연구자료
90-03-011

대량 통계처리의 표준화 방안

1 9 9 0

경제기획원 조사통계국
자 료 관 리 과

일 러 두 기

이 책자는 각종 통계조사의 조사표를 전산 입력한 후 자료처리 및 집계과정에서 수행하게 되는 오류검정 및 수정, 수준점검, 요약집계, 통계표의 작성 등 전산처리업무에 관한 관련자료를 정리 편집하여 작성한 것입니다.

이 책의 발간목적은 현재의 통계자료처리 및 집계방식이 많은 노력과 시간이 투입되는데 이를 효율적으로 개선하기 위한 방안으로 작업의 표준화 및 범용화 추진과 더불어 통계자료처리 및 집계 시스템을 구축하기 위한 이론적 배경을 제시하는데 있습니다.

통계조사 관련직원 및 전산처리 프로그래머들에게 배부하니 통계처리업무 개선에 많은 도움이 되길 바라며 이에 대한 의견제시와 문의사항은 자료관리과로 연락 바랍니다.

1990년 11월

자료관리과장 최 돈 철

목 차

Ⅰ. 서 론

1. 집계 System의 개요

1967년 조사통계국에서 국가 주요기본통계의 종합관리를 위하여 computer system을 도입함으로써 우리나라 전산 system의 서막을 열게 되었다.

통계 조사업무는 일반 행정통계와 달리 천문학적인 input data를 기초로 방대한 양의 결과표별 집계 항목과 cross table의 표측·표두계급수, 구성비·평균·분산 등의 계산을 통하여 조사결과의 처리과정에서 야기되는 error를 방지하기 위해서는 완전한 data check system을 설계하고 효율적인 집계처리에 필요한 program을 개발해야 하는바 이에 따르는 업무부담이 기하급수적으로 가중되는 실정에 있다.

통계집계는 처리형태에 따라 통계집계와 통계해석으로 구분되는데 통계집계는 통계의 집단으로서의 특성을 파악하기 위하여 집단의 개별 data를 통하여 집단으로서의 모습을 찾아내는 작업이며, 통계해석은 집계자료를 이용하여 비율·지수의 산출, 시계열분석, 상관관계 해석등 집단현상을 분석하는 작업으로서 집계와 해석은 상호 보완적인 표리관계를 형성하고 있으나 전산처리라 관점에서 보면 통계해석은 다양한 계산 routine과 program을 통하여 library화 하여야 하므로 특별한 computer지식이 없어도 접근이 가능한 반면 통계집계는 system화나 library화가 어렵기 때문에 필요한 통계표생산을 위하여 이용자 스스로 program을 작성하여야 하므로 상당한 정도의 전산지식이 필요한 부문이라는 점에 차이가 있다.

통계집계 결과의 library화와 system화가 어려운 이유는 이용자 중심으로 간결한 program과 제표논리가 전제되어야 하기 때문에 실무면에서 전적으로 program작성자의 판단에 의지하여야 하며 동일

한 통계집계일지라도 data량·통계표수·집계 항목수에 따라 처리시간이 현저하게 다르며 처리시간의 제약으로 획일적인 기법 적용이 불가능하기 때문에 사안마다 개별적인 집계 system을 구상하여야 하는 어려움이 따르고 통계집계용 범용 program이나 program package는 소량의 data처리나 간단한 통계표 작성에 적합하고 대량의 data처리나 통계표 작성에는 처리시간의 과다 소요로 사실상 적용이 불가능한 실정이다.

이와같은 system적인 정비와 범용 system의 적용이 곤란한 통계집계를 효율적으로 처리하기 위해서는 대량 data집계를 중심으로 program작성과정에서 노력과 시간의 절약이 절실히 요청되므로 집계과정에서 대량 생산방식인 기능별 집계system을 개발·적용할 필요가 있다.

2. 집계 system 개발의 필요성

집계 system을 설계함에 있어서 신속한 연산속도, 최소한의 memory점유율, 용이한 error의 재연산, 간편한 operation을 목적으로 하는 효율적인 program 작성과정에서 수록·debug·수정이 용이하도록 하되 수록은 program 작성지침을 표준화하고 풍부한 macro와 subroutine을 확보하여 일반적이고 대칭적인 program을 통하여 이용자의 이해와 debug·수정이 용이하도록 하여야 한다.

효율적인 program일수록 hardware의 write·debug에 많은 시간을 필요로 하는 상반된 관계에 있기 때문에 효과적인 수단을 선택하기 위해서는 적용업무의 성격을 명확히하고 그림 1-1과 같은 평가기준에 의하여 대량집계에 적합한 표준집계 system을 개발하여야 한다.

집계 싸이클	데이타량	평가기준우선도						
		1	2	3	4	5	6	7
1회한 처리	대량	연산처리	error 처리	operat-ion	debug	writ-ing	점유 memo-ry	수정
	소량	debug	writ-ing	operat-ion	error 처리	연산처리	점유 memo-ry	수정
반복 처리	대량	연산처리	opera-tion	error 처리	수정	debug	writ-ing	점유 memo-ry
	소량	debug	수정	operat-ion	연산처리	writ-ing	점유 memo-ry	error 처리

그림 1-1 programming 평가기준

3. 통계조사표의 제표순서

통계조사집계 system의 검토에 앞서서 그림 1-2와 같은 집계의 흐름에 대한 충분한 이해가 있어야 하기 때문에 통계조사집계 전반을 기획하는 자세로 신중하게 고찰하지 않으면 아니된다.

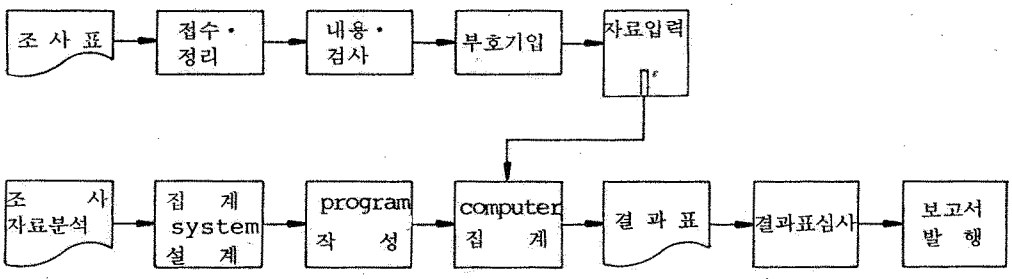


그림 1-2 통계조사의 집약절차 흐름도(제표순서)

가. 조사표 설계

통계조사는 조사표를 통하여 조사목적을 질문사항으로 구체화하고 질문사항의 표현과 배치, 그리고 집계과정에서 편이를 확보하는 문제를 해결해야 하기 때문에 조사표 설계과정에서 요구되는 다음과 같은 유의사항에 대하여 고찰하고저 한다.

(1) 질문사항의 설정

조사목적을 구체화하기 위해서는 질문사항을 설정해야 하는데 이 과정에서 통계학 지식은 물론 당면과제와 관련되는 경제·사회·인구·심리 등 광범한 학문적지식과 일상의 관습과 상식까지를 망라하여 동원할 수 있어야 한다.

(2) 질문사항의 표현과 배치

일반적으로 대부분의 응답자는 교육수준에 관계없이 근원적으로 통계에 생소하기 때문에 조사표의 기재(응답)에 당황하거나 거부감을 갖게 되며 이를 해소하기 위해서는 통계·경제·사회·심리학적 지식을 동원함은 물론 쉽고 간결한 용어를 사용하여 거부반응을 최소화할 수 있도록 하고 질문사항을 적절히 배치하여 질문사항을 직선적으로 이해하고 소화할 수 있도록 하는 한편 응답자의 교육수준을 지나치게 기대해서는 아니 된다.

(3) 집계상 편이도 확보

수집계나 기계집계를 막론하고 조사내용을 부호화하여 기계입력하는 일련의 작업이 있어야만 비로소 집계에 필요한 원시 data가 형성되기 때문에 조사표 설계과정에서부터 집계효율을 제고할 수 있도록 집계부문 종사자의 의견을 적극적으로 반영하여야 한다.

이상의 조사표설계에 관한 기본원칙을 바탕으로 조사항목배치, 기입항목인쇄, box설계채택, 부동문자인쇄, 기계입력명시, 입력란구획선 인쇄, 조사표인쇄색상, 조사표규격 등 조사표설계에 관한 주요 착안점에 대하여 집중적으로 설명하고저 한다.

(1) 조사항목 배치

조사표상의 조사항목에 대한 응답자의 부담없는 응답이 성공적인 통계조사의 요체이기 때문에 응답자의 심리적 부담을 덜고 응답시간·기재착오·기재누락을 최소화하기 위하여 조사표상 연관항목의 분산배치를 피하고 집중배치하며, 기재항목을 좌→우, 상→하의 순으로 배치하고, 조사항목을 응답순서에 따라 배치하는 조사항목배치 4원칙에 충실하여야 한다.

(2) 기재항목명 인쇄

조사표기입란에 기재착오방지를 위한 항목명칭, 항목번호, data 항목별 단위 등을 인쇄한다.

(3) box설계의 채택

기재사항을 지정된 frame에 기재할 수 있도록 미리 frame을 정하는 방식을 box설계라 하며 이와 같은 box설계는 기재란을 넓게 잡을 수 있어서 기재누락을 방지하고 효과적으로 기계입력할 수 있는 이점이 있다.

(4) 부동문자(不動文字) 인쇄

조사항목 가운데 선다형질문은 조사표상의 응답내용에 일련번호를 부여하여 부동문자로 인쇄함으로서 답변내용을 일일히 기재하는 수고를 덜고 해당 답변번호에 ○표를 하거나 번호를 기입하게 함으로써 응답자의 심리적부담을 덜도록 하여야 한다.

(5) 기계 입력항목의 명시

조사표상에 box설계된 조사항목중 입력대상항목과 입력대상 외 항목이 섞여 있는 경우에는 입력대상항목을 조사표상에 일정한 구획을 설정하여 굵은 선으로 묶어 표시하거나 입력대상외항목을 사선 또는 망상으로 표시하여 입력작업이 용이하게 함으로써 입력 error를 최소화하여야 한다.

그림 1-3은 금액단위 착오를 방지하기 위하여 원단위까지의 frame

을 만들고 천원단위까지 zero를 미리 인쇄하여 사선·망상표시를 함으로써 입력착오를 방지토록 한 것이다.

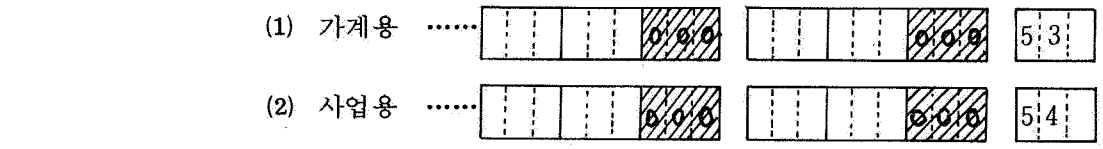


그림 1-3 사선 (또는 망상) 표시 예

(6) 입력란의 구획선 인쇄

입력항목의 box를 수량·금액단위별로 각 column사이에 점선이나 실선으로 구획하고 3 column마다 독특한 선으로 구획하여 응답자의 기재착오를 방지하고 column skip과 같은 입력착오를 방지하는 조치를 취해야 한다.

13 재고액 • 86년 6월 30일 현재 상품재고액은?	※도소매업체만 기입하시오.							
	백억	십억	억	천만	백만	십만	만	천원

그림 1-4 금액단위 구획

(7) 인쇄색상

조사표의 인쇄색상과 색도는 응답자와 입력담당자의 시력보호를 위하여 흑·청·록·자색 계통을 사용하고 있는데 시력보호에 가장 효과적인 색상을 채택하려면 녹색과 자색계통의 색상이 가장 이상적이라 할 수 있다.

나. 접수·정리

집계작업은 중앙집계기관에서 조사표접수, 정리작업으로부터 시작하는데 조사표는 포장된 상태로 수집하기 때문에 포장개수, 포장상태, 포장

내용, 관리상태 등을 면밀히 check한 후 송장과 대조·접수하고 후속되는 제표작업에 편리하도록 지역별 성질별로 분류정리하여야 하는바 내용심사, 부호기입, 입력등의 일련의 작업과 조사표의 보관·관리·운반등에 편리하도록 100～200매(최대 500매) 단위로 편철하고 조사표 훼손을 방지하기 위하여 견고한 표지로 조사표 전후를 보호하되 표지전면에 행정구역, 조사표번호, 조사표매수등 기본적인 속성을 기록하는 총괄표를 첨부하여 조사표 색인에 편리하도록 한다.

다. 내용검사

내용검사는 조사표기재사항 가운데에서 기재누락과 착오가 있거나 항목상호간의 모순등을 검색·정정(수정)하는 작업으로서, 조사표 작성직후에 현지 내용검사와 조사표회수단계에서 중간취합기관에서 실시하는 중간내용검사, 최종적으로 중앙집계기관에서 실시하는 최종내용검사의 3단계 내용검사를 실시하는데 각 단계마다 반복실시할수록 정도 높은 결과를 얻을 수 있기 때문에 예산과 시간이 허락하는 한 반복실시하는 것이 바람직하다.

내용검사는 성격상 개별검사와 관련검사로 구분되는데 개별검사는 응답을 요하는 조사항목에 대한 응답누락 여부, 단답형항목에 대한 복수응답유무, 동문서답식의 응답 여부를 check·정정하는 검사이며 관련검사는 개별검사 결과 개별항목에는 이상이 없어도 항목간 상호연관·조합과정에서 발견되는 모순을 정정하는 검사이다.

이와같이 단계별 내용검사와 성질별 내용검사를 통하여 검색되는 착오사항은 즉각 정정하되 조사표상의 기재내용을 바탕으로 정답을 도출하여야 하며 조사표상에서 정답 도출이 불가능할 경우에는 직접 현지조회과정을 거쳐서 정답을 찾아내야 한다.

라. 부호기입

중앙집계기관에서 최종내용검사가 끝나면 부호기입을 실시하는데 문자로 응답하는 조사항목을 직접 집계하려면 집계효율이 떨어지고 집계

과정에 어려움이 따르기 때문에 조사표 응답내용을 미리 약속한 부호로 변환하여 집계하게 되는데 이와같은 작업을 부호기입이라 한다.

부동문자를 사용한 조사항목에서는 부동문자 앞에 부여한 일련번호로 부호를 대신할 수 있기 때문에 별도의 부호기입이 불필요하나 산업·직업등과 같이 전문지식을 필요로 하는 분류작업에는 응답내용을 가능한한 이해하기 좋게 기입하도록 하고 약속된 부호로 변환하여야 한다.

특히 기계집계에서는 입력작업에 편리하도록 배려하여 입력 error를 최소화하되 입력요원의 reading error방지를 위한 정확한 부호기입, 부호의 자릿수 혼동을 방지하기 위한 leading zero (2 → 02)의 도입, 부호의 column수 최소화를 통한 keyin stroke 수의 감축이 가능한 부호체계를 수립하여야 한다.

마. 입력작업

기계집계는 data 수록매체가 초기의 천공card에서 tape, disk, 등으로 발전하고 다시 광학처리방식으로 발전을 거듭하여 왔는데 현재 통계자료처리에 가장 보편적으로 사용되고 있는 diskette방식에 대하여 설명하고저 한다.

조사표상의 입력항목은 입력작업이 용이하도록 좌 → 우, 상 → 하의 순서로 배열하여 입력 error를 최소화하고 입력능률을 극대화 하도록 하여야 함은 전술한 바와 같다.

그림 1-5 (diskette layout form)과 같이 입력항목에는 행정구역, 조사구번호 등 한번 입력하면 동일구역내에서는 복사사용이 가능한 공통항목과 사업장면적, 종사자수 및 월급여액, 판매액 (수입액), 영업경비, 구입액, 재고액 등 개별적인 특성을 갖는 개별항목으로 구분되는데 공통항목의 경우 란외항목화하여 란외항목이 바뀔 때에만 입력하여 입력속도를 가속화하고 입력 error를 최소화하여야 한다.

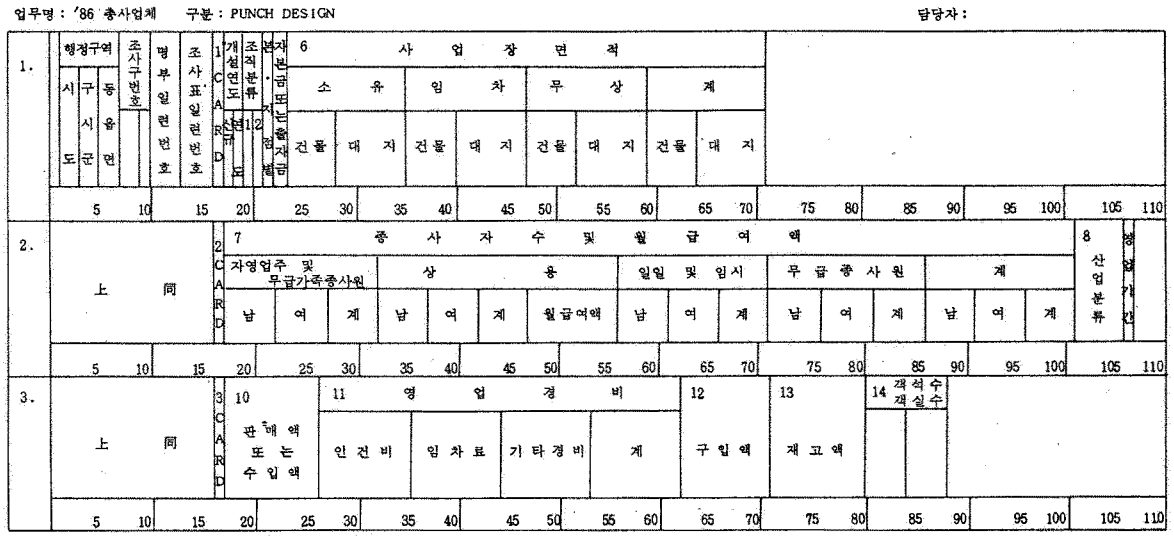


그림 1-5 layout form (punch design)

바. computer집계

원시자료를 tape·disk 등 보조기억장치에 입력하고 이용가능한 통계자료로 집계하고 분석하기 위하여 computer집계를 하여야 하는데 이 때에는 먼저 원시자료중 이용대상자료와 생산하여야 할 통계를 결정하고 그림 1-6과 같은 tape design을 근거로 program을 작성하여야 한다.

한편 computer집계를 위한 program작성에 앞서서 input형식, input부호표, editing요령, 통계표양식 등에 대하여 기획·전산부서간에 충분한 협의가 있은 후에 제공하지 않으면 아니된다.

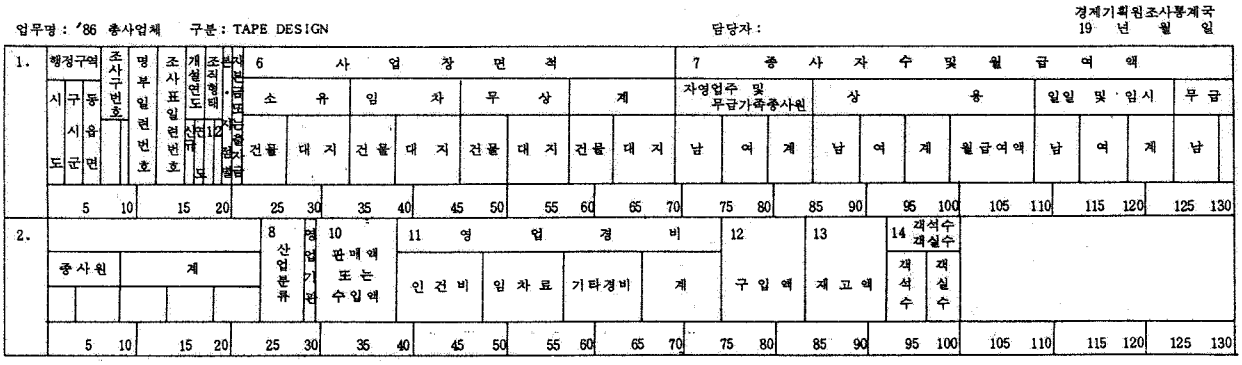


그림 1-6 layout form(tape design)

(1) input형식

입력을 위한 layout form (그림 1-6) 에서 특정한 조사항목의 위치와 길이 (length) 를 column수로 표시하여야 하는데 이와 같은 입력자료의 layout을 기록한 자료를 input형식이라 하고 집계과정에서 필수적인 자료로서 다음에 예시하는 input부호표와 동시에 작성하여야 한다.

(2) input부호표

input부호표란 layout form에서 위치와 길이를 확정한 조사항목에 대하여 부호를 정하고 구체적인 해설을 부가한 표로서 조사항목에 대한 부호이외의 타조사항목과의 관련사항을 상세히 해설하고 특히 조사대상별 기재사항이 서로 다를 때에는 부호의 성립범위를 명시하여야 한다.

program작성자는 부호는 물론 조사항목간 부호의 상관관계등 data 체계를 완전히 소화하여야 하기 때문에 부호표작성시 조사항목 부호 상호간의 관계에 대하여 구체적인 설명을 가함으로써 program작성

에 도움이 되도록 하여야 한다.

그림 1-7 (부호표양식 예)과 같이 위치와 column수란에 layout form (input data)상의 위치(position)와 길이(length)를 기재하고 부호설명란에 각 조사항목의 부호에 대하여 요령있게 정리·기록하되 부호의 종류가 많을 때에는 구체적인 자료를 작성·첨부하고 부호설명란에 참조(연결)번호를 기재하여 program작성에 도움이 되도록 한다.

조사항목	위치	Column수	부호설명
총괄번호	1 - 2	2	수자 01 - 85
행정구역(시·도)	3 - 4	2	행정구역번호 01 - 16 (별첨자료 참조)
취업상태	25	1	1. 취업자 2. 비취업자
노동일수	26	1	1. 50일미만 2. 51-99일 3. 100-150일 4. 151-200일 5. 201-250일 6. 251일이상 blank = 비취업자

그림 1-7 부호표 양식 (예)

(3) editing요령서(data check자료)

data check용 editing요령서는 입력자료의 error를 computer에서 기계적으로 검색·수정하기 위한 program 작성절차

를 기술한 문서로서 일종의 computer용 내용심사요령서라 할수 있다.

data check는 개별check와 관련check로 구분되는데 computer를 통한 data check의 장점은 수작업처리가 불가능하거나 가능하더라도 막대한 인력·시간·예산을 필요로 하고 처리효율이 저하되며 error의 다발로 집계결과의 질이 저하될 경우, 특히 조사항목수가 많으므로서 check내용과 양이 복잡해지고 방대한 조사내용일 경우에 단시간내에 연속처리할 수 있다는데 있다.

check 작업결과 검색된 error의 수정은 computer수정과 인력수정의 두가지 방법이 있는데 computer수정은 상대적으로 능률적이고 경제적이나 부분적으로 인력수정작업을 배제할 수 없는 것이 현실이다.

(4) 통계표 양식

통계표는 표번호, 표명칭, 난외사항, 표측·표두사항을 구성요소로 하고 있는 바 각 구성요소와 조사항목과의 관계, 인쇄 column수, 소수점의 위치, 합계방법등을 program작성에 편리하도록 명시하여야 한다.

통계표 설계는 통계조사의 성패를 좌우하기 때문에 관련분야의 전문가를 동원하여 신중히 설계하여야 하고 통계표 설계에 앞서 조사목적을 충분히 이해하고 조사항목의 조합과 항목분류 정도의 수준결정에 따라 결과표수치의 정도(精度)와 집계처리시간을 고려하여 결정하여야 한다.

위와 같은 사전조치 없이 안전확보를 구실로 막연히 광범위하게 세부적이고 구체적으로 설계하려는 안이한 자세로 기계적으로 조사항목을 조합하고 필요이상으로 항목분류를 하게 되면 결과표수는 천문학적인 양이 되고 집계결과 zero가 과다하게 나타나서 통계표로서의 가치를 상실하게 되기 쉬우므로 사전에 분류정도를 신중히 결정하고 특히 표본조사에서는 필수적인 고려사항임을 명심하여야 한다.

사. 결과표 (통계표) 심사

대체적으로 computer집계라는 이유만으로 결과표를 과신하는 폐단이 있는바 computer는 사람의 의지를 대변하는 수단에 불과하다는 점을 고려할 때 computer 집계결과가 반드시 정확하다는 보장이 없으므로 결과표 공표에 앞서 결과표수치에 대한 면밀한 심사를 하여야 한다.

결과표심사에는 계산기술심사와 타당성심사가 있는데 심사과정에서 순차로 2개 심사방법을 동원하여 결과표심사를 실시하여야 하는 바 구체적인 심사방법은 다음과 같다.

(1) 계산기술 심사

계산기술심사는 computer집계과정에서 발생가능한 error의 유무를 심사하는 것으로서 computer집계과정에서 발생가능한 error는 다시 hardware상의 error와 software상의 error로 분류되는바 hardware상의 error는 computer제작기술의 발달로 hardware에 대한 신뢰도가 높은 수준에 와 있기 때문에 무시해도 좋을 상황이나 software상의 error는 통계기술과 산업사회의 발달로 인한 이용자의 다양한 요구에 부응해야 하는 상황에서 error를 범할 가능성은 높아져가는 실정이다.

software상의 error는 대략 data check누락, coding error, operation miss, 계산기술상의 문제등 4개요인으로 분류되는데 집계규모가 복잡하고 방대할수록 error의 발생율이 크기 때문에 program test과정에서 충분히 확인하였다 하더라도 output 산출과정에서 실제의 output을 대상으로 무작위추출에 의하여 재삼 확인하지 않으면 아니된다.

(2) 타당성 심사

타당성심사는 집계수치의 타당성을 심사하는 것으로서 계산기술 심사과정에서 error가 없다고 판단되는 경우에도 여타 관련 보고서상

수치와의 시계열비교가 가능한가를 판단하고 동시에 분류불능·미상·기타 등 분류상의 특수항목숫자의 과다여부와 불합리한 항목조합 유무를 면밀히 check하여야 한다.

아. 보고서

조사결과는 다수의 이용자가 간편하게 이용할 수 있도록 집계되고 발간되어야 하는 바 조사대상이 광범하고 결과표종류가 다양한 경우에는 조사기획에서 보고서발간에 이르는 전체소요기간이 2-3년의 장구한 시간이 필요하기 때문에 통계이용자의 요구에 효과적으로 대처하기 위하여 잠정집계결과를 공표하거나 보고서를 분책화하여 부분집계결과를 공표하는 등의 방법을 강구하여야 한다.

4. 통계표의 구조

통계표의 구조는 설계자의 발상에 따라 다양하겠으나 표준적인 통계표 구조는 그림 1-8과 같이 표제·두주·표측·표두·표체·각주의 6요소로 구성되는 바 이를 통계표 구조 6요소라 한다.

표제 ← 2.Number of Beds by Medical Facilities

단위 : 개 In each → 두주

	계 Total	종합병원 General Hospital	병원 Hospital	치과병의원 Dental Hospital & Clinic		한방병의원 Oriental Medical Hosp. & Clinic		의원 Clinic	부설의원 Dispensary	조산소 Midwifery Clinic
				병원 Hospital	의원 Clinic	병원 Hospital	의원 Clinic			
1983										
1984										
1985										
1986										
1987										
서울특별시 Seoul-t'ŭkpyŏlshi										
부산직할시 Pusan-jik'alshi										
대구직할시 Taegu-jik'alshi										
인천직할시 Inch'ŏn-jik'alshi										
광주직할시 Kwangju-jik'alshi										
경기도 Kyŏnggi-do										
강원도 Kang-won-do										
충청북도 Ch'ungch'ŏngbuk-do										
충청남도 Ch'ungch'ŏngnam-do										
전라북도 Chŏllabuk-do										
전라남도 Chŏllanam-do										
경상북도 Kyŏngsangbuk-do										
경상남도 Kyŏngsangnam-do										
제주도 Cheju-do										

표측 — 표체 — 표두

그림 1-8 정식통계표의 구조

가. 표제

표제는 표번호(예 : 제○표), 표명칭의 2개부분으로 구성되는 바 결과표집계과정에서 생산가능한 수많은 통계표속에서 필요로 하는 통계표 선정에 결정적인 관건이 되기 때문에 표번호를 계통적으로 부여하고 표명칭은 표측·표두항목의 조합관계와 집계량등을 간결하게 대표할 수 있도록 결정하여야 한다.

특히 표측·표두의 계층관계(그림 1-9)를 적절하게 표현하기 위해서는 계층관계항목을 comma『 , 』로 묶고 병기관계(그림 1-10) 항목을 period『 . 』로 묶음으로써 그 자체만으로도 통계표의 집계내용과 집계항목의 구조를 일목요연하게 알 수 있도록 하여야 한다.

제○표 A,B,C별○○인구

	B				
	C	C	C	C	C
A					

그림 1-9 계층관계 표명

제○표 A.B.C별○○인구

	B				C			
	B	B	B	B	C	C	C	C
A								

그림 1-10 병기관계 표명

나. 두주

두주는 당해 통계표 전체에 대한 주서로서 지역·시기·조사대상등을 정의하여 표두를 보완설명하는 요소로서 그림 1-8에서 표측의 전국·시도가 이에 해당하며 이는 전국 15개시도와 전국합계의 16개 통계표를 작성하여야 함을 의미한다.

다. 표측·표두

표측·표두는 조사사항을 조사목적에 따라서 분류배열한 것으로서 조사사항, 조사목적, 조사방침 등을 고려하여 결정하되 통계표 구성상 표두부문은 계층구조로 표현하는 것이 간편할 뿐만 아니라 표두항목간의

관계가 명확해지기 때문에 특정한 측정수치의 내부적 요인을 표현하는데 적합하며 표측부분은 계층구조로 표현하기가 어려우므로 유사한 속성의 집단으로 묶어서 표현하여야만 집단간의 특성에 따라 서로 다른 점을 비교하는데 용이하다는데 착안할 필요가 있다.

표측과 표두는 바꾸어 배열하여도 통계표로서 변함이 없으나 이용자가 보기에 어렵고 보고서 인쇄시 표두부분이 표측부분에 비하여 항목 수용량을 적게 설계해야 하는 제약때문에 되도록 1 page안에 필요한 내용을 수록하고 불가피할 경우에 2 page 범위내로 압축하여 이용자의 불편을 덜도록 노력하여야 한다.

따라서 일반적으로 표측부분은 길어도 무방하기 때문에 표측의 길이에 제한을 두지 아니하는 반면 표두부분은 보고서 1 page의 폭에 맞도록 설계하는 것이 상례로 되어 있다.

통계표의 표측기준으로 가로를 행 (line), 표두기준으로 세로의 구분을 란 (column), 표측 line은 집계결과 얻어지는 소수행 (素數行 prime line)과 합계·비율·평균 등 집계치의 계산결과 얻어지는 계산치행 (non prime line)으로 구성되며 표두측란은 소수란 (prime column)과 계산치란 (non prime column)으로 구성되는 바 편의상 소수행을 내역, 소수란을 내역항목, 계산치행을 합계항목이라 하기로 한다.

라. 표 체

표체는 집계결과 생산된 통계치를 line과 column의 교차점에 적절하게 배열하는 부분으로서 이때 단위 통계치는 통계표 표체요소로 존재하며 단위요소를 cell (한 칸)이라 한다.

보고서 인쇄시 cell 수치 가운데 zero의 취급에 각별한 주의가 필요한 바 일반적으로 표측·표두의 조합에서 전혀 수치가 없는 경우에는 해당 cell에 『 - 』, 집계결과 수치가 발표 단위에 미달하는 경우에는 『 0 』, 집계결과가 불분명하거나 없을 경우에는 『…』의 기

호로 표시하도록 국제적으로 약속되어 있다.

마. 각주

통계표에서 두주는 표전체에 대한 주석인데 대하여 각주는 line column,cell등 통계표의 구성요소에 대한 주석으로서 총계 수치에 미상수치가 포함된 경우에 미상 수치포함이라는 주석을 부기하는 것처럼 국부적인 column에만 적용하거나 다른 통계표에서 인용 전재하였을 때 그 출처 자료명을 명시하는데 활용한다.

5. 통계집계의 5대 기능과 집계 system

가. 통계집계의 5대 기능

통계표를 집계하기 위해서는 check·분포·합산·가공편성·편집의 5대집계기능이 필요한데 이를 통계집계 5대기능이라 한다.

그림 1-8과 같은 통계표를 모델로 하는 수집계를 가정하면 집계 system설계가 완벽하더라도 기본 data가 부정확할 때 소기의 목적달성이 불가능하므로 정확한 data의 확보를 위해서는 조사표를 computer에 입력하기 전에 입력 error에 대한 면밀한 검사가 선행되어야 한다.

이와 같은 check기능의 보장없이 결과표를 제표한 후에 error 검색을 하고 그 원인을 찾게 되면 자료의 재처리에 소요되는 인력과 시간, 그리고 예산의 낭비가 클뿐 아니라 error 발생원인 규명에 막대한 정력을 소모해야 하기 때문에 computer check가 필요한 바 이와 같은 기능을 check기능이라 한다.

원시 통계자료의 정확성 여부를 computer에서 확인한 다음 단계는 data분포 작업단계로서 조사표를 근거로 입력된 원시적인 input data를 표측·표두항목의 분류에 따라 도수, 금액, 수량을 가산하여 통계표 내역인 표체를 구성하게 되는데 이와 같은 기능을 분포기능이라 한다.

분포작업 다음 단계는 동일 부류간의 합계 산출작업으로서 지역(시도)별 통계수치를 분포과정에서 구하고 동일 부류간 합계를 산출함으로써 통계표의 완성이 가능한 바 이와 같은 기능을 합산기능이라 하고 합산에 이어서 표측·표두의 합계와 평균, 구성비 등을 산출하는 기능을 가공 편성기능이라 한다.

끝으로 보고서의 인쇄체제를 갖추기 위하여 computer의 내부형식(2진수, 부동소수점 등)으로 되어 있는 집계 data를 문자화하고 구독점을 삽입하여 이용자가 편리하게 이용할 수 있도록 하여야 하는데 이와 같은 기능을 편성기능이라 한다.

이상과 같이 집계과정이 check기능으로부터 편집기능에 이르기까지 5개 작업기능을 순차로 실행함으로써 통계표의 작성이 가능해진다.

나. 집계 system

이상과 같은 집계system내에서의 5대기능의 역할은 각 기능을 독립된 program routine으로 간주하여 각 program에 5대기능을 포괄하는 기능복합형과 각기능을 개별 program으로 간주하여 기능을 분산시키는 기능 분산형이 있는 바 이에 근거하여 집계system을 기능 복합형 집계system과 기능 분리형 집계system으로 분류하고 이를 그림으로 표시하면 그림 1-11과 같이 중앙부위의 화살선이 수평적이면 기능분리형, 수직적이면 기능 복합형을 의미한다.

기능분리형 집계system은 이론상 5대기능별로 program을 작성하고 통계표는 1～n표까지 각 기능별로 별도의 program으로 대처해야 하나 실제로는 단일 program으로 통계표 이용자의 다양한 요구에 부응하기가 불가능하기 때문에 이론적인 방법의 제시에 불과하다.

이와 같이 기능분리형과 기능복합형의 2개 집계system을 그림으로 표현하면 집계의 5대기능을 분리하느냐, 복합적으로 포괄하느냐를 구분하는 방법외에 program을 기능분리형에 따라 기능별로 작성

할 것인가, 또는 기능복합형에 따라 표별 program으로 할 것인가에 따라 기능별 집계system과 표별 집계system으로 분류하기도 한다.

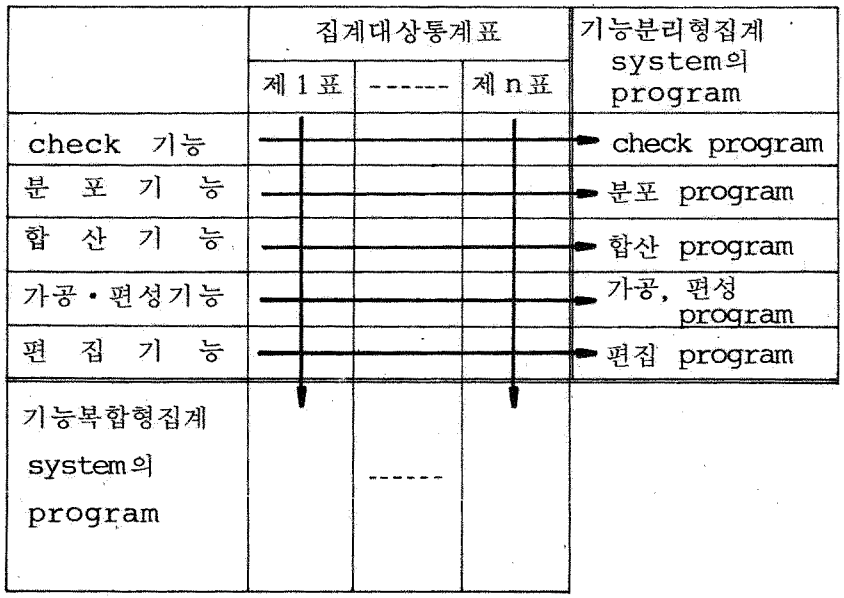


그림 1-11 집계 system의 개념

6. 기능복합형 집계 system

기능복합형 집계system의 기본개념은 그림 1-12와 같이 input data check에서부터 결과표 편집에 이르는 일련의 집계과정을 일괄처리하는데 있기 때문에 통계표의 종류가 다양하여 단일 program으로 대처할 수 없는 경우에는 그림 1-13과 같은 집계 5대 기능을 구비한 여러개의 program을 예비하여야 한다.

그림 1-13에서 data check과정을 기능분리형과 유사하게 독립된 개별 program으로 작성하는 것은 error data의 심사 효율면에서 여러개로 구분하여 심사하고 정정하기 보다는 하나로 묶어서 check

하는 편이 유리하다는 현실적인 요구에 따르기 위한 조치이고 program 1, 2, 3은 분포에서 편집까지의 4개 기능을 cover하는 것으로 통계표종류, 크기(cell수), memory size에 따라 program 수를 달리해야 함을 의미한다.

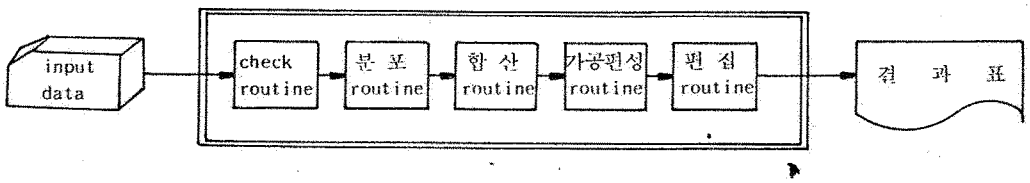


그림 1-12 기능복합형 program의 내부구조

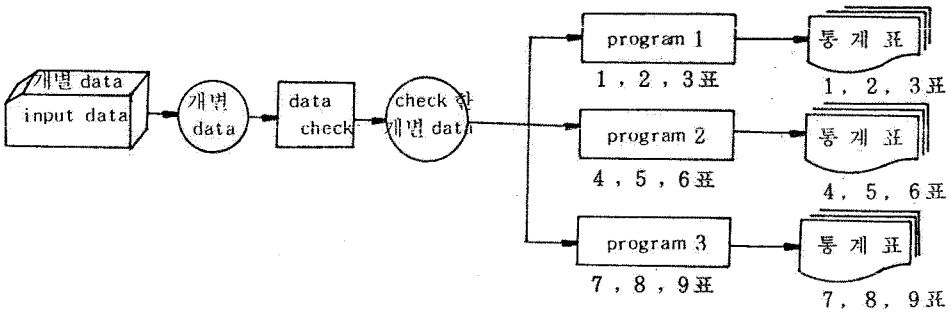


그림 1-13 기능복합형 집계 system의 flow

기능복합형의 특징은 기능분리형과 상대적이므로 기능분리형 집계 system에 대한 설명과정에서 언급하기로 하고 개략적인 특징을 요약하면 다음과 같다.

- o 소량의 input data
- o 간결한 집계항목수
- o 단순한 통계표
- o 소량의 통계보고서

o 최소한의 연산시간

o 복잡한 program(전 집계기능의 단일 program집약)

o 통계표 발표순서에 따른 순차적인 program작성과 집계

이상의 기능복합형 집계system은 업무(행정)통계와 앙케이트 조사집계 등 소규모 집계에 적합한 system이라 할 수 있다.

7. 기능분리형 집계 system

기능분리형 집계system의 기본개념은 그림 1-14와 같이 독립된 각기능을 조합하여 원하는 집계system을 구성하는 것으로서 대량data의 집계처리에 적합한 system이라 할 수 있는 바 기능분리형 집계system의 특징은 다음과 같다.

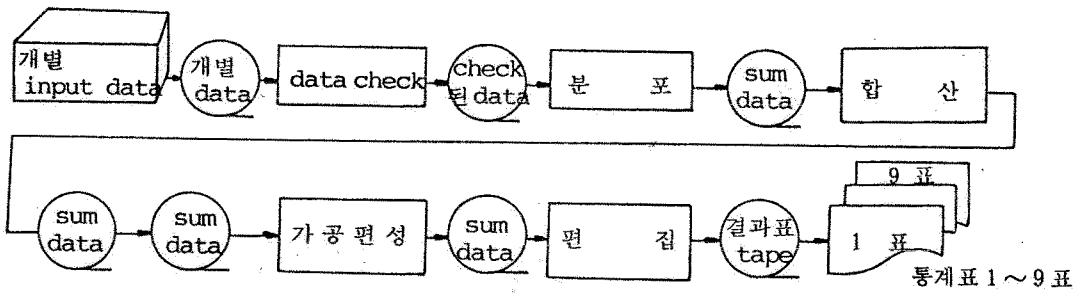


그림 1-14 기능분리형 집계 system의 flow

가. 대량 data 처리회수의 최소화

대규모 통계조사인 인구주택총조사의 경우 data량이 약5천만인·호로서 단위 data에 대하여 check에서부터 편집까지의 처리시간을 50 ms로 가정할 때 전체 data량의 1회 처리시간은 50 ms × 5천만인·호 = 695 hr의 방대한 시간이 소요되는 바 이는 대량 자료처리에 소요되는 시간에 대하여 실감있게 증명하는 사례라 할 수 있다.

이와 같은 대량 data처리에서 최종 통계생산에 이르기까지의 무수한 반복처리를 감안할 때 이에 소요되는 처리시간은 천문학적이기 때문에 가능한한 처리시간을 단축하고 반복처리를 피해야 한다.

일반적으로 집계과정에서 조사대상을 구별할 수 있는 개별 data는 집약된 summary data보다 data량이 많기 때문에 집계기능 가운데 개별 data를 중점 취급하는 check·분포기능이 summary data를 중점 취급하는 여타 3개 기능보다 많은 시간을 필요로 하기 때문에 check·분포기능면에서 개별 data의 처리회수를 최소화하는 조치를 취해야 한다.

기능복합형 집계system은 기능분리형 집계system과 달리 단일 program에 5대기능을 routine화하여 내장하여야 하므로 특정한 기능을 실행하기 위해서는 여타 4개 기능은 기억장치의 일정용량을 점유한 상태에서 휴식상태에 있어야 하며 이로 인한 기억용량의 낭비 등을 감수해야 하는 불이익을 감수해야 한다.

그림 1-14에서 program 1, 2, 3은 통계표의 종류, 크기(cell수)에 따라 3개의 program외에 수 많은 program을 작성하여야 하기 때문에 개별 data 처리회수를 최소화하려면 program수를 최소화하는 방안을 강구해야 하는 바 이때에 단일 program으로 처리하는 것이 가장 이상적이겠으나 실제는 그것이 불가능하기 때문에 단일 program으로 되도록 많은 통계표를 처리(분포)할 수 있도록 설계하여야 한다.

이때의 program은 최소한의 필요 기능만을 수행할 수 있도록 제한하고 그로 인하여 파생하는 memory 여유용량을 다른 통계표의 분포작업에 활용할 수 있도록 하여야 한다.

이와 같은 program작성과정의 기능복합형 program으로부터 분포기능을 독립시켜서 program을 작성하면 여타 기능에 관한 routine과 계산에 필요한 기억용량이 절약되기 때문에 더 많은 통계표의 분

포가 가능한 효과를 거둘 수 있고 통계표상의 합계항목은 내역항목을 근거로 계산하게 되므로 합계항목 cell을 모두 삭제하면 보다 많은 수의 통계표 분포가 가능하고 program당 처리 통계표수가 많아질수록 전체 program수는 감축되므로 이 원리에 근거하여 memory의 경제적인 할당에 대하여 고찰하는 것이 기능분리형 집계system의 기본 개념이라 할 수 있다.

나. 재연산에 대한 유연상 확보

인간사에 완벽을 기하기 어려운 것처럼 computer system도 인간의 의지를 대변하는 수단에 불과하기 때문에 완벽을 기대하기 어려우며 program test를 아무리 반복해도 error를 기피하기란 사실상 불가능할 뿐 아니라 검색error의 computer 재처리가 필연적이나 대량 data의 처리량이 천문학적이기 때문에 재처리문제는 중대과제로 부각되고 error를 최단시간에 재처리하기 위하여 computer system의 error에 대한 유연성 확보가 절실히 요청된다.

예컨대 기능복합형 집계system에서는 5대기능이 단일 program에 압축되어 있기 때문에 합산기능에 error가 발생하면 그 program전체를 재연산해야 하므로 이에 따르는 시간과 예산·인력의 막대한 낭비를 각오하지 않으면 아니 된다.

이때에 routine을 by-path하여 필요한 기능만 재처리해도 program 수정과 test과정이 필요할 뿐만 아니라 program의 수정과 test자체가 완벽한가에 대한 보장이 없기 때문에 기능복합형 집계 system이 재처리에 대하여 경직된 특성이 있다.

기능분리형 집계system에서는 각 기능별로 program이 독립적이기 때문에 error부분이 속하는 program만 간단히 수정하는 것으로 처리할 수 있고, 합계·평균 등의 routine은 집계 흐름의 후단 program에 의존하고 전단 program을 간소화함으로써 error 발생을 최소화하여 재연산 시간을 최소화할 수 있기 때문에 기능복합형

집계system과는 달리 재처리에 대하여 유연성이 있는 system이라 할 수 있다.

다. programmer의 경력·기량에 따른 분업화

계산기와 달리 computer는 이용자가 이용방법(program)을 개발하고 입력하여야만 본래의 기능발휘가 가능하기 때문에 program의 개발과 입력이 용이하여야만 집계system을 효율적으로 활용할 수 있다.

program의 효율성면에서 집계system에 대하여 살펴보면 기능복합형 system에서는 집계 5대기능 전반에 대한 지식과 program작성기술에 숙련되어 있을 때 program작성이 가능한데 대하여 기능분리형 system에서는 개별 data의 check와 분산 등 비교적 고도의 전문지식을 필요로 하는 분야는 error발생시 damage가 큰 program을 상급 전산인, data의 분포·합산 등 처리시간은 많이 소요되지만 내용이 어렵지 않은 program은 중급 전산인, 인쇄 등 초보적인 program을 초급 전산인에게 분장하는 수직 분업체제를 유지함으로써 대량생산이 가능하도록 하여야 한다.

라. program의 간소화, 표준화 및 범용화

각 program이 독립적으로 단일한 기능만 수행토록 되어 있는 기능분리형 집계system에서는 program의 간소화·표준화·범용화가 가능하기 때문에 기능분리형 집계system의 흐름은 program수가 많은 반면 표준화와 범용화가 가능함으로 program을 간결하게 제한할 수 있는 장점이 있다.

마. programmer교육의 단계적 실시

program작성에는 논리적 두뇌의 소유자인 우수 인력이 필요하므로 programmer 양성에 있어서 이론과 실무에 대한 교육이 대단히 중요하며, 집계system에 대한 교육에서도 기능분리형 program과 관련한 집계기능에 대하여 초·중·상급으로 계층화하여 상황에 알맞

는 교육을 실시하여야 한다.

이상의 기능분리형 집계 system은 대량 data의 집계처리를 위하여 인적자원 (programmer)의 적정배분과 지속적인 양성, program의 표준화와 범용화에 따른 programming stuff의 대중화, 기억용량의 적절한 이용 등 integrated한 system이라 결론지을 수 있으며 집계기능에 대한 개념을 이해하기 위하여 소량집계에 의한 응용방안을 통하여 기능별 집계 system을 전제로 한 각 기능의 이론적 배경과 program에 대하여 상세히 설명하기로 한다.

Ⅱ. CHECK PROGRAM의 설계

1. check program의 기본 기능

대량 data를 신속 정확하게 처리할 수 있는 computer의 장점을 살리기 위해서는 정확한 data, 정확한 program, error없는 operation, 완벽한 computer라는 네가지 조건이 전제되어야 하는 바, computer는 일반 계산기와 달리 parity check system을 포함하는 각종 check system이 내장되어 있어 computer system 내부에서 data의 송수와 계산처리과정에서 error의 유무를 hardware에서 check할 수 있도록 되어 있기 때문에 전산처리된 data는 일단은 error가 없다고 해도 좋을 만큼 정확성이 확보된다.

data 자체의 논리적 error는 hardware의 error check system로는 검색이 불가능하기 때문에 미리 program 상에서의 error data 검색을 통하여 여과하므로서 program의 비정상적인 종료나 미궁에 빠짐으로 인한 대외의 계산결과를 방지할 수 있기 때문에 hardware에서의 check는 물론 program에 의한 data자체의 논리적 error를 검색하여 집계 system의 신뢰도를 높여야 한다.

그림 2-1과 같은 연속적인 data처리에서 program A, B, C에 대하여 개별적인 error check를 하면 작업의 중복과 개별 program 간에 통일적인 check 처리절차의 확보가 불가능하기 때문에 check의 엄밀성과 error data 수정의 분할처리에 대한 보장이 어렵게 되므로 error data check 기능의 분산은 가급적 피하여야 한다.

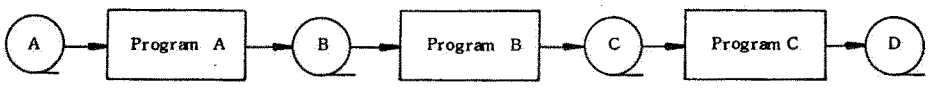


그림 2-1 연속처리 flow

따라서 기능분리형은 물론 기능복합형에서도 data check 기능은 별도의 독립적인 program으로 집약정리하여 처리하는것이 일반적인 예로 되어 있으며 check program의 기능이 다양하고 통계조사의 성격과 결과표의 분류정도에 따라 요구기능에 차이가 있으나 기본적으로 data check·수정, code 변환·정보추가, output data로 group 화한 기능으로 정리할 수 있다.

2. check 기능의 종류

input data의 정확성 제고를 위해서는 computer check 작업 이전에 조사표 기입·내용검사·key in의 3단계 작업에서의 check 체제 정비가 선행되어야 하고 최종검사로서 computer check에 착수하여야 한다.

program에 의한 check의 종류에는 배열구분에 관한 check, total에 관한 check, 존재에 관한 check, 문자 mode에 관한 check로 대별 할 수 있으나 실제 통계 집계과정에서는 위에 나열한 check를 독자적으로 활용하거나 필요에 따라 각각의 기능을 조합하여 활용하고 data의 특성에 따라서는 새로운 방법을 개발 활용하여야 한다.

【 program check의 종류 】

① 배열·구분에 관한 check
- sequence check
- 대상구분 check

② Total에 관한 check
- control total check
- batch total check
- hash total check

- balance total check

③ 존재에 관한 check

- off code check
- range check
- 관련 check
- 검사숫자 check

④ 문자 mode에 관한 check

- blank check
- 숫자 check

가. 배열 구분에 관한 check

(1) sequence check

input data가 특정항목에 대하여 정해진 순서에 따라 배열되어있는가를 검사하는 check기능으로서 행정구역(시도·구시군·동읍면)의 지역번호 data가 약속(sequence)된 순서대로 배열되어 있는가, 단위세대의 data 가운데서 세대주 data가 첫머리에 배열되어 있는가를 check하고 이 약속이 깨어졌을 때에는 전 단계에서의 data 처리에 잘못이 있다고 판단하는 check 방법이다.

이때 data check 이전에 sort를 하면 sort항목의 error code에 의하여 data가 분산되기 때문에 error data의 검색과 수정이 어렵게 되므로 data check 이전의 sort는 피하는 것이 좋다.

(2) 대상구분에 관한 check

input data의 기본속성인 data 종류, 조사년월일, 지역번호등을 computer로 check하는 기능으로 대상구분에 대한 처리범위가 일정하지 않은 경우에는 control card나 operation에 의하여 처리하는 것이 일반적이나 이 check기능은 후술하는 off code check에서 검색이 어려운 대상외 data의 검색에 적합한 방법이다.

전수조사의 시도단위 집계처리과정에서 시도단위 data를 check

할 때 program의 중복작성을 피하기 위하여 표준화된 program으로 지역(시도)번호 01(서울)~15(제주)범위내에 있는가를 check하여 범위 밖의 것은 error가 되도록 program을 작성하여야 하는데 이때에 특정지역번호의 data속에 타지역번호의 data가 섞여 있으면 검색이 불가능하므로 이를 검색하기 위해서는 처리대상 지역번호 nn을 program에 지시하여 input data의 지역번호=nn 라는 대상구분 check 방법을 채택하여야 한다.

나. total에 관한 check

(1) control total check

input data의 특정항목(control 항목)별로 수작업 합계치(control total)와 computer 집계결과치를 상호 대조하는 check 방법으로 계산량이 크거나 복잡한 것은 수작업이 어렵기 때문에 조사구별 종업원수, 행정구역별 사업체수 등 간단한 계산량을 대상으로 한다.

대조작업은 control total을 computer에 입력하여 기억장치내에서 처리하는 방법과 computer에서 control total을 print out하여 eye check하는 방법이 있으나 양쪽 모두 input data의 중복과 탈락의 check가 주안점이 되어야 한다.

(2) batch total check

control total check에서 control 항목 대신에 data를 적당한 크기의 batch(묶음)로 분할하여 batch별 total을 대상으로 check하는 방법으로 control total check에서는 data가 control 항목별로 구분배열되어야 하는데 이와같은 전제가 불가능한 data의 check를 일정규모의 batch로 묶어서 batch check를 실시하여야 한다.

예를들면 1000매분의 data를 batch로 하여 인구수·세대수·사업체수 등을 구하거나, 품목별배열이 곤란한 일일판매액 data를 오

전·오후로 구분하여 batch total을 구하고 check하는 것이 이에 속한다.

(3) hash total check

일반적으로 total이란 금액·수당·인원 등과같이 설정된 합계란에 합계숫자를 연산·수록하는 의미있는 수량을 말하나 hash total은 code번호, 금액계급code·성별code 등과 같이 합계산출이 불가능하고 무의미한 항목을 기계적으로 합계한 total을 같은 자릿수(column수)로 하고 초과하는 상위숫자를 무시하는 작업을 하게 되는데 이는 over flow 부분을 무시한 하위숫자만으로 total이 일치하는 확율은 지극히 적다는데 근거를 두고 있다.

hash total check는 급여계산 업무의 봉급표, control total check에서의 control별 total표 등을 program에 반영할 경우 control data check에 편리하게 활용하는데 예컨데 program에서 control total check를 위하여 동읍면별 인구수를 필요로 한다면 시도번호 2 column, 구시군번호 2 column, 동읍면번호 3 column, 인구수 7 column, 합계 14 column을 하나의 항목으로 간주하여 hash total을 구하고 total 부문에 합산한다.

이때에 hash total이 14 column을 초과하는 over flow 부분을 배제하고 지역별 인구수는 당해지역 code부분에, 합계인구는 합계란에 각각 keyin하게 되는데 이를 punch card 개념으로 표시하면 그림 2-2와 같다.

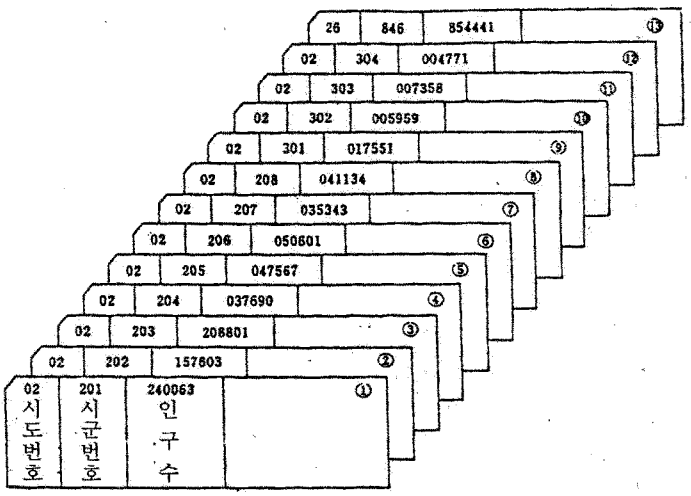


그림 2-2 hash total의 예

다음으로 program에서 지역별 hash total을 수작업과 동일한 약속 아래 계산하고 합계란의 hash total과 비교하여 원시자료의 전기(轉記) error, keyin error, 계산 error등 부정적 요인을 check하게 되는데 지역별구분이 많을 때에는 적당한 간격으로 hash total을 계산하므로서 hash total group별 error검색이 가능하기 때문에 정정이 용이한 장점이 있다.

⑷ balance total check

금액란·수량란과 같은 계량항목에서 합계란과 내역란의 합산치(total)의 일치여부를 check하는 방법으로 여기에서 error발생원인이 합계란과 내역 total의 불일치에 있는가의 여부를 판별할수는 없으나 column skip에 따르는 비정상 숫자의 발견에는 효과적인 check방법이라 할 수 있다.

간혹 합계란 설명으로 column수의 부족현상이 발생하기 때문에 합

계란을 생략하는 경우가 있으나 data check에서는 check의 단서를 놓치는 결과가 되므로 중요항목에서는 생략이나 삭제를 해서는 아니된다.

다. 존재에 관한 check

(1) off code check

input data 항목에는 성별·국적등 code에 관한 항목과 금액·인원등 수량에 관한 계량항목이 동시에 존재하게 되는데 off code check는 code항목의 정의에 따른 code 부여 여부를 check하는 기능을 갖는것으로 예컨데 성별code가 1(남자), 2(여자)로 정의된 경우에 1·2 이외의 code를 error로 처리하는 check 방법이 이에 속한다.

행정구역번호, 산업분류번호등 code 종류가 다양한 경우에는 후술하는 range check를 통하여 off code의 발견이 가능하나 이때에는 결손 code를 미리 파악해야하고 2 column 이상의 code 일때 range check가 집계결과를 흐트러 놓을 수도 있다는 사실에 유의하여야 한다.

예컨데 01(서울)~15(제주도)까지의 2 column 숫자 code로 구성된 행정구역번호의 경우 range check의 하한치 01, 상한치 15에 대하여 그림 2-3과 같이 check하면 바른 data ⓐ에는 01~15이외의 16진수 문자가 개입하게 되므로 range를 막연히 01~15로 정의하는 것만으로는 의외의 위험이 따르기 때문에 그림 2-4와 같이 행정구역번호가 약속된 code인가를 확인한 다음에 check하지 않으면 아니된다.

이때에 program 언어에 따라서는 전체 column이 숫자인가를 check하기 곤란한 경우가 있는바 이때에는 행정구역 code를 구성하는 10단위에서 0~1, 1단위에서 0~9 범위내에 있는가를 각 단위별로 2회에 걸쳐서 range check를 실시하여야 한다.

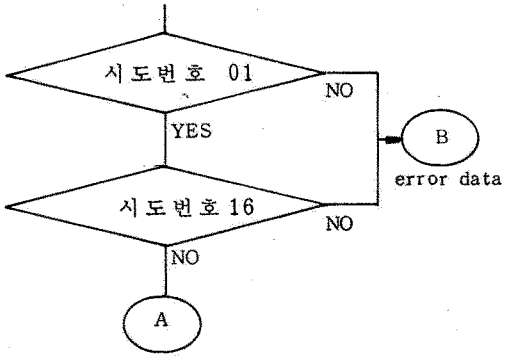


그림 2-3

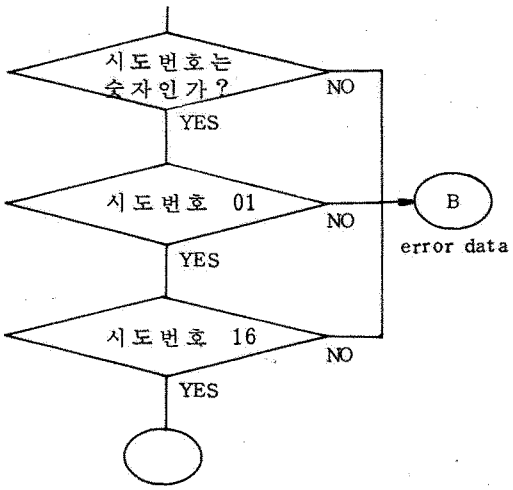


그림 2-4

(2) range check

off code 검색을 위해서는 range check가 가장 유효한 수단임은 전술한 바와 같으나 특히 계량항목에서는 range check 가 주된 check 수단으로 활용되는데 계량항목의 off code check에는 상한치와 하한치에 확정치, 또는 개략치를 부여하는 경우가 있는 바, 후자의 경우에는 범위 밖의 불확실한 data임을 의미하므로 이때에 error로 속단하게 되면 생산 data는 신뢰성을 잃는 위험한 결과를 자초하게 된다.

일반적으로 계량항목은 code 항목과 마찬가지로 올바른 data의 범위가 결정되어 있지 않은 경우가 많기 때문에 그만큼 check에 필요한 수단과 방법을 놓치는 요인으로 작용하게 되는데 다소나마 error data 검색에 도움이 되려면 total check 나, 금액단위 column은 반드시 0, 또는 5일 것 등의 항목성질이나 항목간 연관성을 적극적으로 이용하여야 하며 계량항목일지라도 error는 keyin error에 국한하는 것이 아니므로 주의가 필요하다.

(3) 관련 check

각항목의 off code check에서 error가 검색되지 않았다 하더라도 별개의 관련항목의 조합에 모순된 data가 있게 되는데 이와같은 관련항목간의 모순검색 check를 관련 check라 한다.

예컨대 연령·학력란에서 독자적인 error가 없을지라도 이들 2개 항목의 조합에서 각학력계급 취득에 필요한 하한연령 제한에 근거하여 대학졸업년령이 14세라는 실현불가능한 모순(error)이 부각되었을때 이와같은 모순이 검색되더라도 연령과 학력의 관련항목중 error가 속하는 항목을 찾아내기가 어렵기 때문에 error data를 print out하여 관련정보와 대조하여 판단하지 않으면 아니된다.

(4) double check

전체 input data 항목을 대상으로 동일한 내용의 data가 비정상적으로 연속하는가를 check하는 기능으로 동일내용의 data의 존재를 허용하는 경우가 아니고서는 일반적으로 동일 data가 2매이상 연속하는 사례가 없기 때문에 double check의 존재의미가 있다.

double check는 중복된 data의 검색이 목적이므로 operation miss로 동일한 내용의 중복된 read를 방지하는데 유효하나 중복내용이 연속된 때에만 check가 가능하고 불연속적일 때에는 불가능한 단점이 있다.

(5) 검사숫자 check

keyin error 검색에 유효한 방법으로 code를 바탕으로 계산되는 검사숫자라는 1 column의 숫자를 code 말미에 부가하여 새로운 code를 만들어 사용하는 방법으로 modulus 10과 modulus 11의 2개 방법이 있다.

modulus 11에 대하여 설명하면 code의 1단위숫자×2(배), 10단위숫자×3(배), 100단위숫자×4(배)… 의 요령으로 좌로 1 column 이동할 때마다 승수를 1씩 증가해 나가되 7이 초과할때마다 2로 복귀하여 반복처리하고 각 column의 승수결과를 가산하여 11로 나누어 그값을

정수범위내에서 묶고 잉여치를 구한다음 11에서 잉여치를 공제한 결과 얻는 1단위 숫자를 검사 숫자로 정한다.

code 1974의 검사숫자의 계산예를 들면

1	9	7	4	·····	최초의 code
×	×	×	×		
5	4	3	2	·····	승수

5 + 36 + 21 + 8 = 70

↓

70 ÷ 11 = 6 ····· 4 (잉여숫자)

↓

11 − 4 = 7 ····· 검사숫자

↓

19747 ··· 검사숫자를 첨가한 새로운 5 column code

와같이 7이라는 검사숫자를 만들고 당초 4 column code(= 1974) 말미에 검사숫자 7을 부가하여 새로운 5 column code(= 19747)를 작성사용하되 error check는 program 상에서 위와같은 계산을 통하여 검사숫자와 일치하는가를 check 한다.

이방법은 code의 조사만으로 error 유무를 check 할 수 있기때문에 검사숫자의 부가가 가능한 code 목에서는 활용이 가능하나 검사숫자를 부가할 수 없는 계량항목에서는 활용이 불가능하다.

라. 문자 mode에 관한 check

(1) blank check

input data의 전항목이 blank인가의 여부를 check 하는 방법으로 keyin 단계나 tape 수록단계에서 blank card의 혼입여부를 check하는 기능으로 일반적으로 전 항목이 blank이면 각종 check 과정에서 error로 검색되기 때문에 blank check를 생략해도 무방하다.

(2) 특수문자 check

특정항목을 대상으로 특수문자로 구성되어 있는가를 check하는 방법으로 data의 성격에 따라 특수문자가 숫자·영문자 일 수 있다.

계량항목에서는 숫자를 주로 keyin 하나 조건에 따라서는 blank, 또는 X-skip을 keyin할 경우가 있기 때문에 이러한 항목의 check는 숫자, blank, 또는 x-skip 여부를 확인하여야만 error 검색이 가능하고 column의 중첩을 방지하려면 항목사이에 공백란을 설정하여 그 부분이 blank인가를 check하여야 한다.

이와같이 특수문자의 check는 input data의 성격에 따라 다르기 때문에 각별한 주의가 필요하다.

이상 다양한 check 기능에 대하여 살펴보았으나 결정적이고 완벽한 기능을 구비한 check 방법은 아직 없기 때문에 가장 도움이 되는 check 방법을 조합하거나 data의 특징을 고려하여 error data의 보충율이 높은 program을 작성하는 노력이 필요하다.

3. error data의 수정

각종 check기능을 동원하여 error data를 검색하였을 경우 error data의 처리방법은 조사규모·집계기간·결과표 정도에 미치는 영향을 고려하여 결정해야 하는데 대체적으로 error data를 집계과정에서 제외하는 방법, program에 의한 수정방법, 수작업에 의한 수정방법등 세가지 방법이 있는바 이에 대하여 설명하면 다음과 같다.

가. error data를 집계과정에서 제외하는 방법

error data가 전체 data량에 비하여 무시해도 좋을 만큼 적어서 집계과정에서 제외해도 조사결과에 거의 영향을 미치지 않을 경우에 채택하는 방법으로 program에 의한 수정이나 수작업에 의한 수정이 불가능한 경우에 한하여 채택하는 것이 무난하다.

나. program에 의한 수정방법

program에 의한 수정은 input data의 error code를 집계

결과에 대한 영향이 가장 적은 code로 치환하는 방법으로 정치(定値)수정, 택일 수정, 천이(遷移)수정등 3개방법이 있다.

(1) 정치(定値)수정.

조사내용과 집계결과의 경향을 근거로 일정한 code를 설정하여 수정하는 방법으로 예컨대 각 조사항목에 미상 code를 설정하여 error data는 모두 미상처리하고 미상code가 없을 경우에는 출현빈도가 가장 큰 code를 대상으로 error data를 할당하는 방법을 말한다.

정치수정방법을 채택함에 있어서 미상code를 설정하므로서 문제를 간단히 해결할 수 있는 것으로 생각하기 쉬우나 미상수치가 많으면 통계표 자체가 무의미한 것이 되어 이용가치가 떨어지는 결과가 생기기 때문에 여타 관련항목과의 연관성을 고려하여 타당한 code를 찾아야 한다.

(2) 택일수정

여러개의 대상code 가운데 특정code를 선택하여 설정하는 방법으로 결정code가 일정하지 않다는 점에서 정치수정과 다르고 여러개의 code 선택방법중 한가지 방법을 예시하면 특정항목의 code가 1,2,3, ……, n라 할때 집계자료의 과거계열상에서 1,2,3, ……, n의 구성비를 program 작성과정에서 반영하고 검색된 error data의 code를 구성비에 따라 할당하는 방법으로 성별란 code가 1(남), 2(녀)이고 성비가 50:50이라 할 때 성별불명 data는 성비율에 따라 각 code에 할당하는 방법을 들 수 있다.

(3) 천이(遷移)수정

정치수정이나 택일수정은 미리 결정된 code나 비율에 의한 수정인데 대해서 천이수정은 전산처리된 data의 출현상황과 관련하여 수시로 변동되는 code 가운데 하나를 수정치로 하여 수정하는 방법으로 수정code의 선정은 computer 스스로 해결할 수 있도록 일임하는것이 특징이라 할 수 있는 바 이와같은 천이수정을 그림 2-5의 연령수정을 예로들어 설명하면 다음과 같다.

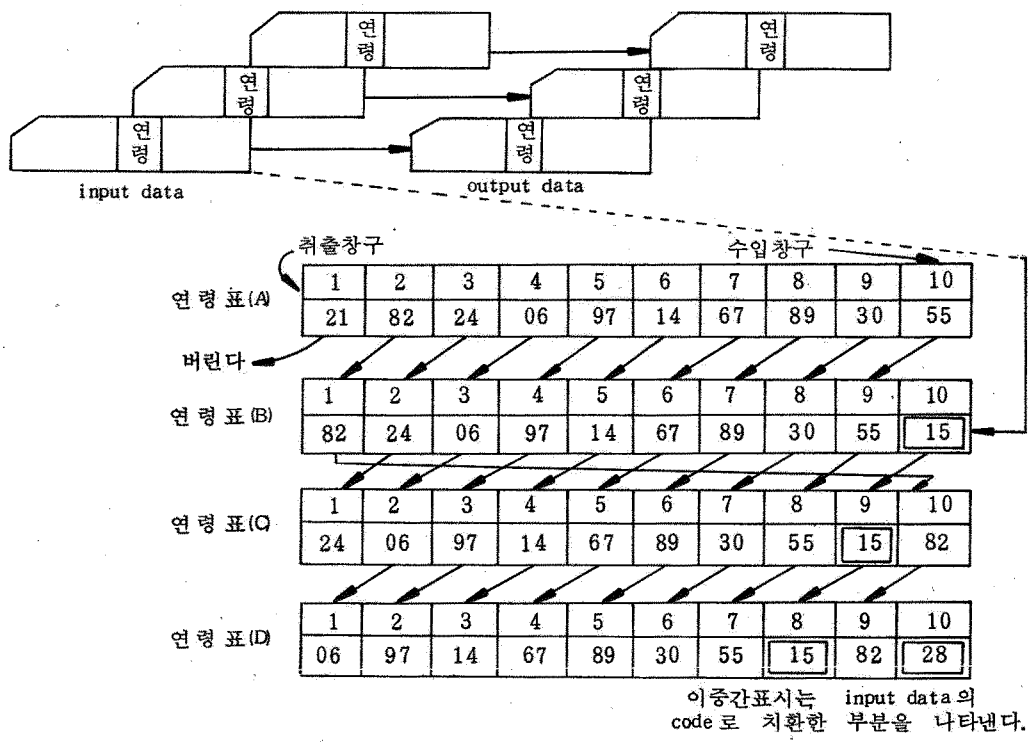


그림 2-5 천이수정방법 (연령 check)

㈎ memory에 연령표(A)와 같이 off code일때의 연령을 배열한 대행열 요소를 편이상 「창구」라 하고, 좌측창구를 취출(取出)창구, 우측창구를 수입(受入)창구, 창구수를 대행열의 길이로 정의하고 대행렬의 길이=10, 1번창구=취출창구, 10번창구=수입창구로 하여 각 창구의 초기값을 난수표에 의하여 부여한다.

㈏ input data ①의 연령은 15로서 바른 code가 부여된다.

㈐ 연령표(A)에서 각 창구값은 좌측으로 1 column씩 이동하고 이때에 취출창구에서 밀려나오는 21을 버린다.

㈑ input data ①의 연령 15를 연령표(A)의 비어있는 수입창구에 받아들여 연령표(B)를 작성한다.

㈒ input data ②를 check하고 연령은 off code이므

로 연령표(B)의 취출창구에 있는 82를 할당한다.

(바) 연령표(D)의 각 창구값을 좌로 1 column씩 이동하고 off code의 경우에 밀려나는 82를 대행열의 lotation을 위하여 수입창구로 다시 보내어 연령표(C)를 작성한다.

(사) input data ③의 연령은 28로서 off code가 아니므로 연령표(C)의 값을 좌로 1 column씩 이동한다.

(아) input data ③의 연령 28을 수입창구에 넣고 연령표(D)를 작성한다.

(자) 이상의 작업을 input data가 없어질때까지 반복한다.

이상 천이수정에서 data에 error가 없을때에는 data처리때 마다 대행열은 1 column씩 좌로 이동하므로 대행열의 길이 이상의 data를 처리하게 되면 대행열요소는 초기값에서 처리된 값으로 치환하게 되며 data의 질적변화(그림 2-5의 연령층 변화)가 대행열에 축적되므로 off code가 존재할 때에는 주변의 유사연령을 할당하는 결과를 낳게 되고 off code data가 계속될 때에는 대행열요소를 순차로 하나씩 할당하고 동시에 대행열이 회전되어 계속 동일한 값의 할당을 방지하게 된다.

예컨대 양로원·기숙사 등에서는 구성연령층이 일정한 범위내에 한정되기 때문에 처리data가 양로원집단에서 기숙사집단으로 이동하면 천이수정의 대행열은 노인연령층의 상태에서 학생연령층으로 급속하게 변화하여 data의 집단적인 특성을 반영하게 된다.

data 특징의 반응도와 수정값의 적정화는 대행열의 길이, 대행열의 종류에 관계되므로 대행열 길이의 단축으로 반응속도를 단축할 수 있으나 단축이 지나치면 off code때의 할당이 동일한 값이되어 바람직하지 못하고 또 적정화는 data의 층별(예 : 성별·취업상황·배우관계중 연령과 함수관계에 있는것)마다 대행열을 작성하여 수정치의 특이성을 방지하여야 한다.

다. 수작업에 의한 수정

program에 의한 수정에는 한계가 있기 때문에 종국적으로는 수작업 수정이 불가피한 항목이 있는바 이와같은 경우 판단자료로서 error list(error 일람표)를 작성하여 수정작업을 용이하게 하여야 한다.(그림 2-6)

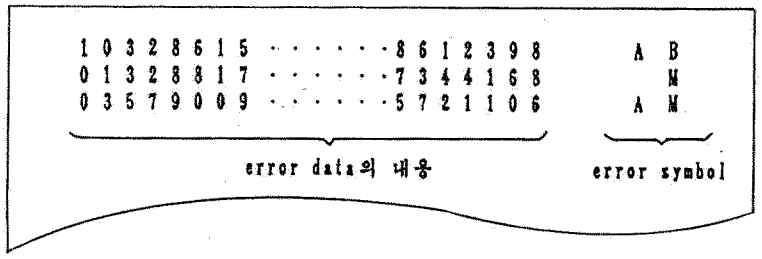


그림 2-6 간단한 error print

error list는 error data의 내용과 error의 종류를 정의하는 error symbol을 약속하여 인쇄한 것으로 error symbol에는 단순히 error의 종류를 표현하는 부호형태와 error내용을 해설하는 문장형태가 있는 바, 주기적(주간, 월간)인 것은 문장체보다는 부호체가 편리하고 부정기적이고 일회한인 것은 문장체가 유리하다.

간단한 error list 예로는 그림 2-6과 같이 error data의 내용과 error symbol을 동일 line에 print 하는 것이나 수작업심사가 용이하도록 하려면 그림 2-7과 같이 변형하여 편집하는 것이 유리하고 수작수정에 대하여는 error data처리의 표준화에서 상세히 설명하기로 한다.

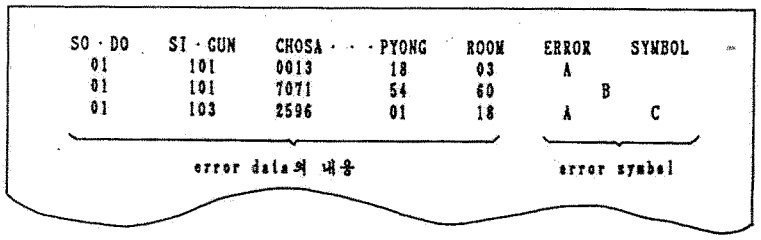


그림 2-7 편집한 error print

4. error data의 처리형태

error data의 처리형태는 error data의 수정형식에 따라서 input data list에 의한 수정 flow, program에 의한 수정 flow, 수정불가능한 data를 제외한 program 수정 flow, error data의 수작업수정 flow, error data의 수작업과 program 병행수정 flow등 5개 flow로 구분된다.

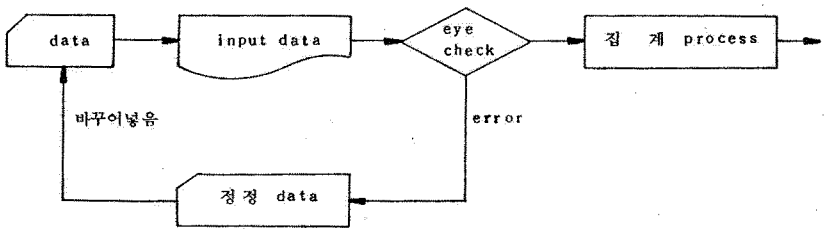


그림 2-8 input data list의 flow

가. input data list에 의한 수정 flow

이 수정형식은 input data 량이 근소할 때 check program 작성노력을 생략하기 위한 방식으로 이 경우에는 수작업 check를 하기 때문에 data량이 많을때에는 오히려 비경제적이나 산업분류code, 봉급표code와 봉급액, hash total의 control data처리 등을 위하여 필요로 하는 constant처리에 대한 check를 수정·치환하여 reprint하고 error를 check하되 error가 없어질 때까지 반복처리해야 한다.

나. program에 의한 수정 flow

조사항목에 미상code가 설정되어 있거나 조사내용이 간단한 경우

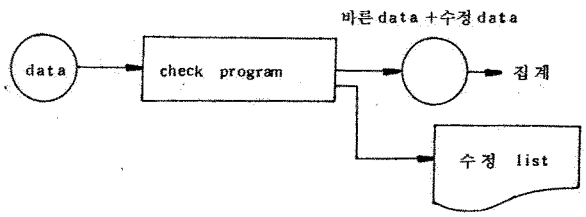


그림 2-9 program에 의한 수정 flow

에 수정 list를 통하여 error data의 수정근거를 제공하므로서 trouble 발생시에 판단자료로 활용한다.

통계조사에서는 program에 의한 집계처리에 선행하여 조사표접수로부터 내용검사, 부호기입에 이르기까지 수작업과정이 필수적으로 따르기 때문에 단순조사에서는 수작업과정에서 대부분의 error가 수정되고 check program에서 처리되는 error는 keyin error에 한정되므로 근소한 작업량으로 축소된다.

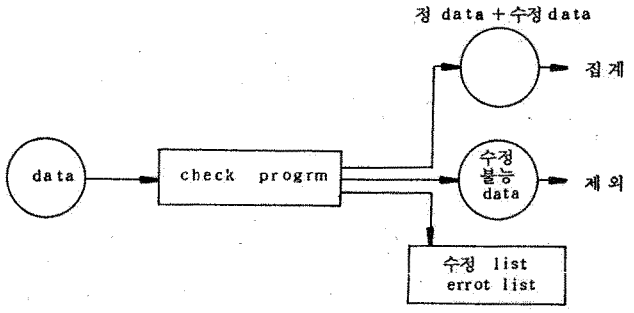


그림 2-10 제외, 수정 병용 flow

다. 수정 불가능한 data를 제외한 program 수정 flow

program으로 수정가능한 data만 수정하고 수정불가능한 data는 집계에서 배제하는 수정방식으로 program에 의한 수정형식과 마찬가지로 수작업 처리과정을 통하여 program으로 수정할 error data가 근소할 경우에는 error list나 수정할 list를 판단자료로 하여 수작업수정이 program을 통한 수정보다 편리한가를 판단하여 결정하여야 한다.

라. 수작업에 의한 수정 flow

이 형식은 program에 의한 수정없이 error list를 근거로 수작업으로 error data를 조사표와 대조·심사하여 수정자료를 만드는 방법으로 error data가 완전히 없어질 때까지 반복처리한다.

실제로는 3회정도의 수작업검색이면 잔존error는 무시해도 좋을만큼 축소되므로 집계과정에서 배제해도 무방하며 이러한 잔존 error

data는 수작업도중에 배제하기도 한다.

이때에 program상에서는 error로 판정된 data 가운데 수정할 필요없는 바른 data가 있는바 이러한 data를 다른 수정대상 data와 더불어 동일한 program으로 check하면 재차 error로 판정되기 때문에 이 형식을 적용할 수 있는 경우는 지극히 한정적인데 주의하여야 한다.

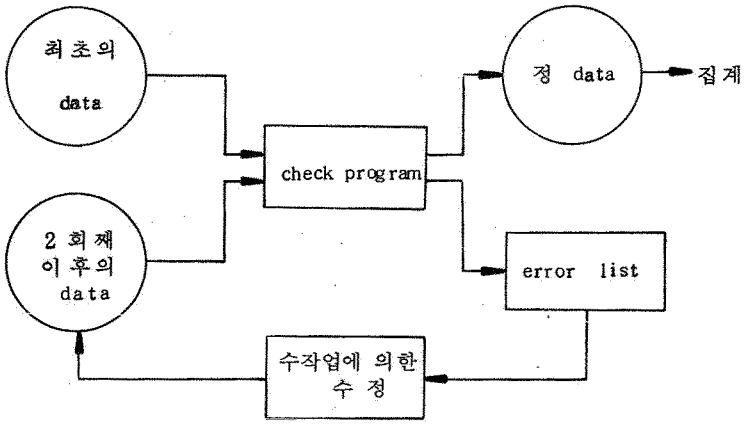


그림 2-11 수작업에 의한 수정 flow

마. 수작업과 program에 의한 수정 flow

전술한 수정형식에서는 정상 data마저도 check program의 check 조건이 엄격하기 때문에 error로 취급되거나, check 조건의 완화로 사실상의 error마저 정상 data로 간주되는 점을 개선하기 위하여 수작업과 program 작업을 동시에 병행동원하여 먼저 check program만으로는 check 조건의 엄격성 때문에 정상 data마저 error 처리되는 것을 방지하기 위하여 error list를 근거로 수작업 수정을 하고 이어서 program check를 하되 수작업에서 정상data로 판정된 data가 error처리 되지 않도록 check조건을 완화한

별개의 program을 개발적용하여야 한다.

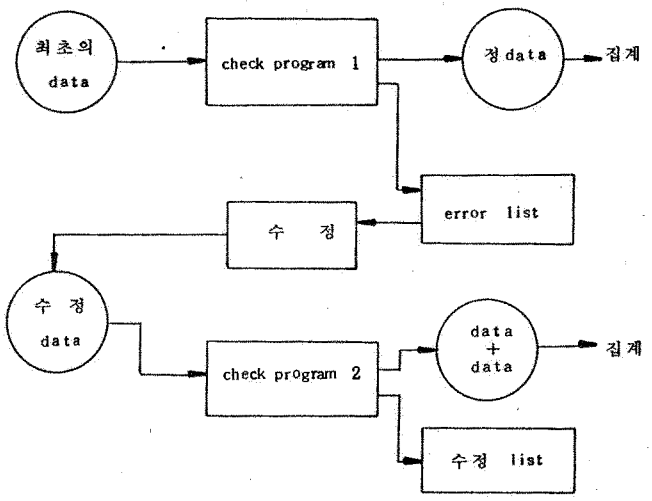


그림 2-12 수작업, program 병용 flow

이 program에는 수작업과정에서 error data를 수정하여 error data가 없을 것이라는 전제아래 check 기능외에 program 수정기능을 추가하는 것이 상례이며 이 형식에서의 문제점은 check program의 check 조건설정으로 귀결되는 바 check 조건을 완화하면 수정 data의 error를 간과하기 쉽고 반대로 check 조건을 강화하면 정상 data까지도 무리하게 수정되는 문제가 발생하기 때문에 일반적으로 error 처리를 3회정도 반복하여 최종수정을 하는 방법을 채택한다.

이와같이 error가 아니면서 강화된 check 조건하에서는 data의 이상처리에 봉착하게 되는 것을 방지하기 위하여 data 수정시 free path 기호를 설정하되 check program 2에서 free path가 있으면 그 data의 check를 생략하는 편법을 원용하면 보다 현실적일 수 있다.

free path 기호는 통상 문자를 사용하여 keyin 여백 column

에 수록하거나 조사년도·연령등 숫자항목 여백에 수록하므로서 check 조건을 완화하는 별도의 조치가 필요없이 정상 data의 무리한 수정을 피할 수 있으나, free path 기호가 부여된 data는 check 대상에서 제외되기 때문에 처음부터 keyin error가 발생하지 않도록 주의할 필요가 있다.

5. error data 처리의 표준화

error data의 각종처리 형식에서 살펴본 바와 같이 error data의 처리형식은 check program에서 요구하는 check 조건에 따라 결정되나 대체적으로 조사의 종류·data량·조사항목수 등이 증대하는 경향이고, data 상호간에 복잡하게 관련되어 수작업 check가 점점 어려워지며, 전산처리기술 향상으로 computer check 쪽으로 이행되어 시간과 노력을 절약하는 경향이기 때문에 표준화된 error 처리형식의 확립이 절실하고 집계과정의 걸림돌로 부각되는 error data 수정의 단순화로 작업량의 경감과 정확도 확보를 위하여 처리 system의 확고한 방향설정이 요구되는 실정이다.

가. 표준화의 필요성

error list에 의한 data 수정은 비단 전산처리 system 에서만의 문제가 아니고 수정·keyin등 수작업과도 불가분의 관계에 있기 때문에 처리상 번거로움이 의외로 클수 밖에 없으며, error data를 처리시한이 임박하여 처리하는 경우가 대부분이기 때문에 시간적인 여유없이 단기간내에 처리하도록 요구되고 이와 관련하여 보기쉽도록 작성된 error list, 용이한 수정작업체제, 능률적인 keyin 작업, 효과적인 처리 system의 확립등이 요청되고 있다.

특히 computer 처리 system의 졸속한 설계는 여타 수작업에 미치는 영향이 크기때문에 이들 상호간에 협력체제를 확립하므로서 원활한 집계가 보장되고 이러한 상황아래서는 error data 처리 syst-

em의 표준화와 범용화가 절실한 바, 집계처리 각부문별로 그 필요성을 열거하면 다음과 같다.

(1) 심사·정정작업

심사·정정작업과정에서 error list는 쉽게 이해할 수 있도록 편집되어야 하고 통상 error data에는 존재하는 error가 2-3개 항목정도에 그치고 그 내용은 단순한 기입착오이거나 keyin error의 경우가 대부분으로 이를 조사표와 대조없이 error list상에서만 직접 부분정정이 가능하도록 정정방법을 통일하여 작업과정에서 정정효율과 정확성을 기하기 위하여 정정문자를 간소화하여야 한다.

(2) keyin 작업

특히 card 단위수정에서는 조사표 전체 column을 다시 keyin하여야 하기 때문에 KES(key entry system) operator의 stroke 작업량 증가가 불가피하고 따라서 각 조사의 집계항목단위 수정방법을 채택하여 stroke수를 경감하고 keyin이 용이하도록 keyin 능율의 향상을 위하여 상하 좌우의 기본원칙에 입각한 list 양식을 마련하고 keyin 문자수를 최소화하여 keyin의 정확성을 유지하여야 한다.

(3) computer 작업

(가) 처리 system flow의 통일

처리 system flow는 조사에 따라 다소 차이가 있으나 전체적으로 대동소이하기 때문에 우선 이를 통일하여 표준화함으로서 program 작성과정에서 시행착오와 전산담당자의 혼돈을 피하여야 한다.

(4) program의 범용화

집계처리과정에서 정정작업·keyin 작업·처리 flow 등을 표준화함므로서 집계처리에 필요한 error list 편집 program이나 수정 program의 범용화를 통하여 program 작성효율을 높이고 집계시간을 단축하며 정확성을 유지하고 수작업부문간의 의사소통이 가능한

범용화의 이점을 최대한 살릴 수 있도록 하여야 한다.

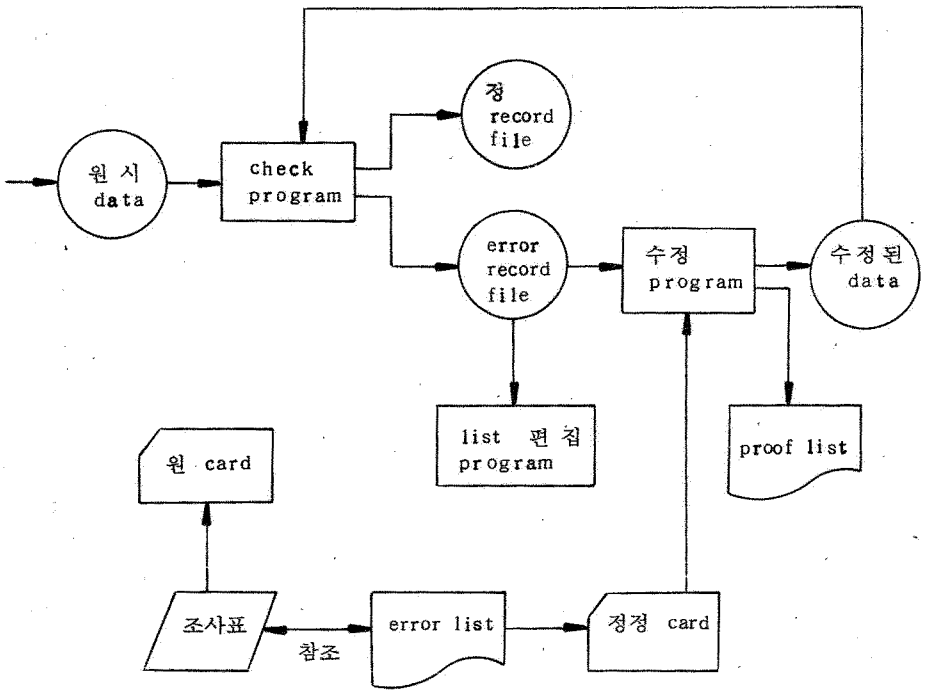


그림 2-13 data check 처리 flow

나. 표준화 system의 개요

(1) 처리 flow

그림 2-13의 data check 처리 flow를 예로하여 처리 system의 표준 flow에 대하여 설명하면 이제까지는 check program에 error list용 편집 routine을 조성하여 입력하는 것으로 되어있으나 이를 분리하여 check program에서는 정상적인 data와 error data를 나누어 output하고 error data에는 error의 종류를 나타내는 error 기호를 부여하여 error list용 편집 program을 통하여 error의 종류와 data의 내용을 특정한 form에 맞추어 print out 한다.

error list의 처리절차는 error의 상태에 따라서 조사표를 다시 keyin하는 것과 error list에 근거하여 정정 card를 작성하되 error list를 근거로 keyin한 정정 data는 수정 program에 의하여 error record file을 수정하고 다시 원래의 data와 같은 형식의 data를 만든다.

수정된 data는 조사표를 다시 keyin한 data와 더불어 check program에 의하여 재차 check하되 원칙적으로 error list 처리는 error가 없어질 때까지 반복하여야 한다.

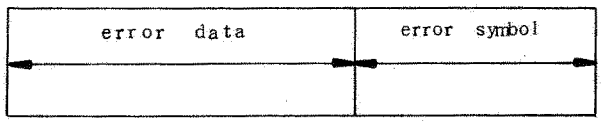


그림 2-14 error record 형식

(2) error record 형식과 error list 양식

error record 형식은 그림 2-14에서 보는바와 같이 error data 말미에 error symbol을 부여하되 error data는 input data 자체를 옮기고 error symbol은 error 종류별로 약속된 error 기호문자를 사용하여 표시한다.

항목번호
정정기입란
error data 표시
error symbol

정정의종류	error data 일련번호	01	02	03	04	05	06
		항목명 1	항목명 2	항목명 3	항목명 4	항목명 5	항목명 6
	x x x	01 xxx	02 xxx	03 xxxx xx	04 x	05 xxx	06 xxxx
	x x x	01 xx	02 xxx x	03 xxxx	04 x x	05 xxx	06 xxxx

그림 2-15

error list 양식은 장표형양식 (그림 2-15) 과 computer editing 양식 (그림 2-16) 으로 구분되는데 장표형식은 대규모 조사집계과정

에서 error list가 대량인 경우에 심사와 정정작업을 용이하게 하기 위하여 미리 항목명과 괘선을 인쇄한 양식이며, computer editing 양식은 조사규모에 관계없이 error list가 소량인 경우에 장표형 양식은 인쇄에 소요되는 시간과 예산이 방대하므로 computer sheet에 간략히 print하여 사용하는 양식을 채택하면 이용자가 보기 어려운 면이 있는 반면에 경제적으로 간단하게 사용할 수 있다는 장점이 있다.

	E.D NO.	01 00	02 000	03 0000	04 0	05 0000	06 000
항목번호 정정기입란 error data 표시 error symbol	xx	01 xx	02 xxx	03 xxxx	04 x	05 xxxx	06 xxx
		01	02	03	04	05	06
	xx	xx x	xxx	xxxx x	x	xxxx	xxx x

그림 2-16 computer editing 양식

정정의 종류에는 record 단위삭제와 항목단위치환이 있는데 record 단위 삭제 「삭제(Delete)」라 표기하고 「D/」 문자를 기입하며 항목단위 치환은 「치환(replace)」이라 표기하고 「R/」 문자를 기입하게 되는데 여기에서 삽입기능을 설정하지 않는 이유는 data 누락시에만 삽입기능이 필요하고 이때에는 부득불 조사표를 다시 key-in 하기 때문이다.

이미 알려진바와 같이 record의 update 작업은 약속된 순서대로 data 배열을 하여야하나 전술한 인쇄양식과 computer 양식의 두가지 장표를 근거로 keyin 하고 서로 다른 두가지 양식의 data를 조합배열해야 하므로 번거롭고 특히 집계작업은 부문별로 다른 group으로 취급하는 것이 편리하기 때문에 몇개부문으로 분할하여 작업하는 것이 상례이다. error record의 일련번호는 error record file의 배열순서에 따라 부여하는 번호로서 이를 근거로 error

record file의 해당 data를 수정하고 항목번호는 code상의 조사항목을 대상으로 좌로부터 순차로 부여하는 일련번호로서 이 번호가 지정하는 바에 따라 항목단위수정이 이루어진다.

(3) 정정순서와 정정 data의 keyin

그림 2-17에 예시된 error list에서 정정순서를 살펴보면 먼저 error symbol을 근거로 list상에서 단순정정이 가능한 것은 직접 정정하고 error list 상에서 간단히 정정할 수 없는 data는 list 상의 행정구역번호·총괄번호·조사표번호 등의 표시항목을 근거로 해당 조사표를 직접 심사하여 정정하되 error의 상황에 따라 조사표내용을 다시 keyin 하거나 error list 상에서 정정하여 항목단위 정정을 결정해야 한다.

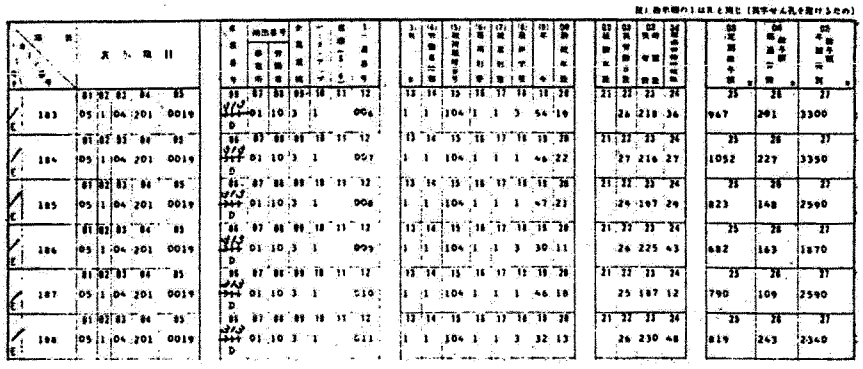


그림 2-17 error list 예

조사표내용을 다시 keyin 할 필요가 있는것에 대해서는 error record file 상에서 해당 record를 삭제하기 위하여 error list의 정정의 종류란에 「D/」 기호를 기입하여 해당 조사표를 key in 부서에 송부하고 항목단위정정에서는 「R/」 기호를 기입하되 정정문자를 항목번호 하단 공백에 유효수자에 이어서 「, (comma)」로 항목간구분이 가능하도록 하고 정정한 error list를 keyin 부서로 송부하여 완결하고 data의 추가가 필요할 때에는 조사표를 다시

keyin 하여 이송해야 한다.

정정 data의 keyin 요령은 error list의 정정종류란에 「 D/ 」, 「 R/ 」 기호를 기입한 것에 대하여 그림 2-18와 같은 정정card를 작성하여 처리하고 「 D/ 」, 「 R/ 」 기호가 없는것은 정상 data로 간주 처리한다.

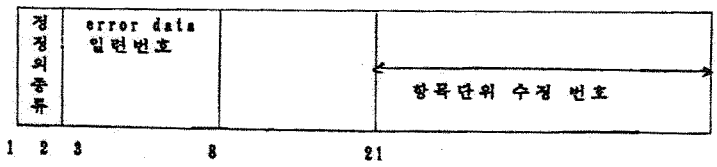


그림 2-18 정정 card 양식

record 삭제는 그림 2-19와 같이 정정의 종류 code와 error record 일련번호를 keyin 하고, 항목단위수정은 그림 2-20과 같이 항목단위수정 정보를 수록하는 21 column 이후의 column을 이용하여 정정대상항목을 keyin 하고 comma로 항목간을 구분하되 항목번호와 정정문자의 구분을 위하여 항목번호는 반드시 2 column을 keyin 하여야 한다.

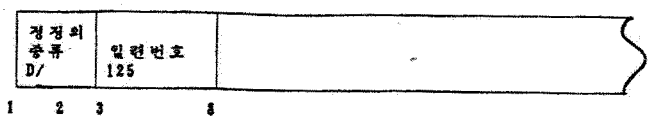


그림 2-19 삭제 card

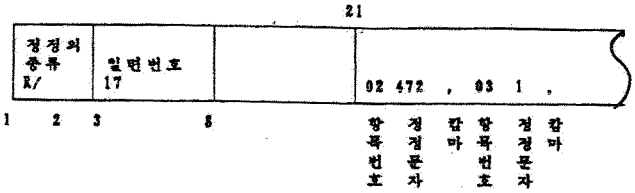


그림 2-20 항목단위 정정 card

(4) 범용 program

이 error data 처리 system에서는 check program 외에 error list 편집 program과 수정 program을 필요로 하는

데 error list 편집 program의 범용화는 error record 일련번호, 항목번호등 정정대상 번호를 부여하고 정정을 단순화하여 항목단위 수정이 용이하도록 해야하며 error data에 이어서 error symbol을 인쇄하는 병기형 print (그림 2-21) 와 error data · error symbol 을 분리하여 상하로 배열하는 분류형 print (그림 2-22) 의 어느 쪽에서나 표현이 가능해야 하고 정정 card 작성이 용이하고 인쇄와 computer editing 모두를 처리할 수 있도록 program을 설계하되 record 단위수정보다는 항목단위수정에 중점을 두고 error record 일련번호와 항목번호에 의하여 수정할 수 있도록 설계되어야 한다.

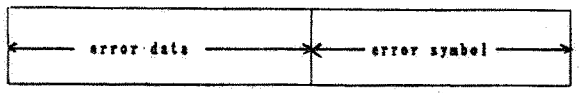


그림 2-21 병기 print 형식

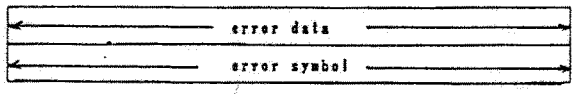


그림 2-22 분리형 print 형식

수정의 종류는 record단위의 삽입·치환·삭제 및 항목단위수정이 있는바, record단위의 삽입과 치환에 대해서는 전술한 바와 같이 수정 card 작성절차가 복잡하기 때문에 error list 상에서 정정이 가능한 record 단위삭제와 항목단위수정에 대하여 언급하고 삽입과 치환은 별도의 조사표를 근거로 입력하여 추가 data로 처리하도록 하고 정정문자는 유효숫자를 입력하는 형식으로 단순화하고 수정된 data는 check program에 의하여 재차 check 하여야 하기때문에 check program의 input data와 동일한 형식으로 작성하고 필요한 경우 수정의 성패를 확인하기 위한 proof list를 작성할 수 있어야 하고 error data 처리의 표준화에 대한 정의는 이용자 스스로 결정할

문제이기 때문에 시행착오가 없도록 가능한 한 단순화하여야 한다.

6. output data의 정보

check program의 output data에는 집계 program에 필요한 정보를 수록하고 당해 data의 정보원(源)을 나타내는 조사표번호와 총괄번호등 보조정보를 수록하여야 하는데 일반적으로 정보원을 규명하는 항목은 집계과정에 직접 필요한 항목이 아닐지라도 집계중 trouble의 해결과 data bank 등 data의 외부이용자를 위하여 반드시 필요한 항목이다.

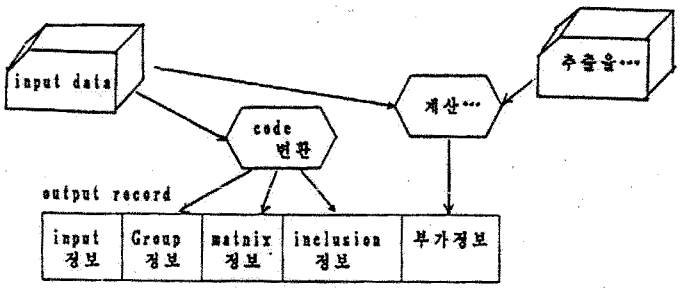


그림 2-23 output 정보

이와같이 축적하고 수록할 필요가 있는 정보를 정리해 보면 check program output에는 input정보·group code정보·matrix code정보·inclution code정보 및 기타 부가정보를 들 수 있다.

가. input 정보

input data의 정보는 computer process는 반복집계처리가 가능하지만 실사를 위시한 수작업은 다시 실시하기 어려우므로 목적하는 바 집계외에 장래의 통계수요변화에 대처하기 위하여 불필요한 항목일지라도 되도록 많은 양을 output에 수록하는 것이 바람직하다.

input data 항목에 따라 code변환을 통하여 group code·matrix code와 같은 별개의 정보형태로 변환하고자 할 때 code 변환에 따르는 progrm error의 방지를 위하여 개별code를 수록

하고 재연산(rerun)할 경우에 check program에서 처럼 수작업부분과의 trouble이 있을때 code변환부분만을 발취하여 집계 program 작성과정에서 간편하게 대처할 수 있도록 하여야 한다.

나. group code 정보

check program의 주요기능중 하나가 code변환기능으로 input data code가 있는것은 조사설계단계에서 전산처리보다는 인력경감면에서 조사표기입·내용검사·부호기입·keyin 등 수작업과정의 편이도에 중점을 두고 설계하여야 하기때문에 전산처리에 부적합하거나 무리가 따르는 경우가 허다하여 check program으로 집계과정을 간소화할 수 있도록 통일된 code체계로 변환하기 위한 조치중 하나가 group code이다.

group code란 특정항목의 code를 몇개 group으로 나누어 code화한 것으로 인구통계의 연령계급별 통계표에서 조사표에 실제연령만 기재하고 5세, 10세등 연령계급별 code를 생략하여 조사표작성과정과 keyin 과정에서 group화작업 없이 집계과정에서 간단히 group화작업을 하도록 한다.

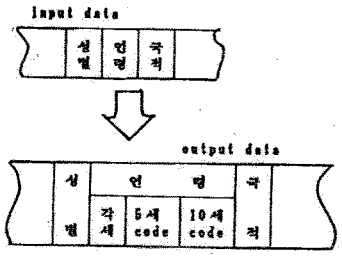


그림 2-24 group code의 예

5세계급, 10세계급등 통계표종류가 많을때에는 집계 program에서 필요로 할 때마다 code변환을 하려면 번거롭고 error를 범할 소지가 크기 때문에 check program에서 code변환을 하여 일괄처리하고 또 통계표에서 불필요할지라도 전산처리상 유리하면 변환code를 하되 직업·산업분류 code, 연령계급별 code등 결과표에서 필수적인 사항이외에 당장 불필요한 구시군 code(예:구부=0, 시부=1, 군부=

2)등도 후일 data이용에 기여할 수 있도로 사전에 변환 code를 부여하는 것이 편리하다.

다. matrix code 정보

통계표작성을 용이하게 하기 위해서는 표측행과 표두열(란)을 일련번호code로 표현하여야 하는데 이와같은 2차원적 code를 matrix code라 하고 통계표의 2차원적행열을 matrix라 한다.

그림 2-25 (통계표의 분포예)의 예에서 통계표를 수작업집계를 가정할 때 좌측의 통계요소를 우측의 유효한 통계표로 정리하려면 연령을 배열할 표두의 교차점 (해당 cell)에 1을 더해야 하는데 이러한 작업을 전산처리할 때에는 수작업을 통하여 표측, 표두의 위치를 결정하는 것과는 달리 표측행과 표두열을 일일히 input data와 비교하여 교차점을 확인해 나가야 한다.

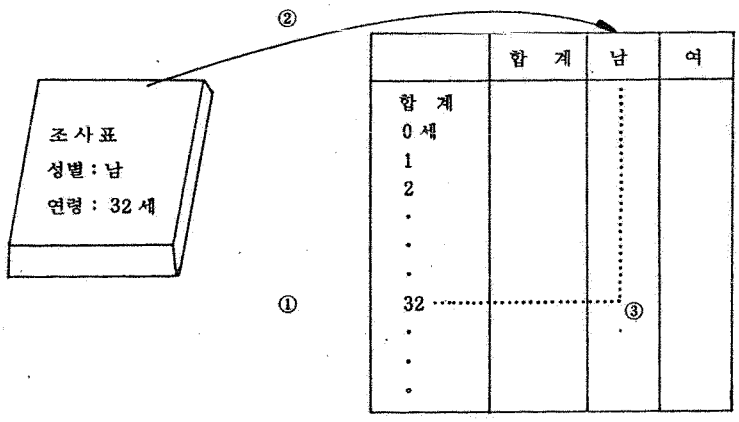


그림 2-25 통계표의 분포

그러나 computer에서 원리적인 방법으로 교차점을 확인하여 matrix를 확정하려면 너무 많은 시간이 소요되기 때문에 input code를 표측행수와 표두열수가 일치하도록 설정하면 첨자식에 의하여 즉시 해당 cell을 찾아내게 되므로 비교시간을 최소한으로 단축할 수 있다.

이를 그림 2-25를 통하여 설명하면 연령 32세의 행은 위에서 34번행,

성별란의 남자는 좌에서 2번열이므로 표측행은 연령 code + 2 (32 + 2 = 34), 표두열은 성별 code + 1 (1 + 1 = 2)로 미리 결정해 놓으면 행과열의 비교없이 즉시 해당 code를 찾아서 matrix를 구성할 수가 있다.

이와같이 통계표의 표측·표두결정에 직접 참여하는 code를 matrix code라 하며 이와같은 matrix code는 표측·표두를 나타내는 일련번호가 이상적이나 간단한 계산식으로 구할 때에는 input code를 이용해도 무방하며 직업분류, 산업분류등과 같이 분류번호가 규칙적으로 부여되어 있을때에는 matrix code란을 설정하는 것이 상식이다.

matrix code는 통계표의 분포를 용이하게 하기 위하여 설정 부여하는 것이므로 후술하는 분포 program에서의 통계표의 memory 할당과 밀접한 관계가 있기때문에 memory 할당이 바뀌면 matrix code도 더부러 바뀌는데 matrix code에 대한 설명은 분포 program 설계에서 상론하기도 한다.

라. inclusion code 정보

통계표작성에 있어서는 표마다 대상 data를 정확히 파악하되 대상 data의 선택이 복잡한 경우에 routine을 잘못 선택하거나, 틀리는 data를 대상으로 할 가능성이 크고 routine 선택에 잘못이 없어도 통계표를 작성할 때마다 선택 routine을 반복하는 것은 번거롭기 때문에 이미 check된 data의 대상범주를 나타내는 특별한 code를 설정하여 정상적인 대상data를 간단히 선택할 수 있는바 이와같은 code를 inclusion code라 한다.

inclusion code는 통계표별로 설정하는 것이 상식이나 data group별로 설정해도 무방하고 통계표를 data별 group으로 분류할 수 있을때에는 data group별로 부여해도 무방하다.

집계된 통계표사이에서 일치하여야 할 항목의 숫자가 틀릴때 그 원

인을 분석해보면 대상data의 선택이 잘못된 것이 대부분인 바 특히, 집계 program을 여러사람이 분담작성하는 경우에 나타나기 쉬운 현상이다.

마. 부가정보

부가정보는 input 항목에 없는 정보를 추가하는 기능을 갖는것으로서 봉급표 code로부터의 봉급액, 행정구역별·조사구별 weight 등과 같이 input data에 수록된 산출근거로서의 기초정보를 보완하기 위하여 constant data와의 대조를 통하여 급여액, weight등 최종정보를 조정하여 output으로 수록하는것 등이 이에 속한다.

7. table search 기법

check program에서는 off code check나 관련 check를 막론하고 table을 이용하여 check하는 사례가 많으며 따라서 여러형태의 table search를 특성에 따라 구분사용하여야만 처리시간의 단축과 program의 효율을 기대할 수 있기때문에 check program의 근본기술인 table search, binary search, binary range search로 구분되는데 각각에 대하여 설명하면 다음과 같다.

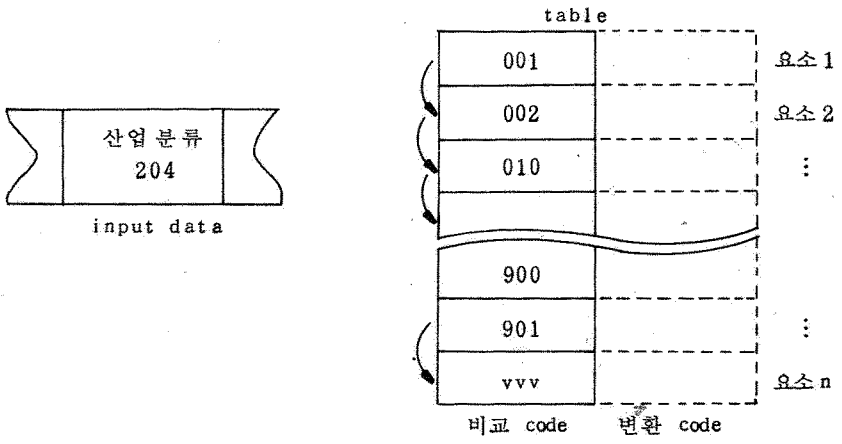


그림 2-26 serial search의 예

가. serial search

table의 요소를 시종 순차대조하여 input 항목 code에 상응하는 요소를 check하는 방법으로 비교 code와 변환 code로 구성되는 table 요소에서 code변환을 필요로 하지 않을 경우에는 변환 code 부분을 생략하거나 program을 작성할때마다 비교code와 변환code를 인접시키지 아니하고 별개의 table을 만드는 경우도 있다.

그림 2-26 (serial search의 구성)에서 산업분류code 204에 대하여 check할때 table 각요소를 정상적인 산업분류code로 구성하여 이 table을 그림 2-27의 flowchart 순서에 따라 search하는 것으로 하면 serial search의 flowchart 순서에 따라 search하는 것으로 하면 serial search의 flowchart는 간단하나 요소수가 증가하면 search 시간이 기하급수적으로 증가하는바 이는 table 요소를 모두 search하여야만 FALSE(찾아내지 못함) 판정이 가능하고 TRUE(찾아냄)도 평균 N/2회를 search해야 하기 때문에 search 시간단축을 위해서는 binary search 등의 대안이나 다음과 같은 개선방안을 강구해야 한다.

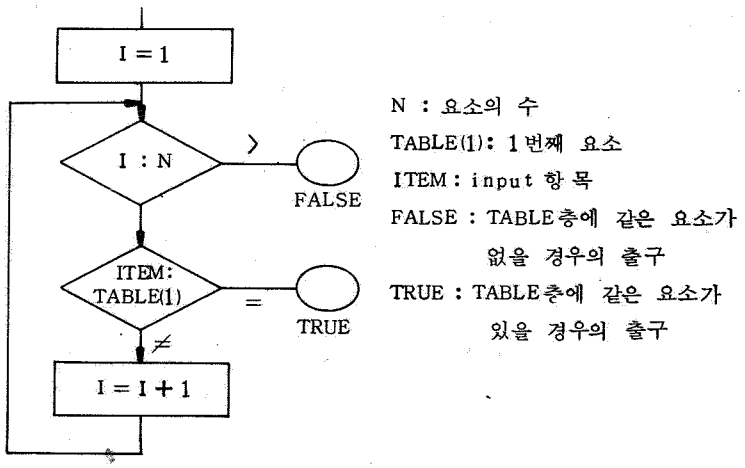


그림 2-27 serial search의 flow

첫째, table 요소를 출현빈도순으로 배열하여 data 중에서 가장 많은 code를 table의 제 1 요소로 하고 순차로 요소를 지정하여 table을 구성하는 방법으로 이때의 출현빈도는 동종의 기존통계자료를 통하여 결정하고, 둘째 FALSE의 신속한 판정을 위하여 요소를 sequential(출현빈도의 적은순)로 배열하여 항목code가 요소보다 적으면 FALSE로 결정하는 방법이 있다.

이상의 개선방법중 택일문제는 조사성격에 따라서 판단할 문제이지만 일반적으로 error data가 정상 data에 비하여 극히 소수임을 감안하여 정상 data를 최소의 회수로 search 할 수 있는 첫째방법이 유리하다.

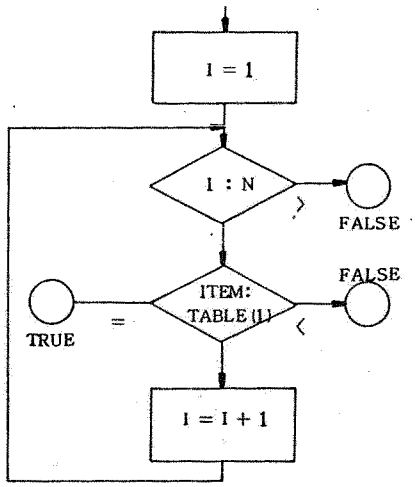


그림 2-28 제 2 방법

나. class search

금액계급, 종업원계급등과 같이 실액(實額)의 범위에 따라서 계급 code를 부여할 경우에 편리하게 사용할 수 있는 search 방법으로 그림 2-29는 금액의 범위에 따라서 계급 code를 부여하여 table요소에 각 계급의 상한치를 set하고 input data의 금액란과 table 요소를 순차로 비교하는 class search 예이며, 그림 2-30의 flow-

chart는 serial search와 유사하나 요소수 N와 1의 비교과정이 생략된 것으로 계급 code 부여와 같은 group code로의 변환은 정상 data 만을 대상으로 하기때문에 table의 특정한 요소에서 TRUE가 되기 마련이다.

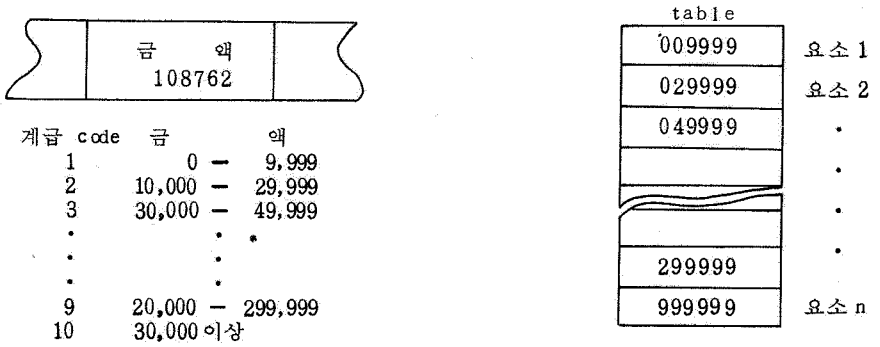


그림 2-29 class search의 예

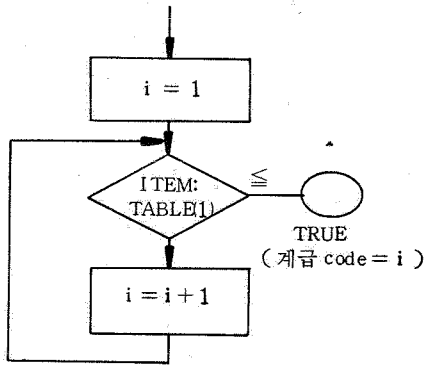


그림 2-30 class search의 flow

이예는 계급code를 TRUE 시점에서의 1의 값으로 부여하는 방법이며, 한편 table에 변환code 부분을 추가한 값을 이용하는 방법이 있는데 이 방법은 불규칙적인 계급code에도 대처할 수 있는 장점이 있다.

다. binary search

binary search는 일명 2분탐색법이라고 하는바 table을 전·후반 영역으로 양분하여 소속영역을 check하고 이를 다시 전·후반으로 양분하여 반복 check 하되 그이상 양분할 수 없을때까지 계속하여 양분이 불가능한 시점이 table에 존재하지 않는 FALSE로 판정하는 search 방법으로 binary search의 도해와 flow는 그림 2-31, 2-32와 같다.

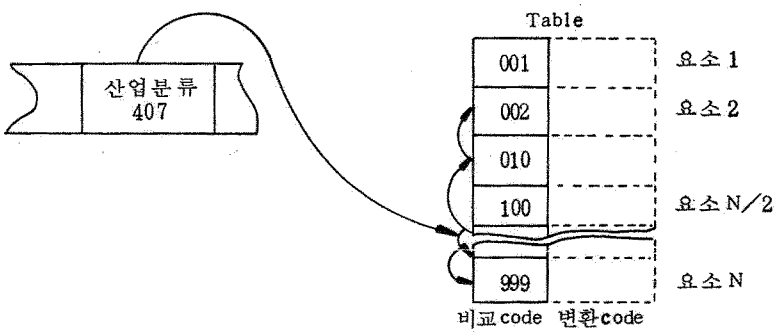


그림 2-31 Binary Search

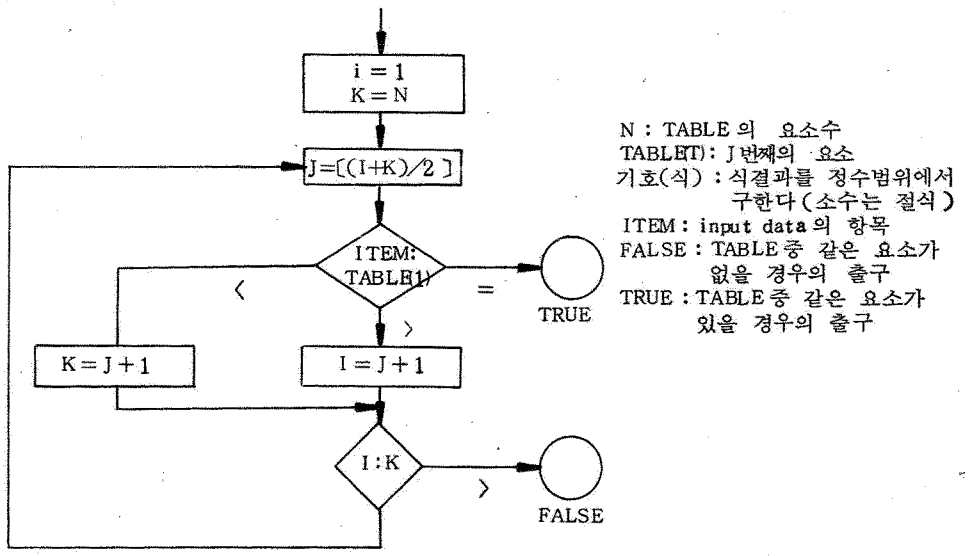


그림 2-32 Binary Search의 Flow

그림 2-33과 같이 table 요소수를 12로하는 input data 16의 binary search의 예를 들면 다음과 같다.

(1) I＝1, K＝12에서
J＝〔(I＋K)/2〕＝(13/2)＝6이 되어 요소 6과 비교한다.

(2) ITEM「16」〉요소6「12」이므로 요소7～12의 check를 위하여 I＝J＋1＝6＋1＝7, K＝12에서 J＝〔(I＋K)/2〕＝(7＋12)/2＝9가 되어 요소 9와 비교한다.

(3) ITEM「16」〈요소 9「18」이므로 I＝7, K＝J－1＝9－1＝8에서 J＝〔(I＋K)/2〕＝(7＋8)/2＝7이 되어 요소7과 비교한다.

(4) ITEM「16」〉요소7「14」이므로 I＝J＋1＝7＋1＝8, K＝8에서 J＝(I＋K)/2＝(8＋8)/2＝8이 되어 요소8과 비교한다.

(5) ITEM「16」＝요소 8의 값이되므로 TRUE가 되어 종료한다. 이때 ITEM 값이 「17」이라면 순서 (1)～(4)까지는 check 하고 요소 8의 값이 「16」〈ITEM 「17」이므로 I＝J＋1＝8＋1＝9, K＝8이 되어 I〉K가 성립되고 FALSE로 판정된다.

이상의 binary search 순서에서 보는바와 같이 TRUE, FALSE의 판정은 적은회수의 작업으로 판단이 가능하기 때문에 탁월한 방법이라 할 수 있으나, sequential배열을 해야하는 단점이 있다.

라. binary range search

binary search는 시간이 빠르고 효율적이나 table 요소가 대량(2-300)인 때에는 table 작성과 table의 정확성 check 등의 search 작업량이 많아지는 단점이 있다.

이와같이 table 요소가 많을 경우에는 요소가 부분적으로 연속번호로 되어 있는것이 많기때문에 상·하한치를 쌍으로 부여하여 요소수를

대폭 줄일 필요가 있고 table작성과 check가 용이하나 off code인 경우에는 각별한 주의가 필요하며 요소의 성격상 range가 아닌 단독일때에는 상·하한치를 같은 값으로 처리하여야 한다.

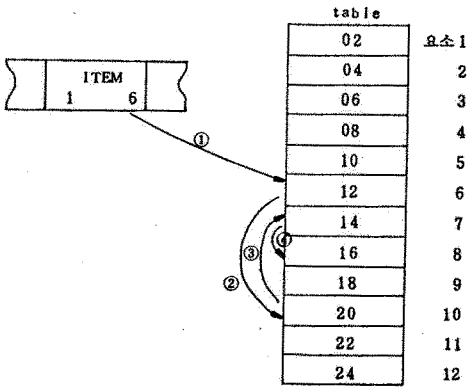


그림 2-33 Binary search의 예

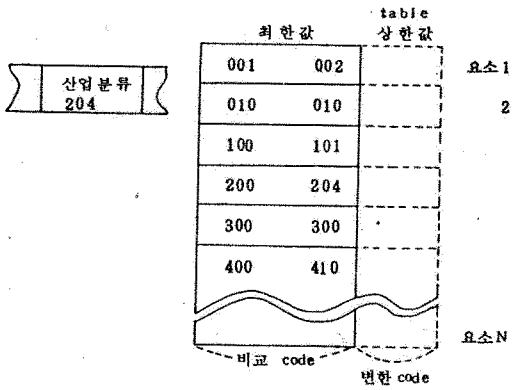
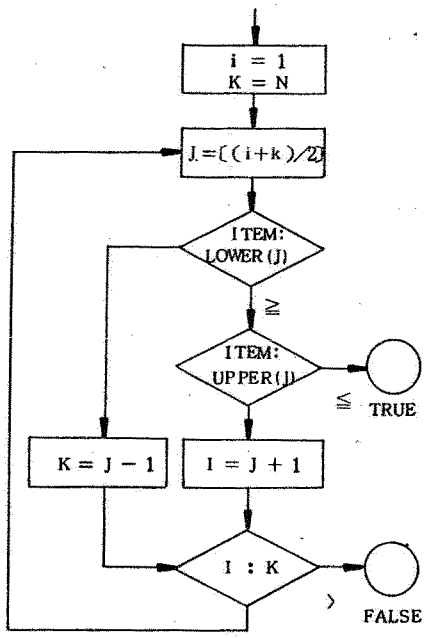


그림 2-34 binary range search



N : table 요소수
LOWER(J): J번째의 하한 값
UPPER(J): j " 상한 값
ITEM: INPUT DATA 항목
FALSE : table 중 같은 요소가 없는 경우 출구
TRUE : table중 같은 요소가 있을 경우 출구
기호[ ] : 식의 결과를 정수범위에서 구한다.(소수절사)

그림 2-35 Binary range search의 flow

Ⅲ. 분포 PROGRAM의 설계

1. 분포Program의 기본개념

통계표작성의 핵심부분은 도수분포를 비롯한 각종 분포표작성으로서 check program은 분포표작성에 필요한 각종부호에 대한 check와 분포표 작성수단으로서의 code 부여가 주기능으로 설명한바 있으나 분포program 본래의 기능은 check program을 통하여 생산한 올바른 data output을 근거로 한 분포표작성에 있는 것이다.

그림 3-1은 분포program과 table image가 computer memory에 축적되는 상태를 도표화한 것으로서 분포program을 실행하면 input data를 read할 때마다 분포표 1～5에 해당하는 table image에 분포하고 input data가 없어질때까지 반복처리한다.

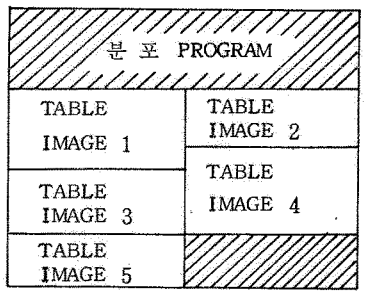


그림 3-1 MEMORY내의 상태

table image는 memory size의 제약으로 결과표와 동일한 수량의 cell의 확보가 불가능하거나, 가능하다 하더라도 비경제적일 경우에는 합계란등 추후에 계산해도 무방한 cell을 배제하고 결과표의 진수(眞髓)로 구성된 table을 memory에 입력하므로서 효율적인 memory운용이 가능하기 때문에 반드시 결과표와 같지 않을수도 있다는 의미에서 결과표란 용어와 구분하여 사용하고저 한다.

대체로 통계조사결과표 table수는 수십종에 이르는 방대한 량이고 결과표크기(cell수)가 크기 때문에 하나의 분포program으로 전체 결과표의 작성이 사실상 불가능하므로 분포program을 여러개의 subprogram으로 분할처리하게 되는데 이와같은 분할처리는 input data의 반복처리가 불가피하여 집계시간이 길어지므로 분포program을 설계함에 있어서 program내용을 단순화하고 program수를 최소화하는 것을 기본으로 하여야 한다.

mechanism상에서 분포란 「어떤 data에 대하여, 어느 cell에, 무엇을 분포할 것인가」로 집약할수 있는바 이를 분설하면 「결과표의 어떤 cell에 분포할 것인가」의 문제는 그림 3-2와 같이 결과표를 2차원행열로 보고 표두를 열, 표측을 행, 분포할 결과표의 단위cell을 P(X, Y)라 할때 X, Y는 표측 X행과 표두 Y열의 교차점임을 의미하므로 그림 3-2의 결과표 program에서 2차원배열로 정의하고 X, Y만 알면 X, Y를 첨자로하여 P를 구할 수 있다.

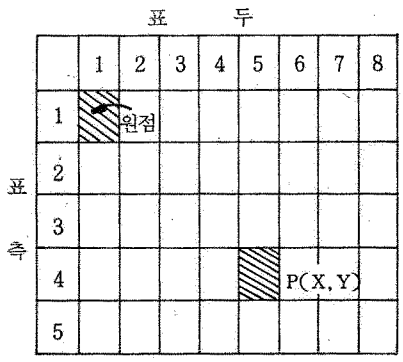


그림 3-2 결과표의 행열

그러나 통계표의 표두·표측은 여러개 항목이 계층구조로 되어 표두항목과 표측항목의 code 그대로는 X, Y의 값으로 이용할 수 없기 때문에 P점을 계산하기 위해서는 표두항목과 표측항목 code를 1부터 시작하는 일련번호로 변환할 필요가 있는바 이와같은 일련번호를 matrix code라 함은 이미 설명한바와 같다.

다음 「어떤 data에 대하여 …」는 각 결과표의 대상data의 선택을 의미하고, 「무엇을 …」은 결과표의 요구에 따른 도수, 계량항목의 분포를 의미한다.

분포program 설계과정에서의 유의사항을 요약하면 다음과 같다.

① Table image의 할당
② 합계란의 설정과 삭제
③ Matrix code
④ Displacement
⑤ 분포 항목
⑥ Inclusion code
⑦ 상위 계급으로의 합산
⑧ Summary record의 printout

2. 분포형태

통계표는 난외항목·표측항목·표두항목의 3개항목요소로 구성되고 data의 분포유형은 단위data가 단위표에 1회 분포하는 경우와 2회 분포하는 경우가 있는바 결과표를 분포형태별로 분류하면 기본분포형(R×S×H), 수평분포형(type1:R×S×h, type2 : R×S×ΣH), 수직분포형(type1: R×s×H, type2 : R×ΣS×H), 혼합분포형(type1: R×s×ΣH, type2 : R×ΣS×h, type3 : R×ΣS×ΣH), zigzag분포형(R×s×h)의 5개류형으로 분류할 수가 있다.

이때 R : 난외항목, S : 표측항목, H : 표두항목, 대문자 : sort가능항목 소문자 : sort 불가능항목을 의미하고 sort가능여부는 그 항목의 sort가 집계상 의미가 있는가의 여부를 의미하는 것이다.

가. 기본분포형 (R × S × H)

기본분포형 결과표는 그림 3-3과 같이 가장 기본적인 분포형식의 표로서 개별data에 대하여 결과표상에서 선택된 특정한 cell에 1회 분포만 가능한 type이다.

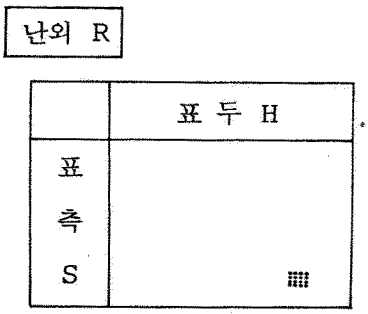


그림 3-3 기본 분포형

나. 수평분포형

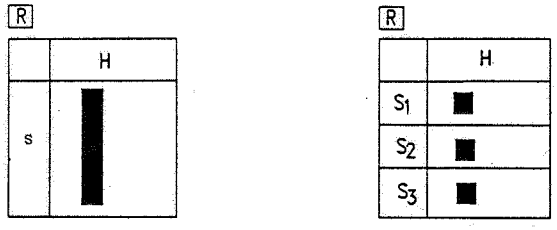
수평분포형 결과표에는 그림 3-4와 같이 type1(R × S × h)과 type2(R × S × ΣH)의 2개 유형이 있는바 type1은 단위 개별data를 표두항목에 따라 분포하는 형식으로 이러한 type의 통계표에서는 input data를 표두항목으로 sort할수 없는 단점이 있으며 type 2는 개별data 내용을 표두분류 ΣH에 따라 단위cell에 분포하는 형식으로 여러개의 기본분포형을 S에 대하여 정리한 유형에 속한다.

type 1 type 2

그림 3-4 수평 분포형

다. 수직분포형

수직분포형 결과표는 그림 3-5와 같이 type1(R×s×H)과 type 2(R×ΣS×H)의 2개 유형이 있는바, type1은 수평분포형 type1에 대응하여 개별data를 표측항목 S에 대응시키는 분포로서 이 유형의 통계표에서는 input data가 S에서 sort가 불가능한 형식이며 type2는 수평분포형의 type2에 대응하여 개별data를 ΣS의 단위cell에 분포하는 형식으로 수평분포형과 같이 기본분포형을 H에 대하여 정리한 유형에 속한다.

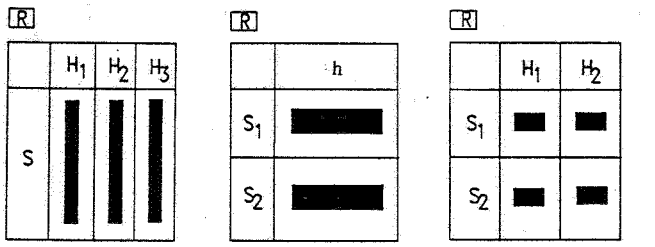


TYPE 1 TYPE 2

그림 3-5 수직분포형

라. 혼합분포형

혼합분포형의 결과표는 그림 3-6과 같이 type1 (R×s×ΣH), type2 (R×ΣS×h), type3 (R×ΣS×ΣH)의 3개 유형이 있는바 type1은 개별data를 표두 ΣH의 여러개 열 S에 대하여 분포하고 type2는 개별data를 표측 ΣS의 여러행 H에 대하여 분포하며 type3은 개별data를 표측 ΣS, 표두 ΣH에 따라 분포하는 형식이다.



TYPE 1(R*s*ΣH) TYPE 2(R*ΣS*h) TYPE 3(R*ΣS*ΣH)

그림 3-6 혼합분포형

마. zigzag 분포형 (R × s × h)

zigzag 분포형은 그림 3-7과 같이 개별data를 zigzag로 S, H에 대하여 분포하는 형식으로 zigzag 분포형을 포함하는 5개분포형태는 개별data의 형식과 깊은 연관아래 유형이 결정되는데 소비지출통계(예)에서 개별data를 세대별 소비항목기준, 단위세대의 소비항목 연결기준 여하에 따라서 기본·수직·수평 등의 분포형태가 결정된다.

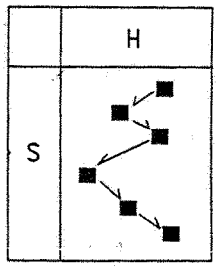


그림 3-7 ZIGZAG 분포형

3. table image 할당

단위 분포program에 수십개 표본의 table image을 확보해야 할 경우 각 표의 table image를 memory에 할당하는 문제는 분포program의 핵심기능으로서 memory 할당결과에 따라 분포program의 성패가 좌우되고 전체 집계system에 심대한 영향을 미치게 된다.

가. 단표(單表)할당

table image의 memory 할당방법은 그림 3-8의 예에서 그림좌측의 표측항목은 연령, 표두항목은 남여별, 난외항목은 행정구역을 지정하는 연령별 남여별 인구에 관한 통계표를 통하여 그림 우측과 같이 전체 행정구역을 망라하는 행정구역별집계와 전국합계를 합한 16매의 결과표생산이 가능한데 전체결과표의 크기(column 수)는 102행(표측)

× 3열 (표두) × 8 column(cell) = 2,448 column이 되고 단위 cell의 크기는 총인구 4천만을 cover할 수 있는 8 column으로 cell의 크기는 전체계열중 가장 큰 전국합계 column 수를 기준으로 하여 결정하여야 한다.

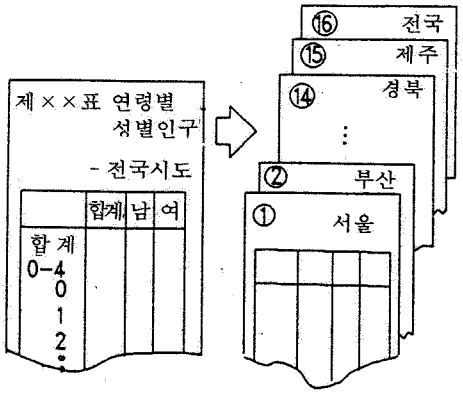


그림 3-8 통계표의 내역

따라서 그림 3-8의 결과표할당은 간단하게는 1 ~ 16까지의 16매분을 memory에 입력하여야 하는데 이때 소요 memory는 2,448 column × 16매 = 39,168 column이 되나 input data를 순차로 입력하지 않더라도 이미 분포장소를 memory에 확보한 상태이므로 input data의 sort를 생략하고 전국합계표는 1 ~ 15의 적산결과이므로 memory 입력을 생략해도 무방한 장점이 있다.

이와같이 할당방법은 집계대상 통계표가 sort 불가능항목으로 구성되거나 input data량의 과다로 sort시간이 장시간 소요될때 주로 사용하는 방법이다.

다음 방법은 시·도의 통계표를 할당하는 방법으로 이때의 결과표 크기가 2,448 column으로 전국규모의 크기 39,168 column에 비하여 memory를 적게 점유하나 input data를 sort하여 행정구역별로 group 화하여 단위 data group을 read할때 이미 분포된 다른 da-

ta group을 수록하는 순차적 연속처리가 가능하다.

이와같이 각행정구역 group별로 table image 내용을 print out하고 zero로 clear한후 group의 분포를 계속 수행하는 control을 control brake(행정구역별)라 하고 input의 행정구역란을 control 항목이라 하는데 이를 환언하면 control항목 내용이 같은 값일 동안은 분포를 계속하고 값이 다르면 print out하게 된다.

control은 집계program을 비롯한 각종 program에서 중요한 개념으로 특히 분포program은 control 설정상황에 따라 성패를 좌우하는만큼 분포program과 control설정에 대한 이해를 돕기 위하여 행정구역별·연령별·성별통계를 예제로 memory의 최소화 방법을 살펴보기로 한다. 전례에서는 control항목을 행정구역(시·도)만을 대상으로 하였으나 표측 연령을 control항목으로 추가하여 2개 항목을 control항목화하면 표측1행의 소요 memory size는 1행(표측)×3열(표두)×8 column(cell)=24 column이 되고 이때는 행정구역·연령으로 control brake되는 것이 마땅하나 이와같이 control항목이 복수일때의 control brake 설정은 복수의 control항목을 단일항목으로 간주하고 동일한 값인가를 조사하여야 한다.

여기에 다시 control항목에 성별을 추가하면 이때의 소요 memory size는 1행(표측)×1열(표두)×8 column(cell)=8 column으로 대폭 축소되는 반면 input data는 행정구역>연령>성별로 sort하여야 하는데 이 기호「>」는 그림 3-9의 sort tree에서 보는바와 같이 성별내용을 적산하여 연령, 연령을 적산하여 행정구역을 구성하는 것을 의미하게 된다.

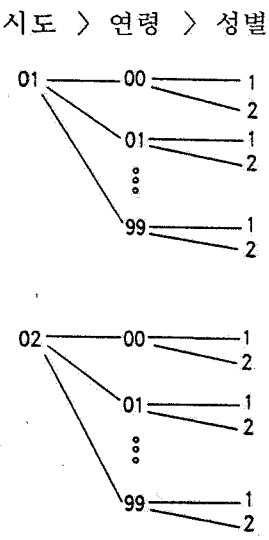


그림 3-9 SORT TREE

기본분포형 (R × S × H) 에서 R = 16 구분, S = 100 구분, H = 12 구분으로 하고 단위 cell 의 column 수를 10 column 으로 가정하면 memory 할당수는 분포형 표현식에서 최대소요 memory 수 (R × S × H × L = 16 × 100 × 12 × 10 = 192,000) 를 산출하고, 이어서 memory 수를 줄일 경우에는 표현식의 대문자항목을 R.S.H 순으로 control 항목에 편입하여 다시 계산해야 한다.

먼저 R를 control 항목화하면 R × S × H × L = 1 × 100 × 12 × 10 = 12,000 이 되어 memory 수가 192,000 에서 12,000 으로 대폭 축소되고 다시 R.S를 control 항목화하면 R × S × H × L = 1 × 1 × 12 × 10 = 120, R.S.H를 control 항목화하면 R × S × H × L = 1 × 1 × 1 × 10 = 10 으로 축소되고 L (length : column 수) 에 착안하여 10 진계산을 2 진화하면 더욱 축소할 수 있다.

나. 복수표의 할당

그림 3-10 과 같은 4 개표본의 통계표 할당 예를 바탕으로 복수표의 할당에 대하여 살펴보면 이때에 표두항목은 표마다 공통부분이 없이

서로 다르고 표측항목은 제 1 표와 제 2 표가 A항목이면서도 1.2로 구분되고 제 3 표와 제 4 표는 B항목이면서 1.2로 분류될때 그림 3-10의 4개표를 memory에 모두 수록할 수 있는가를 판단해야 하므로 각표의 cell수를 계산하지 않으면 아니된다.

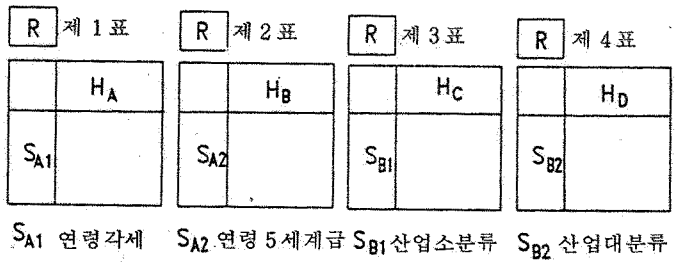


그림 3-10 복수표의 MEMORY 할당 예

이때에 제 1 표 : $M_1 = R \times S_{A1} \times H_A$

제 2 표 : $M_2 = R \times S_{A2} \times H_B$

제 3 표 : $M_3 = R \times S_{B1} \times H_C$

제 4 표 : $M_4 = R \times S_{B2} \times H_D$

합 계 : $M_T = M_1 + M_2 + M_3 + M_4$가 되어 여유 memory에 M_T를 수록이 가능할 때 표의 할당과 같게 되나 memory가 부족하면 난외항목 R는 공통항목이므로 R를 control 항목화하여 계산해야 한다.

이때의 계산결과는 제 1 표 : $M_1 = S_{A1} \times H_A$

제 2 표 : $M_2 = S_{A2} \times H_B$

제 3 표 : $M_3 = S_{B1} \times H_C$

제 4 표 : $M_4 = S_{B2} \times H_D$

합 계 : $M_T = M_1 + M_2 + M_3 + M_4$의 cell수를 필요로 한다.

이로서 할당작업이 종료되나 memory수가 그래도 부족하거나, me-

mory를 더욱 축소할 필요가 있을 때에는 control 항목에 S를 추가하여 R, S를 control 항목화하여야 하는데 표측항목 S는 A . B 항목으로 분류되므로 control brake 범위를 R_1, S_{A1} 또는 R_1, S_{B1} 으로 하는 2개방법을 고려할 수 있다.

이때에 control 항목을 S_{A1}으로 하면

제 1 표 $M_1 = H_A$

제 2 표 $M_2 = H_B$

제 3 표 $M_3 = S_{B1} \times H_C$

제 4 표 $M_4 = S_{B2} \times H_D$

합 계 $M_T = M_1 + M_2 + M_3 + M_4$ 가 되고

S_{B1} 을 control 항목으로 하면

제 1 표 $M_1 = S_{A1} \times H_A$

제 2 표 $M_2 = S_{A2} \times H_B$

제 3 표 $M_3 = H_C$

제 4 표 $M_4 = H_D$

합 계 $M_T = M_1 + M_2 + M_3 + M_4$ 이 되므로 양자중 합계 cell 수가 적은 항목을 control 항목으로 결정하여야 한다.

여기에서 주의할 점은 S_{A1} 와 S_{B2} 를 control 항목으로 선정하였을때

S_{A1} 의 경우 M_1 , M_2 (S_{B1} 의 경우 M_3 , M_4)가

$M_1 = H_A$ $M_3 = H_C$

또는

$M_2 = S_{A2} \times H_B$ $M_4 = S_{B2} \times H_D$ 가 되지 않은 이유는 S_{A1} 또는 S_{B1} 의 부분적인 합계가 각각 S_{A2} 또는 S_{B2} 의 구성단위이기 때문이며 만약 이관계가 성립되지 않을 경우에는 $M_2 = S_{A2} \times H_B$, $M_4 = S_{B2} \times H_D$ 로 표현할 필요가 있다.

이와같이 S_{A1} 으로 S_{A2} 를 대표하고 S_{B1} 으로 S_{B2} 를 대표하는 예로

는 연령 5세계급과 각세, 산업대분류와 소분류의 관계를 들 수 있다.

이상과 같은 할당작업에도 불구하고 4개표에 대한 할당이 동시에 성립되지 않을 경우에는 또 다시 control 항목을 추가하여 $R > S_{A1} > S_{B1}$ 의 순으로 처리하면 합계 cell 수는 각표의 표두부분 cell 수를 합한 것과 같은 수가 되어 cell 수가 현격히 축소되는 반면 제 3.4표의 output record 수가 S_{A1} 배로 증가하기 때문에 오히려 처리시간이 증가하는 결과가 되므로 이방법을 채택할 때에는 record 수가 증가하지 않도록 S_{A1} 과 S_{B1} 중 상위 control 항목 선정에 신중을 기하여야 한다.

마지막으로 program 분할방법을 들수 있는바, control이 같은 표끼리 모아서 제 1.2표를 하나의 program으로 묶고 제 3.4표는 별개의 program으로 통합작성하는 것으로 작성program은 2개이나 output record 수의 증가를 방지하는 장점이 있다.

4. 합계란의 설정과 삭제

control 항목을 도입하여 table image size를 최소화하는 방법 외에 통계표의 합계란을 삭제하여 table image size를 축소하는 또 다른 방법으로 그림 3-11의 통계표에서 합계란(사선부분)이 전체의 53%이기 때문에 합계란의 삭제여부는 memory 할당에 커다란 영향을 미치게 되므로 결과표양식이 크고 결과표수가 많으며 집계data 량이 방대할 때에는 표측·표두의 합계란을 삭제하므로서 table image의 크기를 대폭 축소할 수 있다.

	합 계					남					여				
	계	미혼	기혼	사별	이혼	계	미혼	기혼	사별	이혼	계	미혼	기혼	사별	이혼
합 계															
15～19세															
20～24															
25～29															
30～39															
40～49															
50～59															
60～이상															

전체 cell 수 : 8 line × 3 × 5 = 120
합계 cell 수 : 3 × 5 + 7 (5 + 2) = 64
합계란의 비율 : (64/120) × 100 ≒ 53 %

그림 3-11 통계표의 합계란

합계산출을 위해서는 분포program에 의하여 print out된 summary record를 바탕으로 후술하는 합산program과 가공편성program을 이용하여야 하는데 그에 선행하여 table image를 되도록 축소하고 분포program수를 최소화하여 집계시간을 단축하여야 한다.

합계란의 삭제는 합계routine이 따로 없기 때문에 분포program수를 최소화하고 단순화하여 program을 간결하게 하므로서 program 작성효율향상에 기여할 수 있어야 한다.

그러나 통계표에 따라서는 결과표 난외항목의 종류가 다양하여 합계처리를 간소화하고 합계routine의 단순화를 위하여 합계란 설정이 요구되는 경우가 있는데 그림 3-12에서 표두부분에 합계란이 설정된 예를 통하여 살펴보면 합계란의 설정과 삭제에 대한 판단은 image table의 크기와 matrix code와의 관계라는 측면에서 고찰하지 않으면 아니된다.

「직업상의 지위」란의 code가 그림 3-12와 같이 1～9라 할때 합계란(사선부분)이 없으면 분포점 결정은 code 그 자체를 matrix

code 화하여 사용할 수 있으나 합계란을 설정하면 cell 번호와 code 가 일치하지 않기 때문에 code 를 다시 부여하여야 하고 그에 따른 재계산이 불가피해진다.

이와같이 합계란이 변측적일때는 합계 routine 도 복잡해지기 때문에 합계란의 삭제문제는 분포 routine 과 합계 routine 의 균형점에서 결정하지 않으면 아니된다.

합계	계	자영업주 1 유고업주	자영업주 2 부고업주	자영업주 3 내직	4 가족종사자	계	고용자 5 용역	고용자 6 일반상용	고용자 7 임시직	고용자 8 일용	9 종사상지위에상

그림 3-12 통계표의 합계예

RUN		Memory Size	규모계	
NA	10	column 인천 성 10 × 2 × 2 - 40	40 ×2 - 80	40,160
	11	column 인천 직종 성 10 × 8 × 5 × 2 - 800	800 × 2 - 1,600	
	12	column 학력 직종 성 10 × 4 × 4 × 2 - 320	320 × 2 - 640	
	13	column 산업 제 성 10 ×31 × 3 × 2 - 1,860	1,860 × 2 - 3,720	
	14	column 인천 산업 종 성 10 × 3 × 4 × 5 × 2 - 1,200	1,200 × 2 - 2,400	
	15	column 규 산업 성 10 × 6 × 15 × 2 - 1,680	1,680 × 2 - 3,360	
	17	column 직 산 직 년 성 10 × 4 × 14 × 4 × 4 × 2 - 17,920	17,920	
	18	column 지 성 10 × 20 × 2 - 400	400 × 2 - 800	
	28	연구 년 제 성 10 × 6 × 4 × 3 × 2 - 1,440	1,440 × 2 - 2,880	
	30	연구 규 제 성 10 × 6 × 6 × 3 × 2 - 2,160	2,160 × 2 -4,320	
	33	성 적 위 10 × 2 × 5 × 2 -200	200 ×2 - 400	
	35	인직종 학력 성 10 × 5 × 3 × 2 - 300	300 × 2 - 600	
	37	10 × 2 × 7 × 2 - 280	280 × 2 - 560	
	59	10 × 46 - 460	460 × 2 - 920	
NB	24	학력 직종 성 10 × 3 × 5 × 2 - 300	300 × 2 - 600	48,440
	25	년령 직종 성 10 × 8 × 5 × 2 - 800	800 × 2 - 1,600	
	26	직 년 성 10 × 32 × 4 × 2 - 2,560	2,560 × 2 - 5,120	
	27	규 이 제 성 10 × 6 × 2 × 3 × 2 - 720	720 × 2 - 1,440	

표 3-1 image table 계산표

합계란설정은 통계표가 완전한 형태로 정의되기 때문에 이해가 용이하고 통계표간 check(숫자마춤)가 용이하며 사후처리과정을 단순화할 수 있는 장점이 있고, program수가 많아지고 합계routine에서 error를 범할 우려가 크며 matrix code의 재계산을 필요로 하는 단점이 있다.

이와같이 table image 할당, 합계란 설정·삭제작업을 통하여 table image의 크기가 확정되면 표3-1과 같은 image table 계산표를 작성비치하여야 한다.

5. matrix code 와 displacement

가. matrix code

분포점P(X, Y)의 결정방법은 앞에서 설명한바 있기 때문에 여기에서는 표측항목과 표두항목이 계층구조일 때의 matrix code에 대하여 살펴보기로 한다.

그림 3-13은 표측, 표두 공히 2개 계층구조의 통계표 예로서 표측, 표두계층 길이의 합계를 차원수라 한다면 4차원의 통계표로 정의할 수 있는바 4차원통계표의 분포점P는 4개요소 X_1, X_2, X_3, X_4 에서 구할 수 있기 때문에 이를 P(X_1, X_2, X_3, X_4)라 하고 X_1~X_4를 4차원에서의 matrix code라 한다.

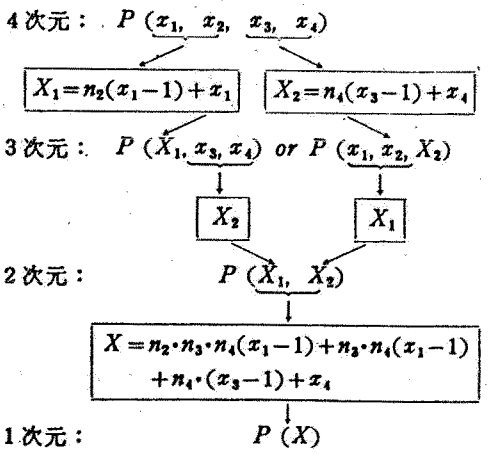
일반적으로 n차의 통계표 분포점은 P(X_1, X_2, … X_{n-1}, X_n) 로 표현되므로 분포program에서 table image를 n차원 배열로 정의할 수 있다면 P에의 분포는 용이하고 n-1 차원까지의 정의는 가능하나 n차원의 정의가 불가능하거나, assembly 언어와 같이 배열개념의 결여로 n차원 배열의 정의가 불가능한 경우에는 정의가 가능한 차원까지 차원수를 줄이지 않으면 아니된다.

그림 3-13의 4차원통계표를 1차원을 축소하여 3차원화하면 각차원에 따라 다음과 같이 전개된다.

지역	성별 / 배우관계 / 연령		남 (1) 미혼 (1)	남 (1) 유배우 (2)	남 (1) 사별 (3)	남 (1) 이혼 (4)	여 (2) 미혼 (1)	여 (2) 유배우 (2)	여 (2) 사별 (3)	여 (2) 이혼 (4)
시부 (1)	15～19세	(1)								
	20～29	(2)								
	30～39	(3)								
	40～49	(4)								
	50～59	(5)								
	60～	(6)								
군부 (2)	15～19	(1)								
	20～29	(2)								
	30～39	(3)								
	40～49	(4)								
	50～59	(5)								
	60～	(6)								

※ 괄호내 숫자는 각항목의 code 임.

그림 3-13 4 차원 통계표의 예



다만, X_1 = 항목 1의 code

X_2 = 항목 2의 code n_2 = 항목 2의 code수

X_3 = 항목 3의 code n_3 = 항목 3의 code수

X_4 = 항목 4의 code n_4 = 항목 4의 code수

다시 그림 3-13의 P점을 2차원으로 축소하여 표현하고저 할때에는

X_1 =지역 code = 1

X_2 =연령 code = 5, $n_2 = X_2$의 종류 = 6

X_3 =성별 code = 2, $n_3 = X_3$의 종류 = 2

X_4 =배우관계 code = 3, $n_4 = X_4$의 종류 = 4이므로

$X_1 = n_2 (x_1 - 1) + x_2 = 6 \times (1 - 1) + 5 = 5$

$X_2 = x_4 (x_3 - 1) + x_4 = 4 \times (2 - 1) + 3 = 7$

따라서 $P(x_1, x_2) = P(5, 7)$이 된다.

그러므로 4차원배열은 불가능하나 3차원배열이 가능한 경우에는 x_1, x_2로부터 x_1을 계산하여 P점을 찾거나 x_2를 계산하여 P점의 위치를 찾아내야 된다.

배열개념이 결여된 assembly 언어의 경우에는 1차원으로 떨어뜨리고 다시 한cell의 column수를 곱해서 P점의 위치를 계산해야 하는데 단시간에 위치계산을 하려면 $x_1 \sim x_4$를 「0」으로 시작하는 code로 하여 $x_1 - 1$, $x_2 - 1$, $x_3 - 1$, $x_4 - 1$ 등의 계산을 생략하고, $n_2 \cdot n_3 \cdot n_4$, $n_3 \cdot n_4$ 등도 image table이 결정과 더불어 확정되므로 계산해 놓도록 하되 이때에 「0」으로 시작하는 code라 할지라도 p점계산에 사용하면 matrix code로 보아야 마땅하다.

이와같이 matrix code 부여방법은 check program을 통한 부여방법외에 분포program에서 수시 작성하는 방법이 있다.

check program을 통하여 작성하는 경우는 그림 3-14와 같이 check 완료 data상에 input정보란과는 별개로 matrix정보란을 설정하되 matrix code는 산업분류, 직업분류와 같이 변환용 table이 크고, code변환이 특수하여 어려우며 여러개의 분포program의 공통사용이 가능할때 분포program의 memory를 절약하기 위하여

check program을 활용하는데 check program에서 matrix code를 부여하면 분포program에서는 그만큼 여유가 생기기 때문에 error를 범할 기회가 적어지는 반면 matrix code를 부여하면 check program을 수정하는 부담이 따르기 때문에 변환하고저할 때에는 주의가 필요하며 이에 대한 대비책으로 재집계 범위와 시간의 절약에 필요한 input정보를 data에 기록해 두어야 한다.

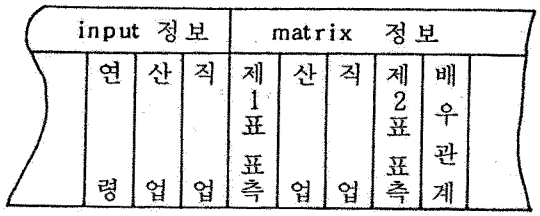


그림 3-14 check된 data상의 matrix code

분포program을 통하여 matrix code를 작성할 때에는 input 정보를 이용할 수 있거나 작성이 간단할때, table image에 합계란이 설정되고 표단위로 matrix code가 독자적이고 통일성이 없을때, 분포program이 하나인 간단한 집계일 때, 특히 소규모집계로 matrix code의 중복부여가 불필요한 때에는 분포program을 통한 matrix code 부여가 유리하다.

나. displaycement

그림 3-15에서 보는바와 같이 분포식에 ΣH, ΣS를 포함하는 분포에서는 분포점P의 계산에 H_1 또는 H_2의 길이(cell수)가 관계할 때가 있는데 table image를 표측 1차, 표두 1차로 하는 2차원으로 정의하면 $S \times H_1$에서의 분포점 $P_1(S, h_1)$은 S=지역 code,

$h_1 = x_1$ = 연령 code 이나 $S \times H_2$의 분포점 $P_2(S, h_2)$는 S = 지역 code, $h9 = n_1 + n_2 \cdot (x_1 - 1) + x_3$ (다만 x_2 = 성별 code, x_3 = 배우자 code, n = H의 길이 =연령 code 의 길이, $n_3 = x_3$의 종류) 가 되어 P_2의 위치는 n_1에 영향을 미치게 되는바 이는 $S \times H_1$의 제 1 cell(A점)과 $S \times H_2$의 제 1 cell(B점)과의 편차로 나타나므로 displacement 라 한다.

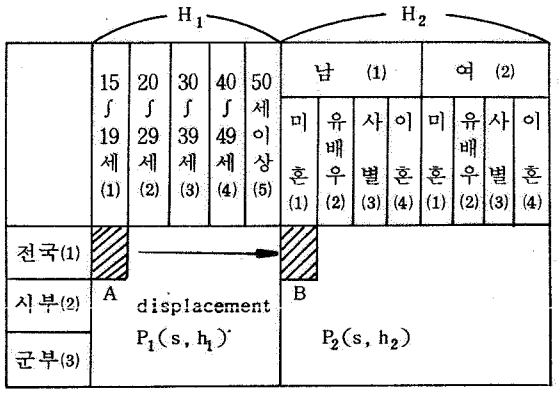


그림 3-15 displacement

6. inclusion code와 분포항목

가. inclusion code

결과표의 요구에 따라 input data의 일부분이 분포대상이 되는 경우에 대상data 선택 code를 inclusion code라 하고「1, 0」의 code로 구성하여 결과표 분포직전에 1 or 0의 영역내에 있는가를 check하여 1이면 분포, 0이면 분포하지 않는 것으로 약속한다.(그림 3-17)

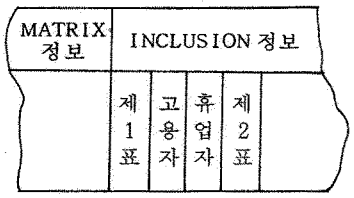


그림 3-16 INCLUSION 정보

따라서 inclusion code는 표별로 설정하는 것이 통일성 유지에 좋으나 중복설정으로 인한 번거로움을 피하기 위하여 사용빈도가 크고 대상 data의 선택이 복잡하며, program miss의 원인이 된다고 판단되는 것에 한하여 code화하는 것이 바람직하다.

그림 3-16에서 제 1, 2 표용의 inclusion code외에 제 1 표 대상 data를 다시 선택하여 고용자 및 휴업자용 code를 부여한 것은

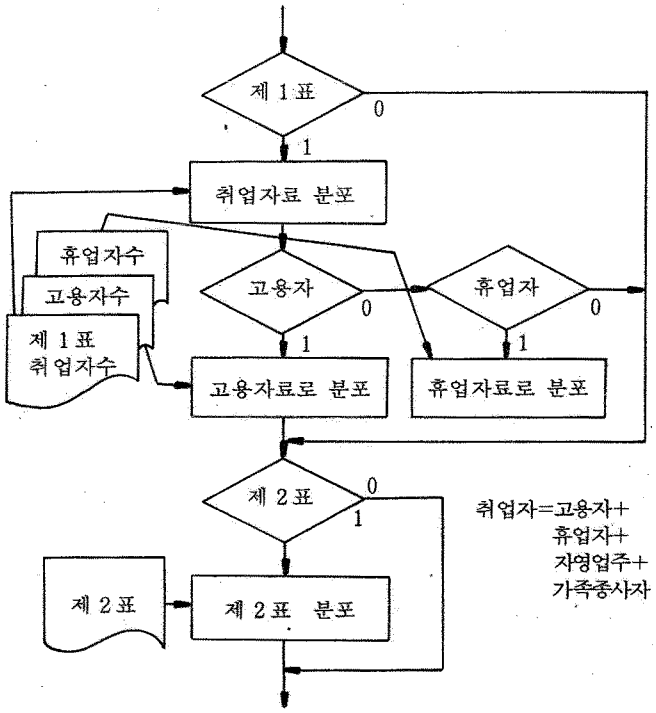


그림 3-17 INCLUSION TEST의 FLOW

1, 2표 이외의 결과표이용에 대처하고 각표의 분포routine 선택에서 miss를 방지하기 위한 조치이며 inclusion code와 matrix code에 대한 특별한 부여방법과 대안이 없기 때문에 상황에 따라 적절히 대응하지 않으면 아니된다.

나. 분포항목

분포는 도수분포와 계량분포로 구분하여 결과표의 요구에 따라서 선택하는데 일반적으로 도수분포는 대상data를 처리할 때마다 1을 분포하고 계량분포는 수량과 금액 등 data의 일부항목을 분포한다.

table image에서의 단위cell의 크기(column수)를 합계란의 column수에 해당하는 분포항목의 최대계산 column수에 맞게 설정하면 계산도중에 column수 부족으로 인한 확장이 불필요하고 overflow로 인한 error를 방지할 수 있다.

7. 상위계급에의 합계

분포program에서는 합계등 추후계산이 가능한 항목을 생략하여 program을 단순화하고 program수의 축소를 목표로 하나 상위계급에의 합계는 table image 할당과 깊이 관련되기 때문에 분포program에서 중요한 개념으로 대두된다.

상위계급에의 합계를 그림 3-18을 통하여 설명하면 표측의 산업분류는 소분류에서 중분류, 중분류에서 대분류, 대분류에서 총계의 순으로 적산하는 계층구조로 되어있고 합계방향은 화살선방향으로 이루어지는바 통계표전체를 table image화할 수 없는 경우에는 S를 control 항목화하여 표측 1행분을 취하고 상위에의 합계를 후속 program에 의존하거나 그림 3-19와 같이 S에 대하여 4행분의 table image를 취하여 분포program에서 합계를 구하는 방법을 채택하고 분포작업순서에서 소분류용 table image에 data를 분포하고 co-

ntrol이 끊기면 소분류 분포결과를 중분류용 table image에 합계하고, 중분류용 control이 끊기면 중분류의 합계결과를 대분류용 table image에 합계하고, 다시 대분류control이 끊기면 대분류의 합계결과를 총계용 table image에 합계하여 상위에의 합계와 동시에 print out하고 zero로 clear하는 순서를 flow로 표현하면 그림 3-20과 같이 되는바 분포program의 flowchart에 대해서는 다음에 구체적으로 설명하기로 한다.

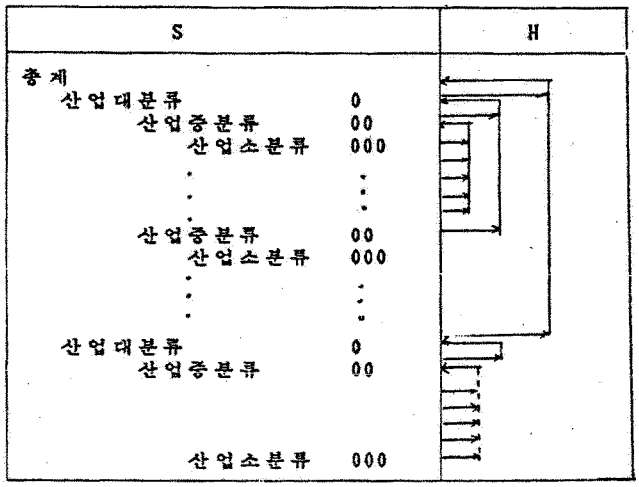


그림 3-18 상위계급으로의 합계

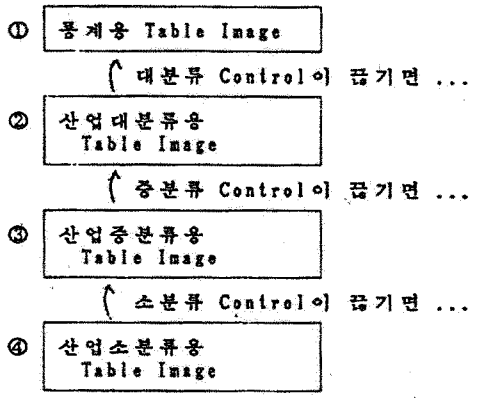


그림 3-19 Table Image의 할당

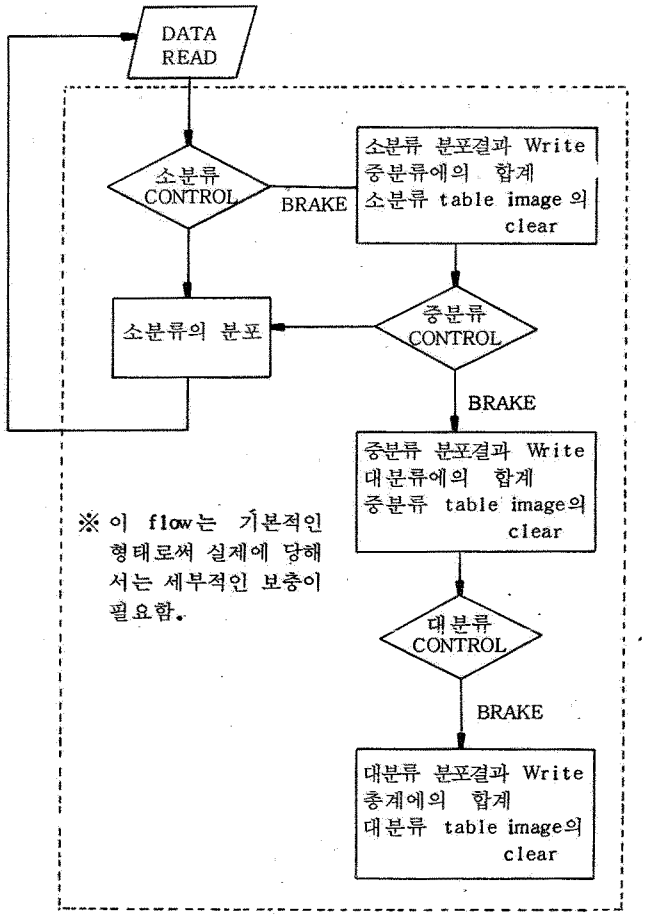


그림 3-20 상위계급에의 합계에 대한 기본적인 flow

8. summary data의 print out

가. RID

summary record는 그림 3-21과 같이 표로 된 항목과 계산수치 항목부분으로 구성되는데 표로 나타낸 항목은 표번호·control항목·record 일련번호등 개개의 record식별을 위하여 record identifier(RID)를 부여하는데 RID의 주기능이 record를 식별하는 것이기 때문에 부여형식은 이용자의 형편에 따라 다르나 그림

3-21 하단을 예로 구성요소에 대한 내용설명을 하면 다음과 같다.

조 사 명 : 조사종류의 약호

년 월 : 통계조사 실시 년·월

표 번 호 : 결과표 번호

control 항목 : 난외항목 R, S 등 분포program의 control용 항목

표측항목 : 표측 행단위 print out 때 control 항목만으로 record 식별이 어려울때 설정

blank : 다른 표의 RID 길이와 통일하기 위한 공란

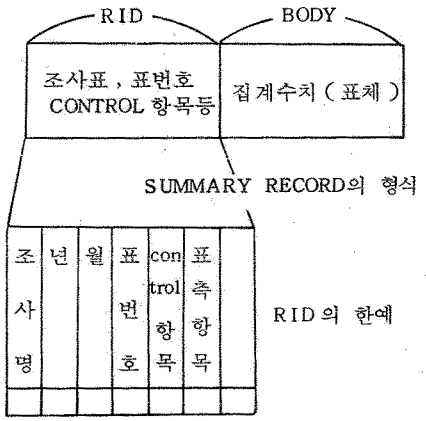


그림 3-21 SUMMARY RECORD의 형식과 RID

나. print out 형식

summary data의 print out 형식은 table image를 그대로 print하는 표단위 print out과 table image를 분할하여 print하는 표측단위 print out의 2개형식이 있는바 그림 3-22를 통하여 설명하면 다음과 같다.

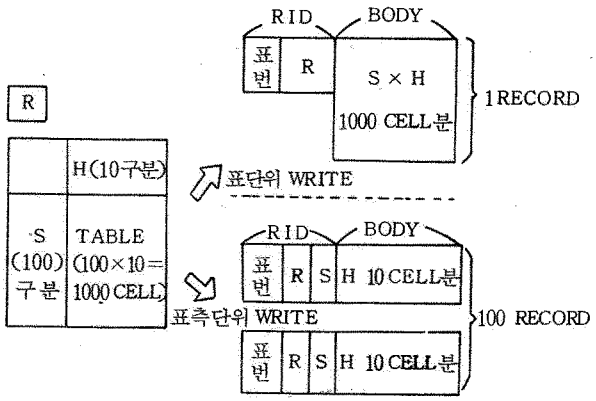


그림 3-22 WRITE의 TYPE

표단위 print out 방식은 그림 3-22 (좌측)과 같이 정의된 table image를 우측상단의 record 형식으로 print out 하는 것으로서 단위 record의 길이(body 부분)가 1,000 cell의 장대 record가 된다.

표단위 print out은 사후처리를 위한 sort와 합산이 간단하고, record 수가 적기때문에 집계시간이 절약되며, print out이 간단하여 program error가 적고 가공편성program의 처리가 용이하며 후속처리의 설계가 용이한 장점이 있는 반면에 장대한 record를 처리해야 하기 때문에 물리적인 i/o error 발생가능성이 크고, 후속 program에서 memory의 제한을 초과하기 쉬우며, program 언어에 따라서는 record length의 제한으로 별도의 처리방법을 강구해야 하는 단점이 있다.

표측단위 print out은 그림 3-22의 우측하단과 같이 표측 1행을 1 record로 하여 100행분을 print out 하는 것으로 RID에 표측항목 S를 추가하여 print out한 record를 식별할 수 있어야 하며 이때에 10 cell 정도의 짧은 record lenth가 되는 것이 특징적이다.

표측단위 print out은 RID가 표측단위이기 때문에 장대record의 처리를 위한 배려를 하지 않아도 좋은 장점이 있고, summary record check와 print out routine이 번거롭고 후속 program처리에서 record수가 많아지는 단점이 있다.

표단위 print out은 한번의 print out명령으로 충분하지만 표측단위 print out은 RID 부분을 각행에 중복되지 않게 이송하면서 print하는 방법과 각행을 제1행에서 중복되지 않게 이송하면서 print하는 방법, 다른 print out영역에 RID 부분과 각행을 이송하면서 print out하는 방법 등 3개방법이 있는 바, 제3방법이 가장 어렵고 제1방법이 가장 쉽게 처리할 수 있는 특성을 가지고 있다.

이상의 표측단위 print out방법중 어느 것이나 표단위 print out에 비하여 번거롭기 때문에 표단위 print out의 제약사항만 해결할 수 있다면 표단위 print out이 가장 유리함은 재론의 여지가 없으나 다만 표측단위 print out의 행전체가 zero인 data에 대한 print out을 생략하는 「zero 삭제」기능이 없는 불리한 면이 있다.

zero 삭제기능에 대해서는 가공편성 program에서 설명키로 한다.

9. 표준 flowchart

정확한 결과표만 얻을 수 있다면 program 작성과정은 아무래도 좋다는 생각은 program이 작성되기까지는 test와 maintenance는 물론 전체 program에 대한 review를 수없이 반복해야 하므로 성공적인 program을 완성하기 위해서는 위험한 발상이 아닐 수 없으며 flowchart도 data의 흐름이 smooth하여야 하므로 임의로 switch를 활용해서는 아니된다.

분포program은 다양한 flowchart의 형성이 가능하나 그림 3-23과 같은 control brake에 중점을 둔 표준 flowchart가 요

구되는바 이에대하여 구체적인 설명을 하면 다음과 같다.

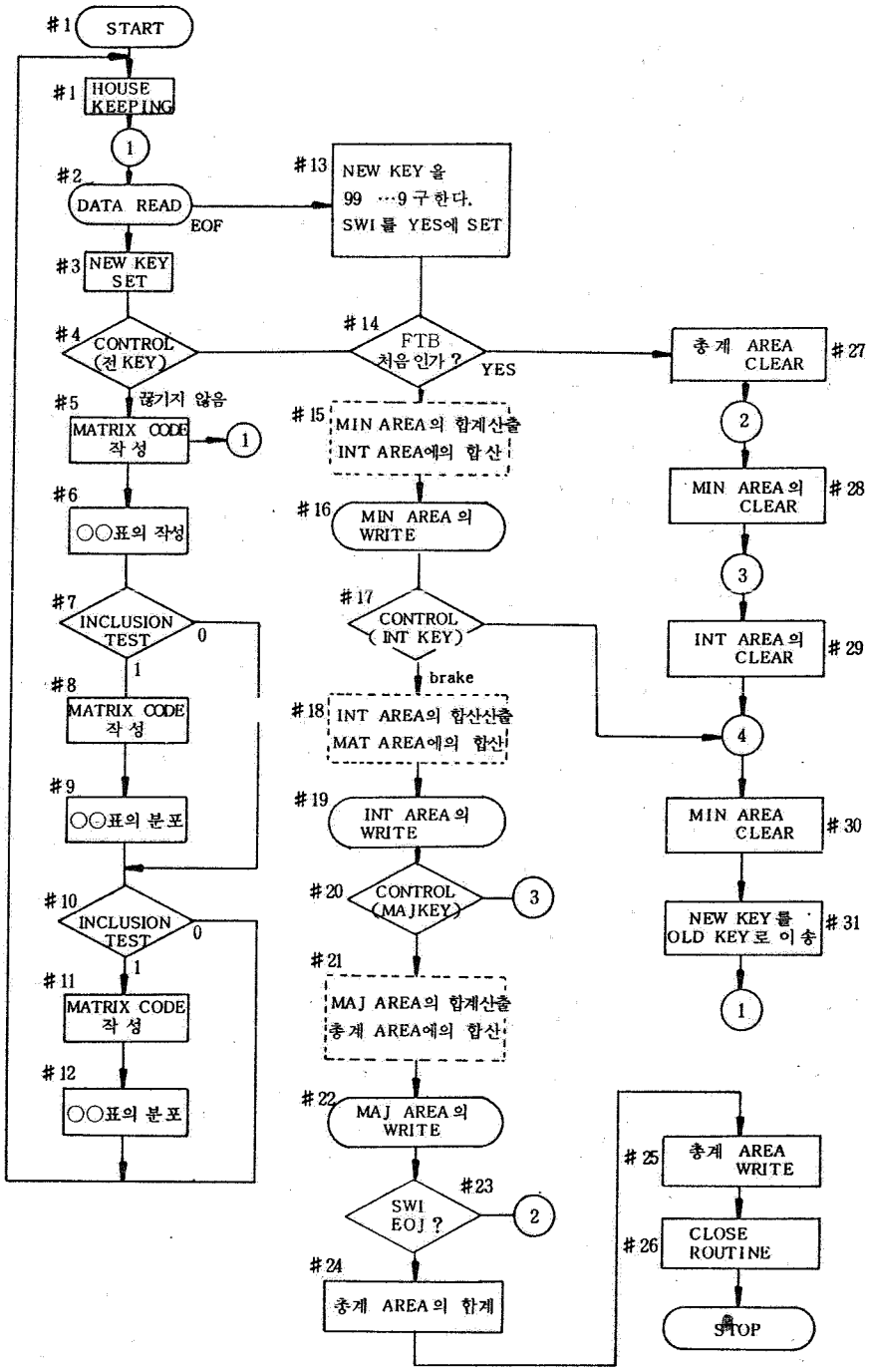


그림 3-23 표준 FLOW CHART

가. flowchart의 개요

분포program의 routine은 대별하여 사후처리 routine · reading routine · control routine · 분포 routine · 합계산출 routine · print out routine · clear routine · control key의 변경 및 EOJ routine 등 9개 routine으로 구성되는데 각각에 대하여 설명하면 다음과 같다.

(1) 사후처리 routine(#1)

일명 house-keeping routine이라 하며 분포 program에서 필요로 하는 switch, constant류의 initialize input file/output file의 open

(2) reading routine(#2)

check 완료 data의 read.

(3) control routine(#3, #4)

분포program에서는 table image가 크기 때문에 control 항목을 설정하고 read된 data의 control 항목과 직전에 처리된 data의 control 항목의 비교로 내용일치여부를 check 조건에 따라 처리 routine을 control하는바, 이와같이 data처리의 흐름을 control하는 routine을 control routine, 신규data의 control 항목을 「new key」, 직전에 처리된 data의 control 항목을 「old key」라 하는바, 표준 flowchart의 장점인 control routine을 십분 활용하기 위하여 「new key」는 input data 항목을 control level의 크기에 따라 좌로부터 순차로 배열할 필요가 있다.

예컨대 그림 3-24와 같이 지역·산업·성별등 input 항목을 #3과 같이 재배열한 다음 전체 key(MIN)와 비교(#4)하여 차이가 발생하면 control brake는 MIN level에서 생겼음을 의미하며 이어서 산업을 제외한 잔여 key(INT)에 대하여 비교하여(#17) con-

trol이 끊기면 지역과 성별중 어디엔가에 error가 있음을 의미하고 다시 MAJ level의 key에서 control brake의 유무를 확인한다(# 20)

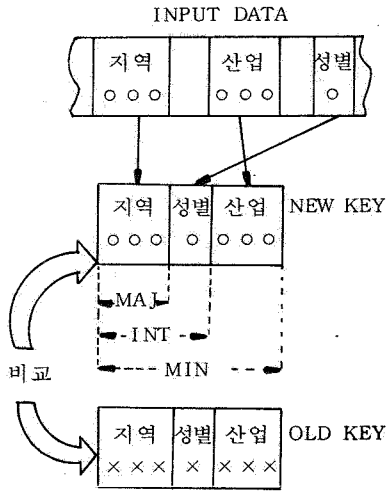


그림 3-24 CONTROL 항목

이와같이 전체적인 key로부터 점차로 key의 범위를 좁혀가면서 차례로 control brake의 유무를 확인하면서 data flow를 smooth하게 하여야 한다.

(4) 분포 routine(# 5 - # 12)

전술한바 matrix code 부여(# 5 · # 8 · # 11), inclusion test(# 7 · # 10), 각표의 분포(# 6 · # 9)등을 망라한 routine이다.

(5) 합계산출 routine(# 15 · # 18 · # 21 · # 24)

분포program에서 합계란 설정을 생략하기 때문에 합계산출 routine이 원칙적으로 불필요하나 통계표에 따라서는 합계란설정이 필요한 때가 있는 바, 이러한 때에는 그림 3-25와 같이 사선부분의 cell을 분포 routine으로 구하고 control이 끊기면 결과표상의 표두·표측합계를 산출하는 방법과 그림 3-26과 같이 각 control

level에서 정의된 table image를 control이 끊길때마다 상위 level image에 합계하는 table image단위로 상위계에 합산하는 방법이 있다.

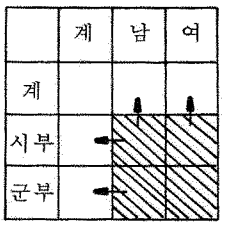


그림 3-25 표내계

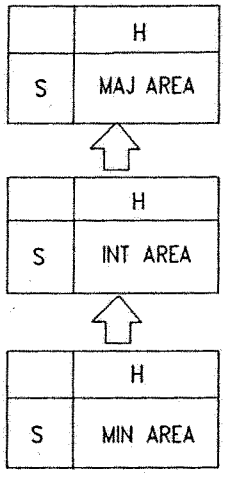


그림 3-26 합산

이때에 table routine이 table image내의 합계를 산출하는데 대하여 table image단위로 level에 합산하는 방법은 table단위로 상위에 적산하는 것이기 때문에 전자와 구별하는 의미에서 합산이라 하기도 한다.

(6) print out routine(#16 · #19 · #22 · #25)

table image의 print out routine으로 표단위 print out과 표측단위 print out이 있다.

(7) clear routine(#27 - #30)

분포routine의 대상인 table image로서의 분포 area와 합산대상인 table image로서의 합산 area는 control이 끊길때마다 total · 합산 · print out이 이루어지는데 print out이 끝난 area를 다음 차례의 분포와 합산을 위하여 clear하는 기능을 수행한다.

(8) control key의 갱신 (# 31)

new key를 old key로 이송하고 다음차례의 control group 처리를 위하여 대비한다.

(9) EOJ routine(# 23 - # 26)

마지막 data의 분포가 끝나면 EOJ(End of Job) routine에 control을 넘겨서 최종적인 total과 합산 및 print out작업을 수행하여 사후처리에 임하게 되는데 여기에서는 사용한 file의 closs와 처리된 data의 count등을 print한다.

나. 주의사항

이상의 표준 flowchart에서 old key의 정의·by-path switch·최종분포결과의 소거·분포program의 type등의 문제에 대하여 유의하여야 하는바 이에 대하여 구체적으로 설명하고저 한다.

(1) old key의 정의와 by-path switch

최초의 data분포에 앞서 table image를 clear하되 사전처리 routine에서 zero로 하는 방법과 table image를 정의할때 zero로하는 방법이 있는바 여기에서는 분포program이 clear routine을 가지고 있으므로 사전처리routine에서 zero로하는 방법을 살펴보기로 한다.

구체적으로 old key를 정의할때 1번 data에서 control이 끊기도록 하되 1번 data가 분포되지 않은 상태이므로 total·합산·print out을 bypath(우회)시키기 위한 switch (# 14)를 설정한다.

key의 정의는 간단하게는 input data에 없는 임의의 문자를 old key의 일부에 설정하는 것으로 충분하고 bypath switch는 통과하는 최초의 1회를 branch로 하고 2회이후는 NOP(No Branch)의 동작을 하게되는데 이러한 기능의 switch를 FTN(First Time Nop)이라 하며 FTB, FTN는 program 작성과정에서 중요

한 기능을 수행하는 것이므로 기억해둘 필요가 있다.

(2) 최종 분포 결과의 소거

최종 input data를 처리하면 EOF(End of file)상태가 되어 이제까지의 분포결과 최종total의 합산과 print out을 하기 위하여 new key의 전부 또는 최좌단의 1 column에 9를 set하고 FTB에 branch하므로서 MIN·INT·MAJ의 각control이 끊기도록 하되 수치 9가 input data에 존재하면 9대신에 input data내에 없는 문자(영문자·숫자·특수문자 등) x를 new key에 set한다.

EOJ의 판단은 flowchart와 같이 switch의 set에 의하거나 new key 좌단에 x의 유무를 확인하는 test에 의해도 무방하다.

(3) 분포program의 type

분포program에는 control이 없기 때문에 input data의 sort가 필요없고 결과표의 전체cell을 table image화하여 모든 data의 분포가 완료된 다음에 print out하는 type(type 1)과 표준flowchart에서 #15-#20, #24-#25, #27, #29-#30을 생략하면 control level이 일계급만 존재하므로 여러개의 control 항목을 전체의 control key에서 비교하여 control이 끊기었을때 합계산출 print out과 memory의 clear가 이루어지는 type(type 2)과 표준 flowchart와 같이 여러개의 control level에서 control이 끊길때 해당 level을 합산·print out과 memory를 하는 type(type 3)이 있다.

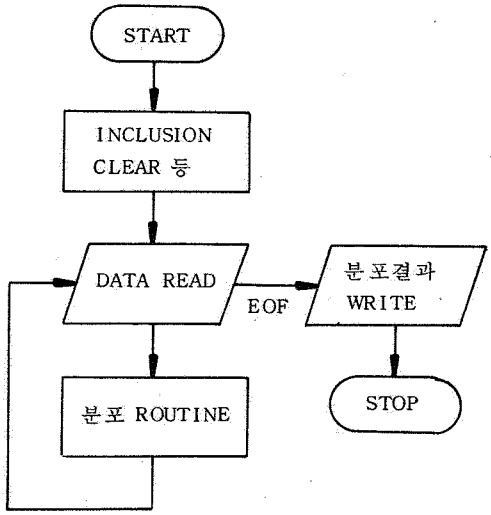


그림 3-27 TYPE 1 의 FLOW

Ⅳ. 합산 PROGRAM의 설계

1. 합산 program의 개요

가. 합산 program의 대상

통계표의 합계산출을 합산이라 하는바 분포program설계에서 언급한바와 같이 표두·표측의 합계(표내계)와 난외항목의 합계(표외계)로 구분되고, 합산program의 합계대상은 표외계에 한하고 표내계는 가공편성program에서 처리하는 것이 상례로서 환언하면 합산처리는 summary record 단위로 형성되기 때문에 분포program에서 표단위 print out일때는 표외계, 표측단위 print out일때는 표내계가 된다.

합산program은 구조상 분포program과 유사하나 분포 program에서는 memory의 일정area에 입력한 data를 적산하여 표번호가 바뀔때나 일정하게 정해진 한도량 내에서 합계를 구하고저 할때 합계를 구하고 합산program에서는 summary data를 대상으로 control key 항목에서 지정한 내용과 동일한 summary data만 선택하여 합산하는데 이때의 summary data는 지정된 내용과 key항목에 대하여 sort되어 있어야 한다.

나. 합산program의 기능

합산program을 그림 4-1의 기본형 분포형식(R×S×H) 통계표를 표단위로 print할때의 summary record형식을 들어 설명하면 난외항목 R(남·여별)을 공통항목으로 하여 합산program에서 남여 합계를 구하려면 summary record는 1표와 2표가 S×H의 구분이 다르기 때문에 가변장(可變長)형식의 record가 된다.

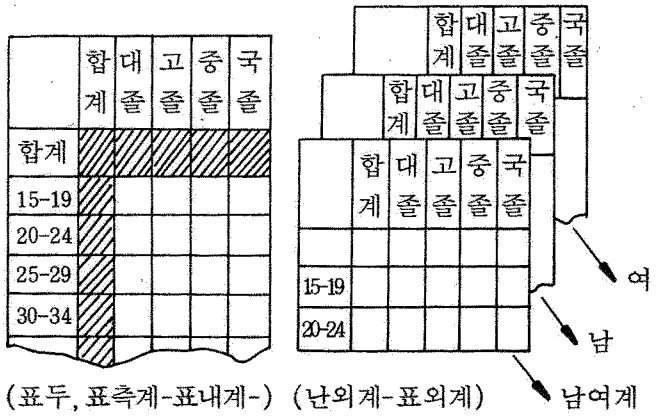


그림 4 - 1 합산의 종류

일반적으로 분포program의 output은 그림4-3과 같이 control이 끊길 때 마다 그 control에 속하는 통계표가 print out되기 때문에 표번호를 기준으로 순부동(順不同)이 되므로 표번호> control 항목순으로 sort하여야 한다.(그림 4 - 2우측 참조)

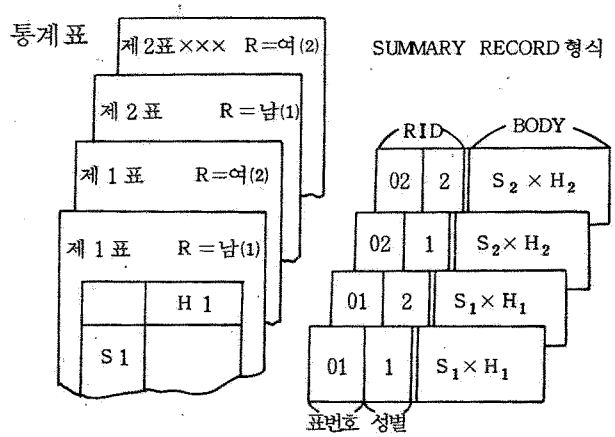


그림 4 - 2 통계표와 SUMMARY RECORD

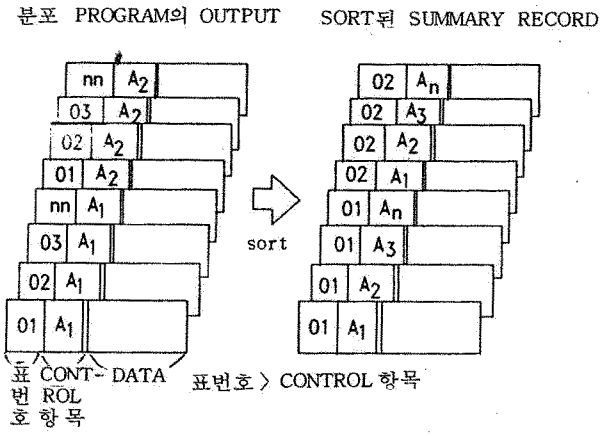


그림 4 - 3 Summary record의 배열

남녀의 summary data를 합계한 남녀계를 다른 시각으로 보면 표번호를 control항목화 하여 표번호가 바뀔 때까지 남여별code를 무시한 합산의 의미도 되기 때문에 control항목으로서의 표번호는 반드시 sort되어야 하고 남녀별code는 sort할 필요없이 control이 합계항목 보다 상위의 항목에 대하여 sort하고 그 control 범위내에서 합산하여야 한다.

이상의 내용을 근거로 a,b,c, …, m,n의 control항목 가운데 m에 대한 합계산출을 일반식으로 표현하면 m을 최하위로 하는 a>b>c …>n>m을 sort key로 하여 sort하고 (m은 최하위에서 제외) control항목을 a,b,c, …m,n으로 하여 합산하여야 하는데 n 항목의 합계는 그림 4 - 4와 같이 n회의 합산이 필요하지만 실제의 통계표 R은 1～3개 항목에 불과하므로 n회의 sort와 합산은 3회정도면 족하다.

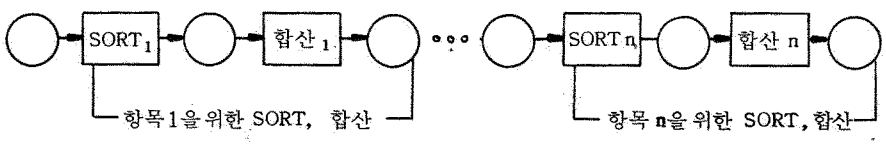


그림 4-4 sort. 합산의 반복처리

2. 합산을 위한 sort

합산의 선행조건인 sort의 순서를 그림 4－5를 예로 하여 통계표를 1개표, summary record를 표단위 print out으로 가정하면 남여·학력별 2개 항목의 합계표작성을 위하여 합계항목 code의 성별을 남＝1, 여＝2로 하고, 학력을 국졸(1)~대졸(4)로 정의하여 file하면 file A는 분포program의 output에서 남여별(2구분)×학력별(4구분)＝8 record로 하여 수록할때 a를 남여별, b를 학력별로 정의하고 먼저 b의 합계를 구하기 위하여 남여별>학력별로 sort하면 file B와 같이 남여별, 학력 code순으로 배열되므로 a(남여별)와 동일한 범위를 합산하면 남자의 학력 record가 합산되어 학력계를 나타내는 code를 0으로 하는 학력별 record, 곧 file C가 작성된다.

다음 a의 합계를 구하기 위하여 학력>남여별 순으로 sort 하여 학력계에 대하여 국졸에서 대졸에 이르기까지 각 항목마다 남여별로 배열하여 file D를 작성하고 b(학력)의 같은 범위를 합산하면 학력별 남여 계를 산출하여 file E를 구성하는데 이때의 최종 record수는 남여별 3구분×학력 5구분＝15 record가 된다.

따라서 sort program은 합산에는 필수적이지만 실제 이용측면에서는 record수의 증가와, sort key의 생략, 배분 program의 활용, merge program의 사용등에 대하여 각별한 주의가 필요하다.

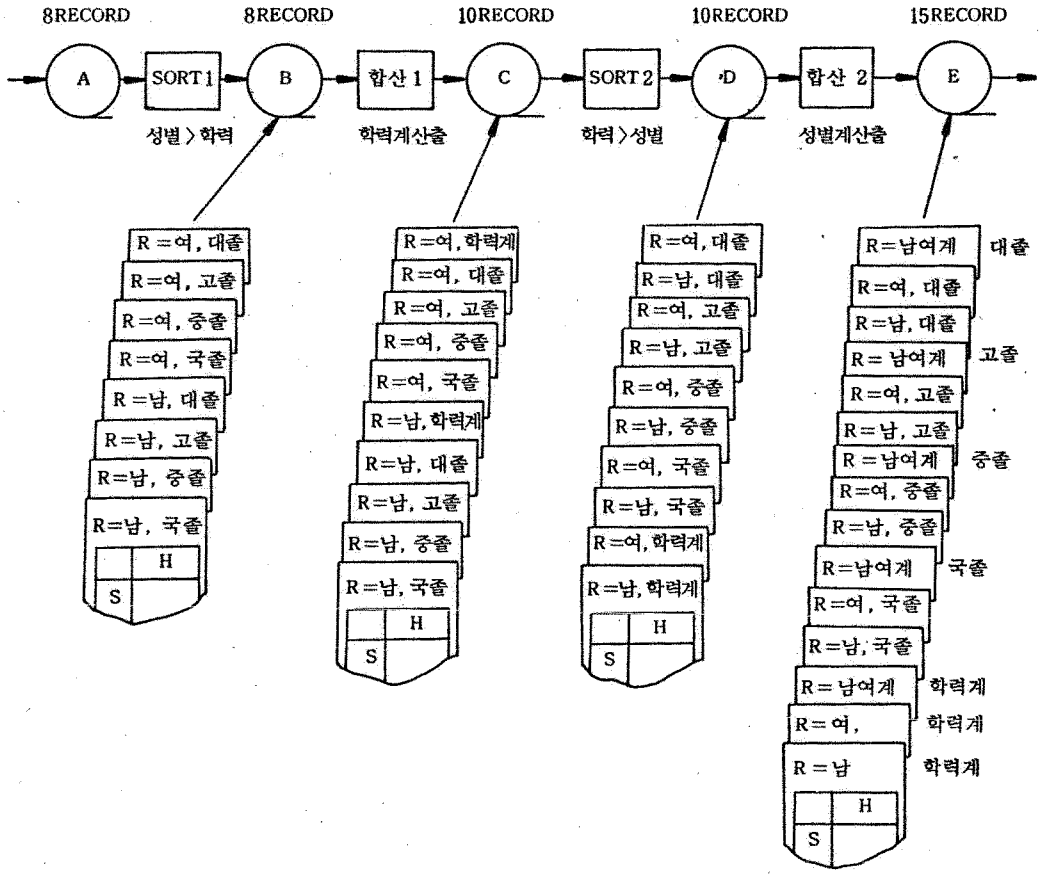


그림 4 - 5 합산을 위한 sort

가. record수의 증가

「남여계」「학력계」를 구하는 방법에는 그림 4-6와 같이 방법 1, 2가 있는바, 양쪽 모두 최종적으로 15 record의 합계를 구할 수 있으나 중간 step의 record수는 반드시 일치하지 않는다.

합산program에서 간혹 합계record 이외는 불필요한 경우가 있기 때문에 output이 적은것이 보통이나 input record수 만큼 많아지므로 방법 1과 같이 학력계를 먼저 구하면 남, 여의 학력계 2개 record의 합계가 output되고 내역record를 합하여 계 10 re-

cord가 되며, 방법 2와 같이 남여별 계를 먼저 구하면 학력별 남여별 합계를 구해야 하므로 4 record가 증가하고 내역 record를 합하면 계 12 record가 되어 방법 1보다 많아지므로 data량이 적은 방법 1이 유리하다는 결론을 얻을 수가 있다.

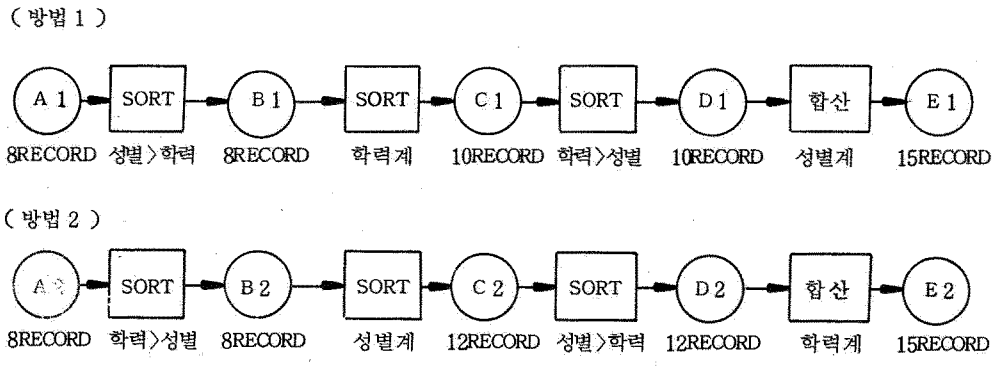


그림 4 - 6 sort의 두가지 방법

data량은 모든 program에서 매우 중요한 point임에는 틀림없으나 특히 sort program에서는 중요항목으로 작용하게 되므로 그림 4 - 7과 같이 분류, 중간조합, 최종조합의 3단계의 분류과정을 거쳐야 하는데 분류단계는 입력file에서 data를 필요한 만큼의 단계로 나누어 memory에 입력하고 지정된 key에 대한 string (일련의 순차data)을 만들어 n개의 중간file에 배분하고 중간조합단계는 분류단계의 string을 순서에 따라 조합하여 n개의 file내에서 최대string이 될때까지 반복처리하고 최종조합단계에서는 중간조합단계 최후의 중간file을 하나의 장대(長大) string으로 완성하여 sort의 output으로 정의한다.

이 과정에서 알 수 있는 바와 같이 sort에서는 순서가 갖추어진 상황에서 중간조합단계가 필요없는 경우에 input data를 최저 2회는 read하여야 하고 통상은 중간조합단계에서 여러차례의 read와 write가 형성되기 때문에 평균 7 - 8회를 read하여야 하고,

sort에서는 data량에 따라서 처리시간에 미치는 영향이 크므로 data량을 최소화 하는데 주력하여야 한다.

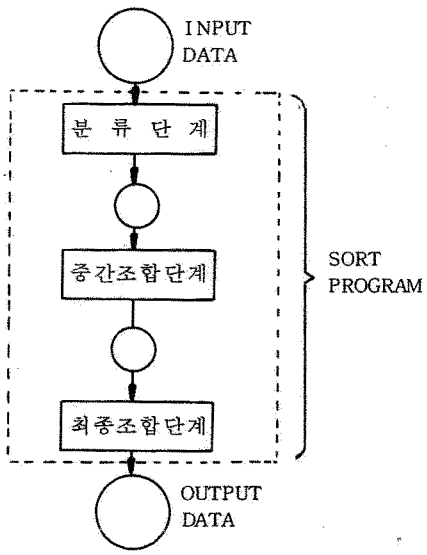


그림 4 - 7 SORT PROGRAM의 구성

또 sort program에서는 data량 이외에 record길이에 의하여 1회 sort량이 정해져 있는바 주로 n개의 중간 file data 수용량에 의하여 제한되기 때문에 이 수용량을 초과하는 경우에는 input data를 몇개 group으로 나누고 group별로 sort하여 그 결과를 merge program을 통하여 하나의 group으로 통합하여야 하므로 이때에는 당연히 처리시간이 증가하는데 이러한 처리시간의 증가를 방지하기 위해서는 합산program에서의 합계산출은 data의 증가가 적은 순으로 처리하여야 한다.

나. sort key의 생략

이미 설명한 바와 같이 합산할 때의 control key지정은 합계를 구하려는 항목을 제외해야 하므로 sort도 합계항목을 최하위 key로하여 지정할 필요는 없다.

일반적으로 합계를 구하고저 하는 항목의 key 지정을 생략하면 그

항목의 code종류가 많을수록 생략효과가 크게 나타나서 sort시간이 단축되기 때문에 sort key를 생략하는 편이 유리하나 sort되지 않은 항목이 포함되기 때문에 error table check에 어려운 면이 있다.

다. 배분 program의 활용

분류항목의 code수에 따라서 sort program의 사용 여부를 검토 결정할 필요가 있는데 sort program에서는 record수 증가에서 설명한 바와 같이 최소한 2회, 평균7～8회에 달하는 data의 read와 write가 이루어져야 하므로 분류대상항목의 code가 적을 때에는 최소한 각 code에 해당하는 data를 보조기억장치에 나누어 배분하는 편이 sort program을 활용하는 편 보다 read와 write회수 즉 data의 이동으로 형성되는 path가 적어지기 때문에 유리하다.

예를 들면 그림4－5의 sort 1은 학력항목을 생략하면 남여별항목만 sort하게 되므로 그림4－8의 배분 program에서 남, 여를 별개의 file로 작성 처리하는것이 sort보다 신속한 처리를 가능하게 한다. 분류대상항목의 code종류가 n인 경우 (n＋1)개의 보조기억장치가 있으면 그림4－9와 같이 sort보다 배분처리 쪽이 처리시간 단축에 기여할 수 있으나 분류항목이 2개 이상인 경우에는 배분처리 방식으로는 처리가 번거로우므로 sort program을 이용하는 편이 유리하다.

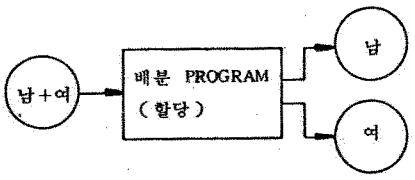


그림4－8 배분 PROGRAM

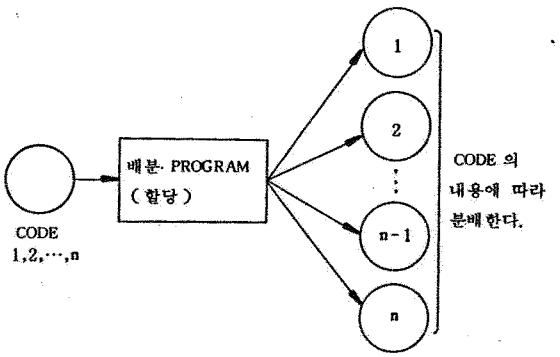


그림4－9 일반적인 분배 PROGRAM

라. merge program의 사용

앞에서 설명한 바와 같이 sort program에서는 한번에 sort 할수 있는 data량에 부과되는 제한을 초과할 때에는 merge program을 사용하여 sort하지 않으면 아니된다.

merge program은 이미 분류되어 있는 몇개의 file을 조합하여 하나의 분류된 file로 정리하는 기능을 가지고 있으나 한번에 조합 가능한 file수에 제한이 있기 때문에 이 제한을 초과하는 file을 조합하는 경우에는 몇번이고 merge를 반복해야 되며 merge의 반복을 위해서는 처리시간 단축에 적합한 merge방법 선정을 검토할 필요가 있다.

그림 4 - 10, 11 와 같이 data량이 일정한 6조의 file을 3조까지 merge하는 program으로 조합하는 예를 들면 그림 4 - 10 과 같이 2조 merge를 3회 실시하여 3조로 merge 하는 방법과 그림 4-11 과 같이 3조 merge와 2조 merge를 병행하여 3조로 merge 하는 방법이 있는바 그림 4 - 10 의 방법에 의한 data의 path는 2 + 2 + 2 (2조 merge 3회) + 6 (3조 merge) = 12 회로 merge program의 사용회수는 4회이고 그림 4 - 11 의 방법에 의한 data 의 path는 3 (3조 merge) + 2 (2조 merge) + 6 (3조 merge) = 11 회로 merge program사용 회수는 3회로서 첫째 방법에 비하여 path의 횟수와 program사용회수가 상대적으로 적기 때문에 처리시간이 단축된다. 따라서 merge대상 file수와 각 file의 data량이 많을 때는 merge방법에 따라서 처리시간에 크게 영향을 미치게 되므로 sort와 관련하여 검토하지 않으면 아니된다.

이상 sort와 연관된 문제를 네가지 핵심부분으로 압축하여 설명하므로서 이론적인 면에 치우친 감이 없지 않으나 system설계 시 집계의 기본방침의 결정방향에 따라서 trouble 대처를 우선 할 것인가 또는 처리시간의 단축을 우선할 것인가 하는 두가지 태도가 있

을 수 있는바 trouble 대처에 주안점을 두는 경우에는 data 량의 증가를 각오하더라도 이해가 용이한 flow를 채택하고 sort key는 생략하지 않는 편이 유리하기 때문에 1회 집계 통계에 활용되는데 반하여 처리시간의 단축에 우선을 두는 경우에는 연간조사, 월간조사등 주기적으로 반복집계하는 통계조사에 유리하다.

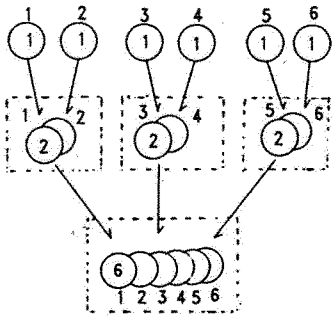


그림 4 - 10 제 1 방법의 merge

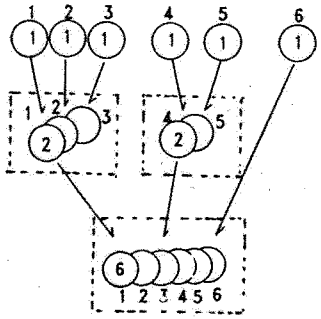


그림 4-11 제 2 방법의 merge

3. 합산 program의 형식

합산 program에는 직열형·병열형·가감산형의 3개 합산형식이 있는바 각 합산형식에 대하여 설명하면 다음과 같다.

가. 직열형 합산

그림 4 - 12에서 보는바와 같이 직열형합산은 input file의 summary record(내역 record)를 순차로 read하여 control 항목 R이 끊길 때까지 합산하는 형식으로 직접합산과 간접합산(group 합산)으로 구분된다.

직접합산은 그림 4 - 12와 같이 input data상의 control 항목 R이 끊길 때까지 합산하는 형식으로 그림 4 - 13 (summary data항목 C의 합계)을 들어 설명하면 control항목으로 표번호, 항목A, 항목B를 지정하고 항목C를 sort할 때에만 control항목으로 지

정하는 것으로 가정할 때 그림 4 - 13 좌하단 flowchart의 처리절차가 형성되는바 이때에 control이 끊길 때까지 input data를 가산하면 항목 C의 합계를 구할수 있다.

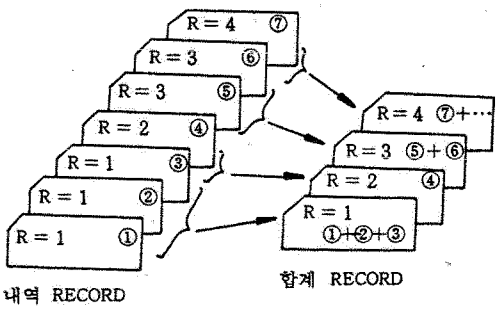


그림 4 - 12 직열형 합산

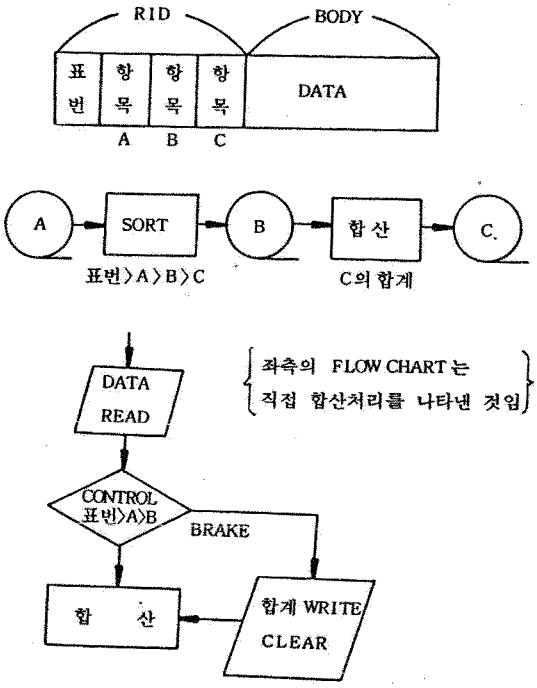


그림 4-13 직열형 (직접) 합산의 예

간접합산 (group 합산) 은 input data상의 control항목 R을 R′로 code변환하고 R′를 새로운 control항목으로 하여 합산하는 형식으로 분류code가 많은 산업분류·상품분류 등에서는 input code 자체를 control 항목으로 설정하면 상위의 합계를 구할수 없기 때문에 새로운 가 (假) code 체계로 변환해야 한다.

더욱이 check program에서 group code를 부여하고 산업 대·중분류를 code화하여 분포program에서 summary record에 입력하여 output하면 group합산은 하지 않아도 되는바 그림 4 - 14의 예제와 같은 summary data의 산업중분류를 구하고 data는

산업소분류 level에서의 output을 가정할 때 표4-1의 산업분류표에서 중분류는 소분류의 100단위, 10단위의 2 column부분이 같은 분류만을 합계하는 것으로 되어 있으나 다만 소분류 410이 예외로 되어 있기 때문에 소분류의 좌측 2 column부분을 control하여 직접 합산 하여도 중분류를 구할수 없기 때문에 이런 경우에는 산업분류를 그림4-14와 같이 생략하지 말고 최하위 sort key로 지정한 다음 합산program에서 input data를 read하여 중분류 code로 변환한 후에야 control이 끊기게 된다. 표4-1의 중분류 code 40~41를 control로 처리하려면 40, 41중 어느 한쪽으로 통일해야 하기 때문에 40으로 통일하여 중분류를 구하면 flowchart는 그림4-15와 같이 되고 이때에 점선으로 표시된 frame속의 fiow는 변환 routine이며, 본래는 control이 끊길 소분류 410을 40으로 변환하고 나머지는 100단위, 10단위의 2 column을 그대로 변환code화 한다.

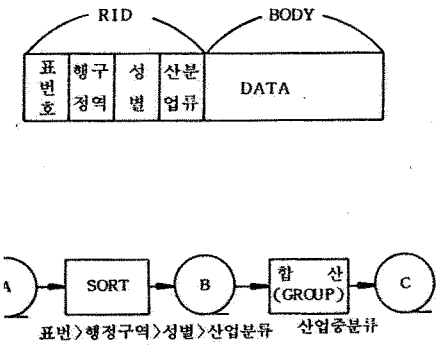


그림 4-14 직열형 (group) 합산의 예

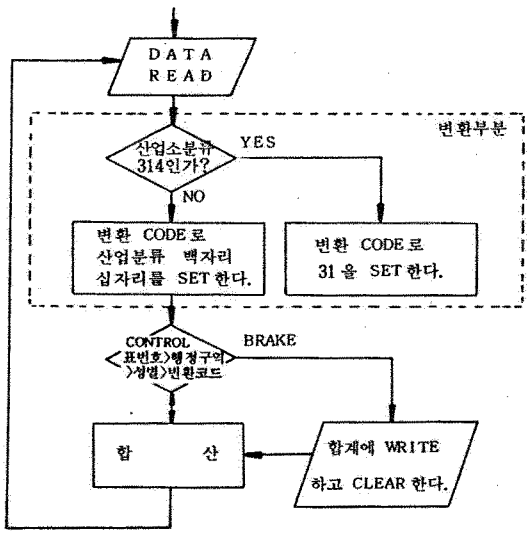
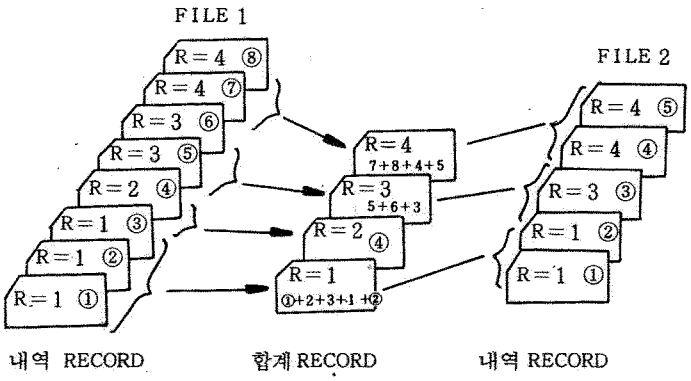


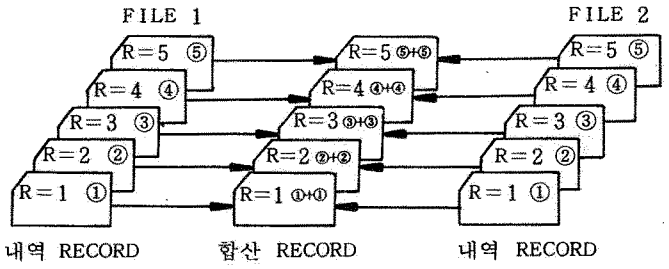
그림 4-15 group합산의 FLOW

나. 병렬형합산

병렬형합산은 2개이상의 input file을 merge하면서 control이 같은 data를 합산하는 형식으로 그림 4 - 16과 같이 file 1과 file 2의 control이 끊길 때까지 양쪽 file에서 data를 read하여 합산하는 merge형 합산은 file 1과 file 2가 R에 대하여 sort되어 있어야 하고 matching형 합산은 input file의 record가 1:1로 대응할 때의 합산형식으로 sequence와 record수에 있어서 file 1과 file 2가 1:1로 대응하는 경우에는 그림 4 - 18과 같이 file 1, file 2의 data를 상호 교대로 read하여 합산하여야만 합계산출이 가능한데 이방법은 sequence와 record수가 일치하기 때문에 합산에 필요한 sort와 control routine이 불필요하고 program도 간단해지나 matching형 합산은 일정한 frame으로 정형화된 통계표를 정기적으로 집계하고 특히 summary record수가 불변일 때의 연보(1년분의 합계)와 같이 지극히 한정적인 용도에 국한할 수 밖에 없으며 대규모집계에서는 record 수의 변동이 잦기 때문에 거의 적용이 불가능하다.



4 - 16 MERGE형 합산



4 - 17 MATCHING형 합산

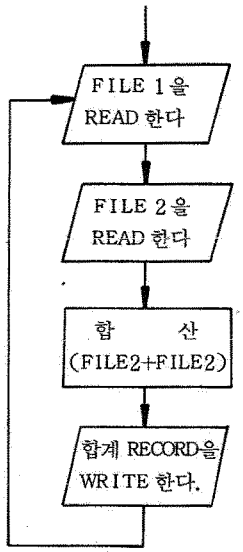


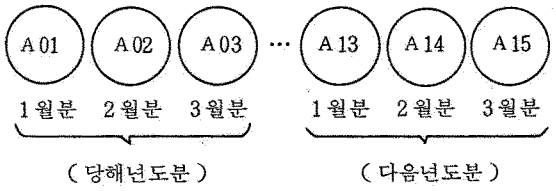
그림 4 - 18 MATCHING형의 FLOW

위와같은 merge형과 matching형의 2개 type의 병열형합산은 연보, 분기보와 같이 월보 file을 반복처리할 때 많이 활용되고 있는 바 그림 4 - 19, 20 에서 보는바와 같이 월보용 summary data를 기간으로 보고서발간을 하려면 월보용 summary data가 사전에 control항목 R에 대하여 sort를 전제로 하여 sort과정을 생략하였다.

직열형합산에서는 merge program에서 단일 file로 정리한 다음 합산이 가능하므로 병열형합산에 비하여 data량이 많아지는 불리함을

감수하지 않으면 아니된다.

(1) 월보용 summary tape



(2) 분기(3개월)보용 PROCESS

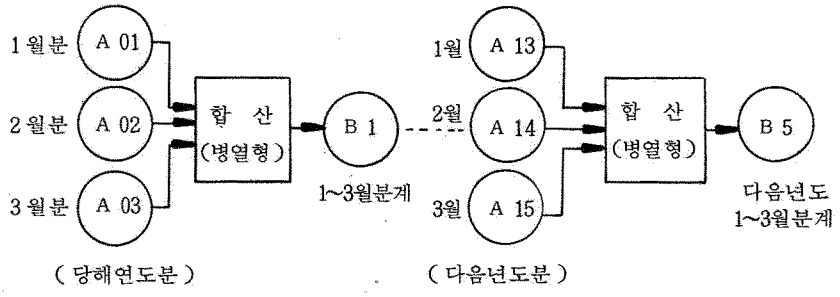
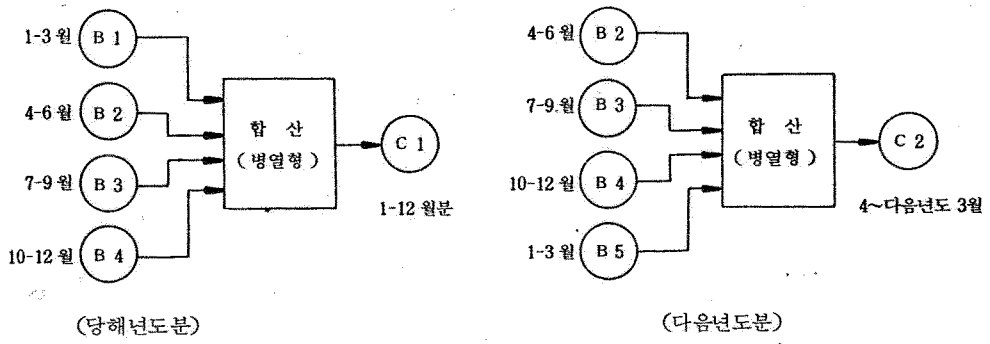


그림 4 - 19 분기보의 flow

(3) 년도보용 PROCESS



[년보다 년도보는 월보 SUMMARY를 사용해도 좋으나
분기보 SUMMARY를 사용하는 편이 능률적 임]

그림 4 - 20 년보, 년도보의 flow

다. 가감산형 합산

가감산형합산형식은 그림 4 - 21 에서 보는바와 같이 가산 file과 감산 file의 merge형 병열합산방식으로 집계과정 보다는 error의 발생으로 rerun을 필요로 할 때에는 위력을 발휘하여 방법이다.

그림 4 - 22 의 통계표에서 연령 20 세, 여자인 data일부의 error로 분포결과 70 이 틀리기 때문에 70 에서 남자 10 세쪽으로 1, 여자 10 세쪽으 2, 남자 20 세쪽으로 2 씩 data를 이동하여야만 바른 data를 얻을 수 있다고 하면 이러한 trouble은 주로 check program error에 근거하기 때문에 집계작업 도중에 비교적 많이 발생하게 된다.

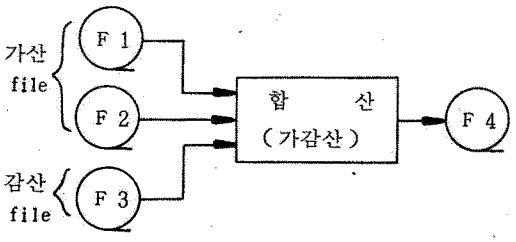


그림 4 - 21 가감산 합산

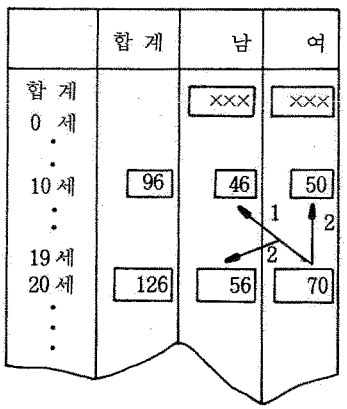


그림 4 - 22 영향을 미치는 cell

이와같은 경우의 rerun 방법은 error data 부분을 집계하여 수작업으로 정정하는 방법과 error data를 정정하여 전체 data를 재처리하는 방법, error data분의 집계와 정정 data분의 집계에서 가감산형 합산을 통하여 정정하는 3 개 방법이 있으나 첫째 방법에서는 합계란과 정정대상이 많으면 인력이 많이 소요될 뿐 아니라 정확성을 기하기가 어렵고 둘째 방법에서는 처리시간이 많이 소요되는 문제

가 있다.

따라서 셋째방법이 rerun시간이 가장 적게 소요되기 때문에 가장 합리적이나 rerun수순을 살펴보면 그림 4 - 23과 같이 첫단계에서 check한 file A로부터 해당하는 error data 5매분을 발취하여 분포와 sort, 합산과정을 거친 다음 file B_1을 형성한다.

둘째단계에서는 original data file A로부터 해당하는 error data 5매분을 발취한 다음 바른 data를 만들기 위하여 check program을 통하여 분포·sort·합산 과정을 통하여 file B1를 형성한다.

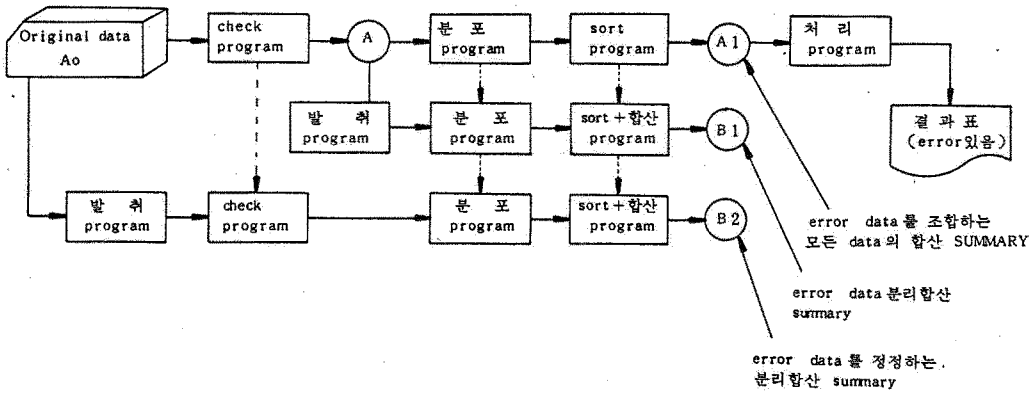
셋째단계에서 file A_1 B_2 B_1을 근거로 가감산합산을 하고 최종단계에서는 합산결과인 file A'_1은 바르게 수정된 summary data이므로 이를 file A_1 대신에 활용하는 과정을 거치도록 되어있다.

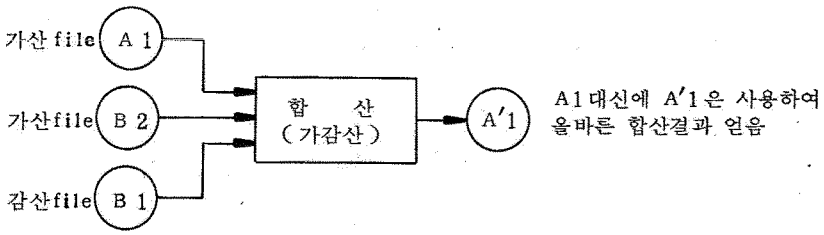
위와같이 error data의 집계와 정정data의 집계를 근거로 가감산형합산을 통하여 정정하는 제3방법의 수순을 살펴보았으나 이 방법의 장점은 rerun시간을 최소화 할수 있다는 점에 있는바 대량의 전체적인 data는 error data 발취에만 이용하고 발취한 error data를 대상으로 단순·간편한 발취program을 필요로 하므로 rerun을 위한 특별한 program이 불필요하고, flow는 복잡하나 logic이 간단하기 때문에 rerun할 필요가 있을때 채택해 볼만한 방법이라 할 수 있다.

예제로 돌아가서 error data의 발취는 20세여자 70매중 5매뿐이므로 연령 20세, 성별 여자라는 2개발취조건으로는 소기의 목적을 달성할 수 없기 때문에 이때에는 총괄번호, 조사표번호등을 근거로 발취하고 집계에 무관한 항목도 output에 수록할 필요가 있다.

또 error의 원인이 규명되더라도 rerun대상 data를 개략적인 범위로 압축해야 할 때에는 error data sheet마다 조사표번호의 지정에 필요한 5매의 error data를 발취하는 대신에 그 범위 내

의 모든data를 error data로 간주하여 그림 4 - 23의 처리를 하는편이 오히려 신뢰성있는 현실적 방법이라 할 수 있다.





A'1 내용

⋮				
10	99	47	52	=
⋮				
20	123	58	65	
⋮				

A1 내용

⋮			
10	96	46	50
⋮			
20	126	56	70
⋮			

B2 내용

⋮			
10	3	1	2
⋮			
20	2	2	0
⋮			

B1 내용

⋮			
10	0	0	0
⋮			
20	5	0	1
⋮			

그림 4 - 23 Rerun의 순서

4. 합산을 위한 RID 설정

분포program에서 print out summary record는 각 통계의 control level에 따라 RID의 내용이 서로 다르기 때문에 합산 program으로 합계를 구하려면 control항목에 대한 sort가 필요하고 control level과 합계를 구하는 항목이 서로 다르기 때문에 RID 설정에 각별한 주의가 필요하다.

그림 4 - 24에서 보는 바와 같이 3개의 summary record를 R×S×H type으로 통일하여 R에 대한 합계를 구하려면 제 1 표의 R는 행정구역 2 column과 직업분류 2 column의 2개항목, 제 2 표의 R는 산업분류 3 column 1개 항목, 제 3 표의 R은 행정구역 2 column과 남여별 1 column의 2개항목일 때 RID상의 항목배열 순서는 좌로부터 표번호에 이어 각표별로 R level의 크고 작은 순서에 따라 배열하는 것이 일반적인 방법이므로 그림 4 - 25 와 같은 배열이 되고 RID 설정은 각표 공히 RID의 길이를 같게하고 RID의 좌측으로부터 R의 순서대로 배열할 때 동일 위치에 있는 각 항목의 column수는 최대 column의 항목에 마추어야 한다.

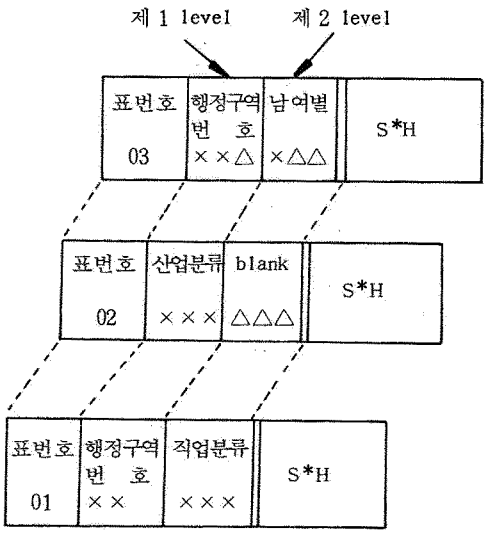


그림 4 - 24 RID의 배열

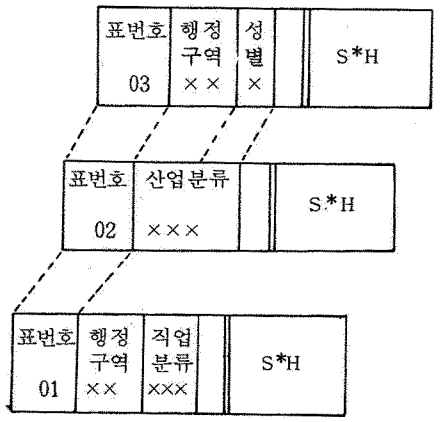


그림 4-25 RID의 SET (미조정)

column의 수정을 하지 아니한 상태에서 각표의 R을 sort하면 그림 4 - 25와 같이 되어 R의 제 2 level (남여별, 직업분류) 합계를 구하는 경우에는 RID의 길이가 같기 때문에 별문제가 없으나 제 1 level (행정구역, 산업분류) 합계의 경우에는 sort key의 지정에 혼란이 야기되고, R의 제 3 level 항목이 추가되는 경우에는 sort 자체가 불가능하기 때문에 R의 합계 level에 마추어서 column을 조정해야 한다.

다음 각표의 합계 level이 일치하지 않으므로 그림 4 - 24의 예에서 R의제 2 level의 합계를 구하려면 표번호 > R의 제 1 level (3 column) 도 sort하여 직열형합산을 하므로서 제 1 표와 제 3 표는 각각 직업분류계와 남여별계를 구할수 있으나 합계가 불필요한 제 2 표의 무의미한 합계가 산출되는 상황을 피하기 위해서는 그림 4 - 26의 flow chart와 같이 합계가 필요한 통계표와 합계가 불필요한 통계표를 선별하고 합계작업이 불필요하고 표두가 많은 경우에는 「합계를 필요로 하는 통계표와 필요로 하지 않는 통계표」로 나누어 group화 하므로서 능률적인 처리를 하여야 한다.

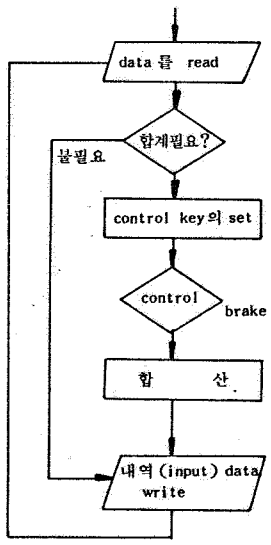


그림 4 - 26 합계의 by-path

5. 표준 flowchart

합산program의 표준flowchart를 직열형합산(직접합산)과 병열형합산(merge 합산)으로 구분하여 설명하면 다음과 같다.

가. 직열형합산(직접합산)

그림 4-27의 직열형합산의 flowchart는 분포program의 표준flow chart와 유사한바 그 이유로는 분포program과 합산program은 취급data가 개별data 또는 summary data인가 여부와, 분포점P를 data별 산출 또는 control이 끊어질 때마다 산출 여부에 따라 결정되는 것이므로 분포program과 합산program은 기본적으로 동일형식의 flow가 될 수 밖에 없기 때문이다.

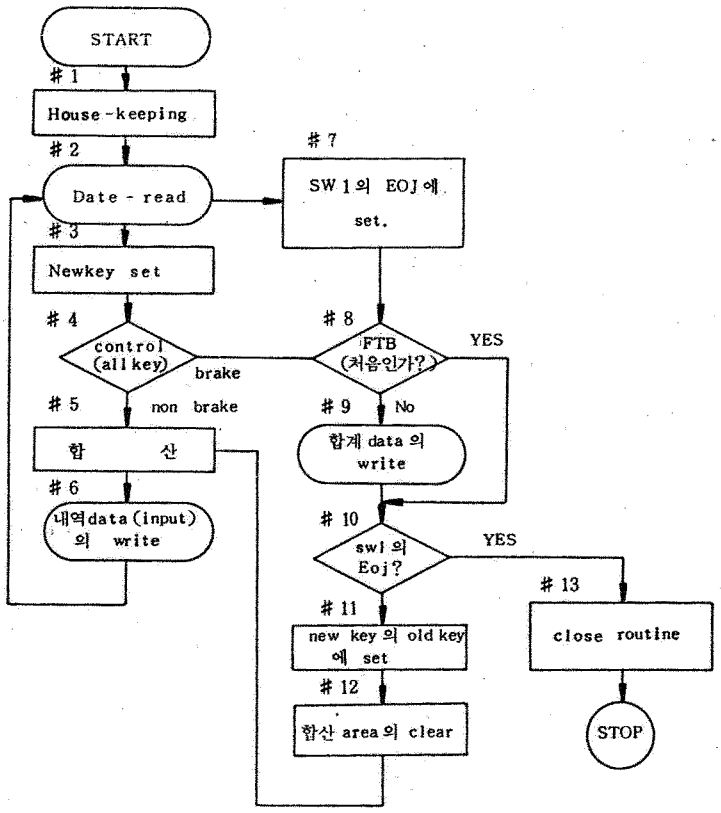


그림 4-27 직열형 합산의 flowchart

그림 4 - 27의 flowchart를 순서에 따라 설명하면 다음과 같다.

#1 … 사전처리 routine : input file, output file의 open.

#2 … input data의 read

#3 … 합계대상항목을 제외한 sort의 control key를 new key로 set.

#4 … 분포program과 동일하게 key 전체를 대상으로한 new key와 old key의 비교.

#5 … 합산routine의 실행

#6 … 내역data의 "print out : 통상 합계와 내역을 동시에 요구하므로 합계산출용 내역data (input summary data)를 print out하되 합계만 요구하는 경우에는 생략.

#7 … EOJ switch의 set.

#8 … control이 처음으로 끊기는가를 판정하는 switch.

#9 … 합계data의 print out : output RID의 합계를 구한 항목에 "T", "0"등과 같은 합계를 의미하는 code의 set.

#10 … EOJ의 판정.

#11 … new key를 old key로 이송.

#12 … 합산area를 input data length만큼 clea, 또는 합산area에 input data를 이송하여 #5의 합산을 by-path.

#13 … 사후처리routine : input file과 output file의 close와 record count의 종료.

나. 병열형합산 (merge 합산)

그림 4 - 28의 병열형합산 (merge 합산)의 flowchart는 input file수를 3개로 제한하고 있으나 이를 n개의 file로 확장하려면 merge형합산의 flowchart에서 특정file의 reading routine이 file수 만큼 있고 그림 4 - 28과 같이 flowchart가 횡으로 장대해지기 때문에 file수와 routine과의 관계가 명백하여 program의 수정과 merge pogram, data 갱신 program작성이 용이하다.

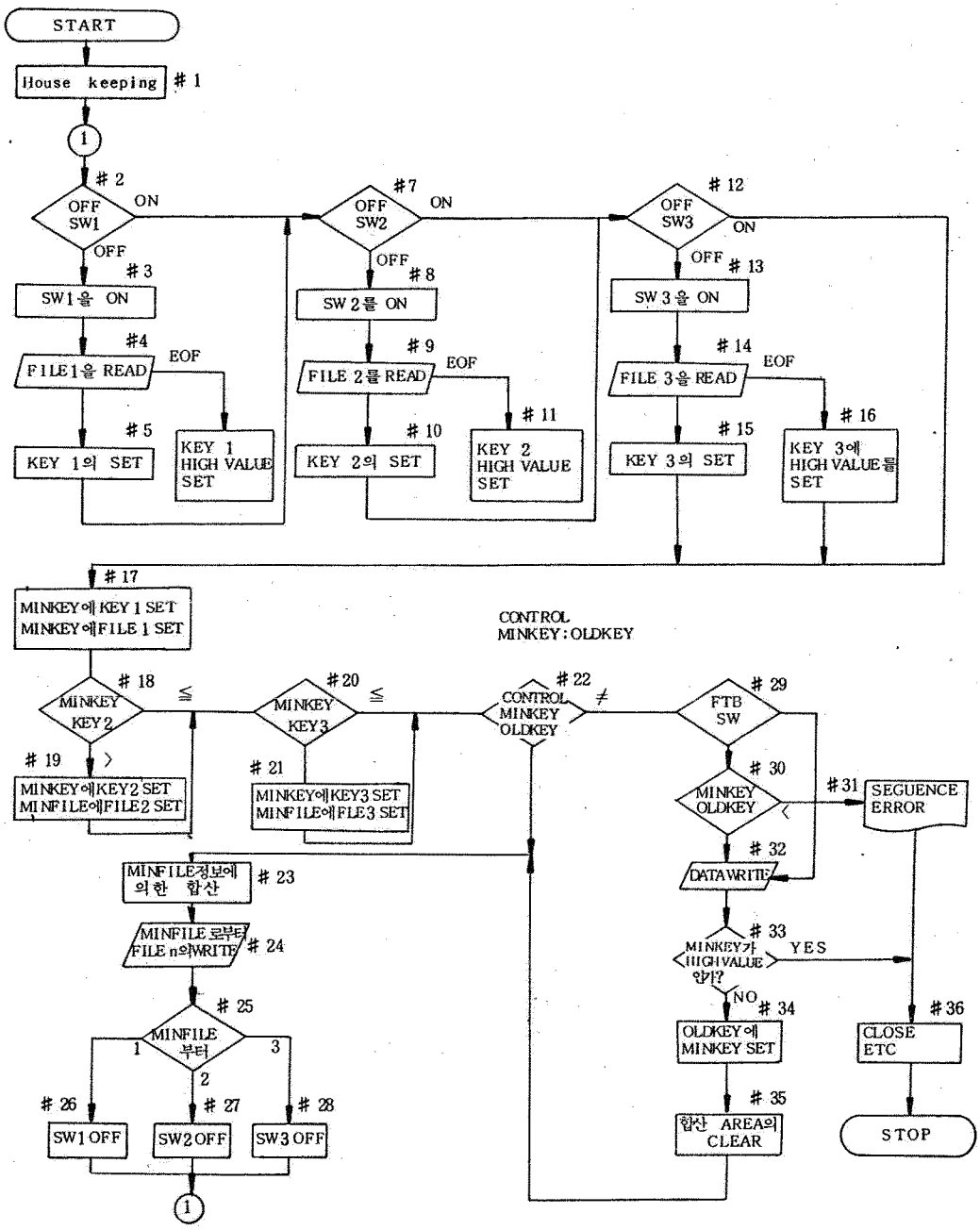


그림 4 – 28 병열형 합산의 flowchart

그림 4 - 28의 flowchart의 순서에 따라 설명하면 다음과 같다.

#1 ····· 사전처리routine : input file과 output file open.

#2 - #6 ··· file 1의 reading관계 routine : SW1(#2)는 초기값을 "OFF"로 하여 reading routine실행 여부의 control기능을 가짐.
따라서 각 input file별로 control항목을 set할 field (KEY 1-KEY 3)를 마련하고 reading routine (#4)을 실행한 다음 file 1의 control 항목 set (#5).
input file1이 EOF가 되면 KEY 1에 그 system에서 가장 높은 분자인 High value(분포 program에서 설명한 "9"도 무방하나 sequence check를 하므로 요주의)를 set. (#6)

#7 - #11 ··· file 2의 reading routine : #2-#6의 기능과 동일.

#12- #16 ··· file 3의 reading routine : file 수가 4개이면 추가되는 1열 file(file 4)의 reading routine 추가, file수가 2개이면 file 3의 reading routine 삭제.

#17 ····· 최저의 key file의 검색을 위하여 MINKEY에 file 1의 KEY 1을 이송하여 최저 key의 file 1으로 간주하고 MINFILE에 "1"을 설정.

#18- #19 ··· MINKEY와 KEY 2의 비교 (#18) : MINKEY > KEY 2면 KEY 1> KEY 2이므로 MINKEY에 KEY 2, MINFILE에 2를 set (#19)

#20- #21 ··· MINKEY와 KEY 3의 비교 (#20) : MINKEY < KEY 3이면 MINKEY와 MINFILE을 갱신.

22 ····· MINKEY와 처리된 OLDKEY를 비교하여 control brake여부 검색 : MINKEY에 최저 key의 set전제.

23 ····· 합산 routine: 가산 file 검색은 MINFILE의 file 번호로 분별.

24 ····· 내역 data의 print out routine : MINFILE 번호에 의하여 대상 file의 data를 분별하고 병열형합산에서는 내역data의 print out은 태무.

25- # 28 ··· 대상 file의 reading routine결정 switch를 set.

29 ····· control brake된 최초 1회는 합산에서 제외되므로 print out routine을 우회.

30- # 31 ··· sequence check routine : merge형 합산에서 control항목에 대한 sort를 전제로 한 각 file의 sort key 지정 error, 합산 program의 control 항목 지정 error의 조기발견, sequence down 시 down된 file 검색 message가 필요한바 예제의 3개 file에 대한 sequence check를 위해서 각 file의 reading routine (#2-#16)을 마련하는 편 보다는 control brake 후에 MINKEY와 OLDKEY를 비교하는 편이 routine이 간단하고 비교회수가 적고 sequence down 시 MINFILE 번호를 근거로 file검색이 가능한 장점이 있음.

32 ····· 합계 data의 print out.

33- # 35 ··· EOJ의 판정 : MINKEY에 high value set 여부 확인 (# 33), EJO가 아니면 OLDKEY에 MINKEY를 이송, 합산 area clear.

36 ····· 사후처리 routine: input file과 output file의 close, record count의 인쇄.

V. 가공편성 PROGRAM의 설계

1. 가공편성 program의 목적

분포program에서는 input table의 작성형식과 결과표 내용(수치)의 정도에 있어서 조사내용과 집계방침에 따라 다르나 전반적인 결과표의 요구사항을 해결하고, print에 필요한 data편집기능을 수행한다.

그러나 분포program 설계에서 살펴본 바와 같이 분포program에서는 집계시간 단축과 program 수 감축에 알맞는 최소한의 내용으로 압축해야 하기 때문에 분포program과 편집program사이에 합산program과 가공편성program을 삽입 처리하여야 한다.

합산program은 summary data상호간에 합계를 구하는 것이므로 summary data내에서의 합계·평균 등의 산출은 가공편성program으로 처리하고, summary record상의 항목배열순서가 결과표에서 요구하는 인쇄순서와 다를때 결과표 인쇄양식에 따라 data배열변환, 표두부분의 분할처리, 표측의 절단·접속등 편집program 기능에 속하는 사항을 처리상의 편이를 위하여 가공편성program에서 처리하는 것이 상례로 되어 있다.

이와 같은 가공편성program의 목적을 간략하게 정리하면 도수 분포된 summary data의 내용을 보충하여 완전한 결과표를 생산할 수 있는 summary data로 변환작성하고, 결과표의 요구내용을 충족하는 data를 배열변환하는등의 처리를 함으로써 편집program 처리를 용이하게 하는 것으로 요약할 수 있다.

2. 가공편성program의 기능

가공편성program의 기능은 결과표의 분포양식과 인쇄양식에 따라 다

르나 대략 다음의 8개기능으로 집약할 수 있다.

① 표내계 (표두계, 표측계등) 산출

② 평균치, 구성비, 지수 계산

③ 분포항목 삭제

④ 파생표 작성

⑤ 분포항목의 배열변환

⑥ record의 절단과 접속

⑦ zero data의 복원

⑧ print 개행 (改行) 문자의 부가

가. 표내계의 산출

가공편성program으로 합계를 구하는 경우는 하나의 분포program으로 점유memory를 절약하기 위하여 결과표수를 최대화 하는 경우와 summary record length를 축소하여 처리시간을 단축하는 경우, 분포program의 기능을 단순화하는 경우를 들수 있는 바 단일한 분포program으로 처리하는 결과표수를 최대화하고저 할때 여러개의 분포program으로 합계란을 생략한 상태에서 분리 집계하게 되는데 가공편성program에서는 생략된 합계를 사후적으로 복원하게 된다.

다음으로 summary record length를 축소하여 처리시간을 단축하는 경우에 분포program의 output record수가 많고 편집program 에 이르는 처리과정이 많을때 결과표의 table image에 합계를 포함하면 record length가 길어지고 합계record를 수록하는 record 수도 증가하여 많은 sort시간이 소요되므로 분포program에서 record length와 record 수의 증가를 피하지 않으면 아니된다.

따라서 분포program에서는 최단의 record length, 최소의 data 량을 편집program의 input으로 하여 본래의 결과표 길이와 양이 되도록 하고 그 사이의 처리step에서는 필요에 따라 단계적으로 증가시키는 것이 바람직하나 그러한 설계는 현실적으로는 불가능하기 때문에 장

래에 필요한 내용은 뒤로 미루는 태도로 설계에 임하는 것이 중요하다.

끝으로 분포program기능의 단순화는 결과표의 합계란을 설정하여 분포program으로 분포·합계기능을 수행하기 보다는 program의 신뢰성과 연관되기 때문에 분포와 합계를 분리하여 분포를 먼저 수행함으로로써 program의 부담을 경감하고 이어서 가공편성program으로 합계를 구하는 것이 경제적인 설계라 할 수 있는 바 합계를 가공편성program으로 구할 때 결과표 양식과 분포program, 가공편성 program에서의 table image는 그림 5-1과 같고 ▨부분이 구하는 합계란이다.

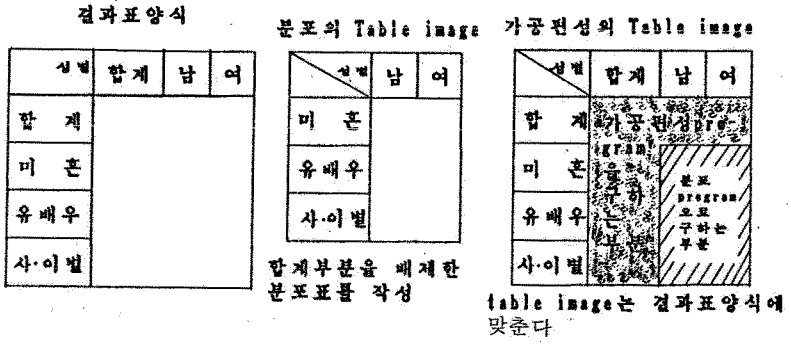


그림 5-1 Table image

나. 평균, 구성비, 지수 등의 계산

평균·구성비·지수등의 계산은 통상 전항목에 걸쳐서 실수값(實數置)이 분포된 다음에 처리 가능한 것으로서 예를 들어 행정구역별 분포를 구하고 전국합계를 합산program으로 구하고저 할때 분포program에서 행정구역별 평균치를 산출하면 전국합계의 평균치는 바르게 구할 수 없기 때문에 이와같은 경우에는 분포program과 합산pro-

gram에서는 실수값 원형상태에서 처리하여 전국합계의 실수값을 구한 다음에 평균치를 계산하여야 한다.

이러한 맥락에서 연보 계산과정에서도 그림 4-19, 20에서와 같이 년간 월보의 summary record를 합산하여 연보용 summary record를 작성하게 되므로 평균치를 내포하는 결과표에서는 월보용 remmary data를 가공편성program input으로서의 합산program output data로 보관한 상태에서 평균치 포함여부에 대한 판단은 매우 번거롭기 때문에 보존data는 합산program의 output과 일괄 결정해도 무방하다.

그림5-2는 평균치를 포함하는 결과표양식 예에서 분포program 에서는 평균소득란을 총소득으로 간주하여 cell을 설정하고 input data의 소득액을 분포한 다음 가공편성program에서는 그림5-3 좌측과 같이 표내계를 산출하여 평균소득을 계산하고 그림5-3 우측과 같이 본래의 총소득란과 치환하여 구성비·지수등을 계산하여야 한다.

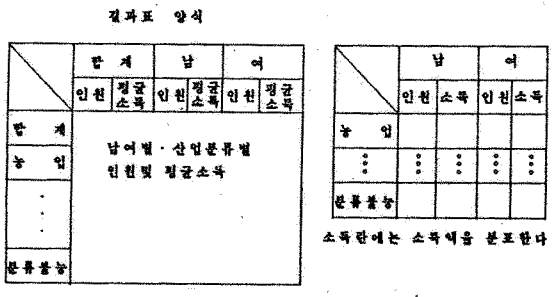


그림 5-2 평균치가 있는 분포

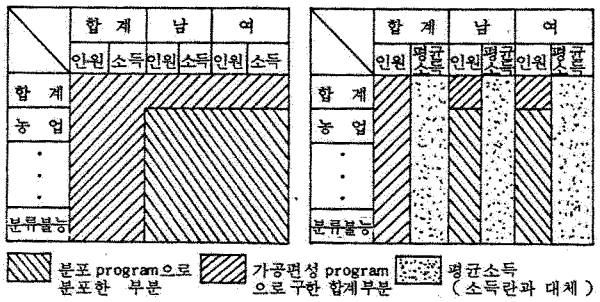


그림 5-3 평균산출 절차

결과표 양식

농 림 수 산 업			비 농 림 수 산 업			
자영업주	가족 종사자	고용인	자영업주	가족 종사자	고 용 인	
					상용	일용

확대

table image

농 림 수 산 업				비 농 림 수 산 업			
자영업주	가족 종사자	고 용 인		자영업주	가족 종사자	고 용 인	
		상용	일용			상용	일용

일부분이 확대된 부분임

그림 5-4 항목 구분의 통일

다. 분포항목의 삭제

분포program에서 분포하는 항목에는 결과표 구성항목 상호간 구분수가 일정하지 않고 구성이 불규칙하거나, 부분적으로 내역표현을 생략하는 상황에서 결과표에 표현할 항목과 표현이 불필요한 항목이 있는 바 이와같은 경우에는 결과표가 요구하는 table image로 분포하면 program이 복잡하기 때문에 program을 단순화 하고 trouble처치를 쉽게 한다는 방침 아래 동일한 항목에서 구분수가 다를 경우 하위 세분류 쪽으로 통일하여야 한다.

그림 5-4 상단 결과표 양식에서 농림업의 고용인란과 비농림업의 고용인란은 구분(분류)이 다르기 때문에 분포program에서는 그림 5-4하단과 같이 농림업의 고용인란을 비농림업의 고용인의 구분 수와 같도록 세분하여 분포하고 가공편성program에서 다시 통합처리하고 결과표양식이 그림 5-5와 같이 총수부분이 종속된 내역보다 세분된 경우에는 종된 내역의 구분을 주된 총수의 구분과 일치시키고 가공편성program에서 합계를 구하면 불필요한 항목은 output에서 삭제된다.

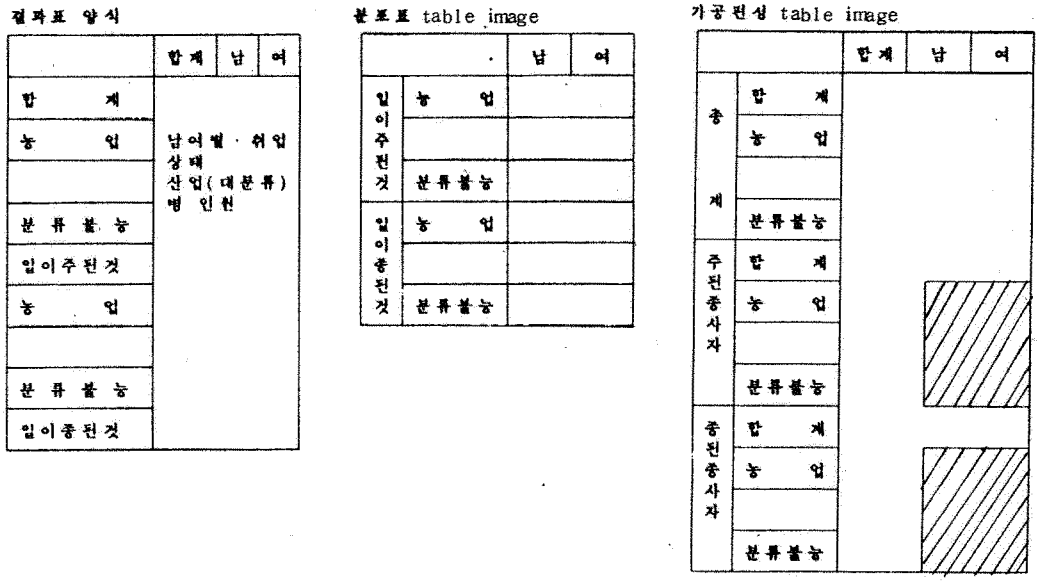


그림 5-5 「합계」가 세분되어 있는 경우

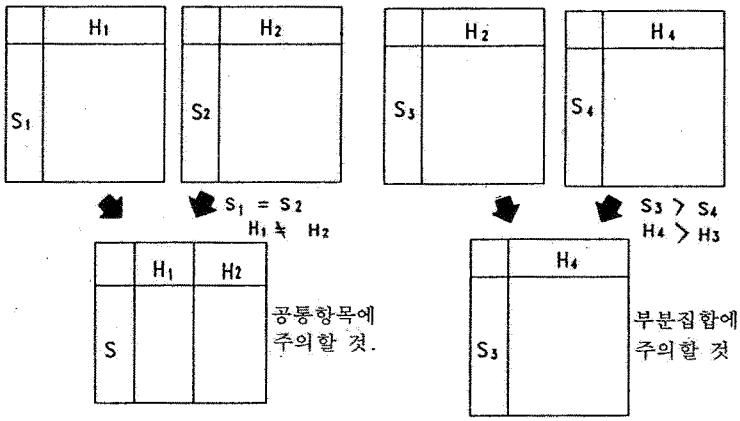


그림 5-6 파생표

위에서 항목의 삭제와 관련하여 분포program에서의 table image 채택방법에 대하여 설명하였으나 분류의 세분화로 소요memory가 커지는 경우와 합계를 구하는 회수보다 분포하는 data가 적은 경우, 합계산출절차가 분포절차보다 번거로운 경우 등에서는 결과표양식 그대로 table image를 채택하여 분포하는 경우가 있는 바 그림 5-5에서는

종된 내역에 관하여 A와 C에 분포하고 주된 총수는 B를 A에 가산하는 절차를 택하는 경우도 적지 않다.

라. 파생표 작성

분포program의 table image는 결과표를 표단위로 대응하여 작성하기 때문에 summary record도 표단위로 수록하는 것이 상례이나 때에 따라서는 여러표에 해당하는 내용을 모아서 하나의 summary data로 하여 수록하는 경우도 없지 않다

그림 5-6 은 2종의 통계표를 하나로 정리하는 요령 예로 표두, 표측 항목이 공통되거나, 부분집합을 형성하는가에 따라 그림과 같이 두가지 방법으로 구분되며 공통항목의 경우에는 그림 5-6 의 좌측과 같이 공통항목을 하나의 항목으로 모아서 R×S×H형으로 형성한다.

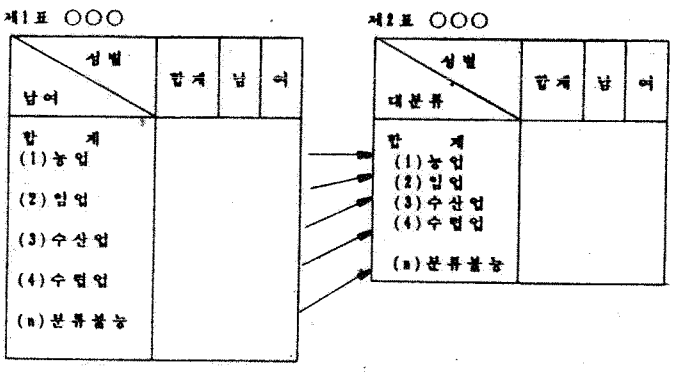


그림 5-7 .부분집합의 파생묘 예

예를 들면 그림 5-7 좌측 제 1 표는 산업 소분류를 표측으로 하는 통계표로서 동시에 산업 대분류를 표측으로 하는 통계표를 필요로 하는 경우 간단하게는 개별적으로 table image를 형성하여 처리할 수도 있으나 소요memory가 커지고 summary record수가 증가하는 단점이 있기 때문에 그림 5-7 제 2표와 같이 제 1표의 산업 대분류를 발

취한 것으로 보고 분포program에서는 제1표의 table image만으로 분포작업을 하고 가공편성program으로 발취하여 제2표를 작성한다.

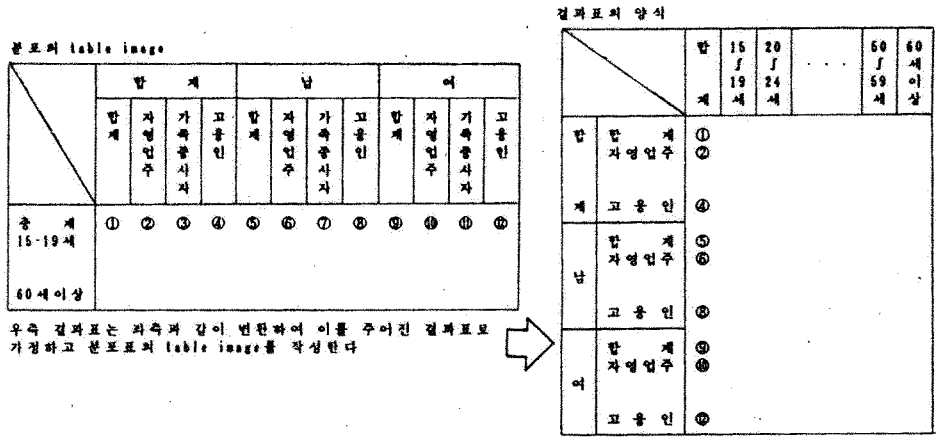


그림 5-8 배열 변환의 예

이 방법은 합리적이긴 하지만 현실적으로 단위 통계표에서 그 일부를 발취하여 별개의 통계표를 구성하는 경우는 극히 희소하고 대개의 경우 여러 종류의 통계표로부터 일부 필요한 부분을 발취하여 재구성하여야 하기 때문에 가공편성program의 처리과정에서 발취하여야 하며 따라서 복잡한 program에서는 독립적인 통계표로 처리하는 것이 상례이나 표1, 2가 실수표와 구성비표 등 별개 형식으로 되어 있는 경우에는 실수표를 분포program으로 처리하고 구성비표는 가공편성program으로 처리하는 것이 편리하다.

마. 분포항목의 배열 변환

table image의 표두, 표측과 결과표의 표두, 표측이 뒤바뀐 경우에 가공편성program에서 표두, 표측을 원상 회복하여야 하는데 이와같은 예는 파생표 작성 이외도 흔한 case로서 분포program table image 형성방법에 의존하여 수정하는 것이 상례이다.

분포program의 설정방침 가운데 control항목의 설정은 통상 R → S → H 순으로 결정하나 제1표의 표측항목 S가 제2표에서 표두항목이 된 경우에는 제2표의 표두를 표측과 치환하여 표측으로 간주하고 table image를 설정하여 S에서 control을 끊고 가공편성 program까지는 뒤바뀐 상태 대로 처리하여야 한다.

결과표양식 예인 그림 5-8에서 우측의 결과표를 분포program에서 구하고저 하는 경우 여타 결과표가 『연령』으로 control되고 있으면 표두항목이 control로 설정되고 summary data는 표두1열씩 수록하는데 이는 표두와 표측을 바꾸어 표측1행씩 수록하는 것과 같고 이방법은 table image작성에 편리한 방법이다.

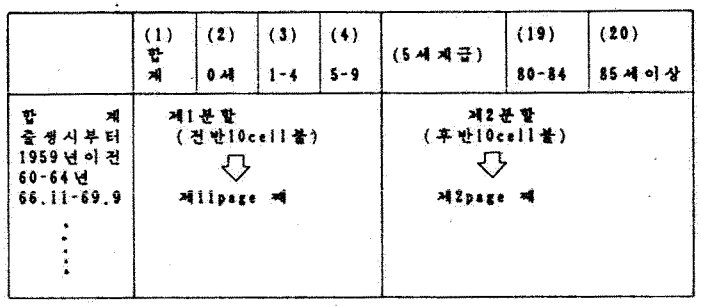


그림 5-9 절단의 예

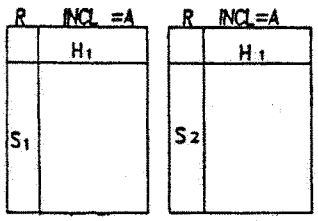


그림 5-10 inclunsion에 의한 Zero data

가공편성program에서는 그림 5-8좌측과 같은 가로형태의 summary record가 read되므로 이를 그림 5-8우측과 같이 세로의 형태로 배열변환을 해야 하기 때문에 당초에 변환처리하는 편이 table image를 이해하는데 편리하다.

바. record의 절단과 접속

record의 절단과 접속은 결과표를 print할때 1 line 또는 1 page분의 결과표 print image에 대한 summary record를 결과표 인쇄양식에 따라서 표두를 분할하고, 표측을 line별로 절단하여 결과표 1 line분의 record를 작성하는 것을 말하는 바 가공편성program의 table image가 그림5-9와 같다고 할때 표측 line당 cell수 : 20cell당 print column수 : 9column으로 가정하면 line당 인쇄data량은 160 column이기 때문에 132 column/line의 system printer로 인쇄하려면 부득이 표두부분을 제 1 분할과 제 2 분할로 분할하여 순차로 인쇄하지 않으면 아니된다.

분할은 물리적인 반분위치가 아닌 표두구분이 명확한 위치에서 행해야 하는데 표두부분의 물리적인 반분위치에서 분할할 경우 관련항목이 좌우page로 분리되어 보기에 불편한 모양새가 되는바 산업분류, 직업분류등과 같이 group별로 분류되는 경우에는 특히 주의하여야 한다.

다음 record접속은 절단과는 반대로 표측 1 line마다 record를 연결하여 1 page분의 결과표 인쇄record를 작성하는 것으로서 가공편성program에서의 record접속 예는 흔한 case가 아니나 편집program에서 page별로 data를 정리하는 편이 유리할 때에 활용한다.

사. zero data의 복원

분포program의 output record에는 inclusion 조건과 write형식이 원인이 되어 그내용이 zero인 data가 write되는 경우가 있는 바 이를 zero data 또는 zoro record라 한다.

inclusion의 조건에 의한 zero data는 한표의 분포program에서는 여하한 inclusion 조건하에서도 발생하지 않지만 복수의 표를 하나의 분포program으로 분포하고 각표의 inclusion 조건이 서로 다를 때에는 분포program의 type(1, 2, 3)에 관계없이 zero data의 발생을 피할 수 없다.

예컨데 그림 5-10과 같은 2개의 통계표 중 제1표에서는 모든 data를 대상(INCE=ALL)으로 하고, 제2표에서는 A라는 data 집단을 대상(INCL=A)으로 한다고 가정하고 control항목을 2개표 공히 R, 인쇄는 표단위 write라 할때 분포routine에서는 제1표는 모든 input data가 분포대상이 되고 제2표에서는 data A만 분포대상이 되므로 control brake시점에서의 table image 내용은 제1표에서는 S×H의 어떤 cell에 분포된 수치가 있고 제2표에서는 control이 끊길 때까지의 data집단내에 data A의 유무에 따라 분포여부가 결정되기 때문에 A가 없을 때 zero인 상태를 수록하면 zero record가 형성되는 바 수록형식에 있어서 전체 data를 대상으로 표단위분포를 하는 경우에도 수록형식에 따라서는 zero record가 발생하는데 그림5-11과 같이 1개표분의 table image에 대한 수록은 표측단위수록으로서 S_2, S_4 line에서 zero record가 발생하게 되는데 이와같은 결과는 S_2, S_4의 2 line에 대한 분포data가 없는데서 오는 결과라 할 수 있다.

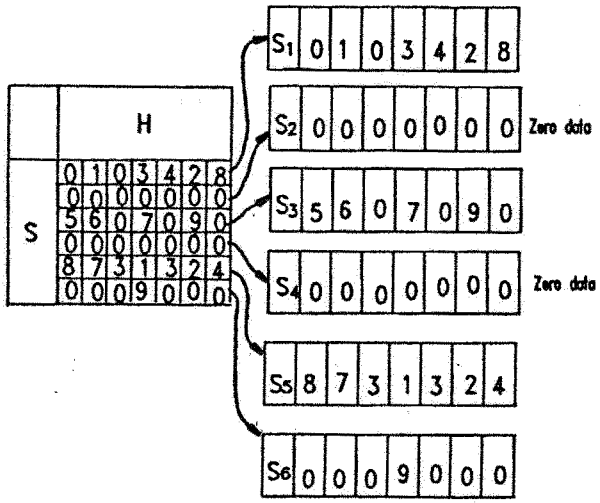


그림 5-11 write table에 의한 Zero record

이와같은 zero data는 계산기술상 아무런 의미가 없는 data 임은 물론, 그만큼 record수와 처리시간이 증가하여 처리실무에 무익하므로 처리시간 단축을 위해서는 zero data의 삭제가 필수적인 바, zero data의 삭제방법은 inclusion조건 측면에서는 write하기 이전에 switch를 통하여 zero record의 분포여부를 판단하는 것이 가장 간단한 반면 write측면에서는 line별로 H에 대하여 line 수만큼의 switch를 설정하고 matrix code로 control하여 그 내용이 zero인가를 check하는 방법이 있으나 번거로운 단점이 있다.

이상과 같이 삭제한 data를 결과표상에 다시 재생하여 print할 필요가 있을 때에는 삭제한 data를 재생하게 되는데 이와같은 작업을 zero data의 복원이라 한다.

복원대상은 주로 표측단위수록일 때의 zero data이고 표측단위수록에서는 복원대상 zero data가 거의 전무한 바 이는 인쇄상태에서

상태에서 결정되므로 전자는 table image를 기준하여 1 page의 표측에서 zero data를 삭제함으로써 압축인쇄되어 다양한 형태의 R 밑에서 숫자의 reading이 어렵기 때문에 표측의 인쇄위치를 정돈할 필요가 있는 반면 표단위에서는 1 page분의 zero data를 삭제해도 page단위로 인쇄가 생략되므로 수치판독에는 어려움이 없다.

zero data의 복원은 편집program에서도 가능하지만 이때에는 본래의 편집기능외에 복원처리기능을 보유해야 하므로 잡다한 기능을 갖는 가공편성program에서 처리하는 편이 편성program의 부담을 덜 수 있다는 점에서는 편집program으로 처리하는 방법은 바람직한 일이 못된다.

아. print 개행(改行)문자의 부가

결과표를 print할 때 그림 5-12와 같이 인쇄하여야 할 대상record(BODY)앞에 특수문자를 부가하여 인쇄행간의 간격과 page의 변환을 지정하는 정보를 carige control charactor(C.C.C)라 하는 바 이와같은 C.C.C는 일종의 편집program기능을 갖는 명령어로서 program의 구성상 가공편성program에 분담시킬 수도 있다.

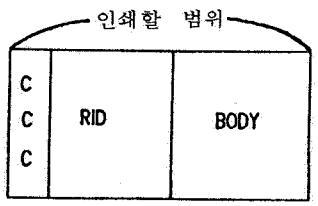


그림 5-12 ccc의 위치

이상 가공편성program의 기능에 대하여 살펴 보았으나 표내계, 평균치, 구성비, 지수등의 계산과 분포항목의 삭제, 파생표의 작성 분포

항목의 배열변환, record의 절단과 접속, zero data의 복원, print 개행항목의 부기등의 제기능은 결과표작성에 기본적으로 필요한 기능으로 결과표에 따라서는 여러가지 기능 가운데 부분적으로 필요한 경우와 모든 기능이 필요한 경우가 있으나 각 기능을 명확히 구분하여 처리할 수 없는 경우도 적지 않다.

이와같이 가공편성program은 여러가지 다양한 처리를 하고 동일한 처리 내용일지라도 반드시 처리방법이 같다고는 할수 없기 때문에 가공편성program전체에 대한 공통성과 처리방법의 정형을 찾기가 쉽지 않다.

3. 가공 program의 처리

가. 2개처리 step

가공편성program의 처리내용을 간단히 표현하면 분포표 image의 data를 연산을 통하여 결과표image의 data로 변환하는 일이라 할수 있는데 실제의 program처리의 경우에는 각표의 결과표에서 요구하는 내용 중 아직 구하지 못한 부분을 구하여 보완하는 step (step 1)과 data형식을 결과표 인쇄형식으로 바꾸는 형식변경 step(step 2)의 2개step이 필요한데 이들은 하나의 program 가운데 내장되어 있는 경우와 2개의 program으로 분할되는 경우가 있다.

2개 step을 하나의 program에서 처리하는 경우는 가장 일반적인 구성방법이며 두개의 program으로 처리하는 경우는 그림5-14와 같이 중간 file A를 연결매체로 하여 처리하는 것으로서 처리내용이 복잡할 때에 기능을 단순화하여 program을 평이롭게 하거나, 기억용량의 제약으로 하나의 장대program으로 처리 불가능하고 파생표를 처리하고저 할때 효과적인 방법이나 실제에 있어서는 step 1, step 2를 over-lay구조로 하여 하나의 program으로 간주하여 처리하

는 경우가 많다.

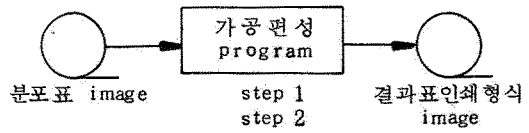


그림 5-13 하나의 program

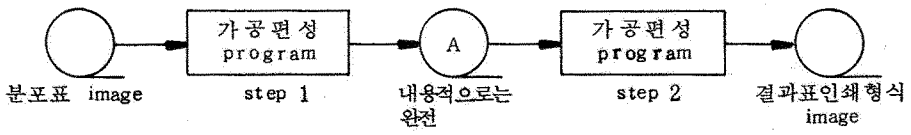


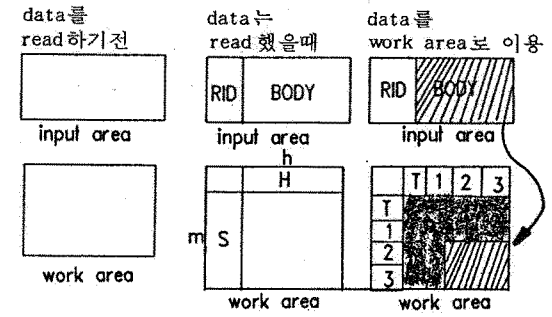
그림 5-14 두개의 program으로 분할

나. step 1의 처리순서

결과표 내용보완작업에는 input data상에 없는 항목과 구분의 추가와, input data상의 내용을 다른 내용으로의 치환등의 경우가 있는데, 표두계·표측계등 표내계산출은 전자의 예(그림 5-1)이고 평균치 산출은 후자의 예(그림 5-2)로서 경우에 따라서 결과표 작성에 양자 모두 필요하거나 한쪽만 필요한 경우가 있다.

그림 5-15 (step 1의 내부처리)에서 보는바와 같이 step 1의 처리순서는 input area의 data를 work area로 이송하고 다시 work area에서 표두계·표측계등의 분포표로서 결여부분을 연산 처리하여 모든 항목에 분포하고 분포된 실수값을 근거로 비율계산, 통계해석계산을 함으로써 step 1의 처리를 완료한다.

work area는 가공편성program의 기능을 수행하기 위한 작업용 area로서 input data를 이용하는 경우, 독립된 작업용area를 확보하는 경우, output area와 겸용하는 경우가 있는 바 memory에 여유가 있다면 독립된 작업area의 확보가 program작성에 편리하다.



양 area를 처리 할 수 있는 최대 memory 확인

표별로 처리하는 표의 work area의(m,n) 를 정의한다.

이동이 끝나면 표 내계, 평균의 산출등을 실시한다.

그림 5-15 step1의 내부처리

다. step 2의 처리순서

결과표 인쇄양식으로 변환하기 위하여 가공편성 program의 기능중 『분포항목의 삭제』부터 『print개행문자의 부가』까지의 기능을 동원하는 경우에 그 처리방법에는 work area를 표측단위에서 보완하는 표측단위보완과 work area의 data를 input record의 특정 control단위에서 보완하는 표단위보완방식이 있다.

표측단위보완은 주로 분포program의 수록형식이 표측단위 summary record를 취급하는 경우로 그림 5-16과 같이 step 1의 work area를 1 line분만 정의하고 data를 read하여 work area에 이송하고 보완처리한 다음 인쇄형식에 따라 수록하는 순서를 반복하는 것으로써 write기능, data의 일부삭제기능, data의 발취기능,data의 배열변환(표두항목의 순차변환)기능, data의 절단(표두분할)기능 등을 수행하나 C.C.C를 부가하는 예는 극히 드물다.

표두를 분할할 경우에는 그림 5-17과 같이 인쇄data를 식별하고 sort key화 하기 위하여 output record의 RID에 분할번호를 부여하되 분할번호의 위치는 RID의 어디라도 무방하나 타표의 분할번호의 위치와 길이에 합치되어야 한다.

표단위보완은 work area의 특정한 control group별로 pool되는 경우로서 zero data의 복원이 이에 해당한다. work area가 표단위인 경우에는 data의 일부삭제, data의 발취, data의 배열변환(표두 표측의 교체변환과 표두 표측내에서의 순차교체변환), C.C.C의 부가, 표두분할, 표분할등 주요기능을 수행한다.

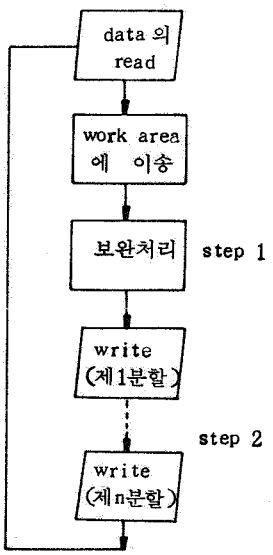


그림 5-16 표측단위 보완

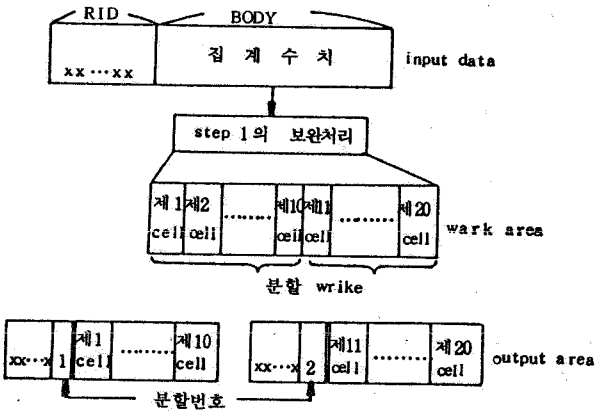


그림 5-17 분할번호의 set

그림 5-18은 표두 표측의 분할 예로서 표두부분을 2분할하여 표측단위로 수록할 때에는 제1분할에 대하여 work area의 모든 line을 output하고 제2분할에 대한 처리를 하되 표두 표측이 동시에 분할되기 때문에 분할번호와 표측 line을 표현하는 표측번호를 RID의 어느 한곳에 set하지 않으면 아니된다.

write순서는 예시한 바와 같이 표두의 좌 우측 반분을 순차로 output하는 외에 line을 대상으로 좌우반분하여 output하는 방법도 있기 때문에 실제에 있어서는 어느쪽을 택해도 무방하다.

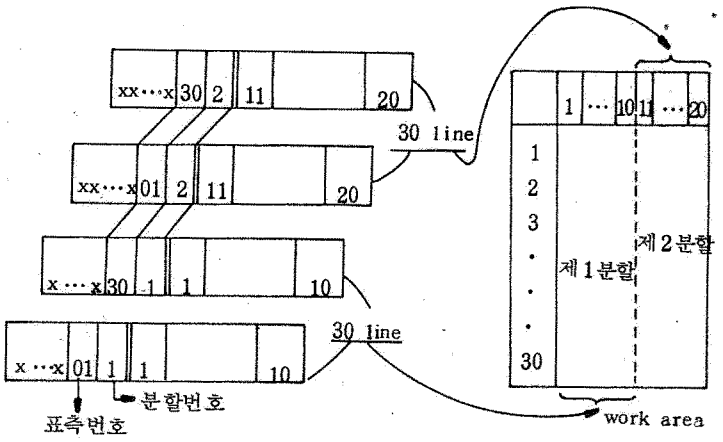


그림 5-18 표두·표측의 분할

4. 가공편성program의 구성

가공편성program의 구성은 대별하면 전표 공통의 처리부분과 각표 공히 독자적인 처리부분으로 구분되며 그 내용은 다음의 routine으로 성립되는 바 그림 5-19는 각 routine의 연관관계를 도표화한 것이다.

- ○ 전표 공통 routine
  - data read routine.
  - data control routine(불필요한 경우도 있음)
  - 처리대상 선택 routine
- ○ 각표 독자적인 routine
  - input data의 work area 또는 output area로 이송 routine
  - 각종 연산 routine
  - work area data의 output area에 이송 routine
  - data의 write routine

가공편성program flow는 data의 처리단위에 따라 크게 나누면 memory에 read단위로 output하는것과, input data를

pool하여 output하는 것으로 나누어진다.

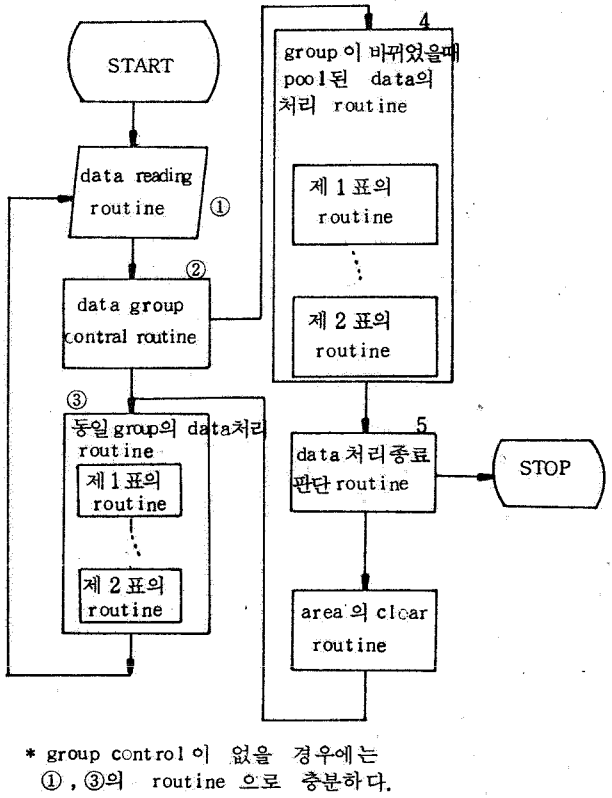


그림 5-19 routine관련도

가. record단위 처리시의 program구성

record단위 별로 output까지의 처리가 가능한 것은 data처리에 필요한 정보를 모두 포함하고 있음을 의미하며 예컨데 summary record가 표단위 또는 표측단위로 수록될 때 zero data의 복원이 없고 표측계는 합산program에서 구하고 표두계만 가공편성 program에서 구하는 경우가 이에 해당하며 이때의 flow chart는 그림 5-20과 같이 program 기술상으로는 단순화할 수 있다.

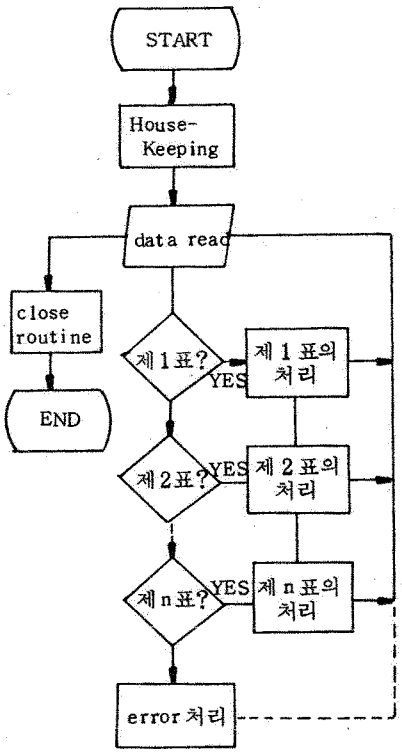


그림 5-20 간단한 flow 의 예

각표별로 data의 처리방법이 다르다는 점에서 memory에 record된 data에 대하여 그 data가 제 몇표의 data인가를 판정하여 각 표를 작성해야 하고 program에서 지정한 표번호와 일치하지 않는 data에 대해서는 error 표시를 한 다음 그 data를 read 하지 아니하고 skip 하거나 처리를 중단해야 하는 바 program test 면에서는 skip 하는 편이 유리하다.

나. pool한 후 처리하는 경우

summary record가 표측단위로 수록된 상태에서 zero data를 복원하고저 할 때에는 특정한 control group내의 input data를 work area에 축적한 후에 처리하는 것이 가장 일반적인 방법

인바 이경우의 program flow는 그림 5-21와 같이 되며 이때의 data flow 경로는 먼저 data를 read하고 직전data의 control과 동일한가를 check한 결과에 따라서 read된 data가 input area의 data를 work area에 축적(그림 5-21의②)할 것인가 또는 work area에 축적된 data에 필요한 처리를 하고 수록(그림 5-21의③)할 것인가를 지시한 다음 read된 input area의 data를 work area에 이송축적하고 다음data를 read하는 순서로 처리해서 이제까지 축적된 data의 control과 상이한 code의 data가 read되었을 때 work area에 축적된 data에 대한 처리를 개시하고 output하여 새로운 control의 첫번째 data를 축적하기 위하여 input area에 있던 data를 work area에 축적하는 작업(그림 5-21의②)을 실행한다.

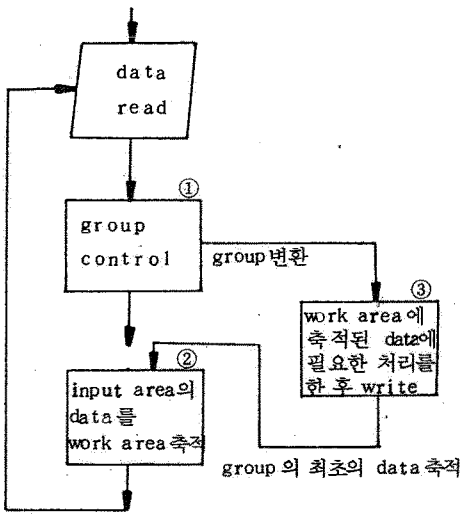


그림 5-21 pool할 경우의 flow

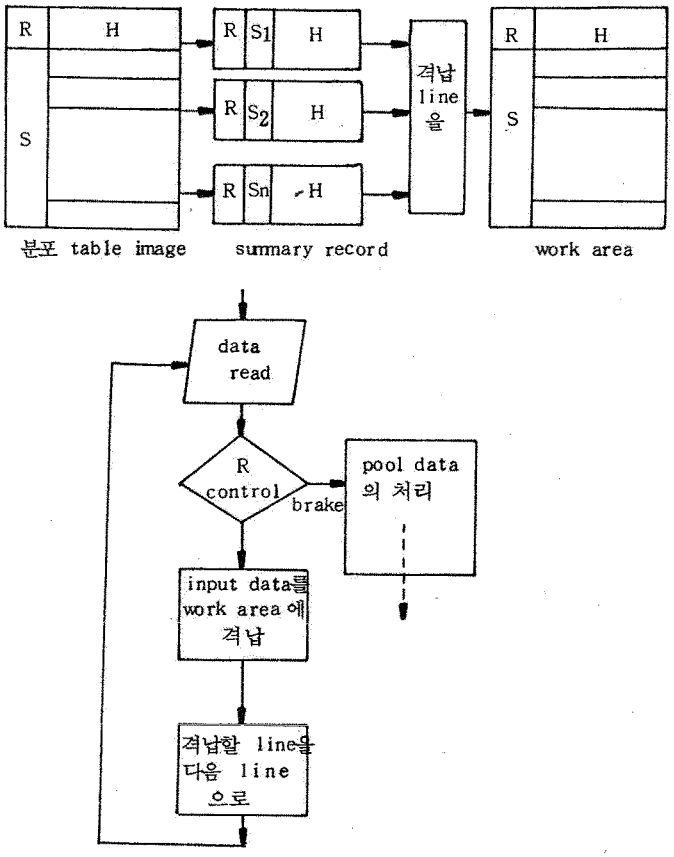


그림 5-22 순차 pool법

(1) pool방법

input area의 data를 work area에 pool하는 방법에는 data를 read하는 순서대로 work area에 축적하는 순차 pool법과 data의 표측항목을 check하여 work area의 어느 line에 격납할 것인가를 결정하여 축적하는 속성 pool법의 2개 방법이 있다.

순차 pool법은 input data에 zero data삭제 없이 data를 격납하는 work area의 image대로 data를 read하는 경우로서 flow는 그림 5-22와 같이 read된 data를 일일히 work

area에 line을 바꾸면서 격납하게 되는데 pool할 때에 input data의 RID $S_1 \sim S_n$과 격납한 work area와의 관계를 check한 상태가 아니므로 zero data에 의하여 code가 삭제되면 축소된 상태로 격납되어 data가 부족한 상태에서 control이 끊긴채 처리될 염려가 있고 또 control R의 sort에 error가 발생하면 work area를 초과하여 program이 파괴될 위험이 따르기 때문에 data의 과부족을 check하지 않으면 아니된다.

속성pool법은 input data에 zero data 삭제가 있거나 data를 격납할 area의 image와 data의 sequence가 상이할 때 사용되며 flow는 그림 5-23과 같은바 속성pool법에서는 input data의 표측 $S_1 \sim S_n$이 work area의 어느 line에 대응하는가를 search routine을 구사하여 check한 다음 격납하게 되므로 input data는 S에 관하여 sort되어 있을 필요는 없다.

분포program에서 표측단위를 write할 때에 RID의 표측부분 즉 $S_1 \sim S_n$을 line번호로 하여 set하면 가공편성program에서 pool될때에 search routine 없이도 처리할 수 있기 때문에 처리시간을 단축하고 program을 간소화할 수 있다.

또 program error를 조기발견하려면 pool의 중복을 check해야 하는데 그림 5-23과 같이 work area의 line과 대응하여 판독문자를 설정하여 pool이면 1, 격납되지 않았으면 0으로 약속하고 1의 line pool되는 data가 있을때 error로 처리하게 되는데 이때의 error 원인은 주로 control 항목 R의 설정이나 sort의 잘못이 대부분이다.

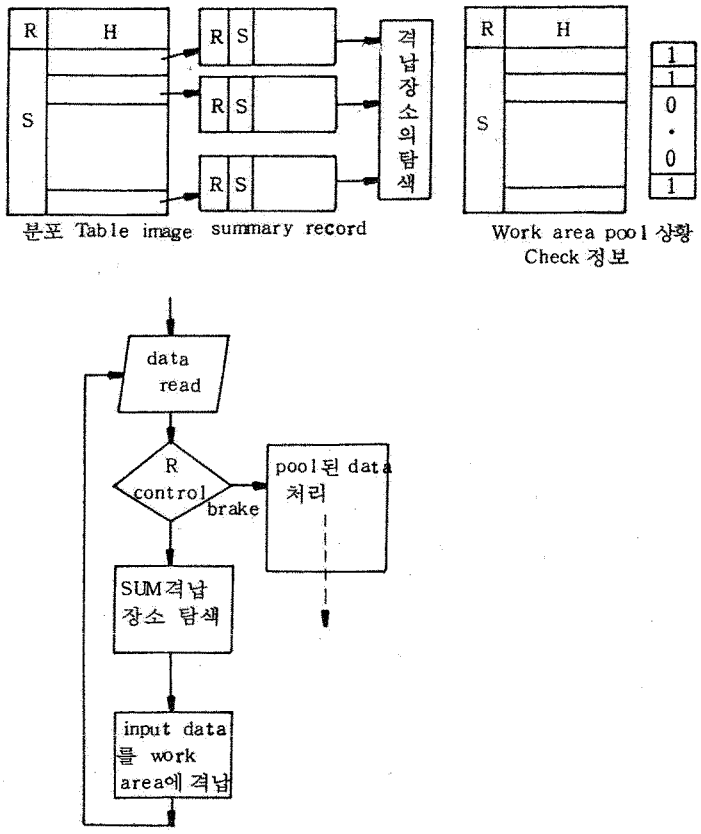


그림 5-23 속성 pool 법

(2) pool 방식의 program 구성

그림 5-24는 pool방식 program flowchart로서 가공편성 program은 표별로 처리routine이 작성되어야 하므로 input data의 표번호를 근거로 해당표의 처리routine에 branch하게 되는데 pool되는 경우 그림 5-24의 flow는 제1표의 최종control R가 pool된 상태에서의 data처리는 제2표의 최초 data가 pool되기 전에 이루어져야만 함에도 불구하고 예시된 flowchart에서는 이 과정이 생략된 상태이기 때문에 엄격하게 올바른 flowchart라 할 수 없으며 완전한 flowchart에 대해서는 추후에 언급하기로 한다.

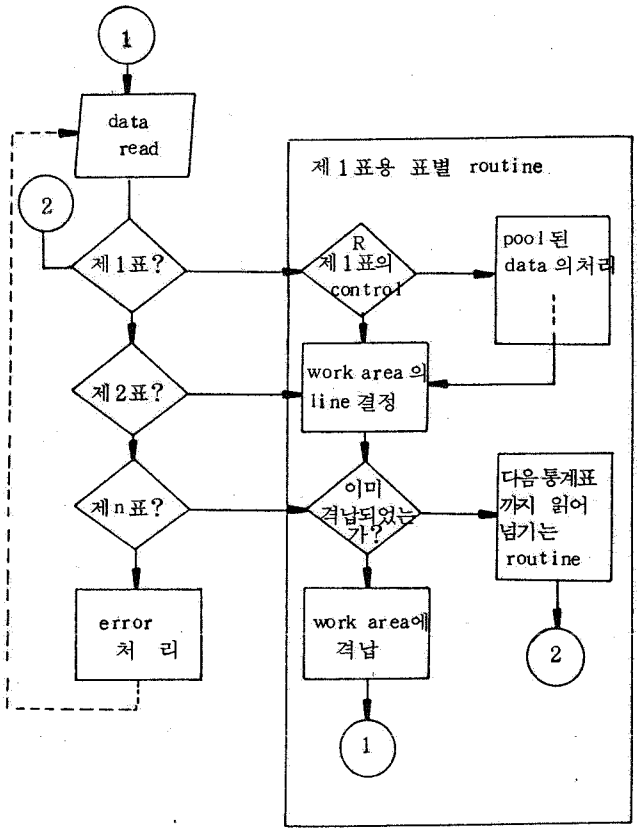


그림 5-24 pool 방식의 program flow

5. 가공편성 program과 subroutine

가. subroutine에 대한 고찰

가공편성 program에서는 주로 결과표 전체를 처리하나 부분처리일지라도 단일 가공편성 program의 처리가능한 표수는 분포program에 비하여 비교할 수 없을만큼 많은바, 가공편성에서 처리가능한 표의 table image를 동시에 확보할 필요 없이 표별로 over lap이 가능하므로 memory size에 따른 계약이 가공편성 program에 크게 작용하지 않는다.

가공편성 program은 방대한 양의 결과표 처리에서 통일된 처리

방법이 없기 때문에 표별처리 routine을 programming하여야 하며 방대한 량의 coding test 자체도 결코 용이한 것이 아니다.

따라서 세부 사항에 이르기까지 반복 coding하는 수고와 번잡을 피하기 위하여 처리의 표준화와 공통화가 절실한바, 이를 위하여 data를 정해진 방법으로 이송하고, 표두 표측등의 합계산출은 처리내용과 처리방법이 같은 반면 이송data의 cell수나 합계량이 다르기 때문에 routine을 subroutine화 하여야만 소기의 목적을 달성할 수 있다.

따라서 그림 5-25와 같이 표에 따라서 다른 값을 subroutine의 parmeter로 하여 해당 subroutine을 호출하면 각표의 처리 routine이 간소화되어 program의 논리를 벗어나지 않을뿐만 아니라 test도 autolysed된 subroutine하에서 parameter만 test하면 충분하다.

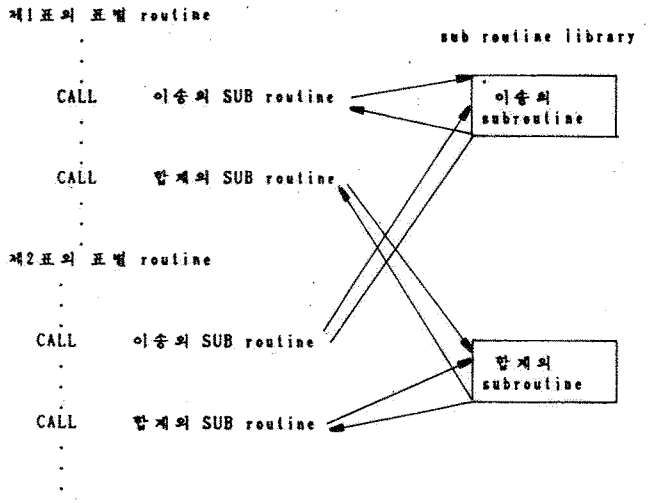


그림 5-25 subroutine의 link

나. subroutine의 예

가공편성 program의 기본기능인 data의 이송과 합계의 산출에 대한 subroutine화 방법에서 program작성 logic에 대해

서만 설명하는데 그쳤으므로 본항에서는 subroutine화 logic에 대하여 예시를 통하여 구체적으로 설명하고자 한다.

(1) data의 이송

data의 이송이라함은 분포의 table image data를 cell 단위로 연속적으로 work area에 이송되는 것을 말하며 합계란이 생략된 table image로부터 합계란이 있는 table image로 cell 단위로 연속 이송하는 경우와 동일 항목내에 대, 중, 소계 등이 변칙적으로 산재하는 상황에서 data를 이송하는 경우가 있는바, 먼저 합계란이 배제된 table image로부터 합계란을 갖는 table image로 cell 단위의 연속 이동의 예를 들면 그림 5-26과 같이 합계란의 위치가 표측항목의 제1line, 표두항목의 제1열에 위치하고 여타의 부분에는 합계란이 생략되는 형식으로 통계표 작성에 많이 활용되는 이송방식 가운데 하나이다.

input area

	미혼	유배우	사·이별
남	①	②	③
여	④	⑤	⑥

input area①-⑥을 work area의 ①-⑥으로 이송한다

work area

	합계	미혼	유배우	사·이별
합계				
남		①	②	③
여		④	⑤	⑥

그림 5-26 이송의 예

이 경우의 subroutine이론은 그림 5-27과 같으나 1차원으로 정의된 것은 assembly 언어와 같이 area의 정의가 1차원인 경우이며, 2차원으로 정의된 것은 assembly 언어와 같이 area의 정의가 1차원으로 되어 있는 경우이고, 2차원의 것은 COBOL, FORTRAN 등과 같은 compiler의 경우를 들수 있다.

subroutine의 parameter로는 h : 분포표 image의 표두 cell 수, s : 분포표 image의 표측 line수, D : 송출측의 area명, D′: 수입측의 area명으로서 D·D′가 program에서 미리 결정되어 있

는 경우에는 치정이 불필요하다.

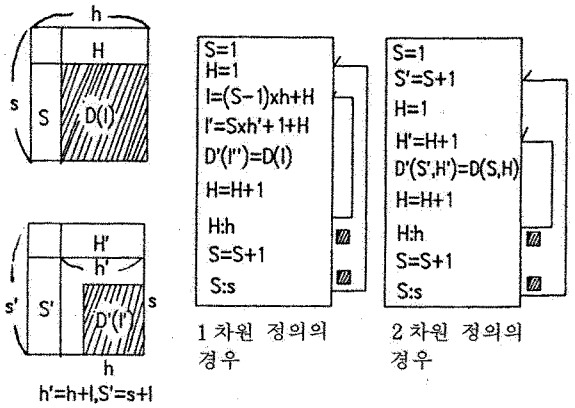


그림 5-27 이송 SUBROUTINE

다음으로 동일항목내에 대, 중, 소계등이 변칙적으로 존재할 때의 data이송의 경우에는 data의 연속이송이 불가능하고 합계란을 배제한 table image에서 합계란을 갖는 table image로 1 cell씩 연속이송하는 경우처럼 단순roof에 의한 이송이 불가능하기 때문에 data 수취 area에 대하여 cell단위로 input data를 입력할 부분이 합계부분인가를 판정하는 control정보를 설정하여 이송을 제어한다.

예컨대 control 정보를 data 입력가능 부분 : 1, 입력불가능 부분 : 0로 약속했을 때 그림 5-28의 control정보는 0 1 1 0 1 1이 되며 이때의 control 정보 1 column은 수취쪽의 단위 cell에 대하여 부여하게 된다.

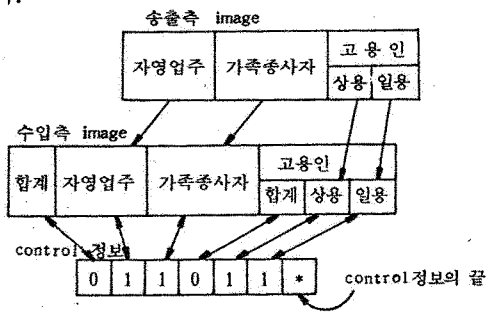


그림 5-28 합계가 있는 이송의 예

처리 routine 에서는 이 control 정보를 참조하여 1이 나타날 때에 해당 cell에 data를 전송하고 0이 나타날 때에는 수취측의 cell을 하나씩 skip 하여야 하는데 이때의 일반적인 logic 은 그림 5-29와 같다.

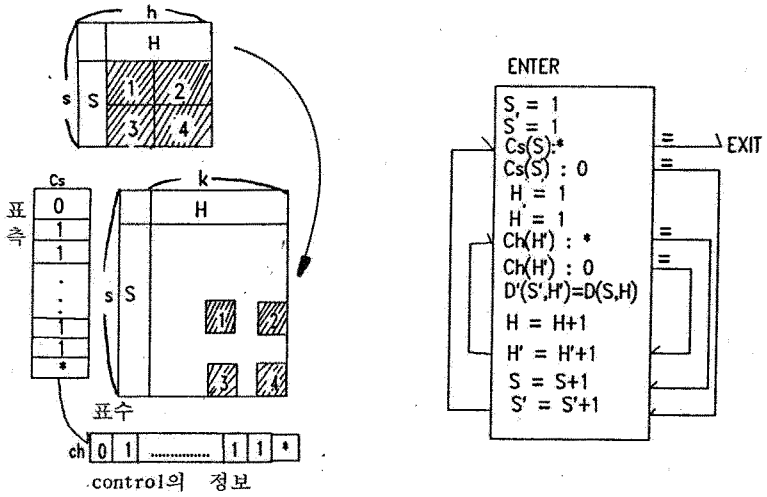


그림 5-29 일반적인 합계가 있는 이송

(2) 합계 산출

합계 산출 routine을 subroutine화 하는 방법은 합계 부분을 중심으로 지시하는 방법과 각 cell 단위로 지시하는 방법이 있다. 먼저 합계 부분을 중심으로 지시하는 방법을 살펴보면 이 처리방법은 어느 부위의 합계 부분에 어떤 cell과 어떤 cell을 가산할 것인가를 지시하는 방법으로 그림 5-30 의 예를 통하여 합계를 control하는 경우를 들면 총수 부분 : A, 총수에 더할 cell : 1, 아무런 처리 없는 cell : 0, control 정보의 종료 : *로 약속한다면 control 정보는 그림 5-30과 같이 되지만 이러한 구분만으로는 그림 5-31의 합계 부분이 대, 중, 소계 등 여러 level로 구성된 경우에는 처리가 불가능하기 때문에 control 정보에 대하여 각 level을 표시하고 각 level의 총수에 어느 부분을 가산할 것인가를 명확히 해야한다.

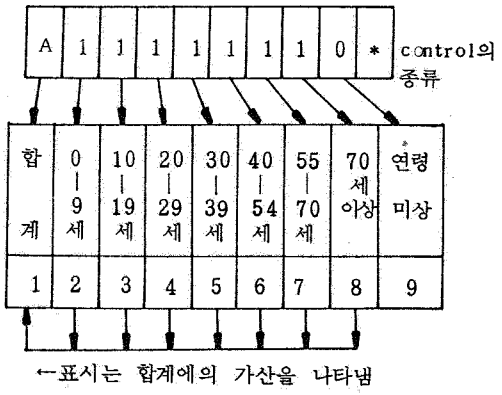


그림 5-30 단순한 합계

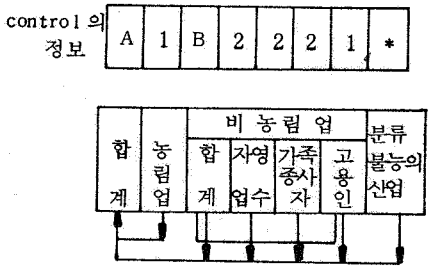


그림 5-31 중간계를 갖는 경우

control 정보의 표시방법은 내역부분 : 2 column, 총수부분 : A～I (영문자), 총수에 가산하지 않는 내역 : 0, control 정보의 종료 : * 로 변환하여 내역부분의 level과 총수부분의 level을 각각 1→A, 2→B, ··· 9→I로 대응시켜서 개체한 control 정보를 사용한 합계 산출은 합계에의 가산에서 내역 level이 총수 level과 같거나 그 이하일 때 가능하며, 총수부분에의 가산 작업은 동일한 level이거나, 그 이상의 level의 총수가 되었을 때 종료하게 된다.

A 1 1 A 1 1이라는 control 정보가 있다 할때 최초 A에의 가산은 다음의 A를 검출하므로서 종료하고 제 5 cell은 제 4 cell에 가산하는 지정 방법으로 그림 5-31의 예에서 control정보는 A 1 B 2 2 2 1 *이 되며 그림 5-32는 level의 결정방식 logic를 flowchart화한 것이다.

다음 각 cell단위로 지시하는 방법은 합계를 구하고저 하는 table image의 cell에 번호를 부여하고 어떤 cell을 합계에 가산할 것인가를 cell번호에 의하여 지시하는 방법으로 그림 5-33의 경우는 ①, ⑤의 cell을 합계란으로 하고 각각 ①=②+③, ⑤=③+④+⑥이라하면 이것을 다음과 같이 표현을 바꾸어 control 정보화 하여야 한다.

제 1 cell : 합계란 → /

제 2 cell : 제 1 cell로 이송 → 1/

제 3 cell : 제 1, 5 cell로 이송 → 1,5/

제 4 cell : 제 5 cell로 이송 → 5/

제 5 cell : 합계란 → /

제 6 cell : 제 5 cell로 이송 → 5/

data종료 : → *

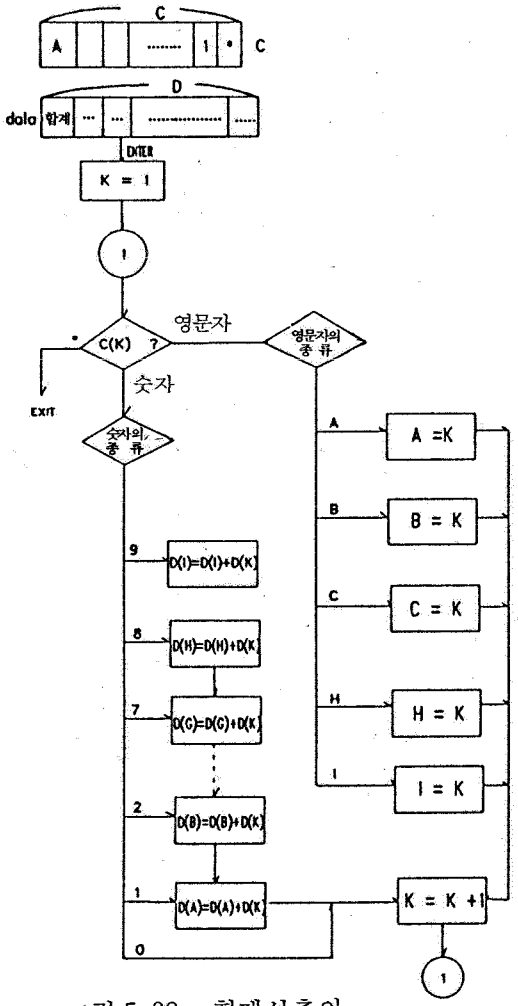
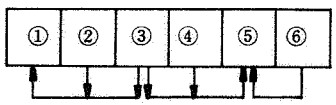


그림 5-32 합계산출의 flow chart



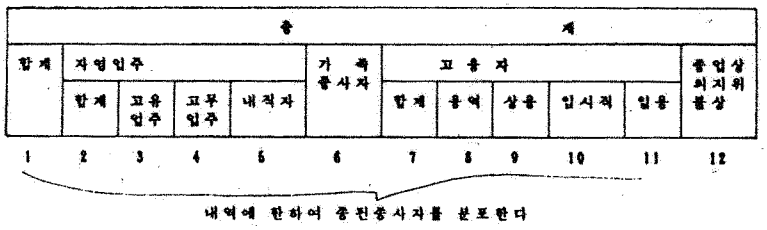
←: 총수에의가산을 의미한다.
①-⑥은 cell 번호임.

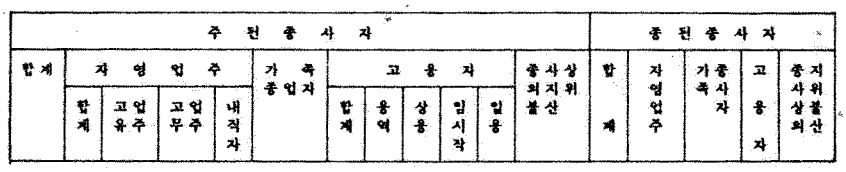
그림 5-33 임의위치에 있는 합계

위의 control 정보기호 가운데 숫자는 cell번호, 기호 /는 cell의 구획을 표시하고 합계 cell은 가산 대상이 아니므로 cell 번호를 기재하지 아니하고 다만 /로 표시하였으나, 만약 합계란을 또 다른 합계란에 가산할 경우에는 내역과 마찬가지로 가산할 cell의 번호를 기재하고 control 정보가 끝나면 예에 따라서 *로 종료표시를 한다.

이 방법은 합계 부분을 중심으로 지시하는 방법에 비해서 합계 부분의 위치에 구애받지 않는 장점이 있으며, 가산 명령을 이송 명령으로 치환하면 이송의 subroutine이 되어 융통성이 있으나 control 정보가 복잡한 단점을 가지고 있다.

각 cell 단위로 지시하는 방법에 대하여 특징적인 예를 들면 그림 5-34의 표두 부분에서 「총수」가 내역인 「일에 종사하는자」보다 상세한데 이와 같은 때에는 총수를 내역인 일에 종사하는자로 가정하여 분포하고 25～29의 초기값을 zero로 한 다음 각 cell 단위로 지시하는 방법에 의하여 합계를 구하면서 정당한 위치에 이동하여야 한다.





CELL번호	가산대상	CELL번호	가산대상	CELL번호	가산대상
1	/	11	1,7,25,28/	21	1,7,9,13,19/
2	/	12	1,25,29/	22	1,7,10,13,19/
3	1,2,25,26/	13	/	23	1,7,11,13,19/
4	1,2,25,26/	14	/.	24	1,2,13/
5	1,2,25,26/	15	1,2,3,13,14/	25	/ (*)
6	1,25,27/	16	1,2,4,13,14/	26	/ 또는25CELL 에에
7	/	17	1,2,5,13,14/	27	/ *표을설치해도
8	1,7,25,28/	18	1,6,13/	28	/ 부당하다
9	1,7,25,28/	19	/	29	/
10	1,7,25,28/	20	1,7,8,13,19/		*

그림 5-34 복잡한 합계가 있는 예

이 방법의 program logic은 그림 5-35와 같이 간단하나, 표현상의 제약으로 control 정보의 숫자는 2 column 이상 일지라도 1 column과 같이 취급하고 있기 때문에 실제로는 program을 작성하는데 있어서 특별한 주의를 기울이지 않으면 아니된다.

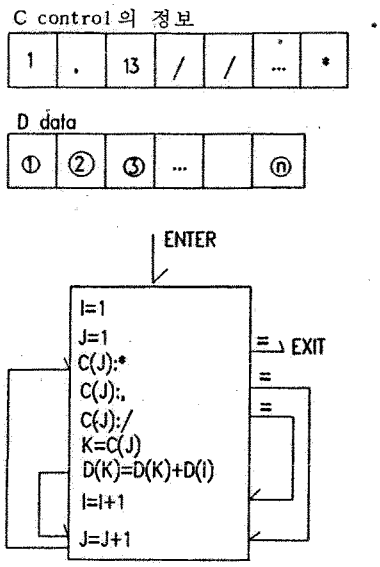


그림 3-35 각 cell 단위로 지시하는 방법의 logic

6. 가공 편성 program의 flowchart

위에서 가공 편성 program에 대하여 설명한바 있으나 가공 편성 program의 기능이 다양하고 복잡하기 때문에 정형화된 처리 patern의 정립이 불가능하고 단편적인 처리가 불가피한 바 이와 같은 정형적인 처리 patern이 없는 program일수록 flowchart의 표준화가 절실하나 개략적인 구성을 이해하기 위하여 그림 5-36의 flowchart를 예로 하여 표별 routine을 over-lay 구조로 하여 표번호가 바뀔 때마다 그 module을 load하는 구성에 대하여 설명을 가하고자 한다.

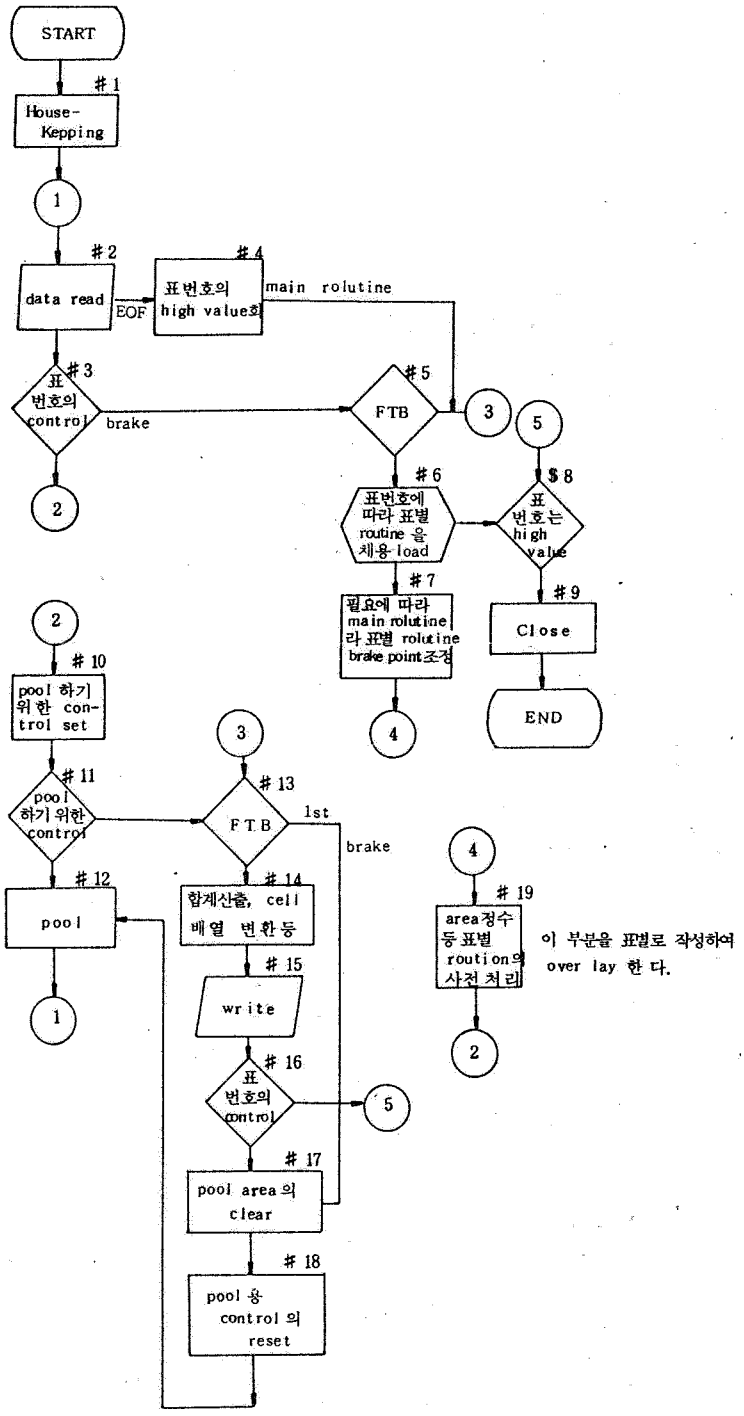


그림 5-36 flow chart의 예

가. main routine 부분

1 : i/o file open 등 사전 준비 처리.

2 :전체 data를 일시에 read 하기 위한 reading area를 input summary record의 최대 length만큼 확보.

3 :표별 module 선정 control.

4 :모든 data를 read한후 최후로 pool된 data의 처리를 위하여 표번호에 high value를 set한다. 다만 이는 control을 끊기 위한 것이므로 어떠한 문자를 넣어도 무방하다.

5 : #6에 의하여 load된 표별 routine과 main routine 간의 branch switch 조정(program 작성 방법에 따라 표별 routine load시 branch switch의 조정이 가능하므로 필요에 따라 이부분을 삽입한다.)

8 :표의 control이 끊겨서 pool된 data를 처리한후 처리하여야할 다음표가 있는가의 여부 check.

9 : program을 종료하기 위한 처리.

나. 표별 routine 부분

10 :표별로 data를 pool하기 위하여 control 항목의 set. (control 항목은 표에 따라 다르다)

11 : pool하기 위한 control이 끊겼는가의 여부를 check.

12 :data의 pool.

13 :최초의 1회는 branch하는 switch임.

14 ~# 15 : pool된 data의 처리(표에 따라 필요한 기능이 다르다).

16 :같은 표의 처리가 계속되는가의 여부 check(표간 간격은 다음표를 load 하기 위하여 point 5로 이송한다).

17 : pool area의 clear.

18 : pool용의 control 항목의 reset.

19 : pool area의 크기는 표에 따라 다르므로 필요하다면 여기에서 정의해도 무방하며 module에 포함시켜도 무방하다.

Ⅵ. 편집 PROGRAM의 설계

1. 편집program의 목적

편집 program은 computer집계과정의 최종단계 program으로 통계표 작성에 필요한 연산과정을 통하여 이용자가 이해하기 쉬운 형태의 결과표 생산을 기본목적으로 한다.

연산과정에서의 summary data는 계산에 알맞는 형식으로 되어 있을뿐 결과표 인쇄양식이나 보기에 용이한 점 등은 일체 무시된 상태이기 때문에 외부적 체제를 정비하기 위하여 각 cell의 유효 수자 앞에 있는 불필요한 「zero」를 제거하고 comma, 소수점, Cr, $ (\) 등의 기호를 필요에 따라 인쇄하고 행간을 control하며 색인을 부여하여 그림 6-1 상단의 summary data를 하단의 편집된 결과표로 정리 하므로서 최종 통계표를 읽기 용이하도록 하는 기능을 가지고 있다.

표	번	지 역	합 계	남	여
전 국	01	TT	000052800	000040624	000012176
경 기	01	01	000020329	000016857	00003472
충 남	01	02	000010051	000008044	000002007
경 북	01	03	000012306	000008523	000003783
전 남	01	04	000010114	000007200	000002914

summary data

PERSONS BY SEX

	TOTAL	MALE	FEMALE
TOTAL	52,800	40,624	12,716
KYONGGI	20,329	16,857	3,472
CHUNGNAM	10,051	8,044	2,007
KYONGBUK	12,306	8,523	3,783
CHUNNAM	10,114	7,200	2,914

그림 6-1 편집완료된 DATA의 인쇄

2. 편집 program의 기능

편집 program의 기능은 처리대상 summary record의 형식과 결과표 인쇄 양식에 따라서 표두계·표측계 등의 표내계, 평균치, 구성비의 계산, 분포 항목의 삭제, 분포 항목의 배열 변환, data의 절단, zero data의 복원, print개행 문자의 부가, 편집 색인의 부여 등 9개 기능으로 분류되며 편집기능과 색인부여기능 이외의 제 기능은 가공편성 program의 기능과 일치함에도 불구하고 편집 program을 별도 설정하는 이유는 가공편성 program이 잡다한 기능을 수행하여야 하고 program 작성과정에서 coding량의 증가에 따르는 번잡과 위험의 분산을 위하여 편집 program과 가공편성 program을 통합하여 하나의 program으로 만들 때 장황해지는 부담을 덜고자 하는데 있다.

따라서 편집 program과 가공편성 program의 기능이 같은 것은 어느것이 편리한 가를 가름하여 결정하여야 한다.

가. 표내계의 산출과 평균·구성비의 계산

표내계의 산출과 평균·구성비의 계산은 가공편성 program에서 처리 하는것이 상례이나 input summary record가 표 단위 수록 형식이고 합계 란이 설정되어 있는 경우, 표두·표측의 합계 산출은 편집 program에서 처리하는 것이 유리하나 합계란이 제외된 상태에서는 표두합계 산출 이외의 합계를 생략하고 합계 산출 그 자체는 가공편성 program의 합계산출 routine을 이용하는것이 간편하다.

나. 분포 항목의 삭제와 배열 변환

표내계의 산출과 평균·구성비의 계산에서 처럼 표측 line 단위 삭제와 배열변환의 빈도가 잦은 때에는 가공편성 program의 이송 subroutine을 이용하여 간편히 처리할 수 있다.

line단위 표내계 산출, 평균·구성비 계산, 분포항목 삭제, 분포항목의 배열 변환 등 4대 편집 program기능의 수행은 가공편성 program의 작업량 축소가 주된 목적이기 때문에 work load상 무리가 없는 한 가공편성 program으로 처리 하는것이 정도다.

다. data의 절단

data의 절단은 결과표의 인쇄양식에 따라 표두를 분할하고, 표측을 line 단위로 절단하는 기능으로 가공편성 program에서의 표두분할은 대체로 표측단위에서 표내계 산출, 평균과 구성비 계산, 분포항목의 삭제, 배열 변환등의 제기능과 분할기능을 수행하는바 record수의 증가를 고려하면 편집 program으로 처리 하는것이 유리하다. 결과표를 대량 인쇄할 경우에는 그림 6-2와 같이 편집된 data (편집 program의 output)를 인쇄전용 program을 통하여 인쇄 하므로서 인쇄용지 파손, 이중인쇄 등과 같은 trouble이 발생하였을때 수정과 인쇄완료 후 부분보완이 용이하고 대형 계산기의 병행처리와 OFF-LINE장치의 이용이 가능하다.

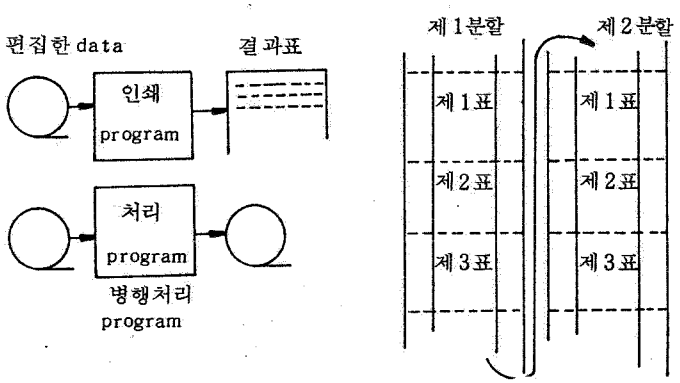


그림 6-2 결과표의 인쇄

결과표 인쇄순서는 제1분할을 대상으로 1～n표의 전체 결과표를 인쇄한 다음 제2분할, 제3분할의 순으로 인쇄하여 각 표 분할간 수치연결을 쉽게 하고 대량 data의 정연한 취급이 가능토록 한다.

data절단의 수록 형식은 그림 6-3과 같은 2개 방법이 있는바 방법 선택은 주로 magnetic tape를 주축으로 하는 주변장치 수에 의하여 결정되나 제1분할 이외의 분할은 분할 번호가 클수록 결과표의 data량이 적어지므로 sort의 input data에서 제1분할을 제외 하는 것이 유리하고 분할 수가 3～4인 경우에는 sort program이 아닌 합산 program의 경우와 같이 발취 program에 의존하는 것이 경제적이기 때문에 분산법을 원용하는 것이 상식이다.

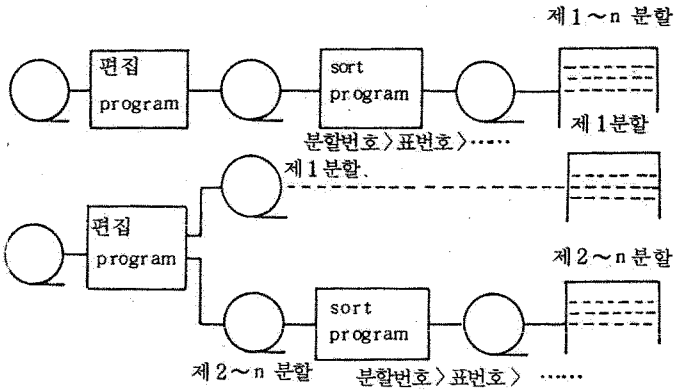


그림 6-3 file의 정리방법

라. zero data복원

zero data복원은 가공편성 program보다는 편집 program에서 처리하는 것이 data증가가 없기 때문에 유리하다.

zero data복원은 가공편성 program에서 work area의 table image를 표별로 정리하고 초기 값을 zero로 한 다음 input를 순차로 pool하는데 대하여, 편집 program에서는 표별로 work area를 설정하고 table image를 정의하는 과정까지는 가공편성 program과 동일하나 초기 값의 설정 방법에 따라 그림 6-4와 같은 2개 방법이 있다.

그림 6-4의 방법 1은 work area에 만들어진 data를 편집하고 표두

분활을 한다는 점에서는 가공편성 program과 동일하나 zero data의 삭제는 R,S,H가 세분되어 있기 때문에 data가 통계표에 분산되는 경우에 가장 유효하고 zero data가 non zero data보다 많을 때 summary record의 처리에 유리하므로 제한적으로 채택하여야 한다.

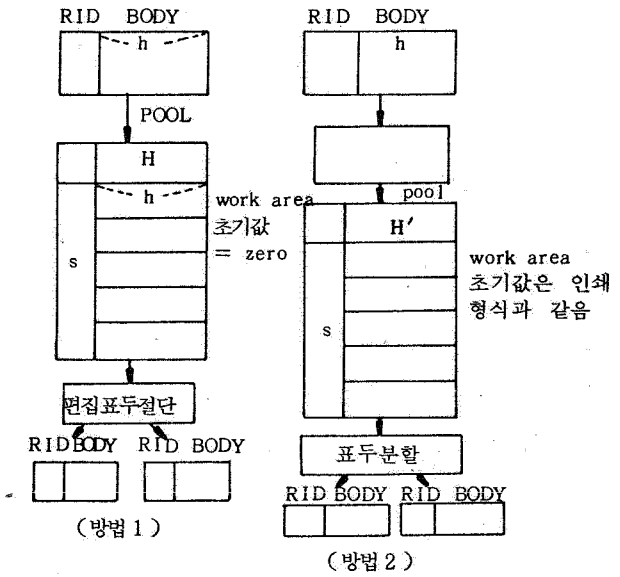


그림 6-4 zero data의 복원

A	B	C	D	E	F	항목
123	1.3	6.5	56	59	0.123	non-zero data의 인쇄
0	0.0	0	-	0	0	zero data의 인쇄
0	0.0	0	-	0	.0	work area의 초기값
0	0.0	0	-	0	.0	
0	0.0	0	-	0	.0	
0	0.0	0	-	0	.0	
			-	0	.0	

그림 6-5 work area의 초기값

이와같이 zero data가 많은 상태에서 방법 1을 선택하면 편집회수는 「line 수×표두 항목 수」가 되어 zero data수와는 관계없이 일정해지므로 non zero data가 하나인 경우에도 매양 동일한 처리시간(pool시간과 다름)을 요하게 되어 비효율적이다.

다음 그림 6-4의 방법 2에 대하여 살펴보면 work area의 초기 값을 zero data의 인쇄문자로 하여 input data를 편집하고 pool 되도록 한 것으로서 work area의 크기는 표측 line수와 input data의 line수가 동일한 반면 표두의 column수가 표두 분할 이전 인쇄양식의 길이가 되며 이 방법에서의 편집 회수는 「non zero data × 표두 항목 수」가 되어 그림 6-5와 같이 work area의 초기 값을 zero data의 인쇄문자로 메꾸기 때문에 non zero data가 적을수록 처리 시간은 단축되어 방법 1보다는 탁월한 방법이라 할 수 있다.

마. print개행(행 간격)문자의 부가

print개행문자의 RID상 위치는 가공편성 program에서 언급한 바와 같이 C.C.C기능이 인쇄 행 간격을 control하고 page 변환을 지시하는 것으로 개행과 page변환에 대해서는 line printer의 carridge tape control용 구멍에 의한 처리도 가능하나 이 방법은 청구서 등 일관된 형식을 처리할 때 line변환과 page변환에는 적합하지만 통계표와 같이 표별로 인쇄형식이 다른 경우에는 부적합하기 때문에 이미 언급한 바 있는 전용 program으로 인쇄할 것을 전제로 하는 data에 line변환이나 page 변환의 정보를 input하는것이 효율적이다.

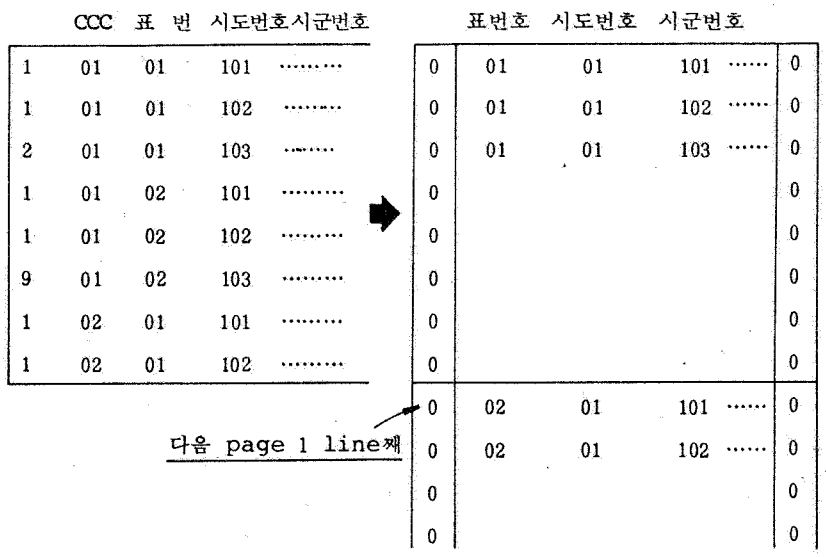


그림 6-6 CCC와 인쇄

C.C.C는 다음과 같은 숫자로 line변환과 page변환에 관한 정보에 대하여 미리 약속하고 필요한 작업을 수행하는 것이 상례이다.

1 : 계속 print.(single space)

2 : 1 line간격의 print(double space)

3 : 1 line print 2 line space (triple space)

4 : 1 line print 3 line space

5 : 1 line print 4 line space

6 : 1 line print 5 line space

7 : 1 line print 6 line space

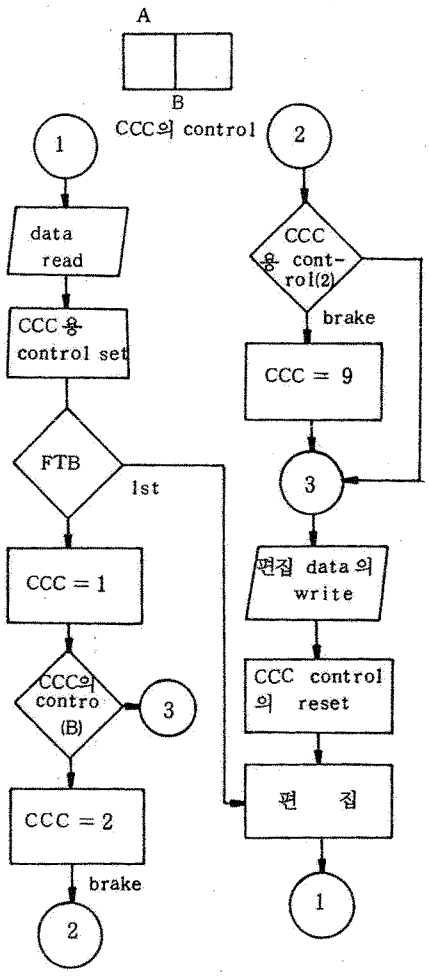
8 : 1 line print 7 line space

9 : 1 line print 다음 page로 변환(page over flow)

그림 6-6과 같이 C.C.C가 지정되어 있다면 단위 page에 6 line의 인쇄가 가능한 바 C.C.C의 set는 input record가 표측단위, 표단위 여부에 따라서 처리 방법을 달리하게 된다.

그림 6-6의 표측단위 input record를 예로하여 flowchart로 표

현하면 그림 6-7 과 같이 되는바 C.C.C설정은 data를 read해야 하기 때문에 data가 pool되는 것을 필요로 하지 않는 경우에도 표측 line 단위로 pool해야 하는 flow가 되는바 그림 6-7의 편집부분이 이에 해당하며 수록하지 않고 pool된 형태이기 때문에 다음 data를 read한 후 C.C.C를 결정하고 앞의 data를 수록하는 구



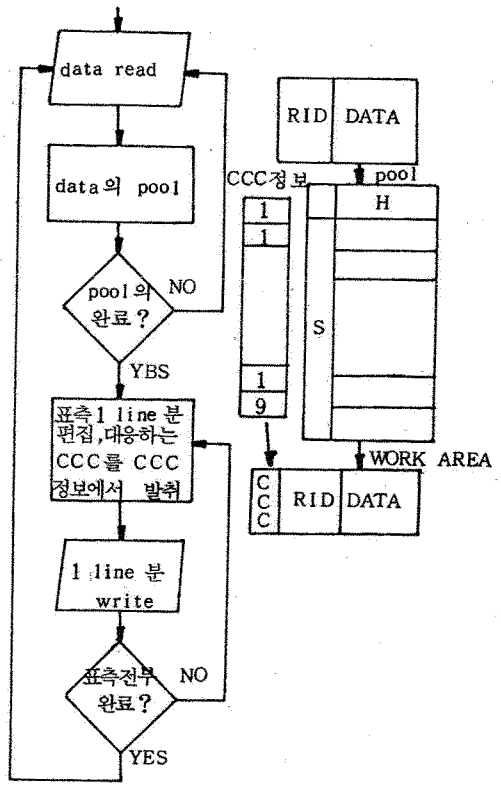


그림 6-7 표측단위의 flow chart 그림 6-8 표단위의 flow chart

조로 되어 있다.

다음 input data가 표단위인 경우는 input data가 표단위·표측단위를 불문하고 pool되는 경우에는 각 line에 대한 C.C.C정보를 memory에 수록하고 각 line을 편집하여 수록할 때에 C.C.C정보를 하나씩 검색하여 output에 set하게 되는데 그림 6-8은 이 때의 flowchart로서 data가 표단위로 pool이 불필요할 때에는 pool관계 명령을 생략한다.

이 방법의 장점은 control항목의 변화에 좌우되지 않기 때문에 C.C.C를 자유롭게 설정할 수 있고 program이 간단하다는 점을 들 수 있다.

바. 편 집

COBOL, PL/1에서는 picture사양에서 편집기능을 수행하나 assembly언어에서는 편집명령이 따로 필요한 바 편집기능은 유효숫자 좌측의 zero삭제, 소수점, comma 삽입, $(/)기호의 설치, 숫자간 임의 문자 설치, minus부호의 부여가 주된 기능이다.

사. 색인부여

이제까지 cell의 인쇄에 치중 하였으나 통계표 인쇄편집에 있어서도 색인의 도움이 절실히 요구 되는바 색인은 그림 6-9에서 보는바와 같이 표제 색인, 표두 색인, 표측 색인이 있다.

표제색인과 표두색인은 인쇄형식에서도 알 수 있는바와 같이 data record와는 별도로 추가되는 것으로서 summary record의 RID에는 표제색인과 표측색인의 정보는 명시적으로 수록되고 표두색인 정보는 표두항목의 표시에 관하여 summary data부분의 각 cell 위치에 따라 부여되나 인쇄된 결과표에서 표두항목의 색인이 없으면 결과표 양식과 인쇄결과를 조합(대조)해 보지 않으면 아니되기 때문에 결과표 종류가 다양하고 인쇄 분량(page)이 방대할 때에는 인력과 시간의 절감을 위하여 색인 부여는 필수적이다.

TABLE 1. POPULATION 15 YS OVER BY EMPLOYMENT STATUS

}표제색인

	TOTAL	주 거	가 족	고 용
TOTAL	4591	911	820	2856
15 - 19	353	3	53	297
20 - 24	675	21	84	569
25 - 29	582	51	104	425
30 - 39	1153	212	214	726
40 - 54	1169	331	223	614
55 - 64	456	191	90	175
65 -	204	102	53	48

}표두색인

표측색인

그림 6-9 색인

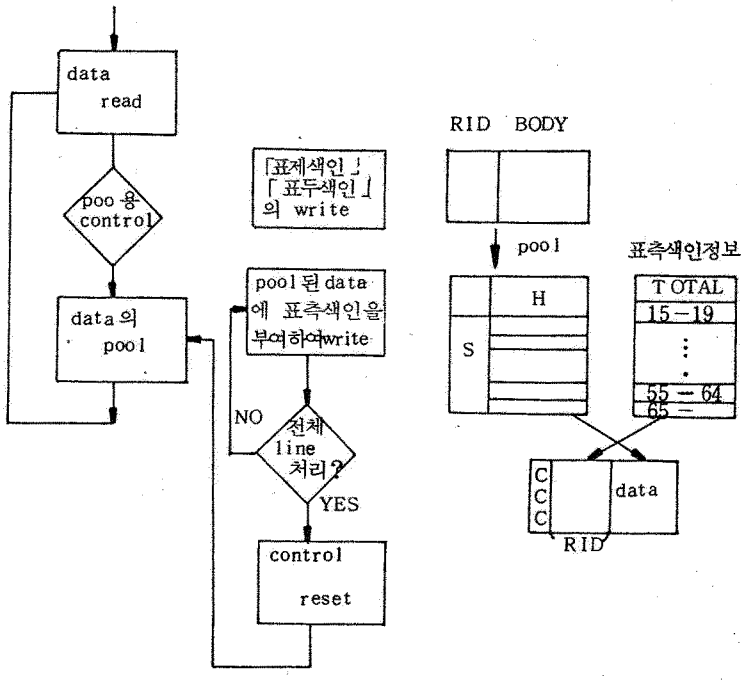


그림 6-10 색인의 flow chart

page단위로 pool되는 경우의 flowchart는 그림6-10과 같은바 이때의 pool은 page단위이므로 pool용 control이 끊기면 data의 수록에 앞서서 표제와 표두의 색인을 수록하여야 한다.

표제색인은 표명이나 표번호 외에도 pool인 때의 control이 일부 포함되는 수가 있으므로 control이 끊길 때마다 색인을 위한 변환을 하여야 하며 표두색인은 표가 바뀔 때까지 일정하게 유지된다.

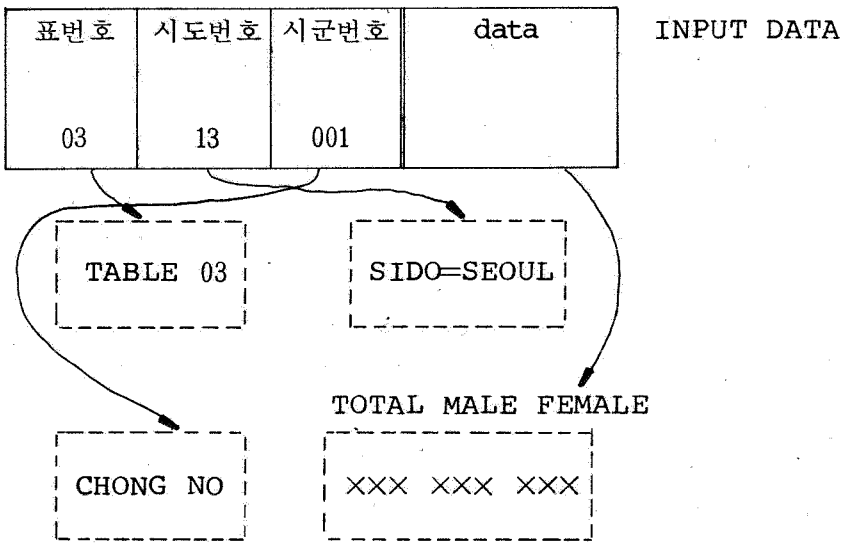


그림 6-11 표제색인의 예

표측색인은 pool된 data의 각 line에 대응하여 설정하게 되는데 각 line을 수록할 때에 하나씩 검색하여 set하는 방법으로 sub-routine을 사용하는 편보다 처리가 빠르다.

3. 편집program의 flowchart

편집 program의 flowchart는 가공편성 program의 flowchart에 편집, 색인과 C.C.C set가 추가될 뿐이므로 편집 program의 골격을 이해하기 위해서는 가공편성 program의 flowchart를 참고하여

야 하며 더부러 편집 program의 C.C.C control은 전체 통계표를 통하여 동일하고, space는 single space, 표두분할을 하며, 표측 단위 record에서 pool되지 않는 조건하에서의 flowchart는 그림 6-12와 같이 되는바 각 event별로 설명하면 다음과 같다.

#1 : i/o file의 open

#2 : 표측 단위 data의 read

#3 : 전표 공통의 C.C.C set에 필요한 control항목을 input area로부터 new key로 set

#4 : 최초의 data는 처리하지 아니하고 work area로 이송하기 위한 switch

#5 : input area로부터 work area로 이송(단위 line의 pool과 동일)

#6 : input area로부터 set된 control을 old key로 이송

#7 : 전 data를 reading 후 표번호를 high value화.

#8 - #14 : C.C.C의 set routine(비교는 전체의 key에서 시작하여 순차로 단축한 후 최종에는 표번호에서 종료).

#15 - #17 : work area의 표번호를 check하는 표별 routine으로의 branch

#18 : 해당표가 아니므로 errer처리 후 skip.

#19 : work area에서 필요한 배열변환과 표두계 산출 수행

#20 - #21 : 제1분할의 처리

#22 - #23 : 최종 분할의 처리

#24 : program종료 판단

#25 : i/o file의 close와 처리된 record의 count를 print

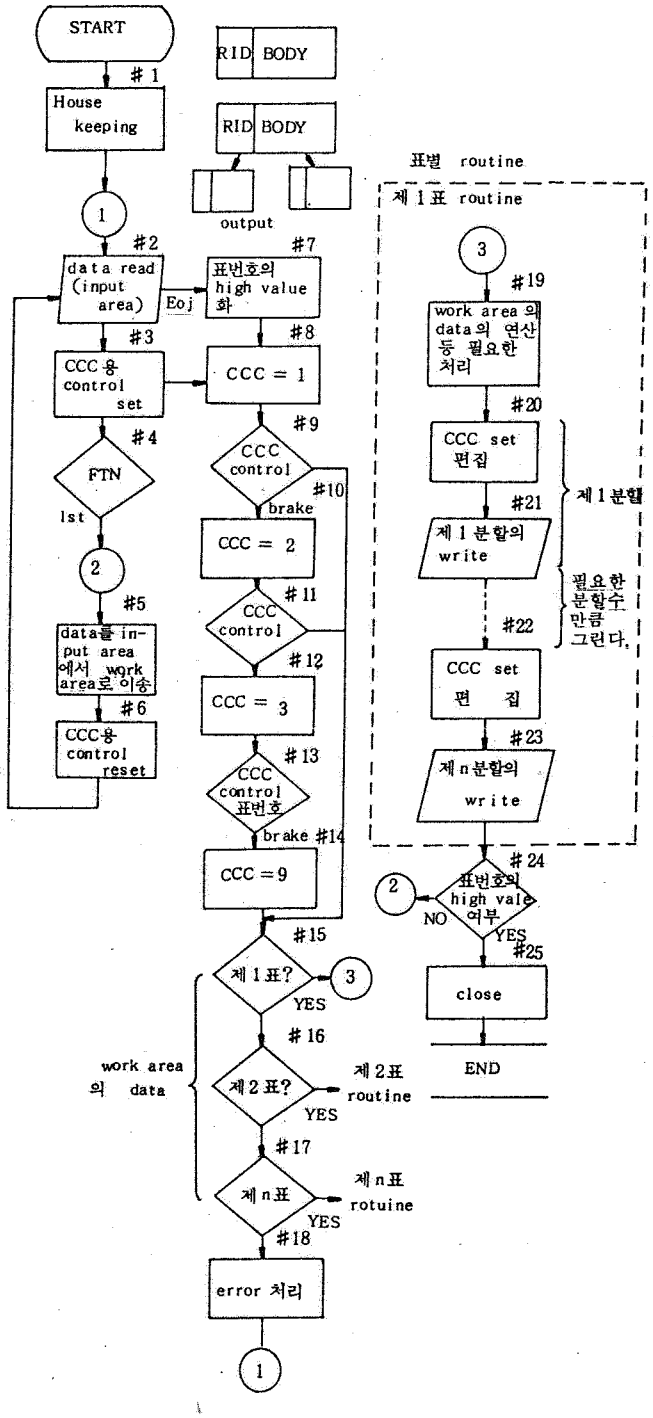


그림 6-12 flowchart의 예

Ⅶ. 집계 SYSTEM의 설계예

1. 집계대상 통계표

기능분리형 집계system의 장점은 대량의 data를 대상으로 다양한 통계표를 작성하는데 소요되는 처리시간의 단축에 있기 때문에 1985년도 인구주택총조사의 집계logic과 처리flow의 개요를 예로 하여 집계system의 설계에 대하여 살펴 보기로 한다.

1985년도 인구주택총조사의 31개 조사항목은 다음과 같다.

인구에 관한 사항

(1) 이 름(본관)
(2) 가구주와의 관계
(3) 성 별
(4) 연 령(관습상의 연령)
(5) 생 년(년도 띠)
(6) 생 월 일
(7) 연 령(만연령)
(8) 종 교
(9) 출 생 지
(10) 일년전 거주지(1세이상)
(11) 5년전 거주지(5세이상)
(12) 교육정도(5세이상)
(13) 경제활동상태(14세이상)
(14) 산 업(14세이상)
(15) 직 업(14세이상)
(16) 혼인상태(15세이상)
(17) 생존자녀수(15세이상 기혼여자)
(18) 사망자녀수(15세이상 기혼여자)

⑲ 총출생자녀수 (15세이상 기혼여자)

주거에 관한 사항

(모든가구)

(1) 주거상태

(2) 주거관계

(3) 주택소유관계

(4) 사용방수

(5) 취사연료

(6) 문화시설

(주택의 주가구)

(7) 가구수 및 총방수

(8) 연령별

(9) 난방시설

(10) 편이시설

(단독주택의 주가구 및 동거가구)

(11) 주가구의 다세대 여부

(12) 동거가구의 전용내역

인구주택총조사 보고서 중 15% 추출집계 결과를 예로 하여 연령 (각세), 생월 (4구분), 남여별 인구 (그림 7-1), 15세 이상 인구의 연령 각세, 성별 배우 관계 (그림 7-2), 산업 (소분류), 종사자 지위 (6구분), 남여별 15세 이상 취업자 수 (그림 7-3), 사회·경제분류등 47개 표에 대한 집계의 설계를 위한 logic과 처리 flow에 대하여 검토하고저 하는바 제1표는 전체 data를 대상으로 하여 표측은 연령 각세에서 5세마다 합계를 구하고 표두는 2개 부분으로 나누어 남녀별 인구수와 출생 월에 의한 인구수를 집계하여야 하기 때문에 동일한 data가 2개 부분에 분포되고 이는 각 부분의 합계치와 일치한다.

제 1 표 연령(각세), 생월(4구분), 남여별 인구

- 전국, 시도

연령(각세)	합계	남	여	남 1-3월 (1)	남 4-6월 (2)	남 7-9월 (3)	남 10-12월 (4)	여 1-3월 (1)	여 4-6월 (2)	여 7-9월 (3)	여 10-12월 (4)
합계											
0-4세											
0											
1											
2											
3											
4											
5-9											
5											
6											
7											
8											
9											
10-14											
.											
.											
.											
95-99											
95											
96											
97											
98											
99											
100세 이상											

- 대상 : 전 인구
- cell 수
  표측 표두
  122 × 11 = 1,342

그림 7-1 통계표 1 (연령, 생월, 남여별 인구)

연령(각세)	15세이상인구(1) 합계	15세이상인구(1) 남	15세이상인구(1) 여	남 미혼	남 유배우	남 사별	남 이혼	여 미혼	여 유배우	여 사별	여 이혼
합계											
15-19세											
15											
16											
17											
18											
19											
20-24											
20											
21											
22											
23											
24											
25-29											
.											
.											
.											
80-84											
80											
81											
82											
83											
84											
85세이상											

- 대상 : 15세이상 전인구
- cell 수
  표측 표두
  86 × 11 = 946

(1) 배우관계 미상 포함

그림 7-2 통계표 2 (연령, 남여, 배우관계 15세이상 인구)

제 3 표 산업(소분류), 종사상지위(6구분) 남여별 14세 이상 취업자수

- 전국, 시도

산업(소분류)	사업체 종사자 지위 합계	고용인	중역	고는용사업인이주있	고는용사업인이주없	가족종사자	가대정직자	남 합계와 같음	여 합계와 같음
합계									
농업									

- 대상 : 14세 이상의 취업자
- cell 수
  표측 표두
  235 × 21 = 4,914

1) 종사상 지위「미상」 포함

그림 7-3 통계표 3

제 4 표 사회경제분류, 산업중분류, 14세이상 취업자수

- 전국, 시도

산업중분류	합계	농림어업자	농고림용어인업	(사회경제분류)	고안직	내직자

- 대상 : 14세이상 취업자
- cell 수
  표측 표두
  61 × 19 = 1,159

1) 사회경제분류「분류불능」 포함

그림 7-4 통계표 4

제 2 표는 15세 이상을 대상으로 표측을 15세 이상 각세에서 5세별로 합계를 구하고 표두는 제 1 표와 마찬가지로 2개 부분으로 나누어 좌측 3 cell 에서 15세 이상의 남녀별 인구, 우측 8 cell 에서 남녀별 배우관계별 인구수를 구하고저 하는것인바 각주에 있는바와 같이 배우관계는 표두의 4개 구분 외에 「미상」이 있기 때문에 「배우자 관계」의 합계와 15세 이상의 인구 수와는 일치하지 않는다.

제 3 표는 14세 이상의 취업자를 대상으로 표측은 산업 소분류, 표두는 남녀별, 종사상의 지위로 구분되고 이때에 각 총수에는 미상이 포함되어 있기 때문에 내역과는 일치하지 않는다.

제 4 표는 제 3 표와 마찬가지로 14세 이상의 취업자를 대상으로 표측은 산업 중분류에 대하여 구하고, 표두는 사회경제분류의 18개 항목만을 집계하는 것으로 하고 총수에는 미상이 포함되기 때문에 내역과는 일치하지 않으며 이상의 4개 표는 공히 행정 역을 두주에 있는바와 같이 란외 항목으로 한다.

이때에 집계에 사용되는 data 량은 전체 인구(약 4200 만)의 15 % 추출 집계량에 해당하는 630 만 record이고 그 중 14세 이상 취업자의 점유비율이 48 % 이므로 14세 이상 취업자의 data량은 300 만 record 이상의 방대한 량이 된다.

2. check program의 설계

각 조사항목에 대한 check는 집계담당자 보다는 조사설계담당자가 주관할 문제로서, 정해진 check 절차에 따르는 program 작성을 가정하고 check process flow 전체를 거론하는 것으로 하여 error data 처리에 표준화된 flow를 채택하는 check처리 flow 는 그림 7-5, check 완료 data의 형식은 그림 7-6, check된 data의 부호는 표 7-1 과 같은 것으로 한다.

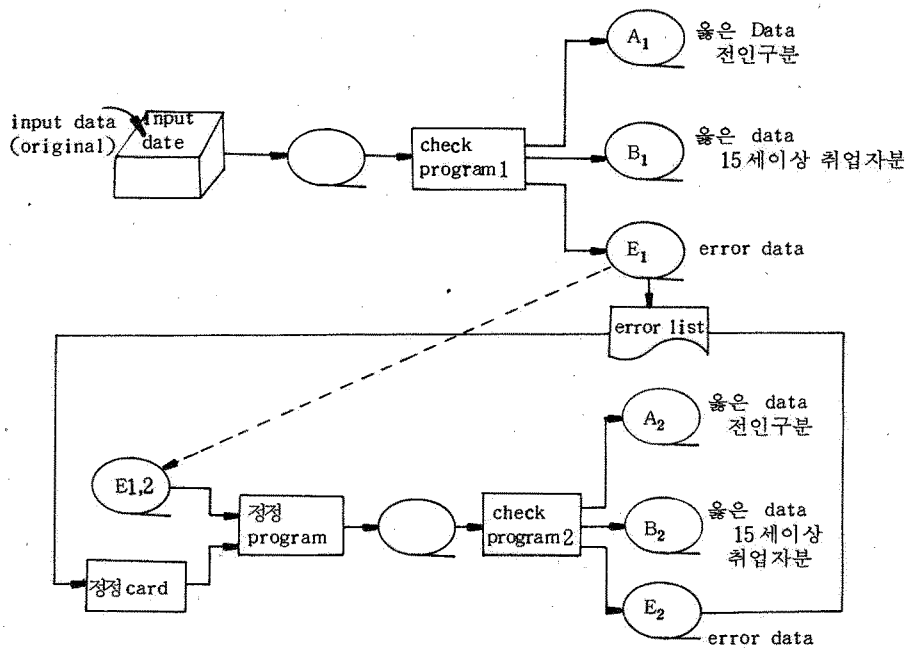


그림 7-5 check 처리의 flow

check처리의 flow에서는 error data가 없어질 때까지 반복하여 정정하는 방법을 채택하고 check program은 제1회 check와 제2회 이후의 check 내용을 바꾸기 위하여 각각 별개로 작성한다.

올바른 data의 output은 $A_1 \sim A_n$ (첨자:연산 횟수)의 전 인구(전 data)와 $B_1 \sim B_n$의 취업자 data로 구분하여 통계표를 전체 data를 대상으로 하는 표(제1, 2표 다만 제2표는 편의상 이에 분류)와 취업자를 대상으로 하는 표(제3, 4표)로 대별하고 분포program의 area관계로 표측항목에서 sort할 필요가 있기 때문에 data는 적은 편이 좋으므로 취업자의 data를 별개의 file로 구성한다.

각 file 항목은 sort를 고려하면 집계에 필요한 항목만으로 압축하여 record length를 축소하여야 하는데 전 인구 file(A)은 check 완료된 master file로 볼수 있기 때문에 조사항목은 모두 output(B)에 수록하고 취업자 file은 집계에 필요한 항목만을

발취해도 무방하다.

그림 7-6의 check 완료 data 형식에서 matrix란과 inclusion란을 설정하지 않는것은 이들의 code란을 설정하지 아니 해도 matrix code의 계산과 data의 선택이 가능하다는 판단이 서기 때문이다.

항 목 명	위 치 (column)	내 용
시 도	1～ 2	01→서울, 02→부산, …… 15→제주
조사표번호	3～ 6	0001～9999 (집계에는 사용하지 않음)
총 괄 번 호	7～ 9	001～999 (집계에는 사용하지 않음)
성 별	10	1→남, 2→여
년 령	11 12～13	5세 code : A→0～5세, B→5～9세, …… T→95～99세 U→100세이상 각세 : 00→0세, 01→1세, …… 99→99세, Blank→100세이상
생 월	14	1→1～3월, 1→4～6월, 3→7～9월, 4→10～12월
배 우 자 관 계	15	15세이상에 대하여 : 1→미혼, 2→유배우, 3→사별, 4→이혼, 5→미상 15세 미만은 Blank
사회경제분류	16～17	01～23 : 자료 1 참조
종 사 상 지 위	18	15세이상 취업자에 대하여 : 1→고용자, 2→임원, 3→유고용업주, 4→무고용업주, 5→가족종사자, 6→가정내직자, 7→미상 15세 미만 및 비취업자는 Blank
산 업 분 류	19 20～21 22～24	대분류 : A ～ N 중분류 : 01～ 46 자료 2 참조 소분류 : 001～173

그림 7-6 check 완료 data의 부호표

3. 분포 program의 설계

가. program의 사양

기능 분리형 system 에서는 분포 program 의 설계가 가장 중요한 바, 그 이유는 분포 program 설계과정의 table image 결정이 check 완료 data의 형식과 file의 구성에 결정적인 영향을 미치고 후속 program의 처리방법을 좌우하기 때문이며 따라서 본문에서는 설명순서에 따라 check program 설계를 선행하는 것으로 하였으나 본래는 분포 program 설계를 선행하고 matrix와 inclusion란의 필요성을 판단하여 file의 분합여부를 결정하여야 한다. 분포 program의 설계에 있어서 table image 의 결정에 선행되어야 함은 이미 언급한 바와 같으나 table image의 결정에는 computer memory가 크고 난외항목, 표측표제의 총수란을 포괄하는 full image를 수용할 수 있는 memory 확보가 가능하다는 전제가 필요한 바 여기에서는 각 통계표의 표측 1 line분의 용량을 소요로 하는 소규모 computer를 가정하였다.

table image를 최소화 하기 위해서는 control의 채택과, 총수의 삭제기능의 동원을 예상할 수 있는데 control은 R → S의 순으로 접근해야 하므로 우선 4개 통계표의 두주가 전국·시도 인것에 착안하여 전국 시도를 control 항목으로 하고 다시 표측항목을 control에 추가하면 2개의 sort가 필요하게 되고 분포 program은 다음과 같이 2개의 program으로 작성하지 않으면 아니된다.

program 1 → 도〉연령의 control로 제1, 2표 처리

program 2 → 도〉산업의 control로 제3, 4표 처리

이때에 2개 program을 하나로 통합하기 위하여 시도〉연령〉산업을 control로 하면 제3표, 제4표의 summary record는 연령의 control이 끊기는 횟수 배 만큼 증가하여 summary record의 처리시간이 급증한다.

제3표를 예로 하여 살펴보면 2개 program의 record수는 16(시·도)×173(산업소분류)=2,768 record 이나, 연령 control을 추가하면 평균연령 67세에 해당하는 횟수만큼 control이 끊긴다고 할때 전체 record수는 16(시·도)×67(연령)×173(산업소분류)=288,391 record가 되어 처리대상 전체 record수는 column을 벗어나는 결과를 빚게된다.

다음으로 총수를 skip하고 표측 단위 line의 table image를 정의하면 그림 7-7과 같이 된다.

제1표 control 시도>연령각세

	남				여			
	1-3	4-6	7-9	10-12	1-3	4-6	7-9	10-12
연령(각세)								

cell수 = 1×8 = 8cell

제2표 control 시도>연령각세

	배우관계미상		남			여		
	남	여	미혼	기혼	사별	이혼	미혼	기혼	사별	이혼
연령(각세)										

cell수 = 1×10 = 10cell

제3표 control 시도>산업소분류

	남							여
	미상	고용원	중역	고용원이있는업주	고용원이없는업주	가구종사자	내직자	과외 내역과 같음
산업 소분류								

cell수 = 1×14 = 14 cell

제4표 control 시도>산업중분류

	미상	1	.....사회 경제 분류.....	18
산업 중분류				

cell수 = 1×19 = 14cell

그림 7-7 table image

제2～4표에서는 표두부분의 합계에 미상(분류 불능)을 포괄하기 때문에 미상의 분포를 위하여 cell을 예비해 두고 가공편성 program에서 내역부분을 더하여 합계를 구하는 것으로 하며, 이때에 각 표의 summary record 형식은 그림 7-8과 같이 정한다.

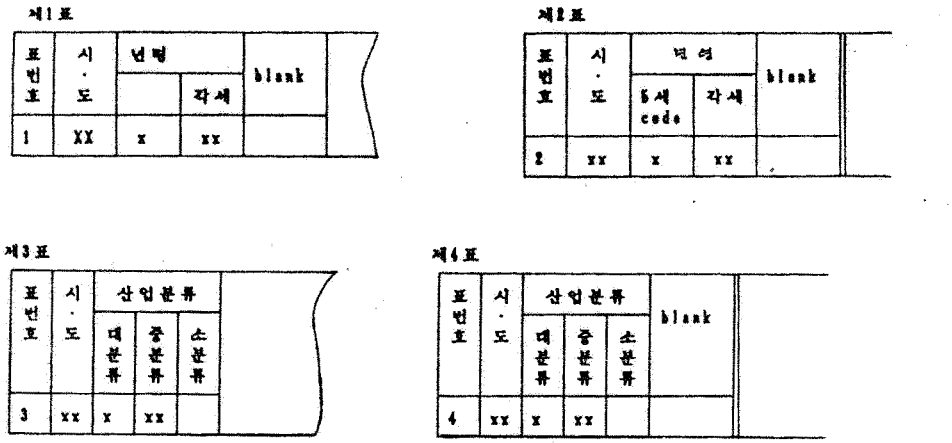


그림 7-8 summary record

RID의 설정에서 주의하여야 할 사항은 합산 program에서 합산 산출을 용이하게 하기 위하여 상위 group code를 부여하여야 하는데 제1표에서는 연령 각세 code 만으로는 5세별 합산이 어렵기 때문에 input data에 부여된 code를 output에서만 이용하는 5세 code를 부여한다.

분포 program은 control이 시·도>연령의 제1, 2표와 시·도>산업 소분류의 제3, 4표의 2개 program으로 하고 처리 flow를 그림 7-9와 같이 정한다.

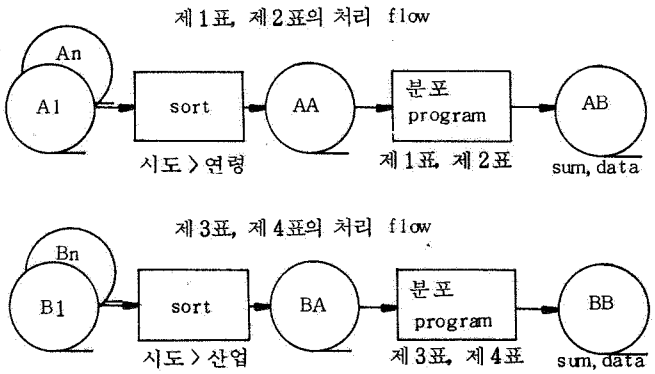


그림 7-9 처리 flow

나. 분포 program 1의 flowchart

분포 program 1에서 제1표와 제2표의 data 분포범위가 다른 점에 유의하여 제1표에서는 15세 미만은 분포하지 아니하고 85세 이상은 절사하는 조건 하에서 control이 끊기지 않도록 해야한다.

control을 중심으로 한 program 1의 flowchart는 그림 7-10과 같다.

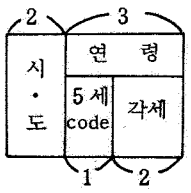
#1 : i/o file의 open.

#2 : input data "AA"의 read.

#3 : data read후 new key의 시도번호 위치를 "99"로 set.

#4 : new key 항목을 다음과 같이 선정하여 set.

#5 : new key와 old key의 전체 column을 비교하여 control brake 여부 check.



#6 : non control brake이면 제1표 분포.
table image는 표측 1 line만 정의하고 있으므로 표두항목에서 분포점 결정. 본문의 15% 추출 data 기준 승수 20/3을 1대신에 대입.

#7 : 제2표는 15세미만을 대상외로 하기 위한 비교임. input 항목의 배우관계 code의 blank 여부 check.

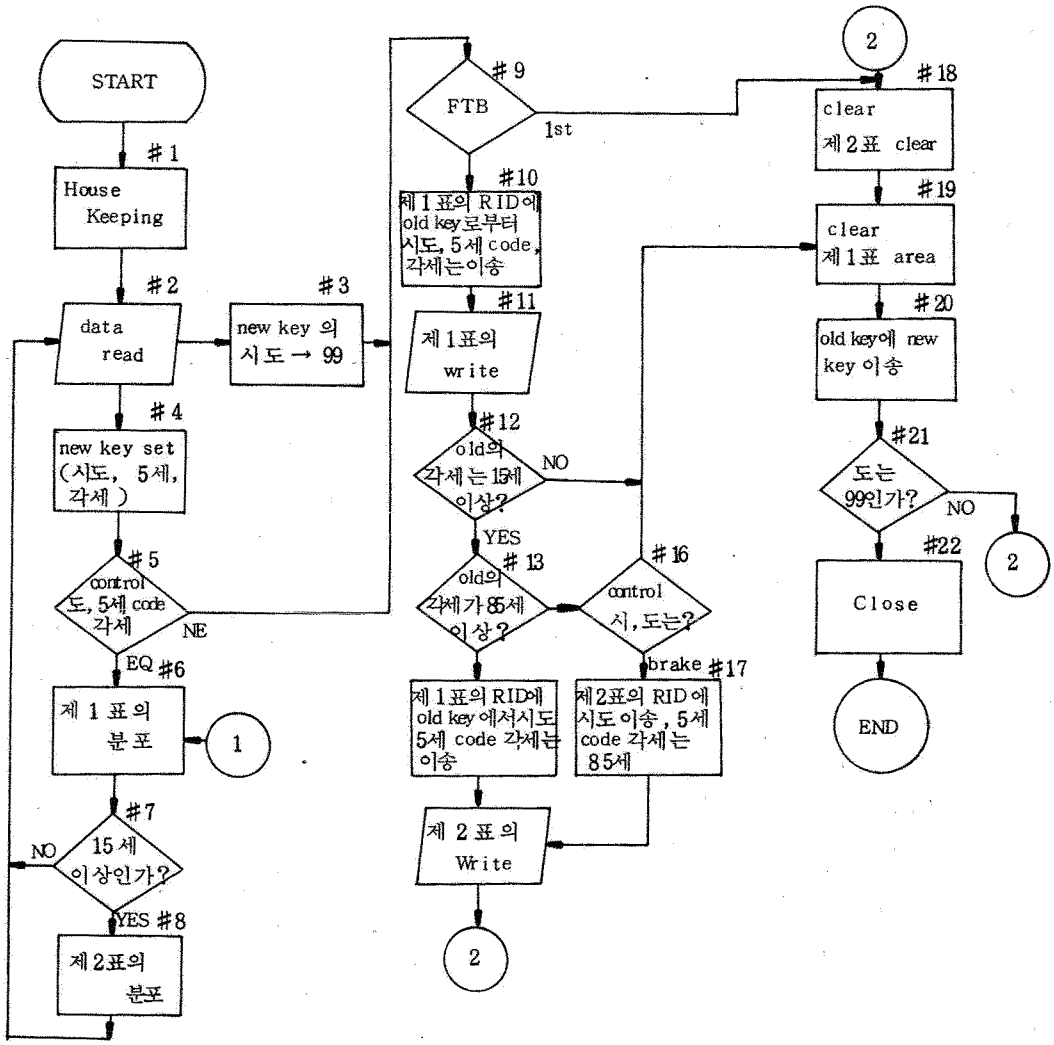


그림 7-10 program 1의 flow chart

#8 : 제2표의 분포 : 배우관계가 미상인 data의 분포가 변측적임에 주의

#9 : 최초 1회의 branch switch : 최초 data read 시의 table image clear

#10 : 제1표의 write를 위한 RID항목의 set : 반드시 old key에서 수취.

#11 : 제1표의 write.

#12 : 제2표 write를 위한 판단 : old key의 연령 15세 미만은 분포대상이 아니므로 제1표의 clear routine에 branch. 제2표의 clear는 미분포 상황으로 불용임.

#13 : 85세 이상의 data 여부 check. 85세 미만은 각세에서 control brake 되기 때문에 제1표와 같이 취급해도 무방하나 85세 이상은 control brake 없이 마무리 가능.

#14-#15 : 15~84세의 각세 control분의 RID set와 write.

#16 : 85세 이상은 다음 시·도 data read 때까지 처리되어야 하므로 시·도의 control brake 여부 check.

#17 : 85세 이상의 write를 위한 RID set, 85세 이상의 표시는 85세의 5세 code, 각세로 대표.

#18-#19 : 제1, 2표의 table image clear

#20 : old key에 new key 이송.

#21 : new key의 시·도 번호가 "99"인가를 check.

#22 : i/o file의 close.

다. 분포 program 2의 flowchart

분포 program 2의 flowchart는 표준적인 것으로 제3, 4표에서는 control level이 다르다는 사실을 고려한다면 어려운 점은 없으며 flowchart는 그림 7-11과 같다.

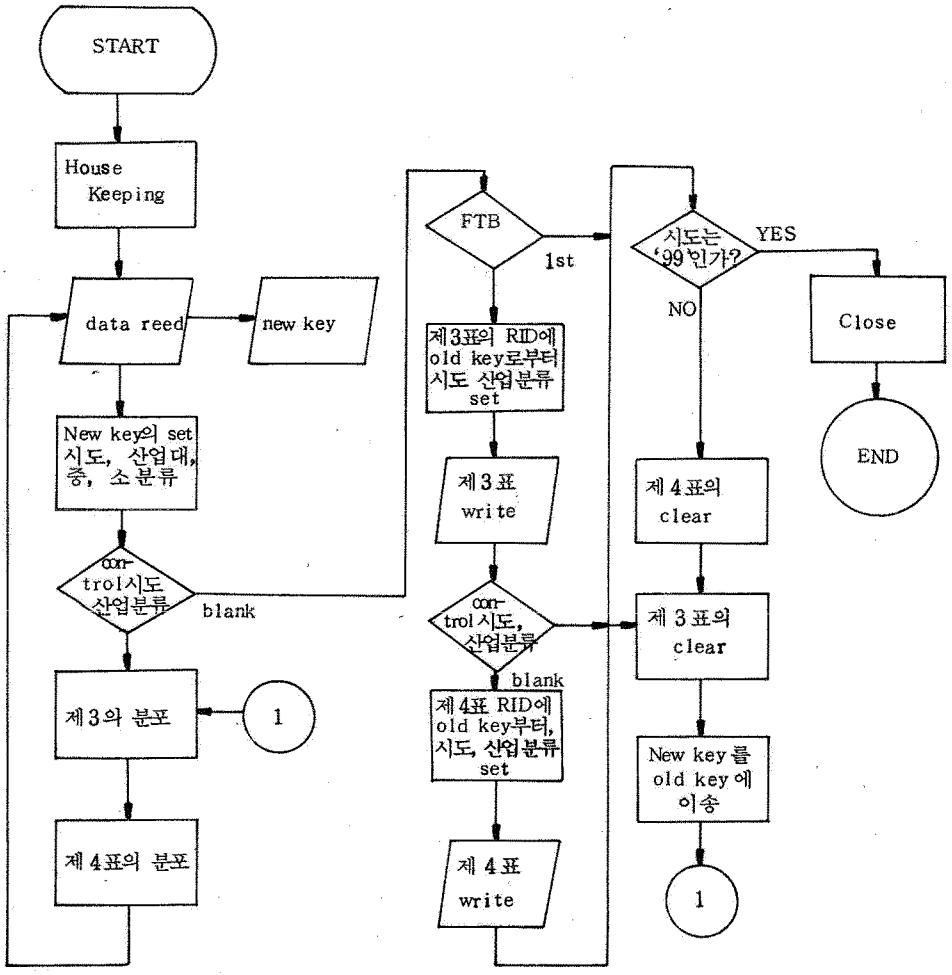


그림 7-11 분포 Program 2의 flow chart

4. 합산 program의 설계

가. program의 사양

표측 항목의 합계는 분포 program에서 구하고 있지 않으므로 합산 program에서 란외항목을 포함하는 모든 합계를 구한다.

합산 program의 합계산출은 대상항목을 최하위의 sort key 로 하거나 무시하고 잔여항목에 대하여 sort 하여 control을 끊는것이므로 각 표의 RID 항목의 위치와 길이는 정돈되어 있어야 하며, 따라서 summary record의 RID 는 연령과 산업분류의 위치와 길이를 일치시키고, 편의상 2개 항목의 각 field를 K_1, K_2, K_3 로 하여 sort key와 control 지정에 원용한다.(그림 7-12)

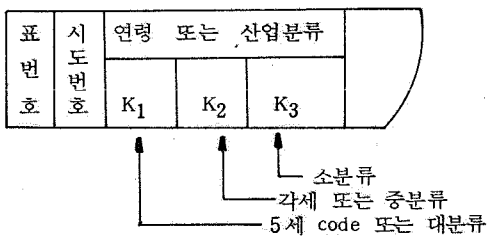


그림 7-12 각 field와 sort key 또는 control의 지정

합산 program output record의 RID는 합계를 산출한 항목을 blank로 메우고 기타는 input record의 RID를 부여하는 것으로 하는데 본래는 합계를 나타내는데 이때에 blank 보다 "T" 가 바람직하지만 K_1에 영문자가 내포 되므로서 sort 할 때에 합계 line이 오지 않을 염려가 있기 때문에 blank로 처리하는 것이다.

각 표의 표측 항목부분의 RID는 그림 7-13과 같고 처리 flow는 그림 7-14와 같이 된다.

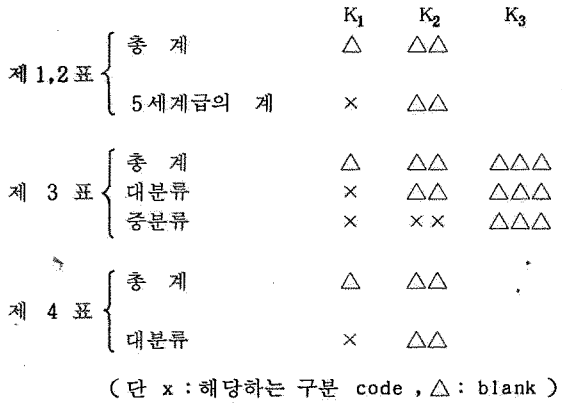


그림 7-13 표측 부분의 RID

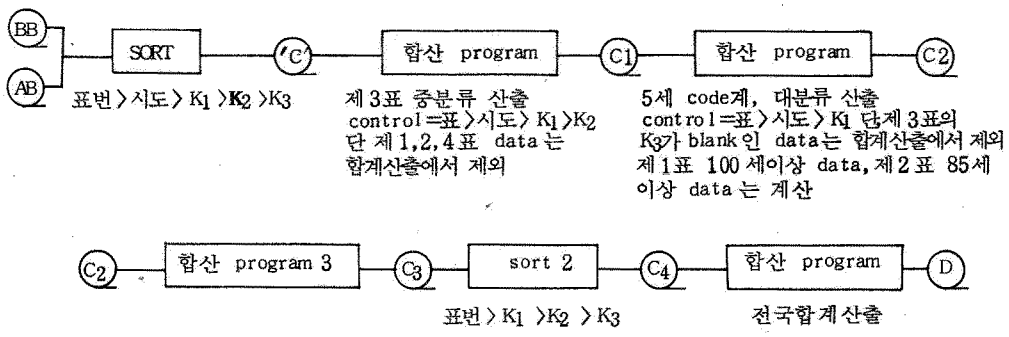


그림 7-14 합산 PROGRAM의 처리 FLOW

나. 합산 program 1의 flowchart

합산 program 1에서는 제 3표의 중분류를 구하는 것이기 때문에 제 3표 이외의 data는 합계산출에 불필요 하므로 input data의 read 즉시 output file에 수록해도 무방하나 중분류 code가 16인 제 3표의 최후 합계가 제 4표 다음에 수록되어 불합리한 결과가 되기 때문에 control brake된 상황에서 합산과 수록이 되지 않도록 하는데 이때의 flowchart는 그림 7-15와 같다.

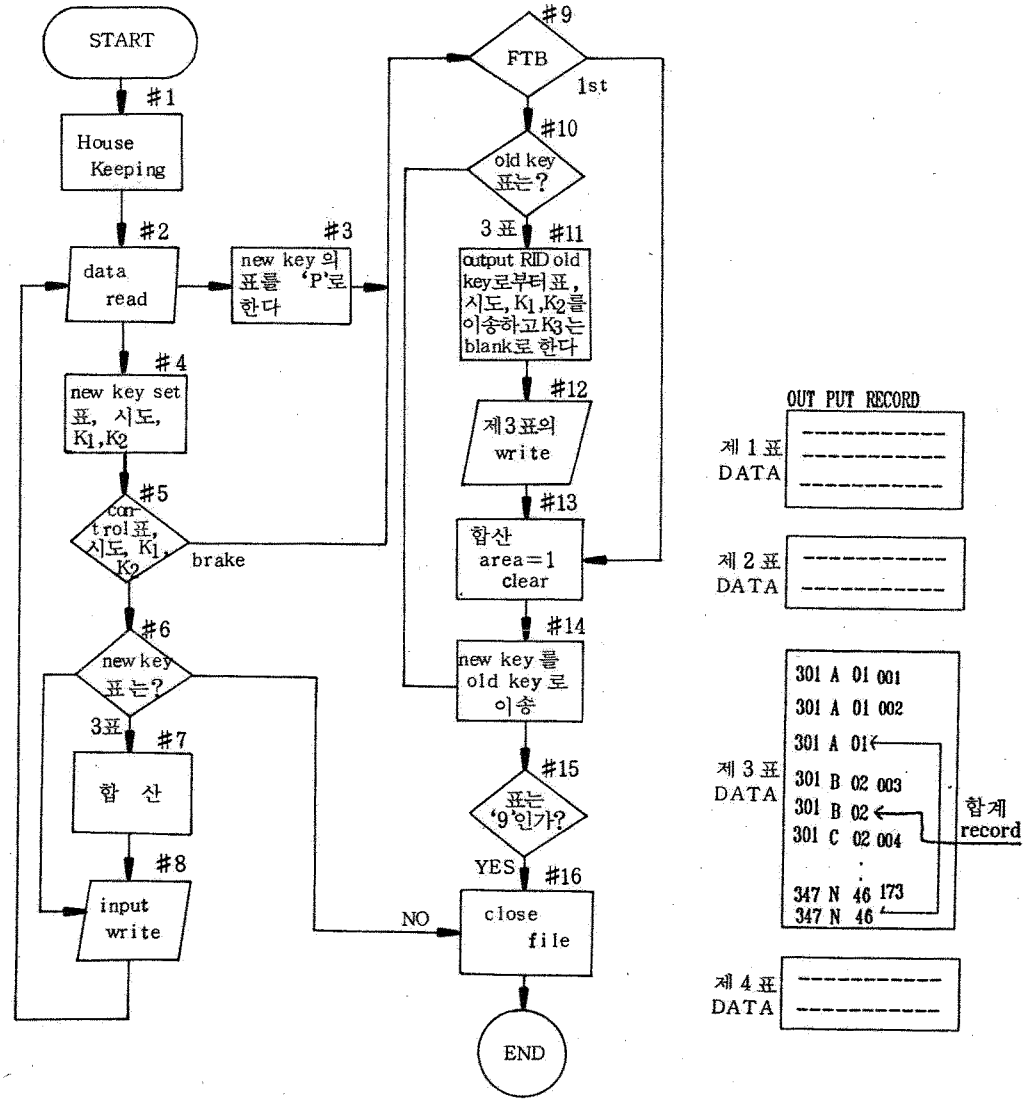


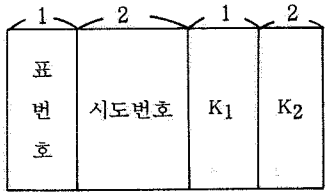
그림 7-15 합산 program 1의 flow chart

그림 7-15의 각 event에 대하여 설명하면 다음과 같다.

#1 : i/o file의 open.

#2～#3 : data의 read. data가 없을 때의 new key 표번호 위치 "9"로 set.

#4～#5 : new key의 set, control brake 여부 check new key는 다음과 같이 배열한다.



#6～#8 : new key에 표시된 input data 가운데 제3표만을 발취하여 합산하고 input data는 모두 output에 write.

#9 : 최초의 branch switch.

#10～#13 : 제3표 이외는 합산하지 않으므로 write를 by-path하고 제3표는 output record의 RID에 old key에서 필요항목을 이송하고 다시 K_3의 위치에 blank를 송출.

#14～#16 : new key를 old key로 이송하고 다시 표번호가 "9"인가를 check, "9"이면 close.

다. 합산 program 2의 flow chart

합산 program 2의 flow chart는 그림 7-16과 같으나 이 program에서 제1,2표의 5세별 합계와 제3,4표의 대분류를 구할 때 제1표의 100세 이상, 제2표의 85세 이상은 5세별로 구하지 아니하고 제3표의 합산 program 1에서 합계 record의 input을 배제할 필요가 있다는 점에 유의하고 flow chart의 #6～#7에서 제3표의 중분류(K=blank)를 합산 routine에서 제외한다.

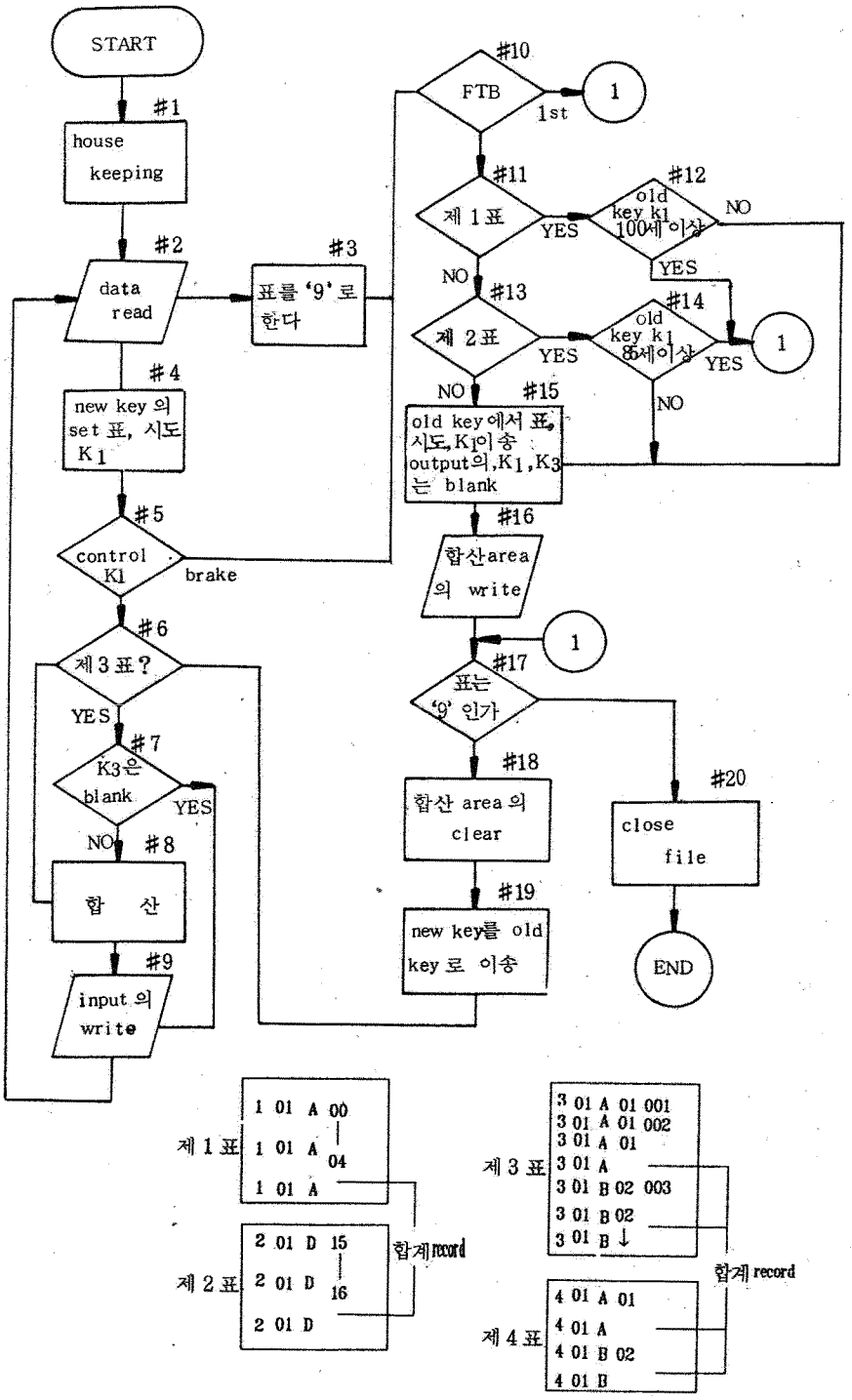


그림 7-16 program flow chart

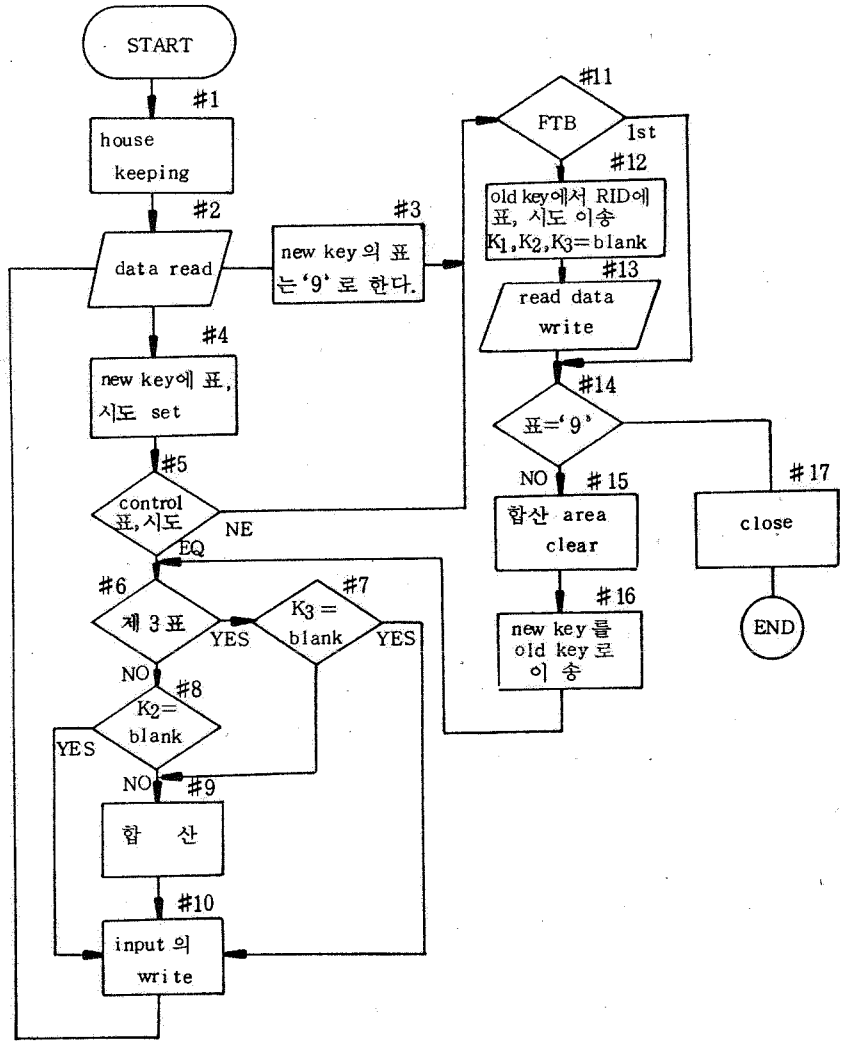


그림 7-17 합산 program의 flow chart

제 1,2 표에서는 100 세 이상과 85 세 이상의 data를 제외하지 아니하고 합산하는 대신에 control brake 후에 수록하지 않도록 하고 있는바 그 이유로는 합산여부를 판단하는 routine (# 11 ~ # 14) 을 # 6 부분에 추가해도 control brake 후에 내용이 zero 인 record를 수록하지 않도록 억제하기 위하여 # 11 ~ # 14 가 필요하므로 복잡한 program 보다는 data 량이 적은 이유로 합산 후에 수록하는 편이 유리하다고 판단되기 때문이다.

라. 합산 program 3의 flow chart

합산 program 3은 표측항목의 총수를 구하는 것으로 flow chart는 그림 7-17과 같고 이미 앞의 program에서 중간 level의 합계를 구하도록 되어 있기 때문에 합계 record를 합산에서 제외하고 총수를 구해야 하므로 flow chart에서는 # 6 ~ # 8 부분에서 K_2, K_3의 blank record를 제외한다.

마. 합산 program 4의 flow chart

통계표의 전국합계 program으로 전국합계를 구하기 위해서 행정구역 시·도를 최하위로 하는 sort (sort 2)가 필요하며 전국합계 program의 표준적인 flow chart는 그림 7-18과 같다.

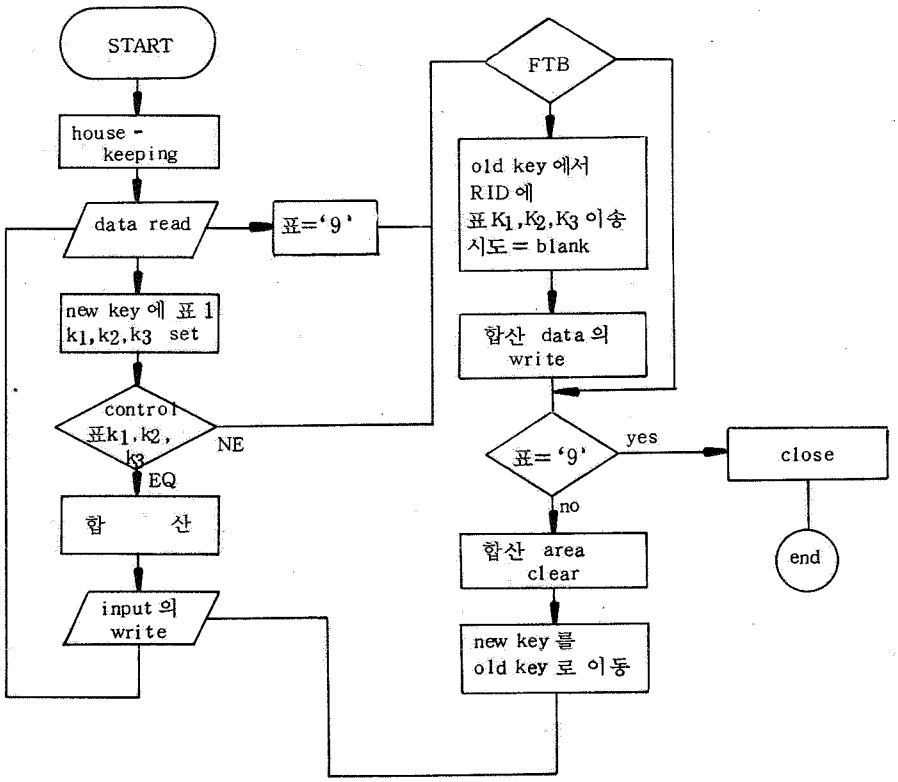


그림 7-18 합산 program 4의 flowchart

5. 가공편성 program의 설계

가. program의 사양

난외항목과 표측항목의 합계산출에 이어 표두항목의 배열변환과 합계산출을 하여야 하는데 표두 분할과 C,C,C의 부여작업은 후속하는 편집 program에서 처리하는 것으로 한다.

4개표의 routine이 동시에 memory되고 over lay 구조를 배제하는 program을 작성하면 표두항목의 work area에의 이송과 합계산출은 가공편성 program에서 설명한바 있는 subroutine을 사용하면 control 정보는 다음과 같다.

제 1표는 그림 7-19와 같이 work area 내에서 4～7 cell의 합계가 제 2 cell, 8～11 cell의 합계는 제 3 cell, 제 2 cell과 제 3 cell의 합계는 제 1 cell이 되고, 제 2표는 그림 7-20와 같이 배우관계의 미상란이 별도로 집계토록 되어있기 때문에 미상란에 미상 이외의 배우관계를 가산하면 15세 이상의 인구를 구할 수 있다.

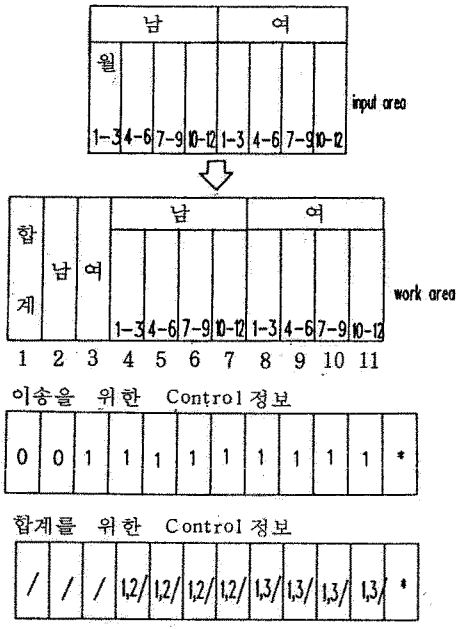


그림 7-19 가공편성 program의 control정보(표 1)

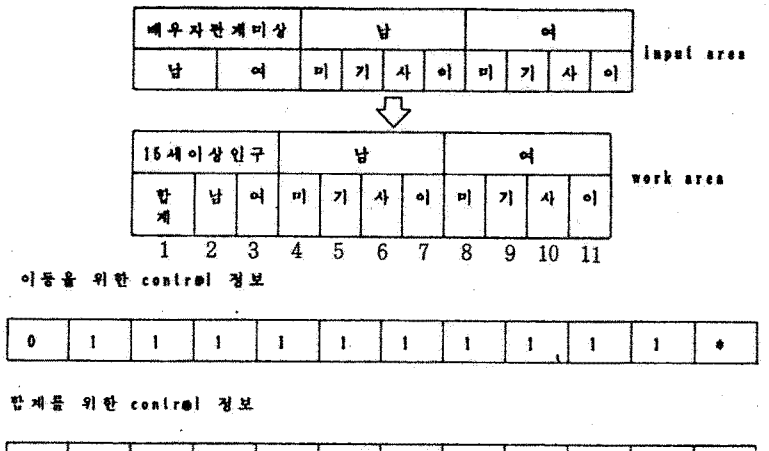


그림 7-20 가공편성 program의 control정보(제 2표)

제 3 표는 그림 7-21과 같이 총수란의 위치가 미상부분을 집계하고 있음을 염두에 두고 합계를 구해야 하며 제 4 표는 그림 7-22와 같이 input record의 제 1 cell에 「분류 불능」이 집계되기 때문에 여기에 잔여부분의 cell을 가산하면 합계를 구할 수 있다.

제 4 표에서는 이 예에서 볼수 있는바와 같이 control 정보로 routine을 제어하지 아니하고 이송과 합계를 roof에 의하여 control 해도 무방하다.

이상으로 가공편성 program의 처리 flow를 정리하면 그림 7-23과 같이 되는바 가공편성 program은 overlay 구조와 C.C.C의 부여를 생략하기 때문에 sort가 불필요 하지만 편집 program의 수행에 선행하여 sort가 필요하므로 sort에 적합하도록 data 의 순서를 갖추어야 한다.

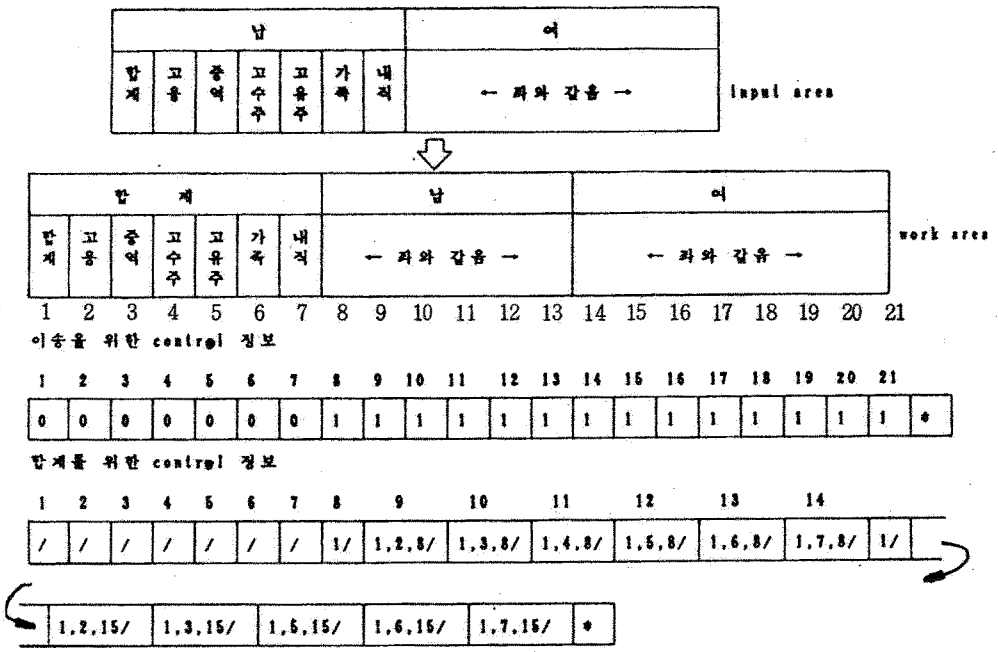


그림 7-21 가공편성 program의 control 정보(제 3 표)

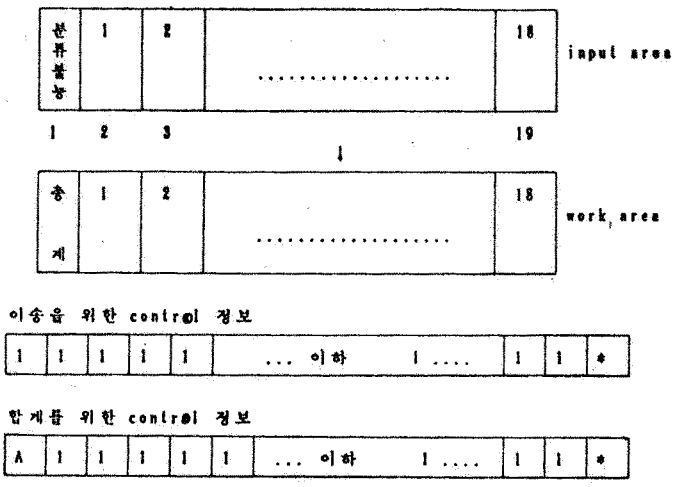


그림 7-22 가공 편성 program의 control 정보(제 4 표)

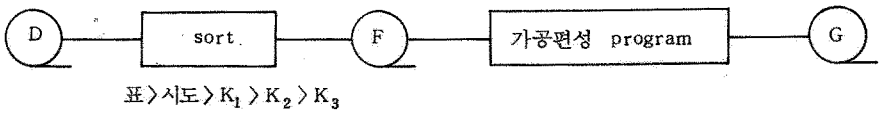


그림 7-23 가공편성 처리 flow

6. 가공 편성 program의 flow chart

가공편성 program의 flow chart를 그림 7-24에 도시한 바에 따라 흐름의 요지를 설명하면 다음과 같다.

1 : i/o file open

2 : data의 read

#3～#6 : 표별 routine에 blanch하기 위한 판단

7 : error data의 검출, 표번호가 error data를 검출하면 print 하고 다음 data를 처리

#8～#9 : 제 1 표용 처리 routine. 이송과 합산은 control 정보에 의하여 subroutine에서 처리.

#10～#11 : 제 2 표용 처리 routine. 이송과 합산은 control정보에 의하여 처리.

#12～#13 : 제 3 표용 처리 routine. 이송과 합산은 control정보에 의하여 처리.

#14～#15 : 제 4 표용 처리 routine. 이송과 합산은 control정보에 의하여 처리.

#16 : close.

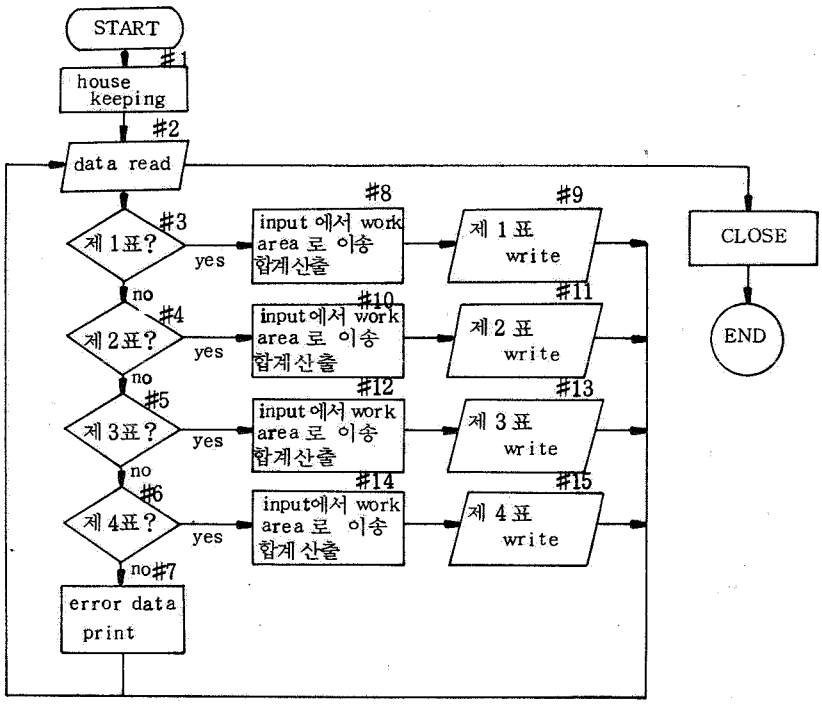


그림 7-24 가공편성 program의 flow chart

7. 편집 program의 설계

가. program의 사양

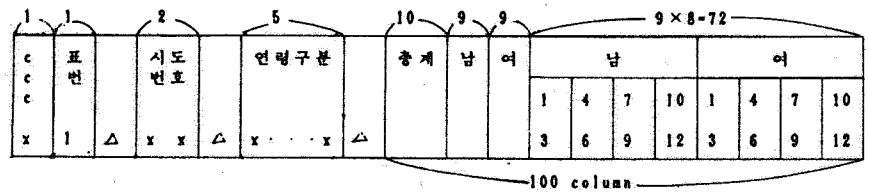
편집 program에서는 편집 외에 C.C.C의 부여와 표두항목의 분활을 수행하는 바, 각 표의 print형식은 다음과 같이 된다.

다만 표두색인은 부여하지 않고 표측색인을 연령과 산업분류에 국한하여 간단히 부여하는 것으로 가정하고 설명키로 한다.

제 1 표에서는 표번호와 행정구역(시·도)번호는 input data를 그대로 사용하고 연령구분은 print할 때에 보다 쉽게 하기 위하여 그림 7-25 와 같은 특수문자로 변환하는 것으로 한다.

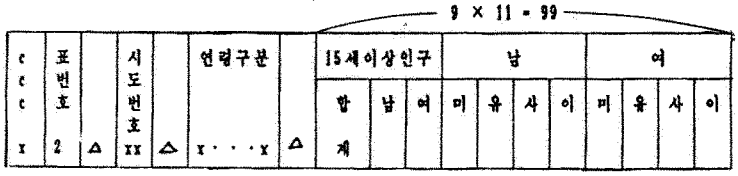
표체부분은 총수부분을 10 column으로 하고 기타는 9 column으로 하여 인근 cell의 숫자와 연계되지 않도록 배려한다.

제 2 표에서는 제 1 표와 마찬가지로 표번호와 행정구역(시·도)번호는 input data code를 그대로 사용하고 연령구분은 그림 7-26과 같이 표측색인을 부여하며, 표체부분의 각 cell은 공히 9 column으로 하고 제 3 표에서는 표두항목의 분할이 필요한 바 그림 7-27과 같이 분할을 3등분하고 cell의 column수를 10 column으로 하여 표측항목의 산업분류는 번호뿐인 간단한 색인을 부여하는 것으로 하며 제 4 표도 제 3 표에서와 같이 표두항목의 분할이 필요한 바 표두항목은 19개항목이므로 그림 7-28과 같이 제 1 분할은 10개항목으로 하고 제 2 분할을 9개항목으로 하되 cell의 column수를 10 column으로 하며 표측색인은 제 3 표에 준하는 것으로 한다.



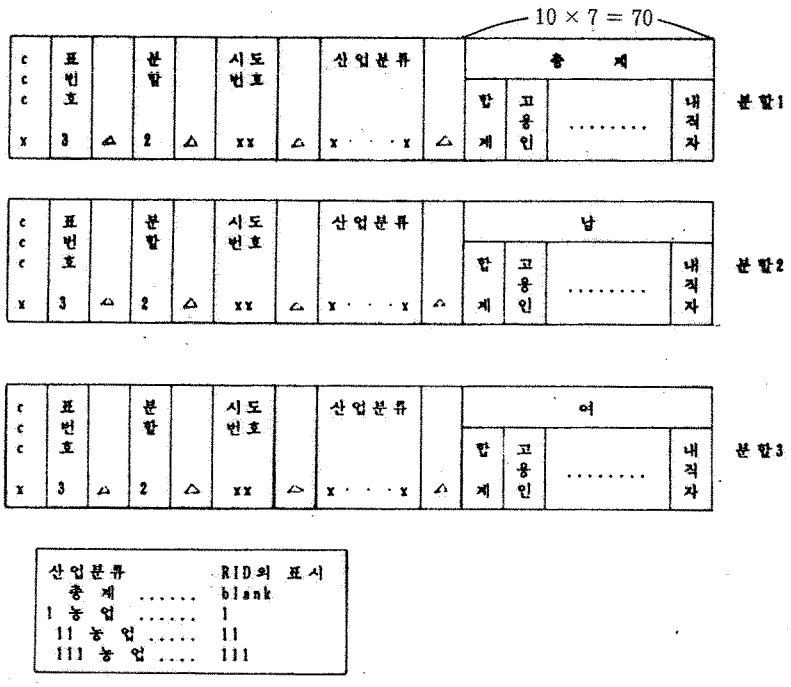
연령구분		RID표시
총 계	.....	blank
0 - 4 세	.....	0 - 4
0	.....	000
.		
.		
.		
95-99	.....	95-99
100세이상	.....	100..

그림 7-25 편집 program 설계사양(제 1 표)



연령구분	
합계	blank
15-19.....	15-19
15.....	15
.	
.	
.	
84.....	84
85세이상	85...

그림 7-26 편집 program 설계사양(제 2 표)



산업분류	RID의 표시
총 계	blank
1 농 업	1
11 농 업	11
111 농 업	111

그림 7-27 편집 program 설계사양(제 3 표)

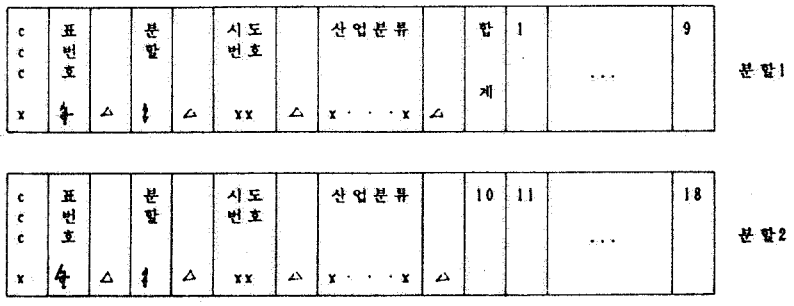


그림 7-28 편집 program 설계사양(제 4 표)

C.C.C는 전체표 공통으로 표번호와 행정구역(시·도)번호 중 어느 하나가 brake 되면 page 변환 (9)을 하고 5세 code나 산업분류 code의 control이 끊기면 double space (2), 그 이외의 것은 single space (1)이 되도록 하고 분할한 record (제 2, 3 분할)는 별개의 tape로 분리하여 수록하되 처리 flow는 그림 7-29와 같다.

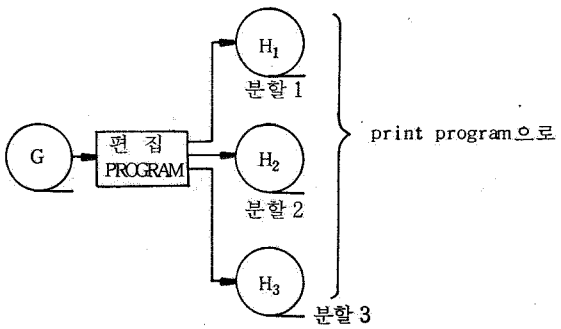


그림 7-29 편집 program 설계사양의 처리 flow

나. 편집 program의 flow chart

편집 program도 overlay 구조가 아니면 그림 7-31의 flow chart와 같이 되는바 이 flow chart의 요지를 설명하면 다음과 같다.

1 : i/o file의 open.

#2～#3 : input data의 read. data처리 후 표번호를 9로 set.

4 : C.C.C를 부여한 control file의 set. new key의 배열은 다음과 같음.

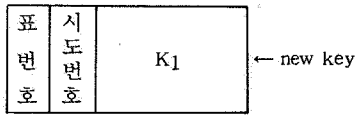


그림 7-30 control field의 set

5 : 최초 data의 work area로의 이송 switch. 최초 1회 = NOP.

6 : C.C.C 결정 시 까지 data 처리를 유보하고 input data의 pool.

7 : new key의 old key 이송.

#8～#12 : C.C.C 결정. C.C.C를 1로 set 하고 표·시도·K_1의 key가 끊기면 C.C.C는 최소한 2가 되며, 시·도의 key가 끊기면 page 변환(C.C.C = 9).
C.C.C는 후속 data read 후에 결정되므로 routine 상으로는 하나의 data를 pool 형식으로 처리하고, C.C.C는 인쇄후 개행이 가능하므로 개행후 인쇄로 전도되면 input data read와 output data수록이 일치하고 pool이 불필요하므로 output time을 read time 보다 1 data씩 천연.

#13～#16 : 각표의 선택 step. 표별 routine은 memory에 동시 입력이 가능하면 간단한 비교로 branch 하고, overlay 구조 하에서는 각 표의 routine을 read.

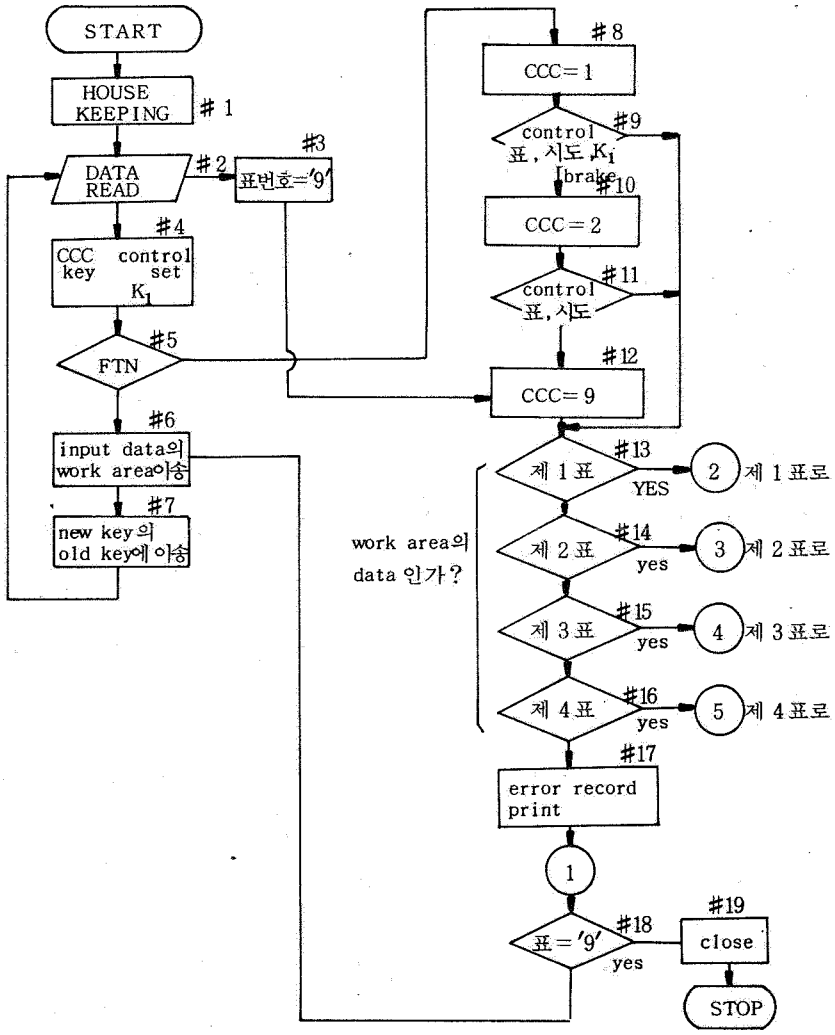


그림 7-31(1) 편집 program의 flow chart

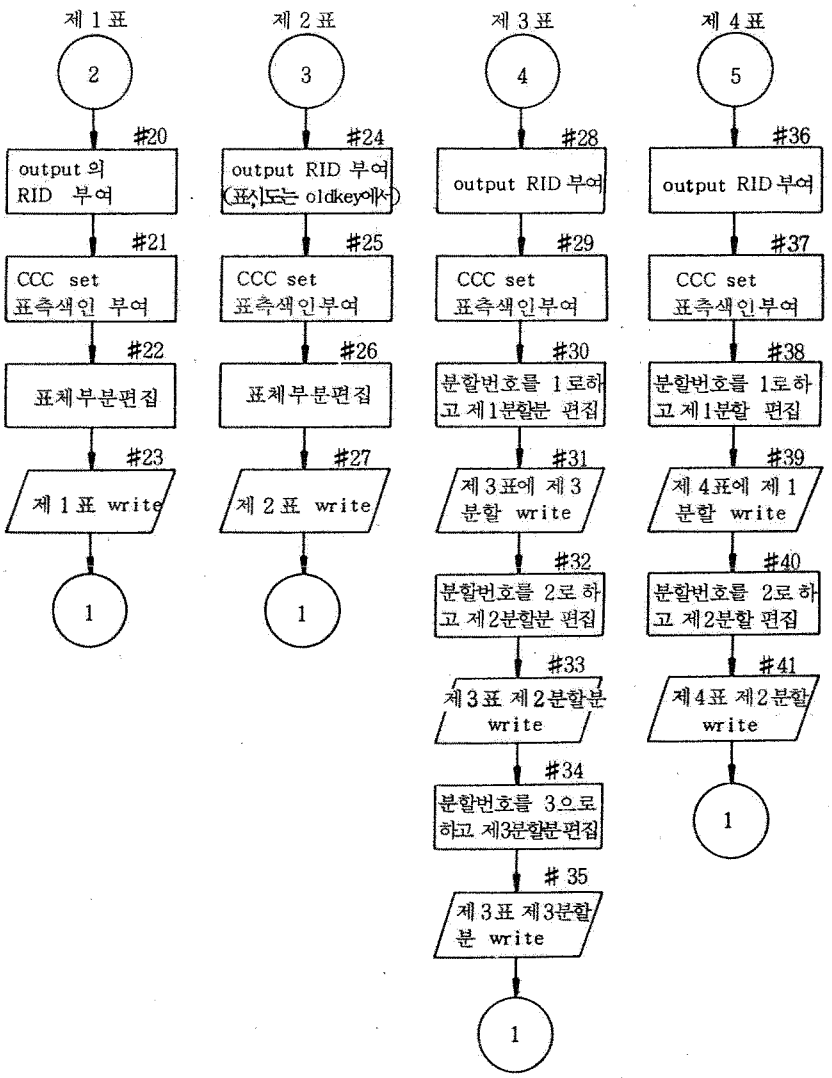


그림 7-31 (2) 편집 program의 flow chart (계속)

17 : 표번호 error data의 print. #13～#16에서 표번호를 비교하여 즉시 check.

18 : 처리대상 data 유무 판정.

19 : 처리대상 data가 없으면 각 file을 close하고 program 종료

#20～#23 : 제 1표의 처리 routine. old key에서 output record의 RID에 「표번·시도번호」와 C.C.C 이송. old key의 K_1, K_2 (연령 5세 code, 각세)을 기초로 표측색인 변환 (그림 7-32). output의 RID에 set. 표체부분 편집시 유효숫자와 좌측의 reading zero를 제거하고 편집후 즉시 수록하되 제 1표는 분활이 불필요하므로 제 1분활 file에 수록.

#24～#27 : 제 1표 처리와 동일한 제 2표의 처리 routine.

#28～#35 : 제 3표의 처리 routine. 제 3표는 3개 표두분활의 분활번호와 대응하는 표체의 편집이 다를뿐 제 1, 2표의 처리와 동일하고 각 분할의 data는 개별 file로 분류 수록.

#36～#41 : 제 4표의 처리 routine. 제 4표의 2개 표두분활 처리형태는 제 3표와 같음.

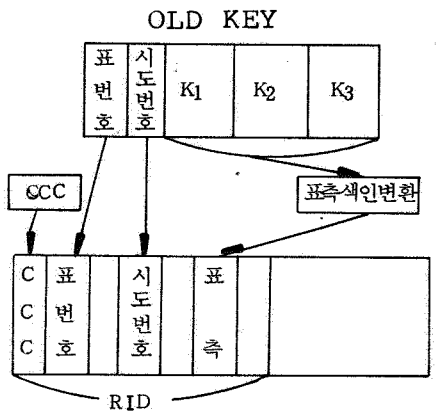


그림 7-32 표측색인의 변환

이상과 같이 분활된 file을 제1분활 부터 순차로 인쇄하여 집계표를 생산할 수 있고, tape 장치 등의 부족으로 분할의 개별 file 작성이 어려울때는 하나의 tape에 수록. sort해야 하는데 표측색인이 부여된 경우에는 sort의 순차처리가 어렵고, 표두색인이 있을 때에는 sort 자체가 불가능한 바 이러한 경우에는 그림 7-33과 같은 형식으로 output 하고 인쇄할 때에 sort를 key field에서 삭탈한다.

이상으로 통계집계 system 설계전반에 대하여 살펴 보았으나 이를 근거로 처리 flow를 정리하면 그림 7-34와 같다.

기능별 집계 system에서는 data check program 이외는 기능적으로 단순하여 다량의 program 수에 비하여 program 작성이 용이하고 범용화된 program package의 원용으로 거의 program을 작성하지 않아도 됨은 물론 기능분리형 system에서 범용 program을 간편하게 작성할 수 있는 장점이 있음을 강조하면서 마무리하고저 한다.

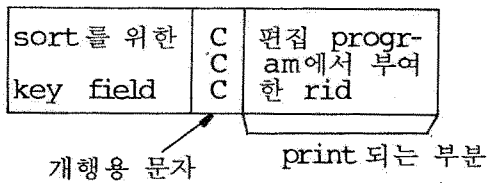


그림 7-33 old key의 제거

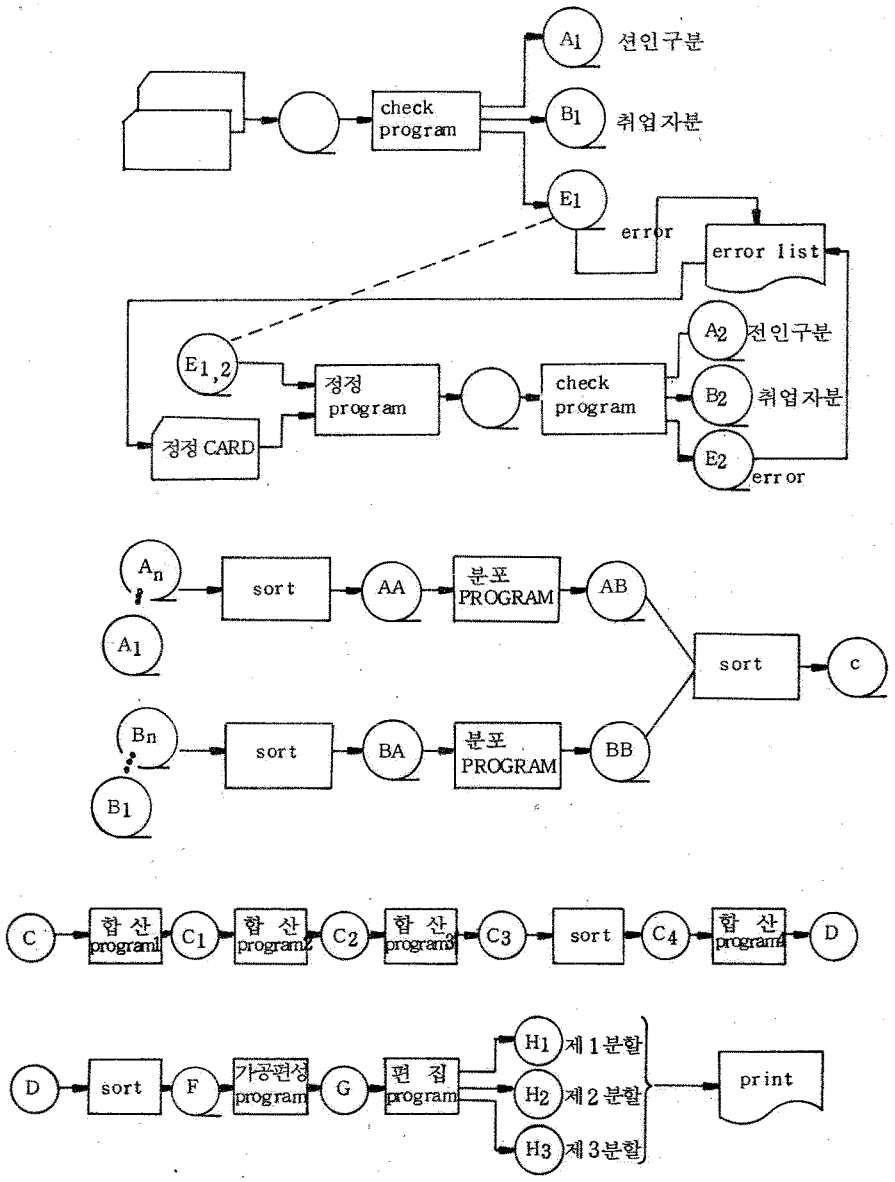


그림 7-34 통계집계 system의 처리 flow