE=(NYR-1)*(NOYRS-1)                                          00042720
      FRATIO=ROWSSN/ERRSSN                                              00042730
      N1 = NOYRS - 1                                                    00042740
      P2 = FVALUE(FRATIO,N1,NDGFRE)*100.0                               00042750
      IF(KPROPT.GT.3)RETURN                                             00042760
      IF (P2.GT.0.1) GO TO 13                                           00042770
      FSTAR = S2                                                        00042780
```

```
      GO TO 15
   13 IF (P2.GT.1.0) GO TO 14
      FSTAR = S1
      GO TO 15
   14 FSTAR = BK
   15 WRITE(NW,16)
   16 FORMAT(//,20X,'MOVING SEASONALITY TEST')
      NMIN1=NOYRS-1
      WRITE(NW,12)ROWSS,NMIN1,ROWSSN,FRATIO,FSTAR,ERRSS,NDGFRE,ERRSSN
   12 FORMAT(43X,6HSUM OF,5X,7HDGRS.OF,9X,4HMEAN/
     142X,7HSQUARES,5X,7HFREEDOM,8X,6HSQUARE,7X,7HF-VALUE,/,
     2 21X,13HBETWEEN YEARS,F17.4,I9,F17.6,F12.3,A2,/,
     3 29X,5HERROR,F17.4,I9,F17.6,/)
      IF (FSTAR.NE.BK) GO TO 18
      IF (P2.LT.5.0) GO TO 20
      WRITE(NW,19) FSTAR
   19 FORMAT(28X,A2,59HNO EVIDENCE OF MOVING SEASONALITY AT THE FIVE PER
     1CENT LEVEL )
      RETURN
   18 WRITE(NW,17) FSTAR
   17 FORMAT(28X,A2,52HMOVING SEASONALITY PRESENT AT THE ONE PERCENT LEV
     1EL )
      RETURN
   20 WRITE(NW,21)
   21 FORMAT(30X,'MOVING SEASONALITY PRESENT AT THE FIVE PERCENT LEVEL')
      RETURN
      END
      SUBROUTINE KWTEST(X,IB,IE,NYR,KPROPT)
C --- THIS SUBROUTINE APPLIES THE KRUSKAL-WALLIS TEST TO X. THE K-W TEST
C --- IS THE NONPARAMETRIC EQUIVALENT OF THE F-TEST.
C --- THE ARRAY X IS DESTROYED IN THE CALCULATION PROCEDURE.
C
      DIMENSION X(IE),NS(12),KOLR(12)
      COMMON /UNITS/ MT,MT2,NW,MP,NG,NF
      COMMON /TESTS/ F(4),CHI,P(4),PROB,TEST1,TEST2,IQFAIL
      COMMON /WORK / K(372)
C --- INITIALIZE.
      DO 1 I = 1,NYR
      KOLR(I) = 0
    1 NS(I) = 0
      DO 2 I = IB,IE
    2 K(I) = I
C --- RANK THE ARRAY X.
      DO 3 I = IB,IE
      XVAL = X(I)
      KVAL = K(I)
      DO 3 J = I,IE
      IF (XVAL.LE.X(J)) GO TO 3
      X(I) = X(J)
      K(I) = K(J)
      X(J) = XVAL
      K(J) = KVAL
      XVAL = X(I)
      KVAL = K(I)
    3 CONTINUE
```

```
C --- CALCULATE THE COLUMN SUM OF RANKS.
      DO 4 I = IB,IE
      L = K(I)-(K(I)-1)/NYR*NYR
      NS(L) = NS(L)+1
    4 KOLR(L) = I-IB+1+KOLR(L)
C --- CALCULATE THE K-W STATISTIC.
      CK = 0.0
      DO 5 I = 1,NYR
    5 CK = CK +KOLR(I)*KOLR(I)/FLOAT(NS(I))
      N = IE-IB+1
      CHI = 12.0*CK/(N*(N+1))-3*(N+1)
      NDF = NYR-1
      PROB = CHISQ(CHI,NDF)*100.0
      IF (KPROPT.GT.3) RETURN
      WRITE(NW,10)
   10 FORMAT(//,20X,'NONPARAMETRIC TEST FOR THE PRESENCE OF SEASONALITY
     1ASSUMING STABILITY')
      WRITE(NW,20)
   20 FORMAT(/,35X,'KRUSKAL-WALLIS',4X,'DEGREES OF',4X,'PROBABILITY',/,
     1  37X,'STATISTIC',8X,'FREEDOM',9X,'LEVEL')
      WRITE(NW,30) CHI,NDF,PROB
   30 FORMAT(/,36X,F11.4,9X,I3,8X,F9.3,'%',/)
      IF (PROB.GT.1.0) GO TO 6
      WRITE(NW,40)
   40 FORMAT(30X,'SEASONALITY PRESENT AT THE ONE PERCENT LEVEL')
      RETURN
    6 WRITE(NW,50)
   50 FORMAT(30X,'NO EVIDENCE OF SEASONALITY AT THE ONE PERCENT LEVEL')
      RETURN
      END
      FUNCTION FVALUE(X,M,N)
C --- THIS FUNCTION CALCULATES F-DISTRIBUTION PROBABILITY LEVELS FOR
C ---   PR(R.V. WITH F-DISTRIBUTION WITH M AND N DEGREES OF FREEDOM > X)
      REAL*8 W,Z,P,Y,D,ZK
      IF (X.LE.0.0) GO TO 112
      IF (X.GT.90.0) GO TO 11
      IF (X.GT.40.0.AND.N.GT.150) GO TO 11
      L =(M/2)*2-M+2
      K = (N/2)*2-N+2
      W = X*DFLOAT(M)/DFLOAT(N)
      Z = 1.0D0/(1.0D0+W)
      IF (L.NE.1) GO TO 3
      IF (K.NE.1) GO TO 2
      P = DSQRT(W)
      Y = 0.31830988618379D0
      D = Y*Z/P
      P = 2.0D0*Y*DATAN(P)
      GO TO 5
    2 P = DSQRT(W*Z)
      D = 0.5D0*P*Z/W
      GO TO 5
    3 IF (K.NE.1) GO TO 4
      P = DSQRT(Z)
      D = 0.5D0*Z*P
      P = 1.0D0-P
```

```
      GO TO 5
    4 D = Z*Z
      P = W*Z
    5 Y = 2.0D0*W/Z
      J1 = K+2
      IF (N.LT.J1) GO TO 8
      IF (L.NE.1) GO TO 7
      DO 6 J = J1,N,2
      D = (1.0D0+DFLOAT(L)/DFLOAT(J-2))*D*Z
    6 P = P+D*Y/DFLOAT(J-1)
      GO TO 8
    7 ZK = Z**((N-1)/2)
      D = D*ZK*DFLOAT(N)/DFLOAT(K)
      P = P*ZK+W*Z*(ZK-1.0D0)/(Z-1.0D0)
    8 Y = W*Z
      I1 = L+2
      IF (M.LT.I1) GO TO 10
      Z = 2.0D0/Z
      K = N-2
      DO 9 I = I1,M,2
      ZK = DFLOAT(I+K)
      D = Y*D*ZK/DFLOAT(I-2)
    9 P = P-Z*D/ZK
   10 IF (P.LT.1.0D0) GO TO 12
   11 FVALUE = 0.0
      RETURN
   12 IF (P.GT.0.0D0) GO TO 13
  112 FVALUE = 1.0
      X = 0.0
      RETURN
   13 FVALUE =SNGL(1.0D0-P)
      RETURN
      END
      FUNCTION TVALUE(N,T)
C --- THIS FUNCTION CALCULATES STUDENT'S T-DISTRIBUTION PROBABILITY
C --- LEVELS FOR PR(STUDENT'S T OF N DEGREES > X )
      X = T*T
      TVALUE = FVALUE(X,1,N)
      RETURN
      END
      FUNCTION CHISQ(X,N)
C --- THIS FUNCTION CALCULATES CHI-SQUARED PROBABILITY LEVELS FOR
C ---     PR( CHI-SQ. R.V. WITH N DEGREES OF FREEDOM > X )
      REAL*8 C,C1,C2,C3,Y
      DATA C/0.797884560802864/
      IF (X.GT.0.0) GO TO 1
      CHISQ = 1.
      RETURN
    1 IF (X.LT.90.0) GO TO 2
      CHISQ = 0.0
      RETURN
    2 C1 = 1.0D0
      C2 = C1
      C3 = 0.0D0
      Y = DBLE(X)
```

```
      M = N/2
      I = M*2-N
      IF (I.EQ.0) GO TO 5
      IF (M.EQ.0) GO TO 4
      DO 3 I = 1,M
      C1 = C1*Y/C2
      C3 = C3+C1
    3 C2 = C2+2.0D0
    4 C2 = DSQRT(Y)
      C4 = SNGL(C2)
      CHISQ = 1.0D0-GAUSS(C4)+C*C3*DEXP(-Y/2.0D0)/C2
      RETURN
    5 Y = Y/2.0D0
      IF (M.EQ.1) GO TO 7
      M = M-1
      DO 6 I = 1,M
      C2 = C2*Y/I
    6 C1 = C1+C2
    7 CHISQ = C1*DEXP(-Y)
      RETURN
      END
      FUNCTION GAUSS(X)
C --- THIS FUNCTION CALCULATES NORMAL PROBABILITY LEVELS FOR
C ---        PR( -X < N(0,1) R.V. < X )
      REAL*8 Y,Z,W
      IF (X.NE.0.0) GO TO 1
      GAUSS = 0.0
      RETURN
    1 Y = ABS(X)/2.0D0
      IF (Y.LT.3.0D0) GO TO 2
      GAUSS = 1.0
      RETURN
    2 IF (Y.GE.1.0D0) GO TO 3
      W = Y*Y
      Z = ((((((((0.000124818987D0*W-0.001075204047D0)*W
     1 +0.005198775019D0)*W -0.019198292004D0)*W +0.059054035642D0)*W
     2 -0.151968751364D0)*W +0.319152932694D0)*W -0.531923007300D0)*W
     3 +0.797884560593D0)*Y*2.0D0
      GO TO 4
    3 Y = Y-2.0D0
      Z = (((((((((((((-0.000045255659D0*Y+0.000152529290D0)*Y
     1 -0.000019538132D0)*Y -0.000676904986D0)*Y +0.001390604284D0)*Y
     2 -0.000794620820D0)*Y -0.002034254874D0)*Y +0.006549791214D0)*Y
     3 -0.010557625006D0)*Y +0.011630447319D0)*Y -0.009279453341D0)*Y
     4 +0.005353579108D0)*Y -0.002141268741D0)*Y +0.000535310849D0)*Y
     5 +0.999936657524D0
    4 GAUSS = Z
      RETURN
      END
      SUBROUTINE LOGAR(X,I,J)
      DIMENSION X(J)
      DO 1 K = I,J
      TMP = X(K)
    1 X(K) = ALOG(TMP)
      RETURN
```

```
      ENTRY ANTILG(X,I,J)
      DO 2 K = I,J
      TMP = X(K)
    2 X(K) = EXP(TMP)
      RETURN
      END
      FUNCTION VARLOG(X,I,J,IOPT)
C --- THIS FUNCTION COMPUTES THE LOG VARIANCE OF X. IF IOPT IS EQUAL
C --- TO 1 THE LOG MEAN IS ASSUMED TO BE 1.
      DIMENSION X(1)
      TMP = 0.0
      IF (IOPT.EQ.1) GO TO 2
      DO 1 K = I,J
    1 TMP = TMP + ALOG(X(K))
      TMP = TMP/FLOAT(J-I+1)
    2 VARLOG = 0.0
      DO 3 K = I,J
      TMP2 = ALOG(X(K)) - TMP
    3 VARLOG = VARLOG + TMP2 * TMP2
      RETURN
      END
      FUNCTION VARIAN(X,I,J,IOPT)
C --- THIS FUNCTION COMPUTES THE VARIANCE OF X. IF IOPT = 0 COMPUTE THE
C --- MEAN , IF IOPT = 1 THE MEAN IS ASSUMED TO BE ZERO, AND IF IOPT = 2
C --- THE MEAN IS ASSUMED TO BE ONE.
      DIMENSION X(1)
      AVE = 1.0
      IF (IOPT.EQ.2) GO TO 2
      AVE = 0.0
      IF (IOPT.EQ.1) GO TO 2
      DO 1 K = I,J
    1 AVE = AVE+X(K)
      AVE = AVE/(J-I+1)
    2 VARIAN = 0.0
      DO 3 K = I,J
    3 VARIAN = VARIAN+(X(K)-AVE)*(X(K)-AVE)
      RETURN
      END
      DOUBLE PRECISION FUNCTION TOTALS(X,I,J,K,IOPT)
C --- THIS FUNCTION CALCULATES TOTALS AND AVERAGES. A TOTAL IS RETURNED
C --- IF IOPT IS ZERO AND AN AVERAGE IF IOPT IS ONE.
      DIMENSION X(1)
      TOTALS = 0.0
      DO 1 L = I,J,K
    1 TOTALS = TOTALS+X(L)
      IF (IOPT.EQ.0) GO TO 2
      FN = (J-I)/K+1
      TOTALS = TOTALS/FN
    2 RETURN
      END
      FUNCTION SDEV(X,I,J,K,IOPT)
C --- THIS FUNCTION CALCULATES THE STANDARD DEVIATION OF X. IF IOPT = 0
C --- THE MEAN OF X IS COMPUTED, IF IOPT = 1 THE MEAN IS ASSUMED TO BE
C --- ZERO, AND IF IOPT = 2 THE MEAN IS ASSUMED TO BE ONE.
      REAL*8 TOTALS
```

```
      DIMENSION X(1)
      FN = (J-I)/K+1
      AVE = 0.0
      IF (IOPT-1) 2,4,1
    1 AVE = 1.0
      GO TO 4
    2 AVE = TOTALS(X,I,J,K,1)
    4 SDEV = 0.0
      DO 5 L = I,J,K
    5 SDEV = SDEV+(X(L)-AVE)*(X(L)-AVE)
      SDEV = SQRT(SDEV/FN)
      RETURN
      END
      FUNCTION VARS(X,I,J,IOPT)
      COMMON /OPT5 / MULADD,IYRT,MA,IWT
      DIMENSION X(1)
      IF (MULADD.EQ.1) GO TO 1
      VARS = VARLOG(X,I,J,IOPT)
      RETURN
    1 VARS = VARIAN(X,I,J,IOPT)
      RETURN
      END
      SUBROUTINE FGEN(ITYPE)
C --- THIS SUBROUTINE GENERATES THE F2 AND F3 TABLES.
      REAL*8 SERNO,TITLE,PCDIF,RAD,MQ,KSER
      REAL*4 IBAR,IBAR2,ISQ,ISD,IMBAR,AORB(2)
      COMMON /UNITS/ MT,MT2,MT1,MP,NG,NFORM
      COMMON /ROTE / TITLE(10),SERNO,KPAGE,IRUN,NEWPG
      COMMON /PRIOR/ KSER,KFORM(20),KFMT
      COMMON /WORK2/ DUM(70),PBAR(12),PSQ(12),AUTOC(14)
      COMMON /TRADE/ DWT(7),D(7),SIGM,LOPT,KDWOPT,LCYR,LAYR,LFDC,LFDR
      COMMON /INPT2/ TDBAR(12),TDSQ(12),SBAR(12),SBAR2(12),SSQ(12),
     1               SSD(12),IBAR(12),IBAR2(12),ISQ(12),ISD(12),
     2               IMBAR(12),CBAR(12),CBAR2(12),CSQ(12),CSD(12),
     3               OBAR(12),OBAR2(12),OSQ(12),OSD(12),OSQ2(12),
     4               SMBAR(12),SMBAR2(12),SMSD(12),OMBAR(12),CIMBAR(12),
     5               CIBAR(12),CIBAR2(12),CISD(12),SMIC(12),V(6),ADR(4),
     6               T1,T2
      COMMON /TESTS/ F(5),P(5),TEST1,TEST2,IQFAIL
      COMMON /OPT7 / RATI(36),RATIS,LTERM,MTYPE,LTER(12)
      COMMON /OPT8 / RATIC,KICOPT,NTERM
      COMMON /OPT9 / LFDA,LYR,LSTMO,LSTYR,LLDA,KFULSM,KPROPT,KCHOPT,
     1               KEXOPT,NY,IFORC,LLDAF,LLAF
      COMMON /MQ3  / PCDIF(3),RAD(2),MQ,QM(3),OUT,MQCD
      COMMON /MQ2  / KPART,KSWV,KSECT,KPRNT,MCD
      DATA AORB/1HB,1HA/
      NW = NFORM
        KHCFM=1
        IF (KFMT .GT. 0) KHCFM=2
      IF (ITYPE.EQ.1) GO TO 1
      NW = MT1
      KPAGE = KPAGE+1
    1 WRITE(NW,100) NEWPG,TITLE,KPAGE,SERNO
      WRITE(NW,101)
      WRITE(NW,102) PCDIF,AORB(KHCFM),MQ,MQCD
```

```
      DO 2 I = 1,NY                                                     00046090
    2 WRITE(NW,103) I,OBAR(I),CIBAR(I),IBAR(I),CBAR(I),SBAR(I),PBAR(I), 00046100
     1                TDBAR(I),SMBAR(I),OMBAR(I),CIMBAR(I),IMBAR(I)     00046110
      WRITE(NW,104) PCDIF,MQ                                            00046120
      DO 3 I = 1,NY                                                     00046130
    3 WRITE(NW,105) I,ISQ(I),CSQ(I),SSQ(I),PSQ(I),TDSQ(I),OSQ2(I)      00046140
      WRITE(NW,106) PCDIF,AORB(KHCFM),MQCD,MQ                           00046150
      DO 4 I = 1,NY                                                     00046160
    4 WRITE(NW,107) I,OBAR2(I),OSD(I),IBAR2(I),ISD(I),CBAR2(I),CSD(I),  00046170
     1                SBAR2(I),SSD(I),CIBAR2(I),CISD(I),SMBAR2(I),SMSD(I) 00046180
      WRITE(NW,108) MQCD,ADR                                            00046190
      IF (ITYPE.EQ.0.AND.NY.EQ.12) KPAGE = KPAGE+1                      00046200
      WRITE(NW,109) MQ,(I,I = 1,NY)                                     00046210
      WRITE(NW,110) (SMIC(I),I = 1,NY)                                  00046220
      WRITE(NW,111) MQ,MCD                                              00046230
      WRITE(NW,112) V                                                   00046240
      N = NY+2                                                          00046250
      WRITE(NW,113) N,(I,I = 1,N)                                       00046260
      WRITE(NW,110) (AUTOC(I),I = 1,N)                                  00046270
      WRITE(NW,114) RATIC,RATIS                                         00046280
      WRITE(NW,115) F(3),P(3)                                           00046290
      IF (KDWOPT.GE.1) WRITE(NW,116) F(4),P(4)                          00046300
      WRITE(NW,117) F(1),P(1),F(5),P(5),F(2),P(2)                       00046310
      IF (ITYPE.EQ.1) GO TO 5                                           00046320
      KPAGE = KPAGE+1                                                   00046330
    5 WRITE(NW,100) NEWPG,TITLE,KPAGE,SERNO                             00046340
      WRITE(NW,118)                                                     00046350
      CALL F3GEN(NW,NY,MQ)                                              00046360
      RETURN                                                            00046370
  100 FORMAT(A1,10X,9A8,A6,12X,4HPAGE,I3,9H, SERIES ,A8)                00046380
  101 FORMAT(//,' F 2. SUMMARY MEASURES')                               00046390
  102 FORMAT( 6X,14HF 2.A: AVERAGE,2A6,A5,                              00046400
     1        47H WITHOUT REGARD TO SIGN OVER THE INDICATED SPAN,/,     00046410
     2        14X,4HSPAN,/,15X,                                         00046420
     3        2HIN,6X,A1,1H1,5X,'D11',5X,'D13',5X,'D12',5X,'D10',6X,'A2',00046430
     4        5X,'C18',6X,'F1',13X,'E1',6X,'E2',6X,'E3',/,              00046440
     5        11X,A7,1HS,4X,1HO,6X,'CI',7X,'I',7X,'C',7X,'S',7X,'P',6X, 00046450
     6        'TD',6X,A3,11X,'MOD.O   MOD.CI   MOD.I')                  00046460
  103 FORMAT(15X,I2,8F8.2,9X,3F8.2)                                     00046470
  104 FORMAT(/,6X,52HF 2.B: RELATIVE CONTRIBUTIONS TO THE VARIANCE OF TH00046480
     1E,2A6,A5,41H IN THE COMPONENTS OF THE ORIGINAL SERIES,/,           00046490
     2        14X,4HSPAN,/,                                             00046500
     3        15X,2HIN,4X,3HD13,5X,3HD12,5X,3HD10,6X,2HA2,5X,3HC18,12X, 00046510
     4        5HRATIO,/,11X,A7,1HS,                                     00046520
     5        3X,1HI,7X,1HC,7X,1HS,7X,1HP,6X,2HTD,5X,14HTOTAL   (X100)) 00046530
  105 FORMAT(15X,I2,2P5F8.2,8H  100.00,2PF8.2)                          00046540
  106 FORMAT(/,6X,14HF 2.C: AVERAGE,2A6,A5,                             00046550
     1        63H WITH REGARD TO SIGN AND STANDARD DEVIATION OVER INDICAT00046560
     2ED SPAN,/,14X,4HSPAN,8X,A1,1H1,                                   00046570
     3        15X,3HD13,14X,3HD12,14X,3HD10,14X,3HD11,15X,2HF1,/,       00046580
     4        15X,2HIN,10X,1HO,16X,1HI,16X,1HC,16X,1HS,16X,2HCI,14X,    00046590
     5        A3,/,11X,A7,1HS,                                          00046600
     6        6(3X,4HAVGE,4X,4HS.D.,2X))                                00046610
  107 FORMAT(15X,I2,6(F9.2,F8.2))                                       00046620
  108 FORMAT(/,6X,30HF 2.D: AVERAGE DURATION OF RUN,8X,2HCI,6X,1HI,7X,  00046630
```

```
     1        1HC,6X,A3,/,39X,4F8.2)
  109 FORMAT(//,6X,21HF 2.E: I/C RATIO FOR ,A7,6HS SPAN,/,18X,12I8)
  110 FORMAT(19X,14F8.2)
  111 FORMAT(/,7X,A7,25HS FOR CYCLICAL DOMINANCE=,I8)
  112 FORMAT(//,6X,111HF 2.F: RELATIVE CONTRIBUTION OF THE COMPONENTS T
     1 THE STATIONARY PORTION OF THE VARIANCE IN THE ORIGINAL SERIES,/,
     2        24X,1HI,7X,1HC,7X,1HS,7X,1HP,6X,2HTD,4X,5HTOTAL,/,
     3       19X,6F8.2,/)
  113 FORMAT(/,6X,59HF 2.G: THE AUTOCORRELATION OF THE IRREGULARS FOR SP
     1ANS 1 TO,I3,/,18X,14I8)
  114 FORMAT(/,6X,42HF 2.H: THE FINAL I/C RATIO FROM TABLE D12:,F12.2,/,
     1        12X,36H THE FINAL I/S RATIO FROM TABLE D10:F12.2)
  115 FORMAT(/,6X,6HF 2.I:,75X,21HSTATISTIC PROBABILITY,/,100X,5HLEVEL,
     1        /,13X,45HF-TEST FOR STABLE SEASONALITY FROM TABLE B 1.,
     2        26X,1H:,F11.3,F8.2,1H%)
  116 FORMAT(13X,50HF-TEST FOR THE TRADING DAY REGESSION IN TABLE C15.,
     1        21X,1H:,F11.3,F8.2,1H%)
  117 FORMAT(13X,45HF-TEST FOR STABLE SEASONALITY FROM TABLE D 8.,26X,
     1        1H:,F11.3,F8.2,1H%,/,
     2        13X,72HKRUSKAL-WALLIS CHI SQUARED TEST FOR STABLE SEASONALI
     3TY FROM TABLE D 8. :,F11.3,F8.2,1H%,/,
     4        13X,45HF-TEST FOR MOVING SEASONALITY FROM TABLE D 8.,26X,
     5        1H:,F11.3,F8.2,1H%)
  118 FORMAT(//,' F 3. MONITORING AND QUALITY ASSESSMENT STATISTICS')
      END
      SUBROUTINE F3CAL(IFAIL)
C --- THIS SUBROUTINE CALCULATES THE QUALITY CONTROL STATISTICS IN TABLE
C --- F3.
C
      REAL*4   IBAR,ISQ,IBAR2,NM
      COMMON /UNITS/ MT,MT2,MT1,MP,NG,NFORM
      COMMON /WORK / TEMP(372)
      COMMON /INPT2/ TDBAR(12),TDSQ(12),SBAR(12),SBAR2(12),SSQ(12),
     1               SSD(12),IBAR(12),IBAR2(12),ISQ(12),TMP2(120),
     2               OSQ2(12),TMP3(96),SMIC(12),V(6),ADR(4),TMP4,TMP5
      COMMON /WORK2/ NM(11),NN,QUAL,KFAIL,NYRS,TMP1(67),PSQ(12)
      COMMON /TESTS/ F(5),P(5),TEST1,TEST2,IQFAIL
      COMMON /OPT5 / MULADD,IYRT,MA,IWT
      COMMON /OPT7 / RATI(36),RATIS,LTERM,MTYPE,LTER(12)
      COMMON /OPT8 / RATIC,KTCOPT,NTERM
      COMMON /MQ1  / STS(372),STSI(372)
      COMMON /OPT9 / LFDA,LYR,LSTMO,LSTYR,LLDA,KFULSM,KPROPT,KCHOPT,
     1               KEXOPT,NY,IFORC,LLDAF,LLAF
      COMMON /MQ2  / KPART,KSWV,KSECT,KPRNT,MCD
      DIMENSION WT(11)
      DATA WT/10.,11.,10., 8.,11.,10.,18., 7., 7., 4., 4./
      IFAIL = 0
      KNY = NY/4
      NM(1) = (ISQ(KNY)/(1.0-PSQ(KNY)))/0.10
      KNY = 12/NY
      NM(2) = (V(1)/(100.0-V(4)))/0.10
      NM(3) = (RATIC*KNY-1.0)/2.0
      FN = LLDA-LFDA+1
      NM(4) = ABS(3.0*(FN-1.0)/ADR(2)-2.0*FN+1.0)/(SQRT(1.6*FN-2.9)*2.5
     17)
```

```
      IF (MCD.EQ.1) GO TO 2                                             00047190
      RMCD = MCD+(SMIC(MCD)-1.0)/(SMIC(MCD-1)-SMIC(MCD))                00047200
      IF (MCD.EQ.NY.AND.SMIC(NY-1).LE.SMIC(NY)) RMCD = KNY*15.5         00047210
      GO TO 3                                                           00047220
    2 RMCD = 1+(SMIC(1)-1.0)/(SMIC(1)-SMIC(2))                          00047230
      IF (RMCD.LT.0.5) RMCD = 0.5                                       00047240
      IF (RMCD.GT.1.0) RMCD = 1.0                                       00047250
    3 NM(5) = (RMCD*KNY-0.5)/5.0                                        00047260
      NM(6) = ABS(RATIS-4.0)/2.5                                        00047270
      NM(7) = SQRT((TEST1+TEST2)/2.0)                                   00047280
      NYRS = (LLDA-LFDA+1)/NY                                           00047290
      NN = 7                                                            00047300
      IF (MTYPE.EQ.5.OR.NYRS.LT.6) GO TO 14                             00047310
      NN = 11                                                           00047320
      N = 2-MULADD                                                      00047330
      SD = SDEV(STS,LFDA,LLDA,1,N)                                      00047340
      AVE = 1-MULADD                                                    00047350
      DO 8 I = LFDA,LLDA                                                00047360
    8 TEMP(I) = (STS(I)-AVE)/SD                                         00047370
      CT1 = 0.0                                                         00047380
      CT2 = 0.0                                                         00047390
      COUNT = 0.0                                                       00047400
      AVE1 = 0.0                                                        00047410
      AVE2 = 0.0                                                        00047420
      AVE3 = 0.0                                                        00047430
      AVE4 = 0.0                                                        00047440
      KLDA = LFDA+NY-1                                                  00047450
      DO 11 I = LFDA,KLDA                                               00047460
      CT = 0.0                                                          00047470
      I1 = I+NY                                                         00047480
      DO 9 J = I1,LLDA,NY                                               00047490
      CT = CT+1.0                                                       00047500
      COUNT = COUNT+1.0                                                 00047510
      DIFF = ABS(TEMP(J)-TEMP(J-NY))                                    00047520
      K = J                                                             00047530
    9 AVE1 = AVE1+DIFF                                                  00047540
      AVE2 = AVE2+ABS(TEMP(K)-TEMP(I))/CT                               00047550
      K = K-2*NY                                                        00047560
      J = K-3*NY                                                        00047570
      IF (J.LT.LFDA) GO TO 11                                           00047580
      AVE3 = AVE3+ABS(TEMP(K)-TEMP(J))/3.0                              00047590
      CT1 = CT1+1.0                                                     00047600
      J = J+NY                                                          00047610
      DO 10 L =J,K,NY                                                   00047620
      I1 = L-NY                                                         00047630
      IF (I1.LT.LFDA) GO TO 10                                          00047640
      CT2 = CT2+1.0                                                     00047650
      DIFF = ABS(TEMP(L)-TEMP(I1))                                      00047660
      AVE4 = AVE4+DIFF                                                  00047670
   10 CONTINUE                                                          00047680
   11 CONTINUE                                                          00047690
      AVE1 = AVE1/COUNT                                                 00047700
      AVE2 = AVE2/NY                                                    00047710
      IF (CT1.EQ.0.0) GO TO 12                                          00047720
      AVE3 = AVE3/CT1                                                   00047730
```

```
      AVE4 = AVE4/CT2
   12 NM(8) = 10.0*AVE1
      NM(9) = 10.0*AVE2
      NM(10) = 10.0*AVE4
      NM(11) = 10.0*AVE3
   14 QUAL = 0.0
      TWT = 0.0
      DO 20 I = 1,11
      IF (I.LE.NN) GO TO 18
      IF (I-10) 15,16,17
   15 QUAL = QUAL+NM(7)*WT(I)
      GO TO 19
   16 QUAL = QUAL+NM(1)*WT(I)
      GO TO 19
   17 QUAL = QUAL+NM(2)*WT(I)
      GO TO 19
   18 IF (NM(I).LT.0.0) NM(I) = 0.0
      IF (NM(I).GT.3.0) NM(I) = 3.0
      IF (NM(I).GE.1.0) IFAIL = IFAIL+1
      IF (MTYPE.NE.3.AND.I.EQ.6) GO TO 20
      QUAL = QUAL+NM(I)*WT(I)
   19 TWT = TWT+WT(I)
   20 CONTINUE
      QUAL = QUAL/TWT
      KFAIL = IFAIL
      RETURN
      END
      SUBROUTINE CHART(A,B,C,IB,IE,IBA,IEA,NOSER,LOG,LYR,NY,IMON,IOPT)
C --- THIS SUBROUTINE PRINTS THE X-11 CHARTS.
      REAL*8 M(17)
      INTEGER*2 L(103),J(8),IJ(4),JI(3),ICARR
      COMMON /UNITS/ MT,MT2,MT1,MP,NG,NFORM
      COMMON /OPT5 / MULADD,MODEL,MA,IWT
      COMMON /WORK4/ VAL(3),P(6),L,JI,IJ,TMP1(47)
      COMMON /SCALE/ BIG,SMALL
      DIMENSION A(1),B(1),C(1),M2(17)
      DATA J/1HX,1HO,1H+,1H*,1H.,1H ,1HE,1HO/
      DATA M/8HJANUARY ,8HFEBRUARY,8HMARCH   ,8HAPRIL   ,8HMAY     ,
     1       8HJUNE    ,8HJULY    ,8HAUGUST  ,8HSEPTEMBE,8HOCTOBER ,
     2       8HNOVEMBER,8HDECEMBER,8H1ST QUAR,8H2ND QUAR,8H3RD QUAR,
     3       8H4TH QUAR,8H 19     /
      DATA M2/3H   ,3H   ,3H   ,3H   ,3H   ,3H   ,3H   ,3H   ,3HR  ,
     1        3H   ,3H   ,3H   ,3HTER,3HTER,3HTER,3HTER,3H   /
      ICARR = J(6)
C     IF (IOPT.EQ.2) ICARR = J(8)
      L(1) = J(5)
      L(103) = J(5)
      DO 1  I = 2,102
    1 L(I) = J(6)
      DO 101 I = 22,82,20
  101 L(I) = J(5)
      DO 2 I = IBA,IEA
      BIG = AMAX1(BIG,A(I))
    2 SMALL = AMIN1(SMALL,A(I))
      IF (NOSER.EQ.1) GO TO 5
```

```
      DO 3 I = IB,IE
      BIG = AMAX1(BIG,B(I))
    3 SMALL = AMIN1(SMALL,B(I))
      IF (NOSER.EQ.2) GO TO 5
      DO 4 I = IB,IE
      BIG = AMAX1(BIG,C(I))
    4 SMALL = AMIN1(SMALL,C(I))
    5 P(1) = SMALL
      IF (LOG.EQ.0) GO TO 7
      WRITE(MT1,1000)
      DELTA = 100.0/(BIG-SMALL)
      DO 6 I = 2,5
    6 P(I) = (I-1)*20./DELTA+SMALL
      GO TO 9
    7 SMALL = ALOG(SMALL)
      WRITE(MT1,1001)
      DELTA = 100.0/(ALOG(BIG)-SMALL)
      DO 8 I = 2,5
    8 P(I) = EXP((I-1)*20.0/DELTA+SMALL)
    9 P(6) = BIG
      IF (IOPT.NE.1.OR.MULADD.EQ.1) GO TO 209
      DO 109 I = 1,6
  109 P(I) = 100.0*P(I)
  209 WRITE(MT1,1002) P
      WRITE(MT1,1003) (J(5),I = 1,103)
      IF (IMON.EQ.0) GO TO 10
      K = IMON+4/NY*12
      WRITE(MT1,1004) M(K),M2(K)
   10 IF (IB.EQ.IBA) GO TO 410
      IE1 = IB-1
      DO 310 I = IBA,IE1
      IF (LOG.EQ.0) GO TO 110
      TMP = A(I)
      GO TO 210
  110 TMP = ALOG(A(I))
  210 KI = INT(DELTA*(TMP-SMALL)+0.5)+2
      JI(1) = L(KI)
      L(KI) = J(7)
      LL = I-(I-1)/NY*NY+4/NY*12
      JYR = LYR+(I-1)/NY
      WRITE(MT1,1005) ICARR,M(LL),JYR,L
  310 L(KI) = JI(1)
  410 CONTINUE
      DO 22 I = IB,IE
      DO 14 K = 1,NOSER
      GO TO (11,12,13),K
   11 VAL(K) = A(I)
      GO TO 14
   12 VAL(K) = B(I)
      GO TO 14
   13 VAL(K) = C(I)
   14 CONTINUE
      IF (LOG.NE.0) GO TO 16
      DO 15 K = 1,NOSER
   15 VAL(K) = ALOG(VAL(K))
```

```
   16 IJ(1) = 0
      DO 19 K = 1,NOSER
      KI = INT(DELTA*(VAL(K)-SMALL)+0.5)+2
      JI(K) = L(KI)
      DO 17 N = 1,K
      IF (KI.EQ.IJ( N)) GO TO 18
   17 CONTINUE
      L(KI) = J(K)
      GO TO 19
   18 L(KI) = J(4)
      JI(K) = JI(N)
   19 IJ(K+1) = KI
      IF (IMON.NE.0) GO TO 20
      LL = I-(I-1)/NY*NY+4/NY*12
      JYR = LYR+(I-1)/NY
      GO TO 21
   20 LL = 17
      JYR = LYR+I-1
   21 WRITE(MT1,1005) ICARR,M(LL),JYR,L
      DO 22 N = 1,NOSER
      KI = IJ(NOSER+2-N)
   22 L(KI) = JI(NOSER+1-N)
      IF (IEA.LE.IE) GO TO 28
      IB1 = IE+1
      DO 27 I = IB1,IEA
      IF (LOG.EQ.0) GO TO 23
      TMP = A(I)
      GO TO 24
   23 TMP = ALOG(A(I))
   24 KI = INT(DELTA*(TMP-SMALL)+0.5)+2
      JI(1) = L(KI)
      L(KI) = J(7)
      IF (IOPT.EQ.3) L(KI) = J(1)
      IF (IMON.NE.0) GO TO 25
      LL = I-(I-1)/NY*NY+4/NY*12
      JYR = LYR+(I-1)/NY
      GO TO 26
   25 LL = 17
      JYR = LYR+I-1
   26 WRITE(MT1,1005) ICARR,M(LL),JYR,L
   27 L(KI) = JI(1)
   28 WRITE(MT1,1003) (J(5),I = 1,103)
      WRITE(MT1,1002) P
      RETURN
 1000 FORMAT(4X,16HSCALE-ARITHMETIC)
 1001 FORMAT(4X,14HSCALE-SEMI-LOG)
 1002 FORMAT(2X,F8.1,5(12X,F8.1))
 1003 FORMAT(7X,103A1)
 1004 FORMAT(1X,A8,A3)
 1005 FORMAT(A1,A3,I2,1X,103A1)
      END
      SUBROUTINE ARIMA(IFB,ITB)
      REAL*8 SERNO,TITLE,TRAN(3)
      COMMON /ROTE / TITLE(10),SERNO,KPAGE,IRUN,NEWPG
      COMMON /INPT1/ SERIES(372),ORIG(372)
```

```
      COMMON /UNITS/ MT,MT2,MT1,MP,NG,NFORM                             00049390
      COMMON /OPT5 / MULADD,IYRT,MA,IWT                                 00049400
      COMMON /OPT9/ LFDA,LYR,LSTMO,LSTYR,LLDA,KFULSM,KPROPT,KCHOPT,     00049410
     1              KEXOPT,NY,IFORC,LLDAF,LLAF,LFD1,LLD1,NSOPT,IDEC,IAG 00049420
     2              ,JAREX                                              00049430
      COMMON /MQ1  / ZZ(372),A(372)                                     00049440
      COMMON /MQ2  / KPART,KSWV,KSECT,KPRNT,MCD                         00049450
      COMMON /MQ4  / B(372),C(372)                                      00049460
      COMMON /MQ5 / TMP(372),ZABC(372)                                  00049470
      COMMON /MQ7  / Z(372)                                             00049480
      COMMON /MQ8  / STPTD(372),STTD(372)                               00049490
      COMMON /ARIM1/ PV(10),PW,ADD,IPDQ(7),IORDER(10),NLOG,MIT,NDP,NSEA 00049500
      COMMON /ARIM2/ NOB,NP,NRD,NSD,MBO,MAX,MAX1,IOPA(10),INC(6)        00049510
      COMMON /ARIM3/ SDEV(4),PROB(3),SSQ,PA(10),FCT(48),IFT,FCTT(3),    00049520
     1               RSQAR,IVALUE,II(3)                                 00049530
      COMMON /MQ10 /RBMSF(372)                                          00049540
      DIMENSION NT(4),LOGS(3),MODELS(21),PT(10)                         00049550
      DATA MODELS/0,1,1,0,1,1,0,   0,2,2,0,1,1,0,   2,1,2,0,1,1,0/       00049560
      DATA DASH,TRAN/4H----,5H NONE,5H LOG ,5HPOWER/                    00049570
C --- SET INDICATOR FOR REFITTING THE BEST MODEL                        00049580
      IREF=0                                                            00049590
      LOGS(1)=1                                                         00049600
      IF (MULADD.NE.1) LOGS(1)=2                                        00049610
      LOGS(2)=1                                                         00049620
      IF (MULADD.NE.1) LOGS(2)=2                                        00049630
      LOGS(3)=1                                                         00049640
      IF (IFB.EQ.2) NLOG=NLOG-1                                         00049650
      IF (PW.LE.0.0) PW = 1.0                                           00049660
C --- IF THE 'ARIMA REPLACEMENT OF EXTREMES' OPTION IS CHOSEN, SET      00049670
C --- THE RESIDUALS TO ZERO                                             00049680
      IF(JAREX.EQ.0.OR.IFB.EQ.2)GO TO 34                                00049690
      DO 27 K=1,372                                                     00049700
      RBMSF(K)=0.0                                                      00049710
   27 CONTINUE                                                          00049720
      SSQ3=0.0                                                          00049730
   34 IF(IFB.EQ.1) GO TO 200                                            00049740
      DO 201 I=1,10                                                     00049750
      IORDER(I)=0                                                       00049760
  201 CONTINUE                                                          00049770
  200 IOR = 0                                                           00049780
      IRD = 16217                                                       00049790
      NSEA = NY                                                         00049800
C --- IF THERE ARE MORE THAN 15 YEARS IN THE SERIES USE ONLY THE LAST 150049810
C --- FOR THE ARIMA FORECAST.                                           00049820
      ITEST = LLDA-LFDA+1-NSEA*15                                       00049830
      KFDA = LFDA                                                       00049840
      IF (ITEST.GT.0) KFDA = LLDA-NSEA*15+1                             00049850
C --- CHECK FOR MAXIMUM LENGTH OF SERIES.                               00049860
      ITEST = LLDA-29*NSEA                                              00049870
      IF (ITEST.GT.0) GO TO 1004                                        00049880
      NDP = LLDA-KFDA+1                                                 00049890
C --- SERIES MUST BE AT LEAST 5 YEARS.                                  00049900
      ITEST = NDP-5*NSEA                                                00049910
      IF (ITEST.LT.0) GO TO 1005                                        00049920
C --- CHECK FOR ZERO OR NEGATIVE VALUES IN THE SERIES. AD1 IS THE       00049930
```

```
C --- CONSTANT ADDED IF THE LOG TRANSFORM IS USED
      AD1 = 1.0
      AVE = 0.0
      DO 20 I = KFDA,LLDA
   20 AD1 = AMIN1(AD1,ZABC(I))
      AD = 0.0
      IVALUE = 0
      IF (AD1.GT.0.0) GO TO 40
      IVALUE = 1
      AD = -1.5*AD1+1.0
      KKLDA = LLDA-3*NY+1
      DO 30 I = KKLDA,LLDA
   30 AVE = AVE+ABS(ZABC(I))
      AVE = AVE/(3*NY)
      IF (AVE.EQ.0.0) AVE = 1.0
   40 CONTINUE
      IFT = 0
      N = 0
      DO 1 I = 1,4
    1 NT(I) = NDP-NSEA*(I-1)
      IF (IFORC.EQ.2) GO TO 4
      MIT = 30
      ADD = 0.0
      DO 3 I = 1,10
    3 PV(I) = 0.1
      PW = 1.0
      GO TO 7
C --- CHECK NUMBER OF ITERATIONS.
    4 IF (MIT.LE.0) MIT = 30
      IF (MIT.GT.50) MIT = 50
      IF (IORDER(1).NE.0) IOR = 1
C --- CHECK MODEL SPECIFICATION.
      DO 5 I =1,7
      IF (IPDQ(I).GT.4) GO TO 1001
    5 CONTINUE
      NLOG = NLOG+1
      IF (NLOG.GT.3) NLOG = 1
      IF (NLOG.EQ.1) GO TO 7
      DO 6 I = KFDA,LLDA
      IF (ZABC(I)+ADD) 1002,1002,6
    6 CONTINUE
    7 IF (NLOG.EQ.1) ADD = 0.0
      KPAGE = KPAGE+1
      WRITE(MT1,100) NEWPG,TITLE,KPAGE,SERNO
  100 FORMAT(A1,10X,9A8,A6,12X,4HPAGE,I3,9H, SERIES ,A8,//)
      WRITE(MT1,101)
  101 FORMAT(31X,'AUTOREGRESSIVE INTEGRATED MOVING AVERAGE (ARIMA) EXTRA
     1POLATION PROGRAM',/)
      IF(IFB.EQ.1) WRITE(MT1,45)
   45 FORMAT(' A5.  ARIMA EXTRAPOLATION MODEL (FORECAST)',//)
      IF(IFB.EQ.2) WRITE(MT1,46)
   46 FORMAT(' A6.  ARIMA EXTRAPOLATION MODEL (BACKCAST)',//)
      WRITE(MT1,102)
  102 FORMAT(33X,'THIS PROGRAM WAS DEVELOPED FOLLOWING THE PROCEDURES OU
     1TLINED IN')
```

```
      WRITE(MT1,103)
  103 FORMAT(36X,22H'TIME SERIES ANALYSIS',' BY G. E. P. BOX AND G. M. J
     1ENKINS.')
      IF (IFB.EQ.1) WRITE(MT1,106) (DASH,I=1,7)
  106 FORMAT(33X,'AVERAGE PERCENTAGE STANDARD',/,
     1       38X,'ERROR IN FORECASTS',/,
     2       33X,7A4)
      IF (IFB.EQ.2) WRITE(MT1,1066) (DASH,I=1,7)
 1066 FORMAT(33X,'AVERAGE PERCENTAGE STANDARD',/,
     1       38X,'ERROR IN BACKCASTS',/,
     2       33X,7A4)
      WRITE(MT1,107)
  107 FORMAT(5X,'MODEL',6X,'TRAN.   ADDITIVE LAST 3  LAST  LAST-1 LAST-2
     1 CHI-SQ. R-SQUARED',14X,'ESTIMATED PARAMETERS')
      WRITE(MT1,108) (DASH,I = 1,31)
  108 FORMAT(24X,'CONSTANT  YEARS  YEAR   YEAR   YEAR   PROB.    VALUE '
     1  /,1X,30A4,A2,/)
    8 L = N*7
      JTEST = 1
      IF (IFORC.NE.1) GO TO 10
      N = N+1
      DO 9 I = 1,7
      IORDER(I) = 0
    9 IPDQ(I) = MODELS(I+L)
      DO 2 I = 8,10
    2 IORDER(I) = 0
      NLOG = LOGS(N)
      IF (NLOG.NE.1) ADD = AD
   10 CONTINUE
      DO 11 I = KFDA,LLDA
   11 Z(I-KFDA+1) = ZABC(I)
      NP=IPDQ(1)+IPDQ(3)+IPDQ(4)+IPDQ(6)
      IF(IPDQ(7).NE.0)NP=NP+1
      IF (NP.GT.10) GO TO 1003
      NRD=IPDQ(2)
      NSD=IPDQ(5)
C
C     CONVERSION ROUTINE.
C
      CALL CONV(PA)
C
C     CHECK THE SPECS CARD
C
      MAX=IPDQ(2)+(IPDQ(5)*NSEA)+IPDQ(1)+(IPDQ(4)*NSEA)
      NOB = NDP-MAX
C
C     DETERMINE THE MAXIMUM POWER OF  B  ON THE RIGHT HAND SIDE OF
C     THE MODEL  (THE REGULAR AND SEASONAL MOVING AVERAGE TERMS).
C
      KK = INC(1)+INC(2)+INC(3)+INC(4)+INC(5)
      MAX2 = 0
      IF(INC(5) .NE. 0) MAX2 = IOPA(KK)
      IF (INC(6) .NE. 0)MAX2=MAX2+IOPA(NP)
C
C     FIND THE MAXIMUM BACK ORDER OF THE MODEL, AND CHECK THAT IT DOES
```

```
C     NOT EXCEED THE NUMBER OF DATA POINTS AVAILABLE
C
      MBO = MAX
      IF(MAX2 .GT. MAX) MBO = MAX2
C
C     LOG DATA IF REQUESTED
C
      CALL TFORM(Z,NLOG,PW,ADD,NDP)
C
C     SET ASIDE THE FIRST -MAX- DATA POINTS OF THE TIME SERIES AND
C     PUT THE REMAINDER OF THE SERIES IN ARRAY -ZZ-.
C
      MAX1 = MAX+1
  111 DO 12 J=MAX1,NDP
      K=J-MAX
 12   ZZ(K)=Z(J)
C
C     SET INITIAL VALUES IN ARRAY  -A- EQUAL TO ZERO.
C
      DO  13 J=1,MAX
   13 A(J) = 0.0
      CALL HAUS59(&21)
      CALL TSMOD(PA,B,&21)
C --- IF THE 'ARIMA REPLACEMENT OF EXTREMES' OPTION IS CHOSEN, OOOO
C --- STORE THE RESIDUALS OF MODEL ONE OR THREE
      IF(JAREX.EQ.0.OR.IREF.EQ.1.OR.IFB.EQ.2) GO TO 31
      IF(IFT.EQ.1) GO TO 28
      KCMAX=MAX
      DO 29 L=1,NDP
      RBMSF(L)=A(L+MAX)
   29 CONTINUE
      IF(IFT.EQ.0) SSQ1=SSQ
      IF(IFT.EQ.2) SSQ3=SSQ
      GO TO 31
   28 SSQ2=SSQ
   31 DO 14 I=MAX1,NDP
      J=I-MAX
   14 C(J) = A(I)
C     USE SUBROUTINE ACOR FOR DIAGNOSTIC CHECK
      CALL ACOR(C)
      DO 15 J=1,NDP
   15 B(J)=A(J)
C --- THE COUNTER IFT IS NORMALLY INCREMENTED IN SUBROUTINE TSFCST
      CALL TSFCST(NT,B,&21)
C --- THIS SECTION REARRANGES THE PARAMETERS IN PDQ ORDER.
      J = 0
      L = 0
      M = 1
      M1 = 0
      M2 = 0
      IF (IPDQ(7).EQ.0) GO TO 151
      K = IPDQ(1)+IPDQ(3)+IPDQ(4)+IPDQ(6)+1
      IF (IPDQ(7).EQ.2) GO TO 150
      M1 = 1
      L = IPDQ(1)+IPDQ(4)+1
```

```
      PT(K) = PA(L)
      L = 0
      GO TO 151
  150 M2 = 1
      PT(K) = PA(1)
      L = 1
  151 K = IPDQ(M)
      IF (K.EQ.0) GO TO 153
      DO 152 I = 1,K
      J = J+1
      L = L+1
  152 PT(J) = PA(L)
  153 IF (M.NE.1) GO TO 154
      M = 3
      L = J+IPDQ(4)+M1+M2
      GO TO 151
  154 IF (M.NE.3) GO TO 155
      M = 4
      L = IPDQ(1)+M2
      GO TO 151
  155 IF (M.NE.4) GO TO 156
      M = 6
      L = IPDQ(1)+IPDQ(3)+IPDQ(4)+M1+M2
      GO TO 151
  156 CONTINUE
      IF (IVALUE.EQ.0) GO TO 158
      DO 157 I = 2,4
  157 SDEV(I) = SDEV(I)/AVE
  158 FCTT(IFT) = (SDEV(2)+SDEV(3)+SDEV(4))/3.0
      II(IFT)=0
      K1=IPDQ(1)+1
      K2=IPDQ(1)+IPDQ(3)
      K3=IPDQ(1)+IPDQ(3)+IPDQ(4)+1
      K4=IPDQ(1)+IPDQ(3)+IPDQ(4)+IPDQ(6)
      L1=1
      L2=IPDQ(1)
      L3=K2+1
      L4=K3-1
      SP=0.
      DO 300 I=K1,K2
      IF (IPDQ(3).GT.0) SP=SP+PT(I)
  300 CONTINUE
      SQ=0.
      DO 301 I=K3,K4
      IF (IPDQ(6).GT.0) SQ=SQ+PT(I)
  301 CONTINUE
      IF (SP.GT..9.OR.SQ.GT..9) II(IFT)=1
      IF(IREF.EQ.0)WRITE(MT1,109) (IPDQ(I),I = 1,6),TRAN(NLOG),ADD,FCTT(
     1                IFT),(SDEV(J),J=2,4),PROB(IFT),RSQAR,(PT(I),I=1,NP)
  109 FORMAT(/1X,1H(,I1,1H,,I1,1H,,I1,2H)(,I1,1H,,I1,1H,,I1,1H),1X,A5,
     1        1X,E10.3,5F7.2,1H%,F8.4,2X,4E11.4,/,78X,4E11.4,/,78X,
     2        2E11.4,/)
      IF(IREF.EQ.1)WRITE(MT1,47)(DASH,I=1,30),(IPDQ(I),I=1,6),TRAN(NLOG)
     1,ADD,FCTT(IFT),(SDEV(J),J=2,4),PROB(IFT),RSQAR,(PT(I),I=1,NP)
      IF (IREF .EQ. 1) WRITE (MT1,6060)(DASH,I=1,30)
```

```
 6060 FORMAT(/,' ',30A4)                                                00052140
   47 FORMAT('-',//,41X,'    FINAL MODEL OF THE MODIFIED ',             00052150
     1'SERIES',/,' ',30A4,/,1X,1H(,I1,1H,,I1,1H,,I1,2H)(,I1,1H,,I1,1H,,I00052160
     21,1H),1X,A5,1X,E10.3,5F7.2,1H%,F8.4,2X,4E11.4,/,78X,4E11.4,/,78X, 00052170
     32E11.4)                                                           00052180
      IF(IREF.EQ.1) GO TO 37                                            00052190
      IF ((IFT.NE.2 .AND. IFORC .NE. 2) .OR.  IFB.EQ.2) GO TO 44        00052200
C --- IF THE 'ARIMA REPLACEMENT OF EXTREMES' OPTION IS CHOSEN, OOOO     00052210
C --- COMPARE MODELS ONE AND TWO                                        00052220
      IF(JAREX.EQ.0) GO TO 35                                           00052230
      IF (IFORC .EQ. 2) GO TO 53                                        00052240
      IF(FCTT(1).GE.FCTT(2)) GO TO 36                                   00052250
C --- MODEL ONE HAS SMALLER FORECAST ERRORS                             00052260
 53   SMSSQ=SSQ1                                                        00052270
      GO TO 35                                                          00052280
C --- MODEL TWO HAS SMALLER FORECAST ERRORS                             00052290
 36   KCMAX=MAX                                                         00052300
      DO 38 L=1,NDP                                                     00052310
      RBMSF(L)=A(L+MAX)                                                 00052320
   38 CONTINUE                                                          00052330
      SMSSQ=SSQ2                                                        00052340
   35 IF (FCTT(1).LT.12.0.AND.PROB(1).GT.10.0.AND.II(1).EQ.0) GO TO 116 00052350
      IF (FCTT(2).LT.12.0.AND.PROB(2).GT.10.0.AND.II(2).EQ.0) GO TO 116 00052360
   44 IF (IFORC.EQ.1.AND.IFT.NE.3) GO TO 8                              00052370
  116 WRITE(MT1,112) (DASH,I=1,31)                                      00052380
  112 FORMAT(1X,30A4,A2,/)                                              00052390
      K = 1                                                             00052400
      DO 76 I = 1,IFT                                                   00052410
      IF (FCTT(I).GT.12.0) GO TO 76                                     00052420
      IF (PROB(I).LT.10.0) GO TO 76                                     00052430
      IF (II(I).EQ.1) GO TO 76                                          00052440
      K = I                                                             00052450
   76 CONTINUE                                                          00052460
      DO 16 I = 1,IFT                                                   00052470
      IF (FCTT(I).GE.FCTT(K)) GO TO 16                                  00052480
      IF (PROB(I).LT.10.0) GO TO 16                                     00052490
      IF (II(I).EQ.1) GO TO 16                                          00052500
      K = I                                                             00052510
   16 CONTINUE                                                          00052520
      NRB=K                                                             00052521
   37 IF(IFORC.EQ.2.AND.JAREX.EQ.1.AND.IREF.EQ.1)NRB=2                  00052522
      IF(IFORC.EQ.1.AND.JAREX.EQ.1.AND.IREF.EQ.1)NRB=IFT                00052523
      DO 17 I = 1,NSEA                                                  00052530
      I1 = (NRB-1)*NSEA+I                                               00052540
   17 ZABC(LLDA+I) = FCT(I1)                                            00052550
      IF(IREF.EQ.1) GO TO 19                                            00052560
      IF (IFORC.EQ.2) GO TO 1800                                        00052570
      J1 = K*7-6                                                        00052580
      J2 = J1+6                                                         00052590
      J3 = LOGS(K)                                                      00052600
      IK=II(K)                                                          00052610
      IF(FCTT(K).LT.12.0.AND.PROB(K).GT.10.0.AND.IK.EQ.0) WRITE(MT1,110)00052620
     1   (MODELS(J),J=J1,J2),TRAN(J3)                                   00052630
  110 FORMAT(15X,'THE MODEL CHOSEN IS (',I1,1H,,I1,1H,,I1,2H)(,I1,1H,,  00052640
     1        I1,1H,,I1,1H),I1,' WITH TRANSFORMATION - ',A5,/)          00052650
```

```
      IF (FCTT(K).LT.12.0.AND.PROB(K).GT.10.0.AND.II(K).EQ.0) GO TO 39
      GO TO 1007
 1800 IK=II(1)
      IF (ITB.EQ.1.AND.FCTT(1).LT.18.0.AND.PROB(1).GT.10.0.AND.IK.EQ.0)
     1WRITE (MT1,110)  (IPDQ(I),I=1,7),TRAN(NLOG)
      IF (ITB.EQ.1.AND.FCTT(1).LT.18.0.AND.PROB(1).GT.10.0.AND.IK.EQ.0)
     1GO TO 19
      IF (ITB.NE.0) GO TO 1008
   18 WRITE(MT1,110) (IPDQ(I),I=1,7),TRAN(NLOG)
      IF (NLOG.EQ.3) WRITE(MT1,118) PW
  118 FORMAT(1H0,14X,'THE POWER IS ',F15.5)
      IF (MIT.NE.30) WRITE(MT1,218) MIT
  218 FORMAT(1H0,14X,'THE MAXIMUM NUMBER OF ITERATIONS IS ',I6)
      IF (IOR.EQ.0) GO TO 518
      WRITE(MT1,418) (IORDER(I),I=1,NP)
  418 FORMAT(1H0,14X,'THE ORDERS FOR THE PARAMETERS ARE ',10I4)
  518 DO 618 I = 1,NP
      IF (PV(I).NE.0.1) GO TO 718
  618 CONTINUE
      GO TO 918
  718 WRITE(MT1,818) (PV(I),I=1,NP)
  818 FORMAT(1H0,14X,'THE INITIAL VALUES FOR THE PARAMETERS ARE
     1 10F6.2)
  918 CONTINUE
      IF (FCTT(1).GE.999.) GO TO 1006
C --- IF THE 'ARIMA REPLACEMENT OF EXTREMES' OPTION IS CHOSEN,
C --- COMPARE MODEL THREE WITH THE BEST MODEL OBTAINED SO FAR
   39 IF(JAREX.EQ.0.OR.IFB.EQ.2) GO TO 19
      IF(K.LT.3) GO TO 41
C --- MODEL THREE IS THE BEST
      KCMAX=MAX
      DO 42 M=1,NDP
      RBMSF(M)=A(M+MAX)
   42 CONTINUE
      SMSSQ=SSQ3
   41 KCMAX1=KCMAX+1
      STDEV=SQRT(SMSSQ/(NDP-KCMAX1))
      TPFSD=2.5*STDEV
C --- REPLACE EXTREMES BY THE VALUES FITTED BY THE BEST MODEL
C --- SET ASIDE THE FIRST KCMAX DATA POINTS OF THE TIME SERIES
C --- AND PUT THE REMAINDER OF THE SERIES IN ARRAY ZZ
      DO 49  I=KFDA,LLDA
      Z(I-KFDA+1)=ZABC(I)
   49 CONTINUE
      IF(IFORC.EQ.1)NLOG=J3
      CALL TFORM(Z,NLOG,PW,AOD,NDP)
      DO 48 L=KCMAX1,NDP
      M=L-KCMAX
      ZZ(M)=Z(L)
   48 CONTINUE
      DO 43 J=KCMAX1,NDP
      IF(ABS(RBMSF(J)).GT.TPFSD) ZZ(J)=ZZ(J)-RBMSF(J)
   43 CONTINUE
C --- SET INDICATOR FOR REFITTING THE BEST MODEL
      IREF=1
```

```
      N=K-1
C --- REDEFINE Z
      DO 32 J=KCMAX1,NDP
      K=J-KCMAX
      Z(J)=ZZ(K)
   32 CONTINUE
C --- PERFORM INVERSE TRANSFORM
      CALL DFORM(Z,NDP,PW,NLOG,ADD,Z)
C --- STORE ORIGINAL SERIES
      DO 50 M=1,NDP
      RBMSF(M)=ZABC(M)
   50 CONTINUE
C --- REDEFINE ZABC
      DO 33 I=KFDA,LLDA
      ZABC(I)=Z(I-KFDA+1)
   33 CONTINUE
      GO TO 8
   19 LLDAF=LLDAF+NSEA
      ITB=0
      IF (IFORC.EQ.2) GO TO 2999
      IF (IFB.EQ.1) IFORC=2
      IF (IFB.EQ.1) ITB=1
      DO 299 J=J1,J2
      IF (IFB.EQ.1) IPDQ(J-J1+1)=MODELS(J)
  299 CONTINUE
      IF (IFB.EQ.1) NLOG=J3
 2999 RETURN
 1001 WRITE(MT2,2001)
 2001 FORMAT(1H0,'ERROR IN THE ARIMA SECTION OF X-11. NONE OF THE MODEL
     1PARAMETERS CAN BE GREATER THAN 4.')
      GO TO 1006
 1002 WRITE(MT2,2002)
 2002 FORMAT(1H0,'ERROR IN THE ARIMA SECTION OF X-11. LOG OR POWER TRANS
     1FORMATIONS CAN NOT BE USED WITH NEGATIVE VALUES. NO TRAN. ASSUMED'
     2)
      NLOG = 1
      GO TO 7
 1003 WRITE(MT2,2003)
 2003 FORMAT(1H0,'ERROR IN THE ARIMA SECTION OF X-11. THE NUMBER OF PARA
     1METERS IN THE MODEL MUST NOT EXCEED 10.')
      GO TO 1006
 1004 WRITE(MT2,2004)
 2004 FORMAT(1H0,'THE NUMBER OF YEARS IN THE SERIES CAN NOT EXCEED 29.'
     1 )
      GO TO 1006
 1005 WRITE(MT2,2005)
 2005 FORMAT(1H0,'THERE MUST BE AT LEAST 5 YEARS IN THE SERIES BEFORE AR
     1IMA FORECASTS CAN BE MADE.')
 1006 IF(IFORC.EQ.1)WRITE(MT2,2006)
      IF(IFORC.EQ.1)WRITE(MT1,3006)
 2006 FORMAT(1H0,'NO ARIMA FORECAST AND BACKCAST MADE',
     1' BY THE AUTOMATIC OPTION.')
 3006 FORMAT(1H0,14X,'NO ARIMA FORECAST AND BACKCAST MADE',
     1' BY THE AUTOMATIC OPTION.')
      IF(IFORC.EQ.1)WRITE (MT1,2009)
```

```
 2009 FORMAT (' USER SHOULD IDENTIFY NEW MODEL OR CAN STILL APPLY',      00053750
     1' REJECTED MODEL IF CONSIDERED SATISFACTORY FOR HIS/HER SERIES',/,00053760
     2' WITH THE OPTION FOR USER-SUPPLIED MODEL.')                       00053770
      IFORC = 0                                                          00053780
      RETURN                                                             00053790
 1007 WRITE(MT2,2010)                                                    00053800
 2010 FORMAT(1H0,'THE AVERAGE FORECAST ERROR MUST BE LESS THAN 12%,'     00053810
     1,' THE AVERAGE BACKCAST ERROR MUST BE LESS THAN 18%,',/,           00053820
     1' THE CHI-SQUARED PROBABILITY MUST BE GREATER THAN 10%',           00053830
     2' AND EVIDENCES OF OVERDIFFERENCING MUST NOT BE PRESENT.')         00053840
      WRITE(MT1,2007)                                                    00053850
 2007 FORMAT(1H0,'THE AVERAGE FORECAST ERROR MUST BE LESS THAN 12%, THE 00053860
     1 CHI-SQUARED PROBABILITY MUST BE GREATER THAN 10%',/               00053870
     2,' AND EVIDENCES OF OVERDIFFERENCING MUST NOT BE PRESENT.')        00053880
      GO TO 1006                                                         00053890
 1008 WRITE (MT1,2008)                                                   00053900
 2008 FORMAT (1H0,'THE AVERAGE BACKCAST ERROR MUST BE LESS THAN 18%, THE00053910
     1 CHI-SQUARED PROBABILITY MUST BE GREATER THAN 10%',/               00053920
     2,' AND EVIDENCES OF OVERDIFFERENCING MUST NOT BE PRESENT.')        00053930
      WRITE (MT2,2010)                                                   00053940
      IF(JAREX.EQ.0) GO TO 1006                                          00053950
      DO 51 N=1,NDP                                                      00053960
      ZABC(N)=RBMSF(N)                                                   00053970
   51 CONTINUE                                                           00053980
      GO TO 1006                                                         00053990
C --- IF THE INITIAL PARAMETER VALUES CAUSE AN OVERFLOW CHOSE ANOTHER    00054000
C --- SET OF INITIAL PARAMETERS USING A RANDOM NUMBER GENERATOR.         00054010
   21 JTEST = JTEST+1                                                    00054020
      IF (JTEST.LT.4) GO TO 22                                           00054030
C --- STOP CALCULATION. THIS MODEL IS ASSUMED TO BE A POOR FIT.          00054040
      JTEST = 1                                                          00054050
      IFT = IFT+1                                                        00054060
      FCTT(IFT) = 999.                                                   00054070
      PROB(IFT) = 0.0                                                    00054080
      WRITE(MT1,113) (IPDQ(I),I=1,6)                                     00054090
  113 FORMAT(/,1X,'THE PROGRAM COULD NOT ESTIMATE PARAMETERS FOR MODEL (00054100
     1',3I2,2H)(,3I2,2H).,/)                                             00054110
      GO TO 44                                                           00054120
   22 DO 23 I = 1,NP                                                     00054130
      IRD = IRD*125                                                      00054140
      IRD = IRD-IRD/2796203*2796203                                      00054150
   23 PA(I) = (IRD/2796203. -0.5)*1.9                                    00054160
C --- THIS SECTION INSURES THAT THE MODEL IS INVERTABLE AND CONVERGES    00054170
      K = 1                                                              00054180
      J = 0                                                              00054190
   24 L = IPDQ(K)                                                        00054200
      IF (L.LE.1) GO TO 26                                               00054210
      PA(J+L-1) = PA(J+L-1)*(1.0-PA(J+L))                                00054220
      IF (L.LE.2) GO TO 26                                               00054230
      DO 25 L1 = 2,L                                                     00054240
   25 PA(J+L-L1) = PA(J+L-L1)*(1.0-PA(J+L-L1+1)/2.0)*(1.0-PA(J+L))/2.0   00054250
   26 J = J+L                                                            00054260
      IF (K.EQ.6) GO TO 111                                              00054270
      IF (K.EQ.3) K = 6                                                  00054280
      IF (K.EQ.4) K = 3                                                  00054290
```

```
      IF (K.EQ.1) K = 4
      GO TO 24
      END
      SUBROUTINE CONV(PA)
      DIMENSION PA(1)
      COMMON /ARIM1/ PV(10),PW,ADD,IPDQ(7),IORDER(10),NLOG,MIT,NDP,NSEA
      COMMON /ARIM2/ NOB,N,NRD,NSD,MBO,MAX1,MAX11,IOPA(10),INC(6)
      DO 3 I=1,N
      IF (ABS(PV(I)).GT.0.001) GO TO 3
      PV(I)=.1
    3 CONTINUE
    5 IF(IORDER(1).NE.0)GO TO 100
      J=0
      N=0
    7 J=J+1
      IF(J.GT.7)GO TO 100
      IF(J.EQ.2.OR.J.EQ.5)GO TO 7
      IF(IPDQ(J).EQ.0)GO TO 7
      IF(J.GT.3)GO TO 33
C     J MUST BE 1 OR 3.
      M=1
    9 K=0
   11 K=K+1
      IF(K.GT.IPDQ(J))GO TO 7
      N=N+1
      IORDER(N)=K*M
      GO TO 11
   33 IF(J.GT.6)GO TO 66
C     J MUST BE 4 OR 6.
      M=NSEA
      GO TO 9
   66 N=N+1
      IORDER(N)=0
  100 INC(1)=IPDQ(1)
      INC(2)=IPDQ(4)
      INC(3)=0
      INC(4)=0
      IF(IPDQ(7).EQ.1)INC(4)=1
      IF(IPDQ(7).EQ.2)INC(3)=1
      INC(5)=IPDQ(3)
      INC(6)=IPDQ(6)
      J=0
      N=0
      K1=0
  110 J=J+1
      IF(J.GT.6)RETURN
      IF(INC(J).EQ.0)GO TO 110
      IF(J.NE.1)GO TO 122
      K1=1
  112 K2=K1+INC(J)-1
      DO 115 I=K1,K2
      N=N+1
      IOPA(N)=IORDER(I)
  115 PA(N)=PV(I)
      GO TO 110
```

```
  122 IF(J.NE.2)GO TO 133
      K1=INC(1)+INC(5)+1
      GO TO 112
  133 IF(J.NE.3.AND.J.NE.4)GO TO 155
      K1=INC(1)+INC(2)+INC(5)+INC(6)+1
      GO TO 112
  155 IF(J.NE.5)GO TO 166
      K1=INC(1)+1
      GO TO 112
  166 IF (J.NE.6) RETURN
      K1=INC(1)+INC(5)+INC(2)+1
      GO TO 112
      END
      SUBROUTINE HAUS59(*)
      COMMON /UNITS/ MT,MT2
      COMMON /MQ1  / ZZ(372),A(372)
      COMMON /MQ4  / S(372),R(372)
      COMMON /MQ5  / Q(10),P(10),E(10),PHI(10),TB(10),B(10,10),D(10,10),
     1               TMP(470)
      COMMON /MQ6  / F(395)
      COMMON /ARIM1/ PV(10),PW,ADD,IPDQ(7),IORDER(10),NLOG,MIT,NDP,NSEA
      COMMON /ARIM2/ NOB,NP,NRD,NSD,MBO,MAX1,MAX11,IOPA(10),INC(6)
      COMMON /ARIM3/ SD(4),PROB(3),SSQ,TH(10),FCT(48),IFT,FCTT(3),RSQAR,
     1               IVALUE
      DIMENSION DELZ(179,10)
      NIT = 1
      SSQ = 0.0
      GA = 0.001
      CALL TSMOD(TH,F,&415)
      DO 90 I = 1, NOB
      R(I) = ZZ(I) - F(I)
      IF (SSQ.GE.1.0E40) GO TO 415
   90 SSQ=SSQ+R(I)*R(I)
  100 GA = GA / 10.0
      INTCNT = 0
C
C     MINIMIZE THE SUM FROM 1 TO  NOB  OF A(T)**2 BY FINDING THE
C     PARTIAL DERIVATIVE W.R.T. EACH PARAMETER OF THIS SUM.
C     D/D(TH)  (SUM A(T)**2)    = SUM (D/D(TH) (A(T)**2)
C                               = SUM ( 2*A(T)*  D/D(TH) A(T) )
C     THE CONSTANT FACTOR 2 IS NEGLECTED.  A(T) = R(I) IN THE NOTATION
C     OF THIS PROGRAM.  THE PARTIAL DERIVATIVE D/D(TH(J)) IS FOUND
C     BY ADDING THE INCREMENTAL QUANTITY P(J) TO EACH PARAMETER TH(J)
C     IN TURN, AND FINDING THE DIFFERENCE QUOTIENT (A(TH+P)-A(TH))/P
C     WHICH IN THE NOTATION OF THE PROGRAM IS (S(I) - F(I))/ P.
C     THE  NP  NUMERICAL PARTIAL DERIVATIVES ARE STORED IN THE ARRAY  Q
C
      DO 130 J=1,NP
      TEMP = TH(J)
      P(J)=0.01*TEMP
      TH(J) = TEMP + P(J)
      Q(J) = 0.0
      CALL TSMOD (TH,S,&415)
      DO 120 I=1,NOB
      DELZ(I,J) = S(I) - F(I)
```

```
  120 Q(J) = Q(J) + DELZ(I,J)*R(I)
      Q(J) = Q(J)/P(J)
C
C     Q WHICH IS THE MATRIX PRODUCT X(TRANSPOSE) * R IS THE VECTOR OF
C     STEEPEST DESCENT ON THE LEAST SQUARES SURFACE WITH PARAMETERS
C     TH(I), I=1,NP.
C
  130 TH(J) = TEMP
C
C     LOOP TO FIND THE MATRIX PRODUCT X(TRANSPOSE)*X = D(I,J)  WHICH
C     IS NEEDED FOR THE ¬LINEARIZED LEAST SQUARES¬ METHOD.
C     FOR REFERENCE SEE : DRAPER AND SMITH.
C
      DO 150 I=1,NP
      DO 151 J=1,I
      SUM = 0.0
      DO 160 K = 1, NOB
  160 SUM = SUM + DELZ(K,I) * DELZ(K,J)
      TEMP = SUM/(P(I)*P(J))
      D(J,I) = TEMP
  151 D(I,J) = TEMP
      IF (TEMP.LT.0.0) GO TO 415
  150 E(I) = SQRT(TEMP)
  666 CONTINUE
C
C     IN ORDER TO COMPROMISE BETWEEN THE ¬STEEPEST DESCENT¬ AND
C     ¬LINEARIZED LEAST SQUARES¬ METHODS, THE MATRIX D(I,J) IS
C     NORMALIZED TO OBTAIN MATRIX B(I,J).
C
      DO 153 I = 1, NP
      DO 153 J=1,I
      IF(ABS(E(I)*E(J)).LT.0.1E-08)GO TO 1
      B(I,J)=D(I,J) / (E(I)*E(J))
      CALL DVCHK(IDIV)
      IF (IDIV.EQ.1) GO TO 415
  153 B(J,I) = B(I,J)
C                              B = SCALED MOMENT MATRIX
C
C     LOOP TO ADD THE QUANTITY GA TO THE DIAGONAL ELEMENTS OF B(I,J)
C     THIS OPERATION IS EQUIVALENT TO IMPOSING THE CONSTRAINT THAT THE
C     PARAMETERS MUST LIE ON A SPHERE IN THE PARAMETER SPACE WHICH
C     HAS CENTRE TH(I) AND WHOSE RADIUS IS A FUNCTION OF GA.
C     THE THEORY OF LAGRANGIAN MULTIPLIERS MAY BE USED TO DETERMINE
C     THE EXACT FUNCTIONAL RELATIONSHIP BETWEEN GA AND THE RADIUS OF
C     THE SPHERE, IF THE READER IS SUFFICIENTLY INTERESTED.
C
      DO 155 I=1,NP
      P(I) = Q(I)/E(I)
      PHI(I) = P(I)
  155 B(I,I) = B(I,I) + GA
C
C     IN THE SUBROUTINE MATIN, THE MATRIX B(I,J) IS INVERTED AND
C     THE INVERSE IS MATRIX MULTIPLIED TIMES P.  THE RESULT IS
C     RETURNED IN THE MATRIX (VECTOR IN THIS CASE) P.
C
```

```
      CALL MATIN (B, NP, P, 1, DET, &415)
C                                P/E = CORRECTION VECTOR
      STEP = 1.0
      SUM1 = 0.
      SUM2 = 0.
      SUM3 = 0.
C
C     LOOP TO FIND THE DOT PRODUCTS P.PHI, P.P, PHI.PHI   .
C
      DO 231 I = 1,NP
      SUM1 = P(I) * PHI(I) + SUM1
      SUM2 = P(I) * P(I) + SUM2
      SUM3 = PHI(I) * PHI(I) + SUM3
      CALL OVERFL(JOV)
      IF (JOV-2) 415,231,415
  231 PHI(I) = P(I)
C
C     FIND THE ANGLE BETWEEN THE STEEPEST DESCENT VECTOR (PHI) AND
C     THE ¬COMPROMISED LINEARIZED LEAST SQUARES VECTOR¬ (P).
C
      TMP1 = SUM2*SUM3
      IF (TMP1.LT.0.0) GO TO 415
      TEMP = SUM1/SQRT(TMP1)
      TMP1 = ABS(TEMP)
      IF(TMP1.GT.0.9999)GO TO 232
      TEMP=57.295*ARCOS(TEMP)
      GO TO 170
  232 TEMP = 0.0
C
C     LOOP TO ADJUST THE PARAMETER VALUES.
C
  170 DO 220 I = 1,NP
      P(I) = PHI(I) * STEP / E(I)
      TB(I) = TH(I) + P(I)
  220 CONTINUE
  215 CONTINUE
C
C     CALL TSMOD AND THEN LOOP TO FIND THE VALUE OF THE SUM OF SQUARES
C     USING THE AMENDED VALUES OF THE PARAMETERS.
C
      SUMB = 0.0
      CALL TSMOD(TB,F,&415)
      DO 230 I=1, NOB
      R(I) = ZZ(I)-F(I)
      SUMB = SUMB + R(I) * R(I)
      IF (SUMB.GE.1.0E40) GO TO 665
  230 CONTINUE
C
C     COMPARE THE SUM OF SQUARES USING AMENDED PARAMETERS(SUMB) WITH
C     THE SUM OF SQUARES USING THE OLD VALUES OF THE PARAMETERS (SSQ).
C
      IF(SUMB - SSQ) 662, 662, 663
  663 IF(AMIN1(TEMP-30.0, GA)) 665, 665, 664
  665 STEP = STEP/2.0
      INTCNT = INTCNT + 1
```

```
      DO 217 I=1,NP
      TEST = (ABS(P(I))) /  (1.0E-20 + ABS(TH(I)))
      IF (TEST.GT.0.004)  GO TO 219
  217 CONTINUE
      IF (INTCNT.GT.20)       GO TO 276
      DO 218  I=1,NP
  218 TB(I) = TH(I)
      GO TO 215
  219 IF(INTCNT - 36) 170, 276, 276
  664 GA = GA*10.0
      INTCNT = INTCNT + 1
      IF(INTCNT - 36) 666, 276, 276
  662 DO 669 I=1,NP
  669 TH(I) = TB(I)
      DO 240 I = 1, NP
      CALL DVCHK(IDIV)
      IF (IDIV.EQ.1) GO TO 265
      IF (ABS(P(I))/(1.E-20+ABS(TH(I)))-0.004) 240,240,265
  240 CONTINUE
      GO TO 276
  265 SSQ = SUMB
      NIT = NIT + 1
      IF(NIT - MIT) 100, 100, 276
 276  CONTINUE
      SSQ = SUMB
      IDF = NOB-NP
      CALL MATIN (D, NP, P, 0, DET, &415)
      DO 7692 I=1,NP
      TMP1 = D(I,I)
      IF (TMP1.LT.0.0) GO TO 415
 7692 E(I) = SQRT(TMP1)
      DO 340 I=1, NP
      DO 340 J = I, NP
      B(J,I) = D(J,I)  / (E(I)*E(J))
  340 B(I,J) = B(J,I)
      IF(IDF) 341, 414, 341
  341 RVARC = SSQ / IDF
C     CALCULATE R-SQUARED
      TOTAL=0.0
      DO 5934 IZ=1,NOB
5934  TOTAL=TOTAL+R(IZ)+F(IZ)
      ANOB=NOB
      TOTAL=TOTAL/ANOB
      ROSS=0.0
      DO 5935 IZ=1,NOB
5935  ROSS=ROSS+(R(IZ)+F(IZ)-TOTAL)*(R(IZ)+F(IZ)-TOTAL)
      ROSS=ROSS/(ANOB-1.)
      RSQAR=1.0-RVARC/ROSS
      IF (RSQAR.LT.0) RSQAR=0
  414 RETURN
    1 WRITE(MT2,2)
    2 FORMAT('0IN SUBROUTINE HAUS59, NEAR ISN 47, THE QUANTITY ',
     1'E(I)*E(J) WAS FOUND TO BE LESS THAN 0.1E-08.')
  415 RETURN 1
      END
```

```
      SUBROUTINE TSFCST(NT,W,*)
      COMMON /WORK / T(48),C(48),CF(48),PS(216),ZP(12)
      COMMON /MQ5  / F(372),TEMP(372)
      COMMON /MQ1  / ZZ(372),A(372)
      COMMON /MQ7  / Z(372)
      COMMON /ARIM1/ PV(10),POW,ADD,IPDQ(7),IORDER(10),NLOG,MIT,NDR,NSEA
      COMMON /ARIM2/ NOB,NP,NRD,NSD,MBO,MAX1,MAX11,IOPA(10),INC(6)
      COMMON /ARIM3/ SDEV(4),PROB(3),SSQ,PA(10),FCT(48),IFT,FCTT(3),
     1               RSQAR,IVALUE
      DIMENSION W(1),NT(4)
      INTEGER FF
      NDRPNF=NDR+NSEA
      DO 9 J=1,NDRPNF
    9 A(J) =0.0
      NDR1=NDR+1
      DO 13 J=NDR1,NDRPNF
   13 Z(J) = 0.0
C     LOG DATA IF REQUESTED
      IF (NLOG.EQ.1) GO TO 18
      DO 16  J=NDR1,NDRPNF
      IF((Z(J)-0.001).GT.0) Z(J) = ALOG(Z(J))
   16 CONTINUE
   18 DO 17  J=1,NDRPNF
   17 W(J) = Z(J)
      NDB=NDR-MAX1
C     USE TSMOD TO DETERMINE C AND CF AND RESIDUALS
      CALL TSMOD (PA,F,&96)
      PS(1)=C(1)-CF(1)
      IF(NSEA .LE. MBO) GO TO 26
      MAX= MBO
      GO TO 29
   26 MAX=NSEA
C     LOOPS THROUGH 42 TO DETERMINE PSI VALUES
   29 DO 35 J=2,MAX
      PS(J) = C(J)-CF(J)
      L=J-1
      DO 32 K=1,L
      KK=K-1
   32 PS(J)=PS(J)+PS(K)*C(L-KK)
   35 CONTINUE
      IF(NSEA .LE. MBO) GO TO 45
      MBOP1 = MBO + 1
      DO 42 J=MBOP1,NSEA
      PS(J)=0
      L=J-1
      I=J-MBO
      DO 38 K=I,L
      KK=K-1
   38 PS(J)=PS(J)+PS(K)*C(L-KK)
   42 CONTINUE
   45 KK=INC(1)+INC(2)+INC(3)
      RMEAN=0.0
      IF(INC(3) .EQ. 0) GO TO 51
      DO 48 J=1,NDR
   48 Z(J) = Z(J)-PA(KK)
```

```
      RMEAN = PA(KK)
   51 KK=KK+INC(4)
      IF(INC(4) .EQ. 1) CONST=PA(KK)
      IF (INC(4) .EQ. 0) CONST=0.0
C     LOOP FOR EACH TIME ORIGIN FOR WHICH FORECASTS DESIRED
      DO 95 IT=1,4
      FF=NT(IT)+1
      LF = NT(IT)+NSEA
C     LOOP TO DETERMINE FORECASTS
      DO 58 K=FF,LF
      A(K) = 0.0
      Z(K) = CONST
      DO 54 J=1,MBO
   54 Z(K)=Z(K)+C(J)*Z(K-J)-CF(J)*A(K-J)
   58 CONTINUE
      DO 62 J=FF,LF
      Z(J)=Z(J)+RMEAN
      L=J+1-FF
   62 ZP(L)=Z(J)
      IF (IT.EQ.1) GO TO 81
      SDEV(IT) = 0.0
      DO 80 J=1,NSEA
      K=J+NT(IT)
      GO TO (75,76,77),NLOG
   75 WT = W(K)-ADD
      ZT = ZP(J)-ADD
      GO TO 78
   76 WT = EXP(W(K))-ADD
      ZT = EXP(ZP(J))-ADD
      GO TO 78
   77 WT = W(K)**(1.0/POW)-ADD
      ZT = ZP(J)**(1.0/POW)-ADD
   78 IF (IVALUE.EQ.0) GO TO 79
      SDEV(IT) = ABS(ZT-WT)+SDEV(IT)
      GO TO 80
   79 SDEV(IT) = SDEV(IT)+ABS(ZT/WT-1.0)
   80 CONTINUE
      SDEV(IT) = SDEV(IT)*100.0/NSEA
      GO TO 95
   81 ITT = NSEA*IFT+1
      IFT = IFT+1
      CALL DFORM(ZP,NSEA,POW,NLOG,ADD,FCT(ITT))
   95 CONTINUE
      RETURN
   96 RETURN 1
      END
      SUBROUTINE DFORM(ZP,NSEA,POW,NLOG,ADD,FCAST)
      DIMENSION FCAST(1),ZP(1)
      DO 100 J=1,NSEA
      A1 = ZP(J)
      GO TO (10,20,30),NLOG
   10 A2 = A1
      GO TO 100
   20 A2=EXP(A1)
      GO TO 100
```

```
   30 XPOW=1./POW
      A2=A1**XPOW
  100 FCAST(J)=A2-ADD
      RETURN
      END
      SUBROUTINE TSMOD(PA,F,*)
      REAL*8 TEMP
      COMMON /WORK /T(48),C(48),CF(48),D(4,4),DS(4,48),TMP(20)
      COMMON /WORK3/ W(372)
      COMMON /MQ1  / ZZ(372),A(372)
      COMMON /MQ7  / Z(372)
      COMMON /ARIM1/ PV(10),PW,ADD,IPDQ(7),IORDER(10),NLOG,MIT,NDR,NSEA
      COMMON /ARIM2/ NOB,NP,NRD,NSD,MBO,MAX1,MAX11,IOPA(10),INC(6)
      DIMENSION PA(1),F(1)
      IF (MBO .EQ. 0) GO TO 68
      DO 3 J=1,MBO
      T(J)=0.0
      C(J)=0.0
    3 CF(J)=0.0
      IF(NRD .EQ. 0) GO TO 17
      D(1,1)=-1.0
      IF(NRD .EQ. 1) GO TO 8
C     LOOPS THROUGH 13 TO EXPAND REGULAR DIFFERENCE TERM
      DO 4 J=2,NRD
    4 D(J,1) = D(J-1,1) - 1.0
    8 DO 13 J=1,NRD
      DO 13 K=2,NRD
      IF(K .GT. J) GO TO 10
      D(J,K)=D(J-1,K)-D(J-1,K-1)
      GO TO 13
   10 D(J,K)=0
   13 CONTINUE
      DO 15 K=1,NRD
   15 C(K)=D(NRD,K)
   17 IF(NSD .EQ. 0) GO TO 43
      MAX=NSD*NSEA
C     LOOPS THROUGH 33 TO EXPAND SEASONAL DIFFERENCE TERM
      DO 20 J=1,NSD
      DO 20 K=1,MAX
   20 DS(J,K)=0.0
      DS(1,NSEA) = -1.0
      MIN=2*NSEA
      IF(NSD .EQ. 1) GO TO 35
      DO 24 J=2,NSD
   24 DS(J,NSEA)= DS(J-1,NSEA)-1.0
      DO 33 J=1,NSD
      DO 33 K=MIN,MAX,NSEA
      IF((K/NSEA) .GT. J) GO TO 30
      DS(J,K)=DS(J-1,K)-DS(J-1,K-NSEA)
      GO TO 33
   30 DS(J,K)=0.0
   33 CONTINUE
   35 L=NSEA*NSD
      DO 38 J=NSEA,L,NSEA
      T(J) = T(J) + DS(NSD,J)
```

```
      IF (NRD .EQ. 0) GO TO 38                                          00058650
      DO 37 M=1,NRD                                                     00058660
      IF (J .EQ. NSEA) T(M) = T(M) + C(M)                               00058670
      N=J+M                                                             00058680
   37 T(N)=T(N)+C(M)*DS(NSD,J)                                          00058690
   38 CONTINUE                                                          00058700
      L=L+NRD                                                           00058710
      DO 40 J=1,L                                                       00058720
      C(J)=T(J)                                                         00058730
   40 T(J)=0.0                                                          00058740
   43 MIN=1                                                             00058750
      MAX=0                                                             00058760
      L = NRD + NSEA*NSD                                                00058770
C                                                                       00058780
C     LOOP TO EXPAND TWO FACTORS INVOLVING PHIS AND COMBINE WITH        00058790
C     PREVIOUS EXPANSION OF DIFFERENCE TERMS TO GET FINAL C VALUES,     00058800
C     AND TO EXPAND TWO FACTORS INVOLVING THETAS TO OBTAIN FINAL        00058810
C     CF VALUES                                                         00058820
C                                                                       00058830
      DO 60 I=1,6                                                       00058840
      IF(I .EQ. 3)L=0                                                   00058850
      IF(INC(I) .EQ. 0) GO TO 60                                        00058860
      MAX=MAX+INC(I)                                                    00058870
      IF(I .EQ. 3 .OR. I .EQ. 4) GO TO 60                               00058880
      DO 48 M=MIN,MAX                                                   00058890
      K=IOPA(M)                                                         00058900
      T(K)=T(K)-PA(M)                                                   00058910
      IF(L .EQ. 0) GO TO 48                                             00058920
      DO 45 J=1,L                                                       00058930
      IF(M .EQ. MIN .AND. I .LE. 2) T(J)=T(J)+C(J)                      00058940
      IF(M .EQ. MIN .AND. I .EQ. 6)T(J)=T(J)+CF(J)                      00058950
      N=J+IOPA(M)                                                       00058960
      IF(I .EQ. 6)T(N)=T(N)-PA(M)*CF(J)                                 00058970
   45 IF(I .LE. 2)T(N)=T(N)-PA(M)*C(J)                                  00058980
   48 CONTINUE                                                          00058990
      L=L+IOPA(MAX)                                                     00059000
      DO 51 J=1,L                                                       00059010
      IF(I .GE. 5)CF(J)=T(J)                                            00059020
      IF(I .LE. 2)C(J)=T(J)                                             00059030
   51 T(J)=0.0                                                          00059040
   60 MIN=MIN+INC(I)                                                    00059050
      DO 66 J=1,MBO                                                     00059060
      CF(J)=-CF(J)                                                      00059070
   66 C(J)=-C(J)                                                        00059080
   68 KK=INC(1)+INC(2)+INC(3)                                           00059090
      DO 70 J=1,NDR                                                     00059100
      W(J)=Z(J)                                                         00059110
   70 IF(INC(3) .EQ. 1)Z(J)=Z(J)-PA(KK)                                 00059120
      KK=KK+INC(4)                                                      00059130
      IF(INC(4) .EQ. 0)CONST=0.0                                        00059140
      IF(INC(4) .EQ. 1)CONST=PA(KK)                                     00059150
C     LOOP TO CALCULATE RESIDUALS                                       00059160
      DO 79 K=MAX11,NDR                                                 00059170
      TEMP = DBLE(Z(K))-DBLE(CONST)                                     00059180
      KK = K - 1                                                        00059190
```

```
      IF (MBO .EQ. 0 .OR. KK .EQ. 0) GO TO 79
      IF(K .GT. MBO) KK = MBO
      DO 74 J=1,KK
C --- THIS SUBROUTINE CHECKS FOR AN OVERFLOW
      CALL OVERFL(JOV)
      IF (JOV-2) 84,73,84
   73 KJ = K-J
   74 TEMP = (TEMP-DBLE(C(J))*DBLE(Z(KJ)))+DBLE(CF(J))*DBLE(A(KJ))
      A(K) = SNGL(TEMP)
   79 CONTINUE
      DO 81 J=1,NDR
   81 Z(J)=W(J)
      MAX1=MAX11-1
C     LOOP TO CALCULATE FITTED VALUES
      DO 83 J=1,NOB
      LL=J+MAX1
   83 F(J)=Z(LL)-A(LL)
      RETURN
   84 RETURN 1
      END
      SUBROUTINE ACOR (C)
      REAL*8 TOTALS
      COMMON /ARIM1/ PV(10),PW,ADD,IPDQ(7),IORDER(10),NLOG,MIT,NDP,NSEA
      COMMON /ARIM2/ NOB,NP,NRD,NSD,MBO,MAX1,MAX11,IOPA(10),INC(6)
      COMMON /ARIM3/ SDEV(4),PROB(3),SSQ,PA(10),FCT(48),IFT,FCTT(3),
     1               RSQAR,IVALUE
      DIMENSION C(1)
      NC = 24
      IF (NSEA.EQ.4) NC = 12
      IF (NC.GE.NOB) NC = NOB-1
      DIV = NOB
C     CALCULATE OVERALL MEAN
      E = TOTALS(C,1,NOB,1,1)
C     LOOP TO CALCULATE DEVIATIONS
      TOT=0.0
      DO  2 J=1,NOB
      C(J)=C(J)-E
    2 TOT=TOT+C(J)*C(J)
C     LOOP TO CALCULATE KTH AUTOCORRELATION
      SUMR = 0.0
      DO 4 K=1,NC
      M=NOB-K
      SUM=0.0
C     LOOP TO DETERMINE SUMS USED IN AUTOCORRELATION
      DO 3 J=1,M
    3 SUM=SUM+C(J)*C(J+K)
      SUM = SUM/TOT
    4 SUMR = SUMR+SUM*SUM/(DIV-K)
      SUMR = SUMR*DIV*(DIV+2.0)
      IDFF=NC-NP
      PROB(IFT+1) = CHISQ(SUMR,IDFF)*100.0
      RETURN
      END
      SUBROUTINE MATIN(A, NVAR, B, NB, DET, *)
      DIMENSION A(10,10), B(NVAR), C(10)
```

```
C
C     THIS SUBROUTINE FINDS THE INVERSE OF A POSITIVE DEFINITE
C     SYMMETRIC MATRIX BY THE PIVOTAL CONDENSATION METHOD.
C     IF NB IS NOT 0, IT MULTIPLIES THE INVERSE MATRIX TIMES B.
C
      DET = 1.0
      DO 3 ICOL = 1, NVAR
      PIVOT = A(ICOL, ICOL)
      DET = PIVOT * DET
C
C     DIVIDE PIVOT ROW BY PIVOT ELEMENT
C
      A(ICOL, ICOL) = 1.0
C
      PIVOT = AMAX1(PIVOT, 1.E-20)
      PIVOT = 1.0 / PIVOT
      DO 1 L=1,NVAR
      CALL OVERFL(JOV)
      IF (JOV-2) 7,1,7
    1 A(ICOL, L) = A(ICOL, L)*PIVOT
C
C     REDUCE NON-PIVOT ROWS
C
      DO 3 L1=1,NVAR
      IF(L1 .EQ. ICOL) GO TO 3
      T = A(L1, ICOL)
      A(L1, ICOL) = 0.
      DO 2 L=1,NVAR
      CALL OVERFL(JOV)
      IF (JOV-2) 7,2,7
    2 A(L1, L) = A(L1, L) - A(ICOL, L) * T
    3 CONTINUE
      IF ( NB.EQ.0 )   GO TO 6
      DO 4  I=1,NVAR
      C(I) = 0.
      DO 4  K=1,NVAR
    4 C(I) = C(I) + A(I,K) * B(K)
      DO 5   I=1,NVAR
    5 B(I) = C(I)
    6 CONTINUE
      RETURN
    7 RETURN 1
      END
      SUBROUTINE TFORM(Z,NLOG,POW,ADD,NDR)
      DIMENSION Z(1)
      GO TO (10,20,30),NLOG
   10 DO 5 J = 1,NDR
    5 Z(J) = Z(J)+ADD
      RETURN
   20 DO 15 J=1,NDR
      Z(J)=Z(J)+ADD
   15 Z(J)=ALOG(Z(J))
      RETURN
   30 DO 25 J=1,NDR
      Z(J)=Z(J)+ADD
```

```
   25 Z(J)=Z(J)**POW
      RETURN
      END
      SUBROUTINE ENDS(STC,STCI,IB,IE,K)
C --- X11 TREND CYCLE END WEIGHTS.
      DIMENSION W5H(8),W9H(32),W13H(72),W23H(242),W23H2(44),STC(1),
     1          STCI(1),W23H1(198)
      EQUIVALENCE (W23H2(1),W23H(199)),(W23H1(1),W23H(1))
C --- 5-TERM HENDERSON
      DATA W5H /
     1 .670, .403,-.073, .0  ,
     2 .257, .522, .294,-.073/
C --- 9-TERM HENDERSON WEIGHTS.
      DATA  W9H /
     1 .581, .424, .185,-.034,-.156, .0, .0, .0,
     2 .298, .354, .282, .126,-.011,-.049, .0  , .0  ,
     3 .086, .242, .315, .259, .120, .0  ,-.022, .0  ,
     4-.029, .102, .255, .324, .263, .120,-.004,-.031/
C --- 13-TERM HENDERSON WEIGHTS.
      DATA  W13H /
     1 .421, .353, .244, .120, .012,-.058,-.092,.0,.0,.0,.0,.0,
     2 .279, .292, .254, .174, .080, .002,-.038,-.043,.0,.0,.0,.0,
     3 .148, .216, .241, .216, .149, .068, .003,-.025,-.016,.0,.0,.0,
     4 .046, .131, .201, .230, .208, .145, .066, .004,-.022,-.009,.0,.0,
     5-.018, .050, .136, .205, .235, .210, .145, .067, .003,-.022,-.011,
     6 .0  ,
     7-.034,-.006, .061, .144, .212, .238, .213, .147, .066, .001,-.025,
     8-.017/
C --- 23-TERM HENDERSON WEIGHTS.
      DATA W23H1 /
     1 .288, .263, .227, .182, .133, .084, .039, .002,-.028,-.049,-.064,
     2-.077, .0  , .0  , .0  , .0  , .0  , .0  , .0  , .0  , .0  ,.0,
     3 .224, .219, .203, .176, .141, .101, .061, .025,-.004,-.024,-.035,
     4-.041,-.046, .0  , .0  , .0  , .0  , .0  , .0  , .0  , .0  ,.0,
     5 .166, .176, .177, .166, .146, .116, .082, .049, .018,-.005,-.019,
     6-.025,-.025,-.022, .0  , .0  , .0  , .0  , .0  , .0  ,.0,.0,
     7 .112, .134, .150, .154, .147, .129, .103, .073, .042, .015,-.004,
     8-.015,-.018,-.014,-.008, .0  , .0  , .0  , .0  , .0  , .0  ,.0,
     9 .066, .095, .119, .136, .142, .137, .121, .098, .068, .040, .015,
     1-.003,-.012,-.013,-.008,-.001, .0  , .0  , .0  , .0  , .0  ,.0,
     2 .027, .059, .088, .114, .131, .139, .134, .119, .095, .067, .039,
     3 .015,-.002,-.011,-.011,-.006, .003, .0  , .0  , .0  ,.0,.0,
     4 .001, .027, .059, .089, .114, .132, .138, .134, .118, .096, .068,
     5 .039, .015,-.003,-.011,-.012,-.006, .002, .0  , .0  , .0  ,.0,
     6-.015, .005, .031, .060, .090, .116, .133, .140, .135, .120, .096,
     7 .068, .039, .015,-.003,-.011,-.013,-.007, .001, .0  , .0  ,.0,
     8-.021,-.010, .008, .034, .064, .094, .118, .136, .140, .137, .120,
     9 .097, .068, .039, .014,-.003,-.013,-.013,-.007,-.002, .0, .0/
      DATA W23H2 /
     1-.019,-.017,-.007, .011, .037, .067, .095, .120, .137, .143, .138,
     2 .122, .097, .069, .040, .014,-.005,-.014,-.015,-.010,-.003,.0,
     3-.011,-.016,-.015,-.005, .013, .039, .068, .097, .122, .138, .144,
     4 .138, .122, .097, .068, .039, .013,-.005,-.015,-.016,-.011,-.004/
      M = K-1
      L = M/2
```

```
      IF (K.EQ.7) RETURN
      DO 5 I = 1,L
      STC(IB+I-1) = 0.0
      STC(IE-I+1) = 0.0
      N = M*(I-1)
      DO 5 J = 1,M
      N = N+1
      IF (K.NE.5) GO TO 1
      WT = W5H(N)
      GO TO 4
    1 IF (K.NE.9) GO TO 2
      WT = W9H(N)
      GO TO 4
    2 IF (K.NE.13) GO TO 3
      WT = W13H(N)
      GO TO 4
    3 WT = W23H(N)
    4 STC(IB+I-1) = WT*STCI(IB+J-1)+STC(IB+I-1)
    5 STC(IE-I+1) = WT*STCI(IE-J+1)+STC(IE-I+1)
      RETURN
      END
      SUBROUTINE F3GEN(NW,NY,MQ)
      REAL*8 MQ,SPAN(4)
C --- THIS SUBROUTINE PRINTS THE F3 TABLE.
      COMMON /WORK2/ QU(11),NN,QUAL,IFAIL
      DATA SPAN/6HTHREE ,6HMONTHS,6HONE QU,6HARTER /
      K = 1
      IF (NY.EQ.4) K = 3
      L = K+1
      WRITE(NW,10)
   10 FORMAT(1H0,6X,'ALL THE MEASURES BELOW ARE IN THE RANGE FROM 0 TO 3
     1 WITH AN ACCEPTANCE REGION FROM 0 TO 1.')
      WRITE(NW,20) SPAN(K),SPAN(L),QU(1)
   20 FORMAT(1H0,3X,'1. THE RELATIVE CONTRIBUTION OF THE IRREGULAR OVER
     1',2A6,' SPAN (FROM TABLE F 2.B).',T99,'M1  = ',F5.3,//)
      WRITE(NW,30) QU(2)
   30 FORMAT(1H0,3X,'2. THE RELATIVE CONTRIBUTION OF THE IRREGULAR COMP
     1NENT TO THE STATIONARY PORTION OF',T99,'M2  = ',F5.3,/,
     2 7X,'THE VARIANCE (FROM TABLE F 2.F).',/)
      WRITE(NW,40) MQ,MQ,QU(3),MQ,MQ
   40 FORMAT(1H0,3X,'3. THE AMOUNT OF ',A7,' TO ',A7,' CHANGE IN THE IRR
     1EGULAR COMPONENT AS COMPARED TO THE',T99,'M3  = ',F5.3,/,
     2 7X,'AMOUNT OF ',A7,' TO ',A7,' CHANGE IN THE TREND-CYCLE (FROM TA
     3BLE F2.H).',/)
      WRITE(NW,50) QU(4)
   50 FORMAT(1H0,3X'4. THE AMOUNT OF AUTOCORRELATION IN THE IRREGULAR AS
     1 DESCRIBED BY THE AVERAGE DURATION',T99,'M4  = ',F5.3,/,
     2 7X,'OF RUN (TABLE F 2.D).',/)
      WRITE(NW,60) MQ,QU(5)
   60 FORMAT(1H0,3X'5. THE NUMBER OF ',A7,'S IT TAKES THE CHANGE IN THE
     1TREND-CYCLE TO SURPASS THE AMOUNT',T99,'M5  = ',F5.3,/,
     2 6X,' OF CHANGE IN THE IRREGULAR (FROM TABLE F 2.E).',/)
      WRITE(NW,70) QU(6)
   70 FORMAT(1H0,3X,'6. THE AMOUNT OF YEAR TO YEAR CHANGE IN THE IRREGUL
     1AR AS COMPARED TO THE AMOUNT OF YEAR',T99,'M6  = ',F5.3,/,
```

```
     2 7X,'TO YEAR CHANGE IN THE SEASONAL (FROM TABLE F 2.H).',/)
      WRITE(NW,80) QU(7)
   80 FORMAT(1H0,3X,'7. THE AMOUNT OF MOVING SEASONALITY PRESENT RELATIV
     1E TO THE AMOUNT OF STABLE',T99,'M7  = ',F5.3,/,
     2 7X,'SEASONALITY (FROM TABLE F 2.I).',/)
      IF (NN.EQ.7) GO TO 1
      WRITE(NW,90) QU(8)
   90 FORMAT(1H0,3X,'8. THE SIZE OF THE FLUCTUATIONS IN THE SEASONAL COM
     1PONENT THROUGHOUT THE WHOLE SERIES.',T99,'M8  = ',F5.3,//)
      WRITE(NW,100) QU(9)
  100 FORMAT(1H0,3X,'9. THE AVERAGE LINEAR MOVEMENT IN THE SEASONAL COMP
     1ONENT THROUGHOUT THE WHOLE SERIES.',T99,'M9  = ',F5.3,//)
      WRITE(NW,110) QU(10)
  110 FORMAT(1H0,2X,'10. SAME AS 8, CALCULATED FOR RECENT YEARS ONLY.',
     1 T99,'M10 = ',F5.3,//)
      WRITE(NW,120) QU(11)
  120 FORMAT(1H0,2X,'11. SAME AS 9, CALCULATED FOR RECENT YEARS ONLY.',
     1 T99,'M11 = ',F5.3)
    1 CONTINUE
      IF (QUAL-1.0) 2,3,3
    2 WRITE(NW,130) QUAL
  130 FORMAT(1H0,/,42X,'*** ACCEPTED *** AT THE LEVEL  ',F5.2)
      GO TO 4
    3 WRITE(NW,140) QUAL
  140 FORMAT(1H0,/,42X,'*** REJECTED *** AT THE LEVEL ',F5.2)
    4 IF (IFAIL.GT.0) WRITE(NW,150) IFAIL
  150 FORMAT(/,42X,'*** CHECK THE ',I2,' ABOVE MEASURES WHICH FAILED.')
      RETURN
      END
      SUBROUTINE COMBFT(KPROPT)
C --- THIS SUBROUTINUE PRODUCES THE COMBINED TEST FOR THE PRESENCE OF
C --- IDENTIFIABLE SEASONALITY.
C
      COMMON /UNITS/ MT,MT2,NW,MP,NG,NF
      COMMON /TESTS/ F(5),P(5),TEST1,TEST2,IQFAIL
      IQFAIL = 1
      TEST1 = 9.0
      IF (F(1)*9.0-7.0) 7,6,6
    6 TEST1 = 7.0/F(1)
    7 TEST2 = (3.0*F(2))/F(1)
      IF (TEST2.GT.9.0) TEST2 = 9.0
      IF (KPROPT.LE.3) WRITE(NW,10)
   10 FORMAT(//,20X,'COMBINED TEST FOR THE PRESENCE OF IDENTIFIABLE SEAS
     1ONALITY')
      IF (P(1).GE.0.1) GO TO 4
      IF (P(2).GT.5.0) GO TO 1
      TEST = (TEST1+TEST2)/2.0
      IF (TEST.GE.1.0) GO TO 4
    1 IF (KPROPT.GT.3) RETURN
      IF (TEST1.GE.1.0) GO TO 2
      IF (P(5).GT.0.1) GO TO 2
      IF (TEST2.LT.1.0) GO TO 3
    2 WRITE(NW,20)
   20 FORMAT(//,30X,'IDENTIFIABLE SEASONALITY PROBABLY NOT PRESENT')
      RETURN
```

```
    3 WRITE(NW,30)
   30 FORMAT(//,30X,'IDENTIFIABLE SEASONALITY PRESENT')
      RETURN
    4 IF (KPROPT.GT.3) GO TO 5
      WRITE(NW,40)
   40 FORMAT(//,30X,'IDENTIFIABLE SEASONALITY NOT PRESENT')
    5 IQFAIL = 2
      RETURN
      END
      SUBROUTINE AGR1(IB,IE,IAGR)
C --- THIS SUBROUTINE IS FOR PRODUCING SEASONALLY ADJUSTED COMPOSITE
C --- SERIES.
C
      REAL*8 SERNO,TITLE
      COMMON /ROTE / TITLE(10),SERNO,KPAGE,IRUN,NEWPG
      COMMON /UNITS/ MT,MT2,MT1,MP,NG,NFORM
      COMMON /INPT1/ SERIES(372),ORIG(372)
      COMMON /AGREG/ O(372),CI(372),OMOD(372),ITEST(5),N,KFORC,TEM(372)
      COMMON /OPT9 / LFDA,LYR,LSTMO,LSTYR,LLDA,KFULSM,KPROPT,KCHOPT,
     1               KEXOPT,NY,IFORC,LLDAF,LLAF,LFD1,LLD1,NSOPT,IDEC,IAG
      IF (IAGR.NE.0) GO TO 2
C --- INITIALIZATION
      N = 0
      KFORC = 0
      IAGR = 1
      DO 1 I = 1,360
      O(I) = 0.0
      OMOD(I) = 0.0
    1 CI(I) = 0.0
      RETURN
C --- PRODUCE THE DIRECT SEASONALLY ADJUSTED COMPOSITE SERIES.
    2 IF (IAGR.LT.0) GO TO 5
      IAGR = 3
      DO 3 I = IB,IE
      SERIES(I) = O(I)
    3 CONTINUE
      IF (ITEST(2).NE.LFDA) GO TO 4
      IF (ITEST(3).NE.LSTMO) GO TO 4
      IF (ITEST(4).NE.LYR) GO TO 4
      IF (ITEST(5).NE.LSTYR) GO TO 4
      RETURN
    4 WRITE(MT2,1000) SERNO
 1000 FORMAT(1H0,'ERROR - SERIES ',A8,' HAS NON OVERLAPPING TIME SPAN. A
     1GGREGATION NOT COMPUTED.')
    5 IAGR = 0
      N = 0
      KFORC = 0
      RETURN
      END
      SUBROUTINE AGR2(IAGR)
C --- THIS SUBROUTINE IS USED TO ACCUMULATE THE INDIRECT SEASONALLY
C --- ADJUSTED SERIES.
      REAL*8 SERNO,TITLE
      COMMON /INPT1/ SERIES(372),ORIG(372)
      COMMON /ROTE / TITLE(10),SERNO,KPAGE,IRUN,NEWPG
```

```
      COMMON /UNITS/ MT,MT2,MT1,MP,NG,NFORM
      COMMON /WORK2/ CIBAR(12),CISD(12),TMP(12),TM1(72)
      COMMON /AGREG/ O(372),CI(372),OMOD(372),ITEST(5),N,KFORC,TEM(372)
      COMMON /OPT5 / MULADD,IYRT,MA,IWT,W
      COMMON /OPT9 / LFDA,LYR,LSTMO,LSTYR,LLDA,KFULSM,KPROPT,KCHOPT,
     1               KEXOPT,NY,IFORC,LLDAF,LLAF,LFD1,LLD1,NSOPT,IDEC,IAG
     2               ,JAREX,KEBACK
      COMMON /MQ4  / STC(372),STCI(372)
      COMMON /MQ7  / STI(372)
      COMMON /MQ8  / STOME(372),UNKN(372)
      COMMON /KCJH /LENGTH,IFTNY
      DIMENSION TEMP(372),DI(24)
      EQUIVALENCE (O(1),TEMP(1))
      DATA DASH/4H----/
      IF (IAGR-2) 1,2,2
    1 IAGR = 2
      ITEST(1) = NY
      ITEST(2) = LFDA
      ITEST(3) = LSTMO
      ITEST(4) = LYR
      IF (IFORC.NE.0.AND.KEBACK.EQ.0.AND.LENGTH.LE.IFTNY) ITEST(4)=ITEST
     1(4)+1
      ITEST(5) = LSTYR
C --- CHECK TO SEE IF THE SERIES ALL HAVE THE SAME SPAN.
    2 IF (IAGR.EQ.4) GO TO 4
      IF (ITEST(1).NE.NY) GO TO 100
      IF (ITEST(2).NE.LFDA) GO TO 100
      IF (ITEST(3).NE.LSTMO) GO TO 100
      IF (IFORC.NE.0.AND.KEBACK.EQ.0.AND.LENGTH.LE.IFTNY) LYR=LYR+1
      IF (ITEST(4).NE.LYR) GO TO 100
      IF (IFORC.NE.0.AND.KEBACK.EQ.0.AND.LENGTH.LE.IFTNY) LYR=LYR+1
      IF (ITEST(5).NE.LSTYR) GO TO 100
C --- ACCUMULATE THE INDIRECT SEASONALLY ADJUSTED SERIES , THE DIRECT
C --- ORIGINAL SERIES, AND THE INDIRECT MODIFIED ORIGINAL SERIES.
      IF (IFORC.EQ.0.OR.KEBACK.EQ.1.OR.LENGTH.GT.IFTNY) GO TO 50
      DO 49 I=1,LLDA
      SERIES(I)=SERIES(I+NY)
      STOME(I)=STOME(I+NY)
      STCI(I)=STCI(I+NY)
   49 CONTINUE
   50 CALL AGR(SERIES,O,IAG,LFDA,LLDA,W)
      CALL AGR(STOME,OMOD,IAG,LFDA,LLDAF,W)
      CALL AGR(STCI,CI,IAG,LFDA,LLDAF,W)
      IF (IFORC.GT.0) KFORC = KFORC+1
      N = N+1
      RETURN
C --- END OF THE INDIRECT ADJUSTMENT.
    4 IAGR = 0
      N = 0
      KFORC = 0
      KFDA = LLD1-NY*3+1
C --- COMPUTE COMPARISON STATISTICS FOR THE TWO METHODS.
      CALL AGGMEA(TEMP,TEM,A1,A2,A3,A4,LFD1,LLD1,MULADD)
      DI(1)=A1
      DI(7)=A2
```

```
      DI(13)=A3                                                         00063060
      DI(19)=A4                                                         00063070
      CALL AGGMEA(TEMP,TEM,B1,B2,B3,B4,KFDA,LLD1,MULADD)                00063080
      DI(2)=B1                                                          00063090
      DI(8)=B2                                                          00063100
      DI(14)=B3                                                         00063110
      DI(20)=B4                                                         00063120
      CALL AGGMEA(STCI,STC,C1,C2,C3,C4,LFD1,LLD1,MULADD)                00063130
      DI(3)=C1                                                          00063140
      DI(9)=C2                                                          00063150
      DI(15)=C3                                                         00063160
      DI(21)=C4                                                         00063170
      CALL AGGMEA(STCI,STC,D1,D2,D3,D4,KFDA,LLD1,MULADD)                00063180
      DI(4)=D1                                                          00063190
      DI(10)=D2                                                         00063200
      DI(16)=D3                                                         00063210
      DI(22)=D4                                                         00063220
      DO 9 I=1,19,6                                                     00063230
      DI(I+4)=(DI(I)-DI(I+2))*100./DI(I)                                00063240
      DI(I+5)=(DI(I+1)-DI(I+3))*100/DI(I+1)                             00063250
    9 CONTINUE                                                          00063260
      WRITE(NG,10)                                                      00063270
   10 FORMAT(1H0)                                                       00063280
      WRITE(NG,20) (DASH,I=1,28)                                        00063290
   20 FORMAT(8X,28A4)                                                   00063300
      WRITE(NG,40) (DASH,I=1,16)                                        00063310
   40 FORMAT(/,38X,'MEASURES OF ROUGHNESS R1 AND R2',                   00063320
     1' FOR SEASONALLY ADJUSTED SERIES',/,37X,16A4//)                   00063330
      WRITE (NG,29)                                                     00063340
   29 FORMAT(48X,'DIRECT',22X,'INDIRECT',17X,'PERCENTAGE CHANGE')       00063350
      WRITE (NG,291)                                                    00063360
  291 FORMAT(48X,'------',22X,'--------',17X,'----------------',/)      00063370
      WRITE(NG,30) DI                                                   00063380
   30 FORMAT(/,51X,'  LAST THREE',17X,'  LAST THREE',17X,'  LAST THREE'/00063390
     1,39X,' FULL SERIES',5X,'YEARS',7X,' FULL SERIES',5X,'YEARS',7X,' F00063400
     2ULL SERIES',5X,'YEARS',//,8X,                                     00063410
     3' R1-MEAN SQUARE ERROR      ',2(5X,2F12.3),5X,2(F11.3,'%'),//,8X,  00063420
     4' R1-ROOT MEAN SQUARE ERROR',2(5X,2F12.3),5X,2(F11.3,'%'),//,8X,  00063430
     5' R2-MEAN SQUARE ERROR      ',2(5X,2F12.3),5X,2(F11.3,'%'),//,8X,  00063440
     6' R2-ROOT MEAN SQUARE ERROR',2(5X,2F12.3),5X,2(F11.3,'%'))        00063450
      WRITE(NG,20) (DASH,I=1,28)                                        00063460
      WRITE(NG,201)                                                     00063470
  201 FORMAT (/8X,'POSITIVE PERCENTAGE CHANGES INDICATE THAT THE INDIRE'00063480
     1,'CT SEASONALLY ADJUSTED',/,8X,'COMPOSITE IS SMOOTHER THAN THE DI'00063490
     2,'RECT SEASONALLY ADJUSTED COMPOSITE.')                           00063500
      WRITE(MT1,101)                                                    00063510
  101 FORMAT (1H1)                                                      00063520
      WRITE(MT1,20) (DASH,I=1,28)                                       00063530
      WRITE(MT1,40) (DASH,I=1,16)                                       00063540
      WRITE (MT1,29)                                                    00063550
      WRITE (MT1,291)                                                   00063560
      WRITE(MT1,30) DI                                                  00063570
      WRITE(MT1,20) (DASH,I=1,28)                                       00063580
      WRITE (MT1,201)                                                   00063590
      RETURN                                                            00063600
```

```
  100 WRITE(MT2,2000) SERNO                                          00063610
 2000 FORMAT(1H0,'ERROR - SERIES ',A8,' HAS NON-OVERLAPPING TIME SPAN. A00063620
     1GGREGATION NOT COMPUTED.')                                     00063630
      IAGR = -1                                                      00063640
      RETURN                                                         00063650
      END                                                            00063660
      SUBROUTINE AGR3(IAGR)                                          00063670
C --- THIS ROUTINE PRODUCES THE INDIRECT SEASONALLY ADJUSTED SERIES.  00063680
      REAL*8 SERNO,TITLE,KSER                                        00063690
      COMMON /ROTE / TITLE(10),SERNO,KPAGE,IRUN,NEWPG                00063700
      COMMON /UNITS/ MT,MT2,MT1,MP,NG,NFORM                          00063710
      COMMON /PRIOR/ KSER,KFORM(20),KFMT                             00063720
      COMMON /TRADE/ DWT(7),D(7),SIGM,LOPT,KDWOPT,LCYR,LAYR,LFDC,LFDR 00063730
      COMMON /WORK / TEMP3(372)                                      00063740
      COMMON /WORK3/ STSIE(372)                                      00063750
      COMMON /INPT1/ SERIES(372),ORIG(372)                           00063760
      COMMON /AGREG/ O(372),CI(372),OMOD(372),ITEST(5),N,KFORC,TEM(372) 00063770
      COMMON /OPT5 / MULADD,IYRT,MA,IWT                              00063780
      COMMON /OPT8 /RATIC,KTCOPT                                     00063781
      COMMON /OPT9 / LFDA,LYR,LSTMO,LSTYR,LLDA,KFULSM,KPROPT,KCHOPT, 00063790
     1               KEXOPT,NY,IFORC,LLDAF,LLAF,LFD1,LLD1,NSOPT,IDEC,IAG00063800
     2               ,JAREX,KEBACK                                   00063810
      COMMON /MQ1  / STS(372),STSI(372)                              00063820
      COMMON /MQ2  / KPART,KSWV,KSECT,KPRNT,MCD                      00063830
      COMMON /MQ4  / STC(372),STCI(372)                              00063840
      COMMON /MQ5  / STMCD(372),STIME(372)                           00063850
      COMMON /MQ6  / STCIME(395)                                     00063860
      COMMON /MQ7  / STI(372)                                        00063870
      COMMON /MQ8  / STOME(372),UNKN(372)                            00063880
      COMMON /MQ10 / STCI2(372)                                      00063890
      COMMON /LARGE/ BIG                                             00063900
      COMMON /KCJH /LENGTH,IFTNY                                     00063910
      COMMON /KCSER/CKHS(372)                                        00063915
      DIMENSION STEX(372),TEMP(372)                                  00063920
      EQUIVALENCE (O(1),TEMP(1)),(STEX(1),OMOD(1))                   00063930
C --- WRITE THE TITLE PAGE.                                           00063940
      KPAGE = 0                                                      00063950
      WRITE(MT1,1000)                                                00063960
 1000 FORMAT(1H1,///,61X,'X-11 ARIMA',//,                            00063970
     142X,'INDIRECT SEASONAL ADJUSTMENT OF COMPOSITE SERIES',//,     00063980
     251X,'STATISTICS CANADA,  SEPTEMBER,1979'///)                   00063990
      WRITE(MT1,1001) TITLE,SERNO                                    00064000
 1001 FORMAT(10X,14HSERIES TITLE- ,10A8,10X,11HSERIES NO. ,A8)       00064010
 1002 FORMAT(//,90X,'DATE ',2A1,1H/,2A1,1H/,2A1)                     00064020
      IF (IFORC.NE.0.AND.KEBACK.EQ.0.AND.LENGTH.LE.IFTNY) LYR=LYR+1  00064030
      WRITE(MT1,1003) LFDA,LYR,LSTMO,LSTYR                           00064040
      IF (IFORC.NE.0.AND.KEBACK.EQ.0.AND.LENGTH.LE.IFTNY) LYR=LYR-1  00064050
 1003 FORMAT(/,42X,16HPERIOD COVERED- ,I2,1H/,I2,4H TO ,I2,1H/,I2,1H.) 00064060
      WRITE(MT1,1004) N                                              00064070
 1004 FORMAT(1H0,//,36X,'THERE ARE ',I5,' COMPONENTS IN THE COMPOSITE.')00064080
C --- INDIRECT SEASONAL ADJUSTMENT.                                  00064090
        DO 888 I=LFD1,LLD1                                           00064092
           TEMP(I)=STCI(I)                                           00064094
  888      STCI(I)=CKHS(I)                                           00064096
      IAGR = 4                                                       00064100
```

```
C --- APPLY A 13- OR 5-TERM HENDERSON TO THE CI                          00064101
      KTCOPT=3                                                            00064102
      CALL VTC(KPART)                                                     00064103
C --- SAVE DIRECT  SEASONALLY ADJUSTED SERIES.                            00064120
      NYY=0                                                               00064130
      IF (IFORC.NE.0.AND.KEBACK.EQ.0.AND.LENGTH.LE.IFTNY) NYY=NY          00064140
      DO 1 I = LFD1,LLD1                                                  00064150
      TEM(I)=STC(I)                                                       00064170
      STCI(I) = CI(I-NYY)                                                 00064180
    1 STOME(I) = OMOD(I-NYY)                                              00064190
      KFMT = 0                                                            00064200
      KPART = 4                                                           00064210
      KDWOPT = 0                                                          00064220
      KSWV = 0                                                            00064230
      IF (MA.EQ.2) MA = 0                                                 00064240
C --- CALCULATE THE INDIRECT SEASONAL FACTORS.                            00064250
          KCLDAF=LLDAF                                                    00064260
          IF (IFORC .NE. 0) KCLDAF=LLDAF-NY                               00064270
          KCLDA=KCLDAF+NY                                                 00064280
      CALL DIVSUB(STS,SERIES,STCI,LFD1,KCLDAF)                            00064290
C --- FORECAST THE SEASONAL FACTORS ONE YEAR AHEAD.                        00064300
      CALL FORCST(STS,0,KCLDAF,KCLDA,NY,1,0.5,1.0)                        00064310
C --- CALCULATE EXTREMES FOR MODIFIED SERIES.                              00064320
      CALL DIVSUB(STEX,SERIES,STOME,LFD1,LLD1)                            00064330
C --- CALCULATE THE MODIFIED SEASONALLY ADJUSTED SERIES E2.                00064340
      CALL DIVSUB(STCIME,STCI,STEX,LFD1,LLD1)                             00064350
C --- MODIFY EXTREMES IN THE SEASONALLY ADJUSTED SERIES FOR CALCULATING 00064360
C --- THE FINAL TREND-CYCLE.                                              00064370
      DO 2 I = LFD1,LLD1                                                  00064380
    2 STCI(I) = STCIME(I)                                                 00064390
C --- APPLY THE VARIABLE TREND-CYCLE ROUTINE TO OBTAIN THE FINAL TREND 00064400
C --- CYCLE D12, BUT INSIST ON A 13- OR 5-TERM HENDERSON                  00064410
      KTCOPT=3                                                            00064411
      CALL VTC(KPART)                                                     00064420
      DO 3 I = LFD1,LLD1                                                  00064430
    3 STCI(I) = CI(I-NYY)                                                 00064440
C --- DIVIDE THE FINAL TREND-CYCLE INTO THE FINAL SEASONALLY ADJUSTED 00064450
C --- SERIES TO GET THE FINAL IRREGULAR D13.                              00064460
      CALL DIVSUB(STI,STCI,STC,LFD1,LLD1)                                 00064470
C --- OBTAIN THE MODIFIED IRREGULAR BY DIVIDING THE FINAL IRREGULAR BY 00064480
C --- THE EXTEMES.                                                        00064490
      CALL DIVSUB(STIME,STI,STEX,LFD1,LLD1)                               00064500
C --- OBTAIN THE UNMODIFIED SI BY DIVIDING THE ORIGINAL SERIES BY THE 00064510
C --- FINAL TREND-CYCLE.                                                  00064520
      CALL DIVSUB(STSIE,SERIES,STC,LFD1,LLD1)                             00064530
C --- OBTAIN THE MODIFIED SI BY DIVIDING THE UNMODIFIED SI BY THE 00064540
C --- EXTREMES.                                                           00064550
      CALL DIVSUB(STSI,STSIE,STEX,LFD1,LLD1)                              00064560
      IF (KPPOPT.GT.3) GO TO 4                                            00064570
C --- WRITE THE UNMODIFIED SI RATIOS D8.                                  00064580
      CALL TABLE(STSIE,LFD1,LLD1,8,1,1,0)                                 00064590
C --- PERFORM ANALYSIS OF VARIANCE ON THE UNMODIFIED SI RATIOS.           00064600
      CALL TABLE(0,0,0,8,2,0,0)                                           00064610
    4 CALL FTEST(STSIE,LFD1,LLD1,NY,0,KPROPT)                             00064620
      CALL KWTEST(STSIE,LFD1,LLD1,NY,KPROPT)                              00064630
```

```
      CALL DIVSUB(STSIE,SERIES,STC,LFD1,LLD1)
C --- PERFORM F-TEST FOR MOVING SEASONALITY.
      CALL MSTEST(STSIE,LFD1,LLD1,KPROPT,NY)
C --- PERFORM TEST FOR THE PRESENCE OF IDENTIFIABLE SEASONALITY.
      CALL COMBFT(KPROPT)
C --- CALCULATE I/S RATIOS.
      CALL VSFA(LFD1,LLD1,NY)
C --- IDENTIFY SI RATIOS THAT ARE MODIFIED.
      EBAR = 1-MULADD
      IF (KPROPT.GT.3) GO TO 7
      DO 6 I = LFD1,LLD1
      TMPE = STEX(I)-EBAR
      IF (TMPE.EQ.0.0) GO TO 5
      TMPE = TMPE/STSIE(I)
      IF (TMPE.LT.0.0001) GO TO 5
      TEMP3(I) = STSI(I)
      GO TO 6
    5 TEMP3(I) = BIG
    6 CONTINUE
C --- WRITE THE FINAL REPLACEMENT VALUES FOR THE SI D9.
      CALL TABLE(TEMP3,LFD1,LLD1,9,1,5,0)
C --- WRITE THE INDIRECT SEASONAL FACTORS.
    7 CALL TABLE(STS,LFD1,LLD1,10,1,1,0)
C --- WRITE THE INDIRECT SEASONALLY ADJUSTED SERIES
      CALL TABLE(STCI,LFD1,LLD1,11,1,2,0)
      IF (KPROPT.GE.3) GO TO 8
C --- DO A TEST FOR RESIDUAL SEASONALITY.
      CALL FTEST(STCI,LFD1,LLD1,NY,1,KPROPT)
    8 IF (IYRT.EQ.0) GO TO 9
C --- IF OPTION SELECTED ADJUST YEARLY TOTALS OF D11 TO EQUAL THE YEARLY
C --- TOTALS OF SERIES.
      CALL QMAP(SERIES,STCI,STCI2,LFD1,LLD1,NY,IB,IE)
C --- WRITE THE SEASONALLY ADJUSTED SERIES WITH REVISED YEARLY TOTALS.
      CALL TABLE(STCI2,IB,IE,11,2,2,0)
      IF (KPROPT.GE.3) GO TO 10
C --- DO TEST FOR RESIDUAL SEASONALITY.
      CALL FTEST(STCI2,LFD1,LLD1,NY,1,KPROPT)
    9 IF (KPROPT.GE.3) GO TO 10
C --- WRITE THE FINAL TREND CYCLE D12.
      CALL TABLE(STC,LFD1,LLD1,12,1,2,0)
C --- WRITE THE FINAL IRREGULAR D13.
      CALL TABLE(STI,LFD1,LLD1,13,1,3,0)
   10 KPART = 5
      RETURN
      END
      FUNCTION FIS(CS,N)
C --- THIS FUNCTION ADJUSTS THE I/S RATIO FOR THE NUMBER OF YEARS
C
      DIMENSION C(8)
      DATA C/1.00000,1.02584,1.01779,1.01383,1.00000,3.00000,1.55291,
     1       1.30095/
      IF (N-6) 1,2,2
    1 FIS = C(N-1)
      CS = C(N+3)
      RETURN
```

```
    2 CS = N*1.732051/(8.485281+(N-6)*1.732051)                          00065230
      FIS = N*12.247449/(73.239334+(N-6)*12.247449)                      00065240
      RETURN                                                             00065250
      END                                                                00065260
      SUBROUTINE WEIGHT(A,B,I1,I2,MQ)                                    00065270
C   THIS SUBROUTINE PRODUCES WEIGHTS FOR THE CENTERED 24-TERM (MONTHLY)  00065280
C   MOVING AVERAGE (CENTERED 8-TERM QUARTERLY MOVING AVERAGE)            00065290
C   FOR THE PRELIMINARY ESTIMATION OF THE TREND-CYCLE.                   00065300
      DIMENSION CENT(25),A(372),B(372),END(6,24)                         00065310
      DIMENSION QCENT(9),QEND(2,8),QEND4(8),QEND3(8)                     00065320
      DIMENSION END12(24),END11(24),END10(24),END9(24),END8(24),END7(24)00065330
      DATA CENT/-0.0112773,-0.0273401,-0.0195570,-0.0053389,             00065340
     1          0.0113162,0.0274075,0.0416667,0.0559258,0.0720171,      00065350
     2          0.0886723,0.1028903,0.1106735,0.1058879,0.1106735,      00065360
     3          0.1028903,0.0886723,0.0720171,0.0559258,0.0416667,      00065370
     4          0.0274075,0.0113162,-0.0053389,-0.0195570,              00065380
     5          -0.0273401,-0.0112773/                                  00065390
      DATA END12/-0.0225546,-0.0234459,-0.0160151,-0.0026792,0.0121628, 00065400
     1           0.0279138,0.0413240,0.0564321,0.0728637,               00065410
     2          0.0913320,0.1064322,0.1145676,0.1058879,0.1067793,      00065420
     3           0.0993484,0.0860125,0.0711706,0.0554195,0.0420093,     00065430
     4           0.0269012,0.0104697,-0.0079986,-0.0230988,             00065440
     5           -0.0312343/                                            00065450
      DATA END11/-0.0106354,-0.0195678,-0.0208123,-0.0144215,           00065460
     1          -0.0026399,0.0121064,0.0272793,0.0432593,0.0595187,     00065470
     2          0.0811315,0.1018984,0.1178834,0.0939688,0.1029011,      00065480
     3          0.1041457,0.0977549,0.0859732,0.0712269,0.0560541,      00065490
     4          0.0400741,0.0238147,0.0022018,-0.0185651,               00065500
     5          -0.0345500/                                             00065510
      DATA END10/0.0019024,-0.0124004,-0.0214279,-0.0229499,-0.0159235, 00065520
     1          -0.0036926,0.0116272,0.0297519,0.0471940,               00065530
     2          0.0692546,0.0931847,0.1151461,0.0814309,0.0957337,      00065540
     3          0.1047612,0.1062832,0.0992569,0.0870260,0.0717061,      00065550
     4          0.0535814,0.0361393,0.0140787,-0.0098514,               00065560
     5          -0.0318128/                                             00065570
      DATA END9/0.0121814,-0.0035929,-0.0177541,-0.0263876,-0.0255157,  00065580
     1          -0.0176449,-0.0039735,0.0141749,0.0338072,              00065590
     2          0.0566876,0.0805875,0.1057635,0.0711520,0.0869263,      00065600
     3          0.1010874,0.1097209,0.1088491,0.1009782,0.0873068,      00065610
     4          0.0691584,0.0495261,0.0266457,0.0027458,                00065620
     5          -0.0224301/                                             00065630
      DATA END8/0.0181990,0.0047570,-0.0107207,-0.0239906,-0.0292988,   00065640
     1          -0.0274358,-0.0176746,-0.0019316,0.0172443,             00065650
     2          0.0407996,0.0654582,0.0895942,0.0651343,0.0785764,      00065660
     3          0.0940541,0.1073239,0.1126322,0.1107692,0.1010079,      00065670
     4          0.0852650,0.0660891,0.0425338,0.0178751,                00065680
     5          -0.0062608/                                             00065690
      DATA END7/0.0192206,0.0109225,-0.0021151,-0.0163613,              00065700
     1          -0.0263849,-0.0309178,-0.0270820,-0.0170193,            00065710
     2          -0.0009008,0.0188178,0.0438290,0.0696580,               00065720
     3          0.0641127,0.0724108,0.0854485,0.0996946,                00065730
     4          0.1097182,0.1142511,0.1104153,0.1003527,                00065740
     5          0.0842342,0.0645156,0.0395043,0.0136754/                00065750
      DATA QCENT/-0.0258462,-0.0208718,0.1250000,0.2708718,0.3016923,   00065760
     1          0.2708718,0.1250000,-0.0208718,-0.0258462/              00065770
```

```
      DATA QEND4/-0.0516923,0.0012821,0.1323846,0.2930256,
     1           0.3016923,0.2487179,0.1176154,-0.0430256/
      DATA QEND3/-0.0036410,-0.0579487,0.0079487,0.1786410,
     1           0.2536410,0.3079487,0.2420513,0.0713590/
      DO 9 I=I1,I2
      B(I)=0.
    9 CONTINUE
      IF (MQ.EQ.2) GO TO 100
      J1=I1+12
      J2=I2-12
      DO 10 I=J1,J2
      B(I)=CENT(13)*A(I)
      DO 10 J=1,12
      B(I)=B(I)+CENT(13-J)*A(I-J)+CENT(13+J)*A(I+J)
   10 CONTINUE
      DO 20 J=1,24
      END(1,J)=END12(J)
      END(2,J)=END11(J)
      END(3,J)=END10(J)
      END(4,J)=END9(J)
      END(5,J)=END8(J)
      END(6,J)=END7(J)
   20 CONTINUE
      DO 30 I=1,6
      L1=J1-I
      L2=J2+I
      DO 30 K=1,24
      M1=L1+I+12-K
      M2=L2-I-12+K
      B(L1)=B(L1)+END(I,K)*A(M1)
      B(L2)=B(L2)+END(I,K)*A(M2)
   30 CONTINUE
      RETURN
  100 J1=I1+4
      J2=I2-4
      DO 11 I=J1,J2
      B(I)=QCENT(5)*A(I)
      DO 11 J=1,4
      B(I)=B(I)+QCENT(5-J)*A(I-J)+QCENT(5+J)*A(I+J)
   11 CONTINUE
      DO 21 J=1,8
      QEND(1,J)=QEND4(J)
      QEND(2,J)=QEND3(J)
   21 CONTINUE
      DO 31 I=1,2
      L1=J1-I
      L2=J2+I
      DO 31 K=1,8
      M1=L1+I+4-K
      M2=L2-I-4+K
      B(L1)=B(L1)+QEND(I,K)*A(M1)
      B(L2)=B(L2)+QEND(I,K)*A(M2)
   31 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE AGR(A,B,IAG,J1,J2,WT)
C  THIS SUBROUTINE COMPOSITES THE COMPONENT SERIES
      DIMENSION A(372),B(372)
      IF (WT.EQ.0.) WT=1.
      DO 10 I=J1,J2
      IF (IAG.EQ.0) B(I)=B(I)+(A(I)*WT)
      IF (IAG.EQ.1) B(I)=B(I)-(A(I)*WT)
      IF (IAG.EQ.2) B(I)=B(I)*(A(I)*WT)
      IF (IAG.EQ.3) B(I)=B(I)/(A(I)*WT)
   10 CONTINUE
      RETURN
      END
      SUBROUTINE AGGMEA(A,B,W,X,Y,Z,I1,I2,MA)
C  THIS SUBROUTINE PRODUCES THE MEASURES OF ROUGHNESS R1 AND R2
C  (MEAN SQUARE ERROR AND ROOT MEAN SQUARE ERROR) OF DIRECT AND
C  INDIRECT SEASONAL ADJUSTMENT OF COMPOSITED SERIES
      DIMENSION A(372),B(372)
      K=I2-1
      W=0.
      X=0.
      Y=0.
      Z=0.
      DO 5 I=I1,K
      W=W+(A(I+1)-A(I))*(A(I+1)-A(I))
    5 CONTINUE
      DIV=I2-I1
      W=W/DIV
      X=SQRT(W)
      DO 6 I=I1,I2
      Y=Y+(A(I)-B(I))*(A(I)-B(I))
    6 CONTINUE
      DIV=DIV+1.
      Y=Y/DIV
      Z=SQRT(Y)
      RETURN
      END
```