



# 로그 램 편 램 (PL/I) 제 2 부

경 제 기 획 원  
조 사 통 계 국



통계청자료실



B0000514

— 차 레 —

서문 (序文)	5
구문과 규칙에 관한 설명 (構文과 規則에 關한 說明, Rules and Syntactic Descriptions)	9
제1 장 : 구문 표시법 (構文 表示法, Syntax Notation)	11
제2 장 : EBCDIC 와 카드-천공 부호의 문자 조 (EBCDIC 와 Card-Punch 符號 (Code) 의 文字 組 (Character Set))	15
대조 순서의 60-문자 조 (對照 順序의 60-文字 組, Collating Sequence)	15
대조 순서의 48-문자 조 (48-Character Set)	18
제3 장 : 열쇠말과 열쇠말 준말 (Key words and Key word Abbreviations)	23
제4 장 : 모양 지정 문자 (模樣 指定文字, Picture Specifi- cation Characters)	31
제1 절 문자-줄 자료를 위한 문자 (文字-줄 資料를 위한 模樣文字, Picture Characters for Character- String Data)	32
제2 절 수치 문자 지정 (數值 文字 指定, Numeric Character Specification) 을 위한 모양 문자	33
숫자나 소수점 지시자 (Digit and Decimal Point Specification) (指示子)	35
영 지우기 문자 (零, Zero Spression Character)	37
삽입 문자 (插入 文字, Insertion Character)	39
음양 기호와 유동 부호 (陰陽 記號와 流動 符號)	

B0000514-  
025764

(Sign and Currency Symbol )	41
대변, 차변, 중복 천공된 기호(貸邊, 借邊, 重複 穿孔된 記號, Credit, Debit, and Overpunched Sign )	45
지수 지정(指數 指定, Exponent Specifier )	47
영국 돈 모양(英國 돈 模樣, Sterling Picture )	48
제 5 장 : 편집 - 지시 서식 항목(編輯 指示 書式 項目, Edit-directed Format Item )	52
제 1 절 자료 서식 항목(資料 書式 項目, Data Format Item )	52
제 2 절 통제 서식 항목(統制 Control )	54
제 3 절 외판 서식 항목( Remote Format Item )	55
제 4 절 서식 항목의 사용법( 使用法 )	55
제 5 절 서식 - 항목의 영자순 열거(英字順 列舉)	55
제 6 장 : 자료 변환(資料 變換, Data Conversion )	68
제 1 절 산수 변환(算數 變換, Arithmetic Conversion)	68
제 2 절 자료의 형식 변환(資料의 形式 變換, Data Type Conversion )	72
제 3 절 고른수 올리기 값의 표( Table of Ceiling Values )	75
제 4 절 산수 연산 결과에 관한 표( Tables for Results of Arithmetic Operation )	76
제 7 장 : 내조립 함수와 유사 변수(內組立 函數와 類似 變數, Built-in Functions and Pseudo-variabl )	87
제 1 절 계산 내조립 함수(計算 內組立 函數, Computational Built-in Function )	89

줄 취급 내조립 함수 ( 줄取扱, String Handling Built-in Function )	89
산수 내조립 함수 ( 算數, Arithmetic Built-in Function )	99
수학 내조립 함수 ( 數學, Mathematical Built-in Function )	110
배열 다루기 내조립 함수 ( 配列 다루기, Array Manipulation Built-in Function )	120
제 2 절 기타의 내조립 함수 ( 其他의 內組立 函數 )	123
제 3 절 유사-변수 ( 類似-變數, Pseudo-Variabl )	126
제 8 장 : ON-조건 ( ON-條件, ON-Condition )	129
제 1 절 이 장의 구성내용 ( 이 章의 構成內容 )	131
제 2 절 계산 조건 ( 計算 條件, Computational Condition )	133
제 3 절 입력/출력 조건 ( 入力/出力 條件, Input/Output Condition )	137
제 4 절 조직 행위 조건 ( 組織 行爲 條件, System Action Condition )	142
제 9 장 : 속성 ( 屬性, Attributes )	144
제 1 절 속성의 지정 ( 屬性의 指定, Specification of Attributes )	144
제 2 절 자료 속성 ( 資料 屬性, Data Attributes )	146
문제 자료 ( 問題 資料, Problem Data )	146
프로그램 통제 자료 ( 프로그램 統制資料, Program Control Data )	147

제 3 절	입구 이름 속성 (入口 이름 屬性, Entry Name Attributes)	148
제 4 절	파일 서술 속성 (파일 記述 屬性, File Description Attributes)	148
제 5 절	범위 속성 (範圍 屬性, Scope Attributes)	149
제 6 절	기억소 종류 속성 (記憶所 種類 屬性, Storage Class Attributes)	150
제 7 절	속성의 열거 (屬性的 列舉)	150
제 10 장	문 (文, Statements)	202
부록 (附錄)	가. 파일 서술 속성과 선택항 (파일 記述 屬性和 選擇項, File Declaration Attributes and Options)	255
부록 나	: 입력/출력 문 형식 (入力/出力 文 形式, Input/Output Statement Formats)	257
부록 다	: 내조립 함수와 유사-변수의 인수와 결과에 관한 요약 (內組立 函數와 類似-變數의 結果에  관한 要約)	259
부록 라	: F-편성자와의 일치성	269
부록 마	: 용어의 정의 (用語의 定義)	275
부록 바	: PL/I 프로그램의 예	291

## 서 문

PL/I 서브세트 언어 ( PL/I Subset language ) 는 작은 용량의 자료 처리 조직에서 사용되도록 설계되었다. 이 서브세트는 자체독립이다; 이것은 프로그래머가 모체인 ( 母体인 ) PL/I 언어를 참고하지 않아도 이를 배울 수 있고 이용할 수 있다는 말이다. PL/I 의 원문을 ( 原文을 ) 수정할 수 있는 설비는 ( 이시 연산 ( 異時 演算 ) 과 편성시 편의시설과 같은 것들 ) PL/I 서브세트에 들어있지 않으나, PL/I 의 프로그램을 짜는 능력의 대부분이 포함되어 있다.

PL/I 서브세트에 그대로 옮겨진 PL/I 의 기본 특성 두가지는 블럭으로 된 구조와 기계 독립성이다. ( 앞으로는 별명이 없는 한 PL/I 서브세트를 간단히 PL/I 이라 부르겠다. ) 이들 특성은 만일 기계의 환경 또는 적용의 환경이 바뀔 때 프로그램 전체를 다시 쓰는 필요를 덜어준다.

하나의 PL/I 프로그램은 수속 블럭과 시작 블럭 ( 手續 블럭과 始作 블럭 Procedure block와 begin block ) 라 불리는 문의 덩어리로 이루어진다. 그리고 각각의 블럭은 프로그램의 한 구역을 정의한다. 수속 블럭이나 시작 블럭은 다른 블럭을 포함할 수 있다. 시작 블럭은 수속 블럭 안에 포함되어야 한다. 각개의 외부 수속은 ( 外部 수속 external procedure<sup>o</sup> ), 이것은 다른 수속에 포함되지 않는다, 따로 따로 편성된다. 동일한 외부 수속이 다른 여러개의 프로그램에 사용되어도 된다.

PL/I 은 다른 어떤 프로그램 언어보다 기계 종속성이 적다. 그렇지만 완전한 독립성이 너무 낭비가 되는 경우는 종속성을 용인하고 있다.

## [ 1 ] 이 책의 이용

이 책은 PL/I 프로그래머를 위한 참고서로 만들어졌다. 이 책은 두 부분으로 나뉘어, 이용자의 필요에 따라 빨리 찾아볼 수 있게 되어 있다.

제 I 부, " PL/I 의 개념 " ( 이는 본편의 자매서가 된다 ) 은 언어의 여러가지 설비와 그들의 상호관계를 설명하는 보기와 논술로 내용을 이루고 있다. 비교 참고를 하기 위한 필요를 줄이고 각 개 장 ( 章 ) 이 그 논제에 대한 완전한 참고로서 독립함을 허용키 위해, 어떤 사항에 대해서는 중복된다.

제 II 부, " 규칙과 구문에 관한 설명 " ( 이것은 바로 본편이 된다 ) 은 특정한 사항에 대한 신속한 인용을 제공한다. 이것은 상호관계에 관한 지식을 적게 포함하고 있으나, 특별한 문제가 신속하게 해답이 얻어지도록 구성되어 있다.

이 책에 있는 ( I, II 부 통털어 ) 구문 언어 ( 構文 言語 syntax language ) 에 관한 설명은, 이 구문 언어란 PL/I 의 요소를 서술하기 위한 것이다. 제 II 부, 제 I 장, " 구문 표시법 " 에 있다.

## [ 2 ] 실무상 고려점

이 책은 D-편성자에 관한 현 설비를 반영한다.

그렇지만, 완전한 실무 지식을 준비하려고 시도된 것은 아니다; 이 책은 IBM 조직/360 원반과 테이프 운영 조직 PL/I 프로그래머 안내서와 연합해서 사용하도록 설계되었다. 실무에 관한 논술은 언어에 관한 완벽한 설명이 요구되는 설비에 제한된다. 보기로서, ENVIRONMENT 속성에 대한 완전한 논술은 기록마디 지향 입력과 출력 화일 구성에 관한 설명으로서 본질적인 것이다.

문구 「조직/360 실무에서」로 나타낸 실무 설비(實務設備)는 IBM 조직/360에 관한 PL/I(부조이전 완전조이전(subset or full set))의 모든 실무에 적용된다.

문구 「D-편성자에서」로 나타낸 설비는 IBM 조직/360 원반과 테이프 운영 조직의 관리하에 있는 IBM D-수준 편성자(PL/I 서비스세트를 위한 것)에 별도로 적용된다.

별도의 책, IBM 조직/360 PL/I 참고서(IBM system/360 PL/I reference manual)는 IBM 조직/360 운영 조직(operating system) 관리하에 있는 F-수준 편성자(PL/I 완전조를 위한 것)에 적용되는 동일한 형식의 실무 지식을 제공한다.

다시 말하면, 전체에 통용되는 것, D-편성자에만 통용되는 것, F-편성자에만 통용되는 실무 지식이 있다는 말이다.

### [ 3 ] 관계 서적

편성, 연쇄 편집(link-edit), 프로그램을 실행하는데 필요한 지식에 대해서는, 독자는 IBM 조직/360 원반과 테이프 운영 조직, PL/I 프로그래머 안내서, Order No.GC-24-9005에 대하여 잘 알고 있어야 한다. (IBM 組織/360 円盤과 테이프 運營 組織/PL/I 프로그래머 案内書 IBM system/360 Disk and Tape Operating system, PL/I Programmer's Guide, Order No.GC 24-9005).

다음 책은 PL/I 프로그래머나 PL/I을 배우고 있는 이에게 유익한 다른 지식을 포함하고 있다.

PL/I 기초(基礎) (PL/I Primer, Order No.GC28-6808)

상업 프로그래머를 위한 PL/I 안내 (Guide to PL/I for Commercial Programmers)

FORTRAN 사용자를 위한 PL/I 안내  
( Guide to PL/I for FORTRAN Users )

才 II 部

構文과 規則에 關한 說明

(Rules and Syntactic Description)

## 제 1 장 : 구문 표시법 (構文 表示法 Syntax Notation )

이 책 전체에 걸쳐서, PL/I 문 -- 또는 요소의 결합 -- 이 논술될 때마다, 문 또는 구 (句 Phrase) 를 쓰는 방법은 정해진 표시법으로 설명되었다.

이 표시법 ( Notation ) 은 PL/I 에 한정된 것 -만은 아니다 ; 이는 어떤 프로그램 언어에서도 구문 -- 또는 구성법 -- 을 기술하는데 사용될 수 있는 표준 표시법이다. 이는 언어가 요구하는 일반적 형식에 관해서 간결하나 정확한 설명을 제공한다. 이는 언어 요소의 의미하는 바를 기술하는 것이 아니라, 그것들의 구조를 설명하는 것이다 ; 이 말은, 이것은 요소가 나타날 수 있는 ( 또는 나타나야 하는 ) 순서, 요구되는 심포 ; 허용되는 선택항을 가르킨다는 것이다.

다음 규칙은 어떤 언어에 관해서도 이 표시법의 사용법을 설명한다 ; 다만 보기는 특별히 PL/I 에만 적용된다.

1. 표시 변수는 ( 表示變數 Notation variable ) 프로그램 언어에 있는 일반적 요소의 이름이다. 표시 변수는 다음으로 이루어 진다.

a. 하위 문자 ( 下位 Lower Case Letters ) 십진숫자 ( Decimal Digits ) 하이픈 ( Hyphens ) 그리고 문자로 시작해야 한다.

b. 하위 문자와 상위 문자 ( Lower Case and upper case letters ) 의 결합, 상위 문자 부분과 하위 문자 부분은

하이픈으로 구분되어야 한다.

\* 문자가 한 부분만으로 쓰였으면 상·하위가 없으나 여기서는 하위라 하기로 약속한다.

사용된 모든 변수는 구문상 또는 의미상으로 정의된다.

보기 :

a. 숫자. 이는 0~9의 숫자의 출현을 말함.

b. 화일-이름. 이는 화일 이름으로 명명된 표시 변수의 출현을 말함.

화일 이름에 관한 보기는 다른 곳에서 주어질 것이다.

c. DO-문 이는 Do 문의 출현을 말함. 상위 문자는 어 열쇠말을 가르키도록 사용되었음.

2. 표시 상수 (表示 常数 Notation constant) 는 표현된 그 문자 자체의 출현을 말함. 표시 상수는 모두 대문자이거나 특수 문자로 된다.

보 기 :

```
DECLARE 표시어 FIXED ;
```

이는 표시 변수 # 표시어 # 화 문자가 그대로 나타나는 DECLARE, FIXED, ; 이 출현함을 말한다.

3. 용어 # 구문 단위 (構文單位 Syntactic unit) # 는 부차적 규칙에서 사용되며, 다음의 하나로 정의된다.

a. 하나의 표시 변수 또는 표시 상수,

b. 표시 변수, 표시 상수, 구문-언어 기호들의 몇개의 집합 그리고 중괄호 ( Brace ) 또는 대괄호 ( Bracket ) 로 둘러 싸인 열쇠말.

4. 중괄호 { } 는 하나의 구문 단위에서 둘 이상의 요소를 포함  
있음을 말함.

보 기 :

표식어 { FIXED  
          } FLOAT

구문 단위를 종으로 써놓은 것은 택일이 됨을 가르킨다. 위  
보기는 변수 "표식어" 는 FIXED 또는 FLOAT 중의 하나가  
뒤에 와야 한다는 것을 말한다.

5. 가로줄 | 는 택일이 됨을 가르킨다.

보 기 :

표식어 { FIXED | FLOAT }

이는 정확히 앞서의 보기와 같은 뜻이다. 두가지 방법이  
택일됨을 나타내기 위해 이 책에서 사용된다.

6. 대괄호 [ ] 는 선택항을 나타낸다.

대괄호에 있는 어떤 것도 나타나도 좋고 안 나타나도 좋다.  
대괄호는 구문 단위를 구분하는 부가적 목적에 이용될 수  
있다.

보 기 :

FILE (파일-이름) [ KEY (식) ]

이는 FILE, 괄호에 싸인 파일 이름이 나타나야 하며, 괄호에  
싸인 식이 붙은 KEY 는 나타날 수도 안 나타날 수도 있음  
을 말한다.

7. 세 점 . . . 은 바로 앞선 구문 단위가 계속해서 한번 이상 출현함을 말한다.

보 기 :

[ 숫자 ] . . .

변수 "숫자"는 대괄호로 싸여 있으므로 출현해도 좋고 안 해도 좋다. 만일 이것이 출현하면, 한번 이상 반복해도 좋다.

8. 밑줄치기 ( Under lining )가 요소와 구문 언어간에 애매성이 초래될 때 요소를 표시하기 위해 사용된다.

보 기 :

연산항 {  $\otimes$  |  $\perp$  } 연산항

이는 두개의 변수 "연산항"은 "그리고" ( " and " )

(  $\otimes$  ) 또는 "또는" ( " Or " ) (1) 중의 하나로 이어짐을 나타낸다.

표시 상수 1은 PL/I 언어의 "또는" 기호와 구문 언어 ( Syntax Language ) "또는"을 구별하기 위해 밑줄이 쳐졌다.

## 제 2 장 : EBCIDIC 와 카드 - 천공 부호의 문자 조

### [ 1 ] 대조순서의 60 - 문자 조

문 자	카드 - 천공	8 - 비트 부호
blank	천공없음	0100 0000
.	12-8-3	0100 1011
<	12-8-4	0100 1100
(	12-8-5	0100 1101
+	12-8-6	0100 1110
	12-8-7	0100 1111
ε	12	0101 0000
\$	11-8-3	0101 1011
*	11-8-4	0101 1100
)	11-8-5	0101 1101
;	11-8-6	0101 1110
7	11-8-7	0101 1111
-	11	0110 0000
/	0-1	0110 0001
,	0-8-3	0110 1011
%	0-8-4	0110 1100
1	0-8-5	0110 1101
>	0-8-6	0110 1110

문 자	카드 - 천공	8 - 비트 부호
?	0-8-7	0110 1111
:	8-2	0111 1010
#	8-3	0111 1011
@	8-4	0111 1100
!	8-5	0111 1101
=	8-6	0111 1110
A	12-1	1100 0001
B	12-2	1100 0010
C	12-3	1100 0011
D	12-4	1100 0100
E	12-5	1100 0101
F	12-6	1100 0110
G	12-7	1100 0111
H	12-8	1100 1000
I	12-9	1100 1001
J	11-1	1101 0001
K	11-2	1101 0010
L	11-3	1101 0011
M	11-4	1101 0100
N	11-5	1101 0101
O	11-6	1101 0110

내  
1  
조

문 자	카드 - 천 공	8 - 비트 부호
P	11-7	1101 0111
Q	11-8	1101 1000
R	11-9	1101 1001
S	0-2	1110 0010
T	0-3	1110 0011
U	0-4	1110 0100
V	0-5	1110 0101
W	0-6	1110 0110
X	0-7	1110 0111
Y	0-8	1110 1000
Z	0-9	1110 1001
∅	0	1111 0000
1	1	1111 0001
2	2	1111 0010
3	3	1111 0011
4	4	1111 0100
5	5	1111 0101
6	6	1111 0110
7	7	1111 0111
8	8	1111 1000
9	9	1111 1001

복 합 부 호	카드 - 천 공
<=	12-8-4, 8-6
	12-8-7, 12-8-7
**	11-8-4, 11-8-4
└<	11-8-7, 12-8-4
└>	11-8-7, 0-8-6
└=	11-8-7, 8-6
>=	0-8-6, 8-6
/*	0-1, 11-8-4
* /	11-8-4, 0-1

[ 2 ] 대조 순서의 48 - 문자 조

문 자	카드 - 천 공	8 - 비트 부 호
blANK	천 공 없 음	0100 0000
.	12-8-3	0100 1011
(	12-8-5	0100 1101
+	12-8-6	0100 1110
\$	11-8-3	0101 1011
*	11-8-4	0101 1100
)	11-8-5	0101 1101
-	11	0110 0000
/	0-1	0110 0001

문 자	카드 - 천 공	8 - 비트 부 호
,	0-8-3	0110 1011
↓	8-5	0111 1101
=	8-6	0111 1110
A	12-1	1100 0001
B	12-2	1100 0010
C	12-3	1100 0011
D	12-4	1100 0100
E	12-5	1100 0101
F	12-6	1100 0110
G	12-7	1100 0111
H	12-8	1100 1000
I	12-9	1100 1001
J	11-1	1101 0001
K	11-2	1101 0010
L	11-3	1101 0011
M	11-4	1101 0100
N	11-5	1101 0101
O	11-6	1101 0110
P	11-7	1101 0111
Q	11-8	1101 1000
R	11-9	1101 1001

문 자	카드 - 칩 공	8 - 비트 부호
S	0-2	1110 0010
T	0-3	1110 0011
U	0-4	1110 0100
V	0-5	1110 0101
W	0-6	1110 0110
X	0-7	1110 0111
Y	0-8	1110 1000
Z	0-9	1110 1001
0	0	1111 0000
1	1	1111 0001
2	2	1111 0010
3	3	1111 0011
4	4	1111 0100
5	5	1111 0101
6	6	1111 0110
7	7	1111 0111
8	8	1111 1000
9	9	1111 1001

내  
2  
조

복합부호	카드 - 천공	60-문자조 대응치
..	12-8-3, 12-8-3	:
LE	11-3, 12-5	<=
CAT	12-3, 12-1, 0-3	
**	11-8-4, 11-8-4	**
NL	11-5, 11-3	¬<
NG	11-5, 12-7	¬>
NE	11-5, 12-5	¬=
..	0-8-3, 12-8-3	;
AND	12-1, 11-5, 12-4	&
GE	12-7, 12-5	>=
GT	12-7, 0-3	>
LT	11-3, 0-3	<
NOT	11-5, 11-6, 0-3	¬
OR	11-6, 11-9	
/*	0-1, 11-8-4	/*
*/*	11-8-4, 0-1	*/*
//	0-1, 0-1	%

주: 48-문자 조를 사용할 때, 다음 규칙이 지켜져야 한다:

1. 콜론을 대신하는 두개의 쉼표는 앞에 오는 문자가 쉼표이면 빈자로 떨어져야 한다.
2. 백분비 부호를 대신하는 두개의 빗금은 만일 앞에 오는 문자가 별표이면 빈자로 떨어져야 하고, 뒤에 오는 문자가 별표이면 빈자로 떨어져야 한다.
3. 연속한 "쉼표 마침표"는 이것이 주석이나 문자줄에 나타날 때, 또는 바로 뒤에 숫자가 올때를 제외하고는 세미콜론을 나타낸다.
4. 48-문자 조가 사용될 때는, 60-문자조의 문자가 사용되어도 좋으나, 그 역은 안된다.

제 3 장 : 열쇠말과 열쇠말 준말  
 ( keywords and keyword abbreviations )

열	쇠	말	준	말	용	도
ABS ( X )					내조립	함수
ADD ( X, y, P, [ q ] )					"	
ADDR ( X )					"	
ALIGNED					속성	
ALL ( X )					내조립	함수
ANY ( X )					"	
ATAN ( X, [ y ] )					"	
ATAND ( X, [ y ] )					"	
ATANH ( X )					"	
AUTOMATIC					속성	
BACKWARDS					"	
BASED ( 지침 - 변수 )					"	
BEGIN					문	
BINARY			BIN		속성	
BINARY ( X [ . P [ , q ] ] )					내조립	함수
BIT ( 길이 )					속성	
BIT ( 값 [ , 크기 ] )					내조립	함수
BOOL ( X, Y, W )					"	
BUFFERED					속성	
BUILTIN					"	

열	최	말	준	말	용	도
BY					DO	문의 절
CALL					문	
CEIL ( X )					내조립	함수
CHAR ( 값 [ , 크기 ] )					"	
CHARACTER ( 길이 )			CHAR		속성	
CLOSE					문	
COLUMN ( W )					서식	항목
CONVERSION					조건	
COS ( X )					내조립	함수
COSD ( X )					"	
COSH ( X )					"	
DATE					"	
DECIMAL			DEC		속성	
DECIMAL ( X [ , P [ , q ] ] )					내조립	함수
DECLARE			DCL		문	
DEFINED			DEF		속성	
DIRECT					"	
DISPLAY					문	
DIVIDE ( X , Y , P [ , q ] )					내조립	함수
DO					문	
EDIT					STREAM I/O	전송 방식
ELSE					IF	문의 절
END					문	

열	쇠	말	준	말	용	도
ENDFILE					조건	
ENDPAGE					"	
ENTRY					속성 또는 문	
ENVIRONMENT			ENV		속성	
ERF ( X )					내조립 함수	
ERFC ( X )					"	
ERROR					조건	
EXP ( X )					내조립 함수	
EXTERNAL			EXT		속성	
FILE					"	
FILE ( 파일 - 이름 )					GET 와 PUT 의 선택항, RECORD I/O 문의 지정	
FIXED					속성	
FIXED ( X [ , P [ , q ] ] )					내조립 함수	
FIXEDOVERFLOW					"	
FLOAT					속성	
FLOAT ( X [ , P ] )					내조립 함수	
FLOOR ( X )					"	
FORMAT ( 서식 - 나열 )					문	
FROM					REWRITE 또는 WRITE 문의 선택항	
GET					문	
GO TO ( GOTO )					문	

열	의	말	준	말	용	도
HIGH ( i )					내조립 함수	
IF					문	
INDEX ( 줄 , 외형 )					내조립 함수	
INITIAL			INIT		속성	
INPUT					속성 , OPEN 문의 선택항	
INTERNAL					속성	
INTO ( 변수 )					READ문의 선택항	
KEY ( 파일 - 이름 )					조건	
KEY ( X )					READ와 REWRITE 문의 선택항.	
KEYED					속성	
KEYFROM ( X )					WRITE와 LOCATE 문의 선택항	
KEYTO ( X )					READ 문의 선택항	
LABEL					속성	
LINE ( W )					서식 항목 , PUT 문의 선택항	
LIST					STREAM I/O 전송 방식	
LOCATE					문	
LOG ( X )					내조립 함수	
LOG 2 ( X )					"	
LOG <del>10</del> ( X )					"	
LOW ( i )					내조립 함수	

열	최	말	준	말	용	도
MAIN					PROCEDURE	문의
					선택항	
MAX (인수)					내조립	함수
MIN (인수)					"	
MOD (XX, X <sub>2</sub> )					"	
MULTIPLY (XX, X <sub>2</sub> , P [, q])					"	
NOCONVERSION					조건	전치 포식어, CON- VERSION을 불능하게 함
NOFIXEDOVERFLOW					조건	전치 포식어, FIX- EDOVERFLOW를 불능하게 함
NOOVERFLOW					조건	전치 포식어, OV- ERFLOW를 불능하게 함
NOSIZE					조건	전치 포식어, SIZE 를 불능케 함
NOUNDERFLOW					조건	전치 포식어, UND- ERFLOW를 불능하게 함
NOZERODIVIDE					조건	전치 포식어, ZER- ODIVIDE를 불능케 함
NULL					내조립	함수
ON					문	
ONSYSLOG					PROCEDURE	문의 선택항
OPEN					문	

열	최	말	준	말	용	도
OPTIONS (나열)					PROCEDURE	문의 선택항
OUTPUT					속성, OPEN	문의 선택항
OVERFLOW					조건	
PAGE					서식 항목, PUT	문의
PAGESIZ (W)					선택항	
PICTURE			PIC		OPEN	문의 선택항
POINTER			PTR		속성	
PRECISION (X, P [, q])					"	
PRINT					내조립 함수	
PROCEDURE			PROC		속성	
PROD (X)					문	
PUT					내조립 함수	
READ					문	
RECORD					"	
RECORD (화일 - 이름)					속성	
REPEAT (출, i)					조건	
REPLY (C)					내조립 함수	
RETURN					DISPLAY	문의 선택항
RETURNS					문	
REVERT					속성	
REWRITE					문	
ROUND (x, n)					"	
					내조립 함수	

열	최	말	준	말	용	도
SEQUENTIAL					속성	
SET					READ 와 LOCATE 문의	선택항
SIGN ( X )					내조립 함수	
SIGNAL					문	
SIN ( X )					내조립 함수	
SIND ( X )					"	
SINH ( X )					"	
SIZE					조건	
SKIP [ ( X ) ]					서식 항목, PUT 문의	선택항
SQRT ( X )					내조립 함수	
STATIC					속성	
STOP					문	
STREAM					속성	
STRING ( X )					내조립 함수, 유사 변수	
STRING ( 줄-이름 )					GET 와 PUT 문의	선택항
SUBSTR ( 줄, i, j )					내조립 함수, 유사 변수	
SUM ( X )					내조립 함수	
SYSTEM					ON 문의	행위 지정
TAN ( X )					내조립 함수	
TAND ( X )					"	
TANH ( X )					"	

열	의	말	준	발	용	도
THEN					IF	문의 절
TIME					내조립	함수
TO					DO	문의 절
TRANSMIT					조건	
TRUNC ( X )					내조립	함수
UNALIGNED					속성	
UNBUFFERED					"	( X )
UNDERFLOW					조건	( )
UNSPEC ( X )					내조립	함수, 유사 변수
UPDATE					속성	
WHILE					DO	문의 절
WRITE					문	
ZERODIVIDE					조건	

## 제 4 장 모양 지정 문자 ( 模 樣 指 定 文 字 picture Specification characters

모양 지정 문자 ( 模 樣 指 定 文 字 picture specification character ) 는 PICTURE 속성에 나타난다. 이들 문자는 당해 자료 항목의 속성을 기술하는데 사용된다. 모양 지정의 대강에 관한 논술은 제1부, 제9장. " 편집하기와 줄 취급 " 에 있다.

모양 지정은 항상 문자-줄 자료 항목 또는 수치 문자 자료의 문자 표현을 기술한다. 문자-줄 모양 항목 ( characterstring pictured item ) 은 영문자, 십진 숫자, 그리고 여타의 특수 문자로 내용을 이룰수 있다. 수치 문자 모양 항목 ( numeric character pictured item ) 은 십진 숫자, 소숫점, 임의의 음양 기호, 여타의 연관 가능한 문자로 내용을 이룰 수 있다. 그러나 이들 여타의 연관 가능한 문자는 수치 문자 변수의 산수 값의 부분은 아니다. 그러나 이 변수의 문자 줄 값의 부분이 된다.

수치 문자 변수에 대입된 산수는 수치 문자 표현으로 변화된다. 영 지우기 또는 문자의 삽입과 같은 편집 작용이 수치 문자 자료 항목에 지정될 수 있다. 편집 작용은 모양 문자-줄 자료에 지정되지 못한다.

수치 모양 지정으로 선언된 변수에 대입되는 자료는 내부 규약 산수 자료이어야 한다 ( 비트 줄과 수치 문자 자료는 이들이 수치 문자 변수에 대입되기 전에 내부 규약 산수도 바뀐다 ).

이 장에 있는 그림은 어떻게 여러가지의 모양 지정이 여기에 값이 대입될 때 그 값의 표현 법을 보여준다.

각 그림은 자료의 원래의 값, 값이 꼬집어내질 변수의 속성, 모양 지정, 수치 문자 또는 모양 문자-줄 변수의 문자-줄 값을

보인다.

주: 문자-줄 값이 수치 문자 변수에 대입될 때 ( I/O 문에 의하거나 중첩 정의에 의하거나 ), 편성자는 줄에 있는 문자가 수치 문자 변수를 위해 선언된 모양 문자와 부합되는가를 검토해주지 않는다. 정확한 문자가 사용되도록 단도리를 하는 것은 이용자의 책임이다.

제 1 절 문자-줄 자료를 위한 모양 문자 ( picture characters for character-string data )

오직 X 모양 문자만이 문자-줄 항목을 지정하기 위하여 사용될 수 있다.

이는 항목내의 자리가 기계에서 허용되는 어떤 내부 비트 배치 형으로도 구성될 수 있음을 지정한다. 이 문자는 수치 문자-줄 항목에 들어가지 못한다.

그림 4 - 1 은 문자-줄 모양 지정의 보기를 보여준다. 이 보기에서, 문자 b 는 빈자를 가르킨다. 대입은 왼쪽에서부터이며, 필요한 채우기는 오른쪽이다.

원시 속성	원시 자료 (상수 풀로)	모양 지정	문자-줄 값
CHARACTER(5)	9B/2L'	XXXXX	9B/2L
CHARACTER(5)	9B/2L'	XXX	9B/
CHARACTER(5)	9B/2L'	XXXXXXXX	9B/2Lbb

그림 4 - 1 모양 문자-줄의 보기

제 2 절 수치 문자 지정을 위한 모양 문자 ( picture ch-  
aracters for numeric character speification)

수치 문자 자료는 수치 값을 나타내야 한다 ; 그러므로, 연관된 모양 지정은 문자 X 를 가질 수 없다. 수치 문자 자료를 위한 모양 문자는 자료의 상세한 편집을 위하여 지정될 수 있다.

수치 문자 변수는 두 종류의 값을 갖는다고 생각될 수 있다. 이들의 사용에 따라서, 이들은 (1) 산수 값과 (2) 문자-줄 값이 된다.

산수 값은 십진 숫자, 소수점의 위치, 음양 기호에 의해 표시되는 값이다.

수치 문자 변수의 산수 값은 변수가 산수 또는 비트-줄 식의 연산 또는 FIXED, FLOAT, BIT 의 어떤 속성을 가진 변수에 대입에서 나타나는 어디에서든지 사용된다. 이런 경우, 수치 문자 변수의 값은 내부 규약 산수 표현으로 바뀐다. 또 산수 값은 다른 수치 문자 변수에 대입으로 사용된다.

문자-줄 값은 모양 지정에 나타나는 모든 숫자와 문자로 표시되는 값이다.

그러나 문자-줄 값은 모양 문자 V 로 지정되는 지시되는 소숫점의 위치는 포함되지 않는다. 수치 문자 변수의 문자-줄 값은 변수가 문자-줄 식의 연산 또는 문자-줄 변수에 대입함에, 또는 수치 문자 변수위에 정의된 문자-줄 변수로 되는 어디에서도 사용된다.

수치 문자 지정을 위한 모양 문자는 다음 범주로 가를 수 있다.

- 숫자와 소숫점 지정자 (指定子 digit and decimal point specifiers)
- 영 지우기 문자 (zero suppression Characters)
- 삽입 문자 (插入 文字 insertion Characters)
- 음양 기호와 유동 부호 (陰陽 記号와 流動 符号 mimeric signs and currency symbol)
- 대변, 차변, 중복 천공된 기호 (貸辺, 借辺, 重複, 穿孔된 記号 Credit, debit, and over punched signs)
- 지수 지정자 (指数 指定子 exponent specifiers)
- 영국돈 모양 (英國돈 模樣 sterling picturs)

이 모양 문자는 여러 방법으로 결합해서 사용될 수 있다. 이들 결합은 지정에 의해 기술되는 자료의 형태에 의존한다. 이들 형태에 관한 논술과 기술 방법이 다음에 설명된다.

수치 문자 모양 지정은 십진 또는 영국돈 자료 중에 하나를 기술할 수 있다. 십진 수치 문자 값은 고정점수 (fixed-point) 또는 부동점수 (floating-point)가 된다. 고정점수를 위한 수치 문자 모양 지정은 하나의 난 (欄 field)으로만 되며, 이 난은 두개의 소난 (小欄 subfield)으로 될 수 있다: 소수의 왼쪽에 있는 숫자를 기술하는 고른수 소난 (整数 小欄 integer subfield), 그리고 소수점의 오른쪽에 있는 숫자를 기술하는 소수 소난 (小数 小欄 bractional subfield).

부동점수를 위한 수치 문자 모양 지정은 두개의 난으로 구성된다: 가수 난과 지수 난 (仮数 과 指数 mantissa field and exponent field).

가수 난은 고정점수 값을 기술하며, 이것은 지수에 의해 10이 제공되고 이것이 가수에 곱해지면 실 값을 나타낸다; 가수 난은 고정점수가 지정되는 것과 같은 방법으로 지정된다. 지수 난은 부호가 있을 수 있는 10의 지수를 기술한다.

영국돈 모양 지정은 세개의 난으로 이루어진다: 파운드 난 (pounds field), 실링 난 (shilling field), 펜스 난 (pence field); 펜스 난은 두개의 소난을 가질 수 있다. 영국돈 모양은 이 장의 끝에서 따로 설명될 것이다.

수치 문자 자료를 위한 모양 지정의 (picture specification) 주요 요구는 각 난은 적어도 숫자 자리를 지정하는 한개의 모양 문자를 가져야 한다는 것이다. 이 모양 문자는 숫자 문자 9가 되어야 할 필요는 없다. 영 지우기 문자 (Z 또는 \*)와 같은 9 이외의 문자도, 숫자 자리를 지정한다. 적어도 이들 문자의 하나가 수치 문자 지정을 정의하는데 사용되어야 한다. 이는 모양 문자 X를 갖지 못한다.

#### [1] 숫자와 소수점 지시자 (digit and decimal point specifiers)

모양 문자 9와 V가 고정점 십진 값을 표현키 위한 수치 문자 지정의 단순한 꼴에서 사용된다.

9 연관된 난의 자리가 십진 숫자로 이루어짐을 지정한다.

V 자료 항목에서 소수점이 이 자리에서 정해짐을 지정한다.

그러나 이는 실제의 소수점이 삽입되는 것을 지정하는 것이 아니다. 대입된 값의 고른수와 소수 부분이 V 문자를 중심으로 배열된다; 그러므로 대입된 값은 어느 끝에서나 0

숫자로 넓혀지거나 또는 어느 끝에서나 잘려질 수 있다. 만일 유의 숫자가 왼쪽에서 잘린다면, 결과는 알 수 없고 SIZE 중단이 만일 SIZE 가 가능하다면 일어날 것이다. 만일 V 문자가 고정점 십진수 값 (또는 부동점 십진 값의 가수 난) 의 모양 지정에 나타나지 않으면, 난 지정의 오른쪽 끝에 V가 있는 것으로 대행된다. 이는 필요하면 대입된 값을 정수로 되게 자를 수 있다. V 문자는 한번 밖에 나타나지 못한다. V는 모양 지정에서 소난 구분자 (区分子 delimiter) 로 생각된다. 이 것은, V 앞에 있는 부분과 이 다음에 오는 부분 (만일 있다면) 은 각각 이 지정의 소난이 되기 때문이다. V는 하나의 문자를 지정하는 것이다.

그림 4 - 2는 수치 문자 보기를 보여준다.

원 시 속 성	원 시 자 료 ( 상 수 폴 로 )	모 양 지 정	문 자 - 줄 값
FIXED (5)	12345	99999	12345
FIXED (5)	12345	99999V	12345
FIXED (5)	12345	999V99	34500
FIXED (5)	12345	V99999	00000
FIXED (7)	1234567	99999	34567
FIXED (3)	123	99999	00123
FIXED (5,2)	123.45	999V99	12345
FIXED (7,2)	12345.67	9V9	56
FIXED (5,2)	123.45	99999	00123

## 그림 4 - 2 모양 수치 문자의 보기

### [2] 영 지우기 문자 ( Zero suppression characters )

영 지우기 모양 문자는 문자-줄 값에서 조건적 숫자 자리를 지정하며 앞서는 영을 ( $\emptyset$ 을) 별표나 빈자로 바꿔 놓는다. 앞서는 영 ( Leading Zero )이란 (1) 고정점 수의 가장 왼쪽 자리 또는 부동점수의 두개 부분의 가장 왼쪽 숫자 자리들에 나타난 영들이거나, (2) 소숫점 자리의 왼쪽, (3) 숫자 1부터 9의 어느것도 앞서지 않은 영을 말한다. 부동점수는 지수 난에서도 앞서는 영을 가질 수 있음을 유의하라.

Z 조건적 숫자 자리를 지정하며 당해 자리에 있는 앞서는 영을 빈자로 바꿔놓게 한다. 당해 자료 자리가 앞서는 영을 갖지 않으면, 이 자리에 있는 숫자는 빈자로 바뀌지지 않는다. 모양 Z는 모양 문자 \*가 있는 난에 동시에 나타나지 못한다. 또 이는 부동 모양 문자 (浮動 模樣 文字 Drifting picture character) 의 오른쪽 또는 9, T, I, R의 오른쪽에 나타나지 못한다.

\* 이는 조건적 숫자 자리를 지정하며 모양 문자 Z가 사용되는 방법으로 사용되며, 앞서는 영을 별표로 바꿔놓는 것이 다름뿐이다.

모양 문자 \*는 동일 난에서 모양 문자 Z와 함께 나타나지 못하고, 부동 모양 문자의 오른쪽 또는 9, T, I, R의 오른쪽에 나타나지 못한다.

주: 모양 문자 V는 오른쪽에 있는 모든 숫자 자리가 영 지우

기 모양 문자로 되어 있으면, 만일 소수부에 있는 모든 자리가 영이며 모든 고른수 자리가 지워졌다면, 이 값의 소수에 있는 영은 지워질 것이다. 그렇다면, 이 자료 항목의 전체 문자-줄 값은 빈자나 별표로 이루어진다.

만일 소수 부분이 어떤 유의 숫자를 가진다면 소수 부분에 있는 숫자는 빈자 또는 별표로 바뀌지 않을 것이다.

그림 4-3은 영 지우기 문자의 사용법의 보기를 보여준다. 그림에서, 문자 b는 빈자를 가르킨다.

원시 속성	원시 자료 (상수 꼴호)	모양 지정	문자-줄 값
FIXED(5)	12345	ZZZ99	12345
FIXED(5)	00100	ZZZ99	bb100
FIXED(5)	00000	ZZZ99	bbb00
FIXED(5)	00100	ZZZZZ	bb100
FIXED(5)	00000	ZZZZZ	bbbbbb
FIXED(5,2)	123.45	ZZZ99	bb123
FIXED(5,2)	001.23	ZZZV99	bb123
FIXED(5)	12345	ZZZV99	34500
FIXED(5)	00000	ZZZVZZ	bbbbbb
FIXED(5)	00100	*****	**100
FIXED(5)	00000	*****	*****
FIXED(5,2)	000.01	*****	**01

그림 4-3. 영 지우기의 보기

### [3] 삽입 문자 ( Insertion characters )

모양 문자 쉼표 ( , ), 점 ( . ), 빈자 ( B ) 는 삽입 문자이다.

이들은 지정된 문자를 당해 자리에 삽입된다. 이들은 숫자 자리를 가리키지 않고, 숫자 사이에 끼워진다. 각 문자는 이것이 지워지건 아니건 간에 문자 자리를 가르킨다.

쉼표 ( comma ) 와 점 ( point ) 는 조건적 삽입 문자이다 ; 영 지우기 문자 내에서는, 이들 역시 지워질 수 있다. 빈자 ( blank ) 는 부동 문자와 연결되었을 때 조건적 삽입 문자가 된다, 그 밖에는 이는 무조건적 삽입 문자이다. 이는 빈자 또는 부동 문자가 당해 자리에 나타남을 말한다.

주 : 삽입 문자는 문자-줄 값으로만 이용된다. 이들은 자료 항목의 산수 값에 대해서는 영향을 주지 않는다.

이 는 쉼표를 영 지우기 작용이 안 일어났을 때 당해 자리에 삽입시킨다. 만일 영 지우기가 일어났을 때, 쉼표는 쉼표 자리의 왼쪽에 지워지지 않은 숫자가 있을 때,

또는 V가 이것의 왼쪽에 나타나며 소수 부분이 어떤 유의 숫자를 가질 때는 당해 자리에 삽입된다. 그 외의 영 지우기가 일어났을 경우, 세 문자 중에 하나가 쉼표 자리에 삽입된다. 쉼표 대신에 놓아지는 문자는 쉼표 앞에 오며 숫자 자리를 지정한 첫번째 모양 문자에 좌우된다 :

- 만일 이 문자 자리가 별표이면, 쉼표 자리는 별표로 된다.
- 만일 이 문자 자리가 부동 음양 기호 ( Drifting sign ) 또는 부동 유동 부호 ( Drifting Currency symbol ) ( 뒤에 나옴 ) 이면, 부동 문자 줄이 쉼표 자리를 포함하고 있는 것으

로 간주되며, 그래서 일어나는 행위는 부동 문자의 그것과 같다.

- 만일 이 문자 자리가 별표도 부동 문자도 아니면, 쉼표 자리는 빈자로 된다.
  - 이는 쉼표 모양 문자가 사용되는 같은 방법으로 사용된다, 다른점은 점 ( . ) 이라는 것이다. 이 문자는 고정점 십진수의 모양 지정에서 소수점으로 작용하지 않으며 자료 항목의 산수 값의 일부가 아니다. 그런 기능은 모양 문자 V에 의해서만 수행된다. V가 없으면, 이는 난의 최오른쪽 숫자 자리의 오른쪽에 있는 것으로 대행된다. 그리고 소수점은 이것에 따라 운용되며, 점 삽입 문자가 다른 곳에 나타나는 경우도 같다.
- B 이는 빈자가 수치 문자 난의 문자-줄 값의 당해 자리에 삽입됨을 지정한다.

점을 (또는 쉼표) V와 연결해서 쓰면, 정수와 소수사이에 나타나므로 소수점의 역할을 하게된다. (인쇄의 경우에). 만일 이것이 V 바로 앞에 오면, V의 왼쪽에 유의 숫자가 올 때만 이것이 나타난다. 만일 이것이 V의 바로 뒤에 오면, V의 오른쪽 숫자 전부가 지워지면 이것도 지워질 것이다.

삽입 문자 B, 쉼표 (Comma), 점 (point)는 숫자 자리 앞에 와야 한다.

그림 4 - 4는 삽입 문자의 쓰임을 보여준다.

원시 속성	원시 자료 (상수 플로서)	모양 지정	문자-출 값
FIXED(4)	1234	9,999	1,234
FIXED(6,2)	1234.56	9,999V.99	1,234.56
FIXED(4,2)	12.34	ZZ.VZZ	12.34
FIXED(4,2)	00.03	ZZ.VZZ	bbb03
FIXED(4,2)	00.03	ZZV.ZZ	bb.03
FIXED(4,2)	12.34	ZZV.ZZ	12.34
FIXED(4,2)	00.00	ZZV.ZZ	bbbbbb
FIXED(4,2)	67.89	9,999,999.V99	0,000,067,89
FIXED(7,2)	12345.67	**,999V.99	12,345,67
FIXED(7,2)	00123.45	**,999V.99	***123.45
FIXED(9,2)	1234567.89	9.999.999V,99	1.234.567,89
FIXED(6)	123456	99.999.9	12.345.6
FIXED(6)	001234	ZZ,ZZ,ZZ	bbb12,34
FIXED(6)	000000	ZZ,ZZ,ZZ	bbbbbbbb
FIXED(6)	000000	**,**,**	*****
FIXED(6)	123456	99B99B99	12b34b56
FIXED(3)	123	9BB9BB9	1bb2bb3

그림 4 - 4 . 삽입 문자의 보기

[4] 음양 기호와 유동 부호 (Signs and currency symbol.)

모양 문자 S, +, -는 음양 기호를 지정한다. 모양 문자 \$

는 유동 부호이다.

이들 모양 문자는 고정적 ( 固定的 static ) 또는 부동적 ( 浮動的 drifting ) 방식으로 사용될 수 있다. 부동 문자 ( drifting character ) 는 영 지우기 문자와 비슷하다. 그러나, 한개의 부동 문자가 항상 ( 전체의 난이 지워지지 않는 한 ) 부동 문자 줄 끝에 또는 첫째 유의 숫자 바로 왼쪽에 남아 있다.

이들 문자의 고정적 사용은 음양 기호, 유동 부호, 또는 빈자가 항상 당해 자리에 남아 있음을 지정하는 것이다.

부동적 사용은 앞서서 영음·지음을 지정하는 것이다. 이 경우, 이 모양 문자와 연관되어 지워진 가장 오른쪽 자리는 음양 기호, 빈자, 또는 유동 부호로 될 것이다. 부동 문자는 줄로서 나타나야 한다. 줄이란 V 와 삽입 문자 쉼표, 점, 또는 B 중의 하나가 포함될 수 있는 부동 문자의 연속을 말한다. 줄의 마지막 부동 부호 다음에 오는 삽입 문자 쉼표, 점, 또는 B 의 어떤 것도 부동 문자 줄의 일부로 간주된다. 그러나, 뒤에 오는 V 는 부동 문자 줄을 끝내고 이 줄의 일부가 아니다. 부동 문자 줄은 숫자 자리, 삽입, 문자, 또는 V 의 뒤에 오지 못한다. 부동 문자 줄이 어떤 난에 나타나면, 영 지우기 문자 ( Z 또는 \* ) 는 동일 난에 나타나면 안된다.

쉼표, 점, B 가 있는 부동 문자 줄의 자리는 다음의 하나가 될 것이다.

- 만일 유의 숫자가 왼쪽에 나타나면 쉼표, 점, 빈자
- 만일 오른쪽의 자리가 그 난의 가장 왼쪽 유의 숫자가 된다면, 부동 문자

만일 그 난의 가장 왼쪽 유의 숫자가 오른쪽으로 둘째 자리 이상이면, 빈자 만약 부동 문자 줄이 부동 문자를  $n$  개 가지면, 이 줄은  $n - 1$ 의 조건적 숫자 자리가 된다. 부동 문자의 가장 왼쪽과 연관된 자리는 부동 문자나 빈자만이 될 수 있고, 숫자는 안된다.

만일 부동 문자 줄이 어떤 난에 지정되면, 다른 부동 문자는 한번만 나타날 수 있으며, 그 다른 문자는 고정적인 음양 기호 또는 유동 부호를 말한다.

단지 한가지의 음양 문자만 한 난에 나타날 수 있다.

고정적 문자로 사용되는 한개의  $s$ ,  $+$ ,  $-$ 는 부동점수 지정의 가수와 지수난의 왼쪽과 고정점수 지정의 왼쪽 또는 오른쪽에 나타날 수 있다.

만일 부동 문자줄이 그 안에  $V$ 를 포함하고 있으면,  $V$ 는 앞쪽 부분을 소난으로서 구분하고,  $V$  뒤에 오는 모든 숫자 자리는 두번째 소난을 시작하는 부동 문자 줄의 일부가 되어야 한다.

$V$  뒤에 오는 모든 숫자 자리가 부동 문자로 되어 있으면, 소난에서의 영 지우기는 정수와 소수가 모두 영일 경우에만 일어날 것이다. 그러면 그 결과는 빈자가 될 것이다. 만일 유의 소수 숫자가 있다면, 전체 소수 부분은 지워지지 않은 채로 나타날 것이다.

이는 유동 부호 (Currency symbol) 임을 지정한다. 만일 이 문자가 두번 이상 나타나면, 이는 부동 문자 (drifting character)가 된다; 아니면 고정적 문자 (static character)가 된다.

racter)가 된다.

• 고정적 문자는 그 문자가 당해 자리에 놓인다. 고정적 문자는 지정된 난의 제일 왼쪽 또는 제일 오른쪽에 있어야 한다. 이 문자의 부동 사용법에 관해서는 앞을 보라.

S 이는 만일 자료 값이  $\geq 0$ 이면 양 기호 문자 (+), 아니면 음 기호 문자 (-)를 지정한다. 이 문자는 부동적 또는 고정적으로 사용될 수 있다. 이 규칙은 유동 부호의 그것과 같다.

+ 이는 만일 자료 값이  $\geq 0$ 이면 양 기호 문자 (+), 아니면 빈자가 됨을 지정한다. 이 문자는 부동적 또는 고정적으로 사용될 수 있다. 그 규칙은 유동 부호의 그것과 같다.

- 이는 만일 자료 값이  $\geq 0$ 이면 음 기호 문자 (-), 아니면 빈자로 됨을 지정한다.

주: 문자 +, -, S, ~~S~~ 전후에 있는 \*를 가진 모양은 변수가 0의 값을 가질 때 이들 문자를 \*로 바꿔 놓는다.

그림 4-5는 부동 모양 문자의 사용법의 보기를 보여준다.

원시 속성	원시 자료 (상수 풀로서)	모양 지정	문자-출 값
FIXED(5,2)	123.45	<del>S</del> 999V.99	<del>S</del> 123.45
FIXED(5,2)	001.23	<del>S</del> ZZZV.99	<del>S</del> bb1.23
FIXED(5,2)	000.00	<del>S</del> ZZZV.ZZ	bbbbbb b
FIXED(5,2)	123.45	<del>S</del> <del>S</del> 9V.99	<del>S</del> 123.45
FIXED(5,2)	001.23	<del>S</del> <del>S</del> 9V.99	bb <del>S</del> 1.23

원시 속성	원시 자료 (상수 플로서)	모양 지정	문자-출 값
FIXED(5,2)	012.00	998	128
FIXED(2)	12	<del>888</del> , 999	bb b8012
FIXED(4)	1234	<del>888</del> , 999	b81,234
FIXED(5,2)	123.45	S999V.99	+123.45
FIXED(5,2)	-123.45	S999V.99	-123.45
FIXED(5,2)	-123.45	+999V.99	b123.45
FIXED(5,2)	123.45	-999V.99	b123.45
FIXED(5,2)	123.45	999V.99S	123.45+
FIXED(5,2)	001.23	+++B+V.99	bbb+1.23
FIXED(5,2)	001.23	---9V.99	bbb1.23
FIXED(5,2)	-001.23	SSS9V.99	bb-1.23

그림 4-5 부동 모양 문자의 보기

[5] 대변, 차변, 중복천공된 기호

( credit, debit, and overpunched signs )

문자의 짝 CR (credit) 와 DB (debit) 는 부동점 수치 문자 자료 항목의 기호를 지정하며 보통 사무용 보고서 형식에 쓰인다.

모양 문자 T, I, R 는 고정점 수치 문자 자료 항목의 당해 숫자 자리에 중복천공된 음양 기호를 지정한다. 만일 이 숫자 자리가 빈자이면, 값은 양의 영으로 대행된다. 중복천공 음양 기호는 숫자와 한자리에 천공되는 12-천공 (양일 때) 또는 11-

천공 (음일 때)이다. 이는 산수 자료의 음양 기호를 가리킨다.

중복천공 문자는 난의 마지막 숫자 자리에만 나타날 수 있다.  
CR 이는 만일 자료 값이 영보다 작으면 당해 자리가 문자 CR로 됨을 말함. 그렇지 않으면, 그 자리는 두개의 빈자로 된다. 문자 CR은 난의 오른쪽 뒤에만 나타날 수 있다.  
DB 이는 당해 자리에 문자 DB가 나타나는 것 외는 CR과 같은 식이다.

주: CR 또는 DB 전후에 \*가 있는 모양은 이 변수가 영의 값을 가지면 \*\*로 바뀌진다.

T 이는 입력에서 당해 자리가 음양 기호가 중복 천공이되어 있음을 지정한다. 이는 또 중복천공이 문자줄 값에서 이루어짐을 지정한다.

I 이는 입력에서 당해 자리가 값이  $\geq 0$ 이면 +로 중복 천공됨을 지정한다; 아니면, 이는 중복 천공이 안된다. 이는 또 만일 값이  $\geq 0$ 이면 중복 천공이 문자줄 값에서 이루어짐을 지정한다.

R 이는 입력에서 만일 값이  $< 0$ 이면 -로 중복 천공됨을 지정한다; 아니면, 이는 중복 천공이 안된다.

이는 또 만일 값이  $< 0$ 이면 중복천공이 문자 줄 값에서 이루어 짐을 지정한다.

주: 모양 문자 CR, DB, T, I, R는 동일 난에서 다른 어떤 음양 기호와도 사용되어서는 안된다.

문자 T, I, R로 된 숫자는 인쇄가 안될지도 모른다. 만일 이런 경우면, 빈자가 이 숫자 대신에 나타날 것이다.

그림 4-6은 CR, DB, 중복천공 문자의 보기를 보여준다.

원시 속성	원시 자료 (상수 폴로서)	모양 지정	문자-출 값
FIXED(3)	-123	<del>S</del> Z.99CR	<del>S</del> 1.23CR
FIXED(4, 2)	12.34	<del>S</del> ZZV.99CR	<del>S</del> 12.34bb
FIXED(4, 2)	-12.34	<del>S</del> ZZV.99DB	<del>S</del> 12.34DB
FIXED(4, 2)	12.34	<del>S</del> ZZV.99DB	<del>S</del> 12.34bb
FIXED(4)	1021	999I	102A
FIXED(4)	-1021	Z99R	102J
FIXED(4)	1021	999T	102A

그림 4-6 CR, DB, T, I, R 모양 문자의 보기

[6] 지수 지정 (指数 指定 exponent specifiers)

모양 문자 K와 E는 부동점 십진수를 기술하는 수치 문자 지정의 지수 난을 구분한다. 지수난은 항상 뒤쪽이며 K와 E는 동일 지정에서 나타나지 못한다.

K 이는 지수난이 이 자리의 오른쪽에 나타남을 지정한다. 이는 수치 문자 자료 항목에서 하나의 문자를 지정하는 것이 아니다.

E 이는 당해 자리가 문자 E로 되며, 지수 난의 개시를 지정한다.

지수 값은 문자-줄 값에서 첫째 난(가수 mantissa)의 첫번째 유의 숫자가 이 지정의 첫번째 숫자 지정자(specifier) 자리에 나타나도록(이것이 영 지우기 문자일지라도) 조정된다.

그림 4-7은 지시 구분자(区分子 delimiters)의 용법의 보기를 보여준다.

원시 수성	원시 자료 (상수 플로서)	모양 지정	문자-줄 값
FLOAT(5)	.12345E06	9V.99999E99	1.23450E05
FLOAT(5)	.12345E-06	9V.99999ES99	1.23450E-07
FLOAT(5)	.12345E+06	9V.99999KS99	1.2345+05
FLOAT(5)	-123.45E12	S999V.99ES99	-123.45E+12
FLOAT(5)	001.23E-01	SSS9.V99ES99	+123.00Eb-3
FLOAT(5)	001.23E+04	ZZZV.99KS99	123.00+02
FLOAT(5)	001.23E+04	SZ99V.99ES99	+123.00E+02
FLOAT(5)	001.23E+04	SSSSV.99E-99	+123.00Eb02

그림 4-7 부동점수 모양 문자의 보기

[7] 영국돈 모양 (sterling picture)

다음 모양 문자는 영국돈 자료를 위한 모양 지정에서 사용된다.

y 이는 앞서는 영 또는 앞서지 않는 영이든 영을 빈자로 바꿔놓음을 지정한다. 당해 자리가 영이 아니면, 빈자로 바뀌지 않는다.

- 8 이는 BSI 단문자 (單文字 single-character) 로 표현으로 되는 실링 숫자 자리를 지정한다. 10 실링은 12-천공 (ε) 로 표현되고 1.1부터 19까지의 실링은 각각 A부터 I까지의 문자로 표현된다.
- 7 이는 BSI 단문자로 표현되는 펜스 (pence) 숫자 자리를 지정한다. 10 펜스는 12-천공 (ε) 으로 표현되고 11 펜스는 11-천공 (-) 으로 표현된다.
- 6 이는 BSI 단문자로 표현되는 펜스 숫자 자리를 지정한다. 10 펜스는 11-천공 (-) 으로 표현되고 11 펜스는 12-천공 (ε) 으로 표현된다.
- P 이는 당해 자리가 펜스 문자 (pence character) D 를 가짐을 지정한다.
- G 이는 영국돈 모양의 개시를 지정한다. 이는 수치 문자 자료 항목에서 한개의 실 문자를 지정하는 것이 아니다.
- H 이는 당해 자리가 실링 문자 S 를 가짐을 지정한다.
- M 이는 한개 난의 개시를 지정한다.  
 . 이는 수치 문자 자료 항목에서 한개의 실 문자를 지정하는 것이 아니다.

영국돈 자료 항목은 고정점 십진 자료 항목으로 간주된다. 산수 연산에 참가할 때는, 이들은 고정점수의 펜스로 나타낸 값으로 변화된다. 영국돈 모양은 다음의 일반 형식을 가진다.

PICTURE \ G [ 편집 - 문자 - 1 ] . . .

M 파운드-난

M [ 분리자 - 1 ] . . . 실링-난

M [분리자 - 2] . . . 펜스-난

[편집-문자-2] . . .

"편집 문자 1"은 다음의 고정적 모양 문자의 하나 이상이 될 수 있다.

8, +, -, S

"파운드 난"은 다음의 모양 문자로 될 수 있다.

Z, y, \*, 9, #, 8, +, -, S

마지막 4 문자 (8, +, -, S)는 부동 문자이어야 한다. 쉽 표는 삼입 문자로 사용될 수 있다.

"분리자 1"은 다음의 모양 문자의 하나 이상으로 될 수 있다.

l, ., B

"실링 난"은 다음과 같이 될 수 있다.

{ 99 | yy | ZZ | y9 | z9 | yZ | 8 }

모양 문자 Z는 만일 이 문자의 왼쪽에 있는 난 전체가 (파운드 난을 포함해서) Z나 y 이외의 다른 숫자 자리를 가지지 않을 때만 나타날 수 있다.

"분리자 2"는 다음 모양 문자의 하나 이상이 될 수 있다.

l, ●, B, H

"펜스 난"은 다음의 형식을 취한다.

{ 99 | yy | ZZ | y9 | z9 | yZ | 7 | 6 }  
[ [ V | V . | . V ] 9 | Z | y ] . . .

마지막 9는 다음의 하나로 대체될 수 있다.

T, I, R

"편집 문자 2"는 다음이 될 수 있다.

1. 하나의  $\text{\textcircled{X}}$  그리고 / 또는 P, 또는
2. 하나 이상의 문자 B와 결합한 하나의  $\text{\textcircled{X}}$  그리고 / 또는 하나의 P, 또는
3. 위에 배치와 결합한 하나의 CR, DB, +, -, S  
파운드, 실링, 펜스 난은 각각 적어도 하나의 숫자 자리를 가져야 한다.

영 지우기는 전체의 지정에 기초해서 수행된다. 즉 난의 각각에 독립하지 않는다. 만일 개개의 난에 있는 영이 빈자로 바뀌게 하려면, 모양 문자 y가 사용되어야 한다. 분리자 빗금 (slash /), 점 (point •), H는 만일 소난에 있는 이들의 왼쪽 모든 문자가 빈자일 때만 자워진다. 하나의 영국돈 지정에서, 음양 기호는 하나만 있어야 한다. 이 음양 기호는 "편집 문자 1", 펜스 난에 있는 T, I, R, "편집 문자 2", 또는 펜스난의 부동 문자 줄에 의해 지정될 수 있다.

그림 4-8은 영국돈 모양 지정의 용법의 보기를 보여준다.

원시 속성	원시 자료 (펜스 값으로)	모양 지정	문자 줄 값
FIXED(4)	0534	GMZM.8M.99V .9CR	b2.4.06.0bb
FIXED(4)	0019	GMZM.ZZM. ZZp	bb.b1.07b

그림 4-8 영국돈 모양 지정의 보기

## 제 5 장 : 편집 - 지시 서식 항목 (編輯 指示 書式 項目 EDIT-directed format item)

이 장은 GET나 PUT 문의 서식 나열 (Format list) 에 나타날 수 있는 편집 - 지시 서식 항목의 각개를 설명한다.

세개 유형의 (類形的) 서식 항목이 있다. 자료 서식 항목 (資料 書式 項目 Data format items), 통제 서식 항목 (統制 書式 項目 Control format items), 외딴 서식 항목 (외딴 書式 項目 Remote format item).

이 장에서 세개의 유형이 각각 논술되고 서식 항목이 각 유형에 나열된다. 서식 항목에 관한 자세한 설명을 영자 순으로 배열되었다.

### 제 1 절 자료 서식 항목 (Data format items)

자료 서식 항목이란 한개의 자료 항목의 외부 형식 (External format) 이다.

입력에서, 흐름의 자료는 문자들의 연속된 줄이라 볼 수 있다; 모든 빈자와 따옴표도 문자로 취급된다. GET 문에 있는 각개 자료 서식 항목은 줄로부터 취득될 문자 수를 지정하며 그들 문자가 해석되는 방법을 기술한다. 줄은 따옴표에 둘러싸이면 안되고 비트 줄을 표시하기 위해 문자 B를 사용해서는 안된다.

출력에서 줄에 있는 자료는 서식 나열에 의해 지정된 형식 (form) 을 취한다. PUT 문에 있는 각개 자료 서식 항목은 문자 형식 (Character form) 으로된 당해 자료 항목이 놓이지

는 난의 넓이를 지정하며 값이 가질 형식을 기술한다. 둘러 싸는 따옴표 ( apostrophes )는 삽입되지 않으며, 비트 줄을 가르키는 문자 B도 없다.

앞서는 빈자는 출력 줄에서 자료 항목을 분리하기 위해 자동으로 삽입되지 않는다. 줄 자료는 넓이가 지정된 난에서 왼쪽 맞추기가 된다. 산수 자료는 오른쪽 맞추기다. 산수 자료인 경우 지정된 난의 넓이가 그들(수)을 수용하는 한 앞서는 빈자는 나타나지 않을 것이다. 불충분한 난 넓이 지정에 의한 자르기는 ( Truncation ) 산수 항목에서 왼쪽, 줄 항목에서는 오른쪽에서 이루어진다.

이 진수 자료의 값은 입력과 출력 모두 십진수 형식으로 표현됨을 유의하라.

다음은 자료 서식 항목의 열거이다.

고 정 점 수 서 식 항 목 (Fixed-point format item)	F ( 지 정 )
부 동 점 수 서 식 항 목 (Floating-point format item)	E ( 지 정 )
비 트 - 줄 서 식 항 목 (Bit-string format item)	B ( 지 정 )
문 자 - 줄 서 식 항 목 (Character-string format item)	A ( 지 정 )

체 2절 통제 서식 항목 (統制 書式 項目 Control format items)

통제 서식 항목은 입력과 출력 화일 (File)에 적용된다. 이들은 외부 매체로부터 들어오는 또는 나가는 서식 항목 지면의 (紙面의) 체제 만들기를 지정한다.

다음은 통제 서식 항목의 열거이다.

지면 넘기기 서식 항목 (Paging format item)	PAGE (PRINT 화일에만)
줄 건너기 서식 항목 (Line skipping format item)	SKIP (지정) (PRINT 와 비-PRINT 화일에)
줄 자리 서식 항목 (Line position format item)	LINE (지정) (PRINT 화일에만)
자리 서식 항목 (Column position format item)	COLUMN (지정) (PRINT 와 비-PRINT 화일에)
자리 띄기 서식 항목 (Spacing format item)	X (지정) (PRINT 와 비-PRINT 화일에)

통제 서식 항목은 이것이 자료 나열이 다 소비되기 전에 오지 않으면 효과가 없다.

서식 항목이 서식 나열에서 감지될 때만 실행되는 반면 PUT 문의 선택 항목은 어느 자료도 전송되기 전에 실행된다는 점을 제외하고는, PAGE, SKIP, LINE 서식 항목은 PUT 문의 당해 선택 항목의 효과와 같다.

### 제 3 절 외판 서식 항목 (Remote format item)

외판 서식 항목 (Remote format item)은 외판 서식 항목과 대치될 서식 나열로 구성된 FORMAT 문의 명찰 (Label)을 지정한다.

외판 서식 항목은 다음과 같다.

R (문-명찰-지시자) [ R (Statement-Label-Designator) ]

“문-명찰-지시자 (文-名札-指示子)”란 명찰이거나 첨자 없는 요소 명찰 변수이다.

### 제 4 절 서식 항목의 사용법

앞에서 열거된 것 중에서 PAGE와 외판 서식 항목을 제외하고는, “지정”은 하나 이상의 식으로 될 수 있다. 그런 식은 십진 고른 상수이어야 한다.

보기: F(4), A(5), SKIP(2), COLUMN(60)

### 제 5 절 서식 항목의 영자순 열거

#### (1) A 서식 항목

A 서식 항목은 :

### A [ (난 - 넓이) ]

문자-줄 서식 항목은 문자 줄의 외부 표현을 기술한다. 이는 문자 줄에만 사용될 수 있다. 문자 줄은 다른 어떤 서식 항목으로도 전송되지 못한다. 변환이 수행되지 않음.

일반 규칙:

1. "난-넓이" ("Field-Width") (때로는 W로 표시되기도 함)는 0보다 크고 256보다 작은 부호 없는 십진 고른 상수 (Decimal integer constant) 이어야 한다. 이는 줄 (String)을 포함하는 (또는 포함할) 자료 흐름에서 문자 자리수를 지정한다.
2. 입력에서, 지정된 문자 수는 자료 흐름에서 얻어지며 자료 나열에 있는 연관된 요소에 매입된다. 난의 넓이는 입력에서 언제나 요구된다. 만일 따옴표가 흐름에 있으면, 그들은 줄에 있는 문자로 취급된다.
3. 출력에서, 난의 넓이는 반드시 지정될 필요는 없다: 이 경우에, 연관된 줄의 길이가 사용되며, 이 자료 항목은 그 난을 완전히 채운다. 둘러 싸는 따옴표는 삼입되지 않는다.

### (2) B 서식 항목 (B Format item)

B 서식 항목은:

#### B [ (난 - 넓이) ]

비트-줄 서식 항목은 비트 줄의 외부 표현 (External Representation)을 기술한다. 개개 비트는 문자 0과 1로 표현된다. 이 서식 항목은 비트 줄에만 사용될 수 있다: 비트 줄은 다

른 어떤 서식 항목으로도 전송되지 못한다.

일반 규칙 :

1. "난-넓이"는 0보다 크며 65보다 작은 무부호 십진 고른 상수이어야 한다. 이는 비트 줄을 포함하는 (또는 포함할) 자료 흐름에서 문자 수를 지정한다.
2. 입력에서, 비트 줄의 문자 표현은 지정된 난의 어디에 있어도 좋다. 첫번째 빈자 아닌 문자가 감지되면, 이 문자부터 (다음 문자가 빈자전 아니건) 목적 난 (Target field)에 있는 비트 수에 이를 때까지 변환된다. 비트 줄의 전후에 나타날 수 있는 빈자는 무시된다. 난의 넓이는 입력에서 언제나 요구된다. 0 또는 1 이외의 어떤 문자도 (삽입되어 있는 빈자, 따옴표, 문자 B들도) Conversion 조건을 일으킬 것이다.
3. 출력에서, 비트 줄의 문자 표현은 지정된 난에서 왼쪽 맞추기가 된다. 그리고 필요한 자르기와 빈자로 채우기는 오른쪽에서 일어날 것이다. 따옴표나 문자 B는 삽입되지 않는다. 만일 난의 넓이가 지정되지 않으면, 연관된 줄에 선언된 길이가 사용되며, 자료 항목은 난을 완전히 채운다.

### (3) COLUMN 서식 항목 (COLUMN Format item)

COLUMN 서식 항목은 :

COLUMN (문자-자리)

자리 (또는 문자 자리) 서식 항목은 현재 사용되는 줄 (Line) 또는 블럭 내에서 지정된 문자 자리로 (Character Position)

자료 항목의 자리여기를 통제한다. 이는 GET 와 PUT 문에서 PRINT 와 비-PRINT (非-PRINT) 화일에 사용된다.

일반규칙 :

1. "문자 자리" (때로는 W로 표시되기도 함)는 0보다 크고 256 작은 십진 고른 상수이어야 한다.
2. 입력에서, 문자 자리는 현재의 기록마디로부터 읽혀질 다음 난의 시작을 지정한다. 만일 지정된 문자 자리가 이미 지나쳤으면, 현재의 기록마디가 끝나고 다음 기록마디가 지정된 문자 자리에서 읽히기 시작한다. 출력 화일에서, 빈자가 그냥 지나간 수만큼 놓아진다. 만일 자료가 이미 지정된 문자 자리 또는 그 뒤로 놓아졌으면, 현재의 줄이 끝나고, 새로운 줄이 시작한다. 그러면 빈자가 다음 난이 새로운 줄의 지정된 문자 자리에서 시작하도록 자료 흐름에 삽입된다. 비-PRINT 화일에서, 블록 크기가 줄의 크기로 취급된다.
3. 만일 지정된 문자 자리가 현재 기록마디의 오른쪽 문자 자리를 넘어서면 (이는, W가 기록마디 크기를 넘는 경우이다), 문자 자리는 1로 대행된다.
4. COLUMN 서식 항목은 이것이 자료 나열이 다하기 전에 오지 않으면 효과가 없다.

(4) E 서식 항목

E 서식 항목은 :

E ( 난 - 넓이, 소수 - 자리수 [, 유의숫자 - 자리수] ) E ( Field-Width, Number-of-Fractional-Digits [, Number - of-Significant-Digits] )

부동점수 서식 항목은 부동점수 형식으로 된 십진 상수 자료의 외부 표현을 기술한다.

일반 규칙 :

1. "난의 넓이", "소수 자리수", "유의숫자 자리수" ( 때로는 W, D, S로 차례로 표현되기도 함 )는 무부호 십진 고른 상수이어야 한다. 난의 넓이는 33보다 작아야 한다. "난의 넓이"는 난에 있는 전체 문자 수를 지정 한다. "소수 자리수"는 가수 ( Mantissa )에서 소수점 다음에 나타나는 자리수를 지정한다. "유의숫자 수"는 가수에 나타나야 하는 숫자의 수를 나타낸다.
2. 입력에서, 자료 흐름에 있는 자료 항목은 지정된 난 안에 어디든지 있을 수 있는 음양 기호가 붙어도 좋은 십진 부동점수이거나 고정점수의 상수이다. 만일 자료가 고정점수이면, 지수는  $\emptyset$ 으로 대행된다.

이 수의 외부 형은 :

$$\left[ \begin{array}{c} + \\ - \end{array} \right] \text{가수} \left[ \left\{ \begin{array}{c} [E] \{ + 1 - \} \\ E \{ + 1 - \} \end{array} \right\} \text{지수} \right]$$

가수 ( Mantissa, 仮数 )는 십진 고정점 상수이어야 한다.

- a. 숫자는 지정된 난 어디에서든지 나타날 수 있다: 빈자가 난 안에서 수의 전후에 나타나도 된다. 만일 전체 난이 빈자이면, 이의 값은 양의  $\emptyset$ 으로 대행된다.

소수점 (decimal point)이 없으면, 소수 숫자 수 (d) (number of the fractional digits)는 가수 자리에서 대행된 소수점의 오른쪽의 문자 자리 수를 지정한다. 만일 소수점이 자료에 나타나면, 이는 소수 숫자 지정 수를 무시한다. "난의 넓이"로 표시된 값은 붙어있는 빈자, 지수 자리 (Exponent position, 붙을 수 있는 음양 기호의 자리, 있을 수 있는 문자 E의 자리, 가수에 있을 수 있는 소수점은 모두 포함한다.

- b. 지수는 3 자리수를 넘지 못하는 십진 고른 상수이다. 지수와 지수 앞의 음양 기호 또는 문자 E가 없으면, 영의 지수가 대행된다.
  - c. 가수의 음양 기호는 난 넓이에 항상 계상(計上)해야 한다. 이것이 양수에서 빈자일 때도 같다.
3. 출력에서, 내부 자료는 부동점수로 변환된다. 그리고 지정된 난에 있는 외부 자료 항목은 다음과 같은 일반 형식을 갖는다.

[ - ] { s - d 자리수 } . { d 자리수 } E { + | - } 지수

- a. 지수는 두 자리수의 십진 고른 상수이며, 모두 영이 될 수도 있다. 지수는 가수의 앞서는 숫자가 영이 안되도록 자동으로 맞추어 진다 (물론 가수가 영이 안되는 경우).
- b. 만일 위 숫자의 형식이 출력에서 지정된 난을 채우지 못하면, 이 수는 오른쪽 맞추기가 되며 왼편은 빈자로 채워진다. 만일 유의 숫자 자리수가 지정되지 않으면, 이는 소수 자리수 더하기 1로 된다. D-편성자에서, 양수든 음수든, 난의 넓이는 유의 숫자 수 더하기 6 한 것과

같거나 커야 한다(양의 기호는 써지지 않으나, 제상은 되어야 한다)。그렇지만, 만일 소수 숫자의 수가 영이면, 소수점은 써지지 않고, 앞에서 설명한 난의 넓이는 하나 줄어든다。

주:출력에서, E. 서식으로 편집된 자료는 마지막 숫자 자리에서 사사오입되며(四捨五入 Rounding), 잘리지 않는다。

(5) F 서식 항목

F 서식 항목은:

F ( 난-넓이 [, 소수숫자-자리수 [, 척도-인자] ] )

( F ( Field-Width [, Number-of-Fractional-digits [, Scaling-Factor] ] ) )

일반규칙:

1. "난넓이", "소수숫자 자리수", "척도-인자" ("尺度-因子")는(때로는 W, d, p로 표시되기도 함) 십진 고른 상수이어야 한다. 단지 P만이 음양 기호가 붙을 수 있다: 다른 것은 무기호여야 한다: W는 33보다 작아야하며 기호를 제상해야 하며, 기호가 빈자일 경우도 그렇다.
2. 입력에서, 자료 흐름에 있는 자료 항목은 지정된 난 안에 어디든지 있어도 좋은 음양 기호가 붙을 수 있는 십진 고정점 상수의 문자 표현이다. 빈자가 수의 전후에 나타나도 좋다. 만일 전체 난이 빈자이면, 이는 영으로 해석된다. 소수 자리수가 지정되지 않으면 영으로 대행된다.

만일 척도 인자가 지정되지 않고 소수점이 난에 나타나지 않으면, 소수 자리수는 난에서 대행된 소수점의 오른쪽에 있는 자리수를 지정한다. 만일 소수점이 자료에 나타나면, 이는 소수 자리수를 위한 지정보다 우선한다.

만일 척도 인자가 지정되면, 이는 척도인자로 10을 제공해서 자료 항목의 값을 곱한다. 그러므로, 만일 P가 양수이면, 수는 마치 소수점이 주어진 자리보다 P번 오른쪽에 있는 것과 같다. 만일 P가 음수이면, 이 수는 마치 소수점이 주어진 자리보다 P번 왼쪽에 있는 것과 같다. 주어진 소수점의 자리는 실점 (actual point) 이거나 대행된 점이거나 둘 중에 하나를 가르킨다.

출력에서, 내부자료는 필요하다면 고정점수로 변환되고, 이 외부 자료는 십진 고정점수의 문자 표현이 되며, 지정된 난에서 오른쪽 맞추기가 된다.

만일 서식 항목에서 난의 넓이만이 지정되면, 수의 고른수 부분만이 썬진다: 소수점은 나타나지 않는다.

만일 난의 넓이와 소수 자리수가 함께 지정되며 척도 인자가 없으면, 고른수 부와 소수부가 둘다 기록되며 소수점이 오른쪽 d 자리 앞에 삽입된다. 실 소수 숫자가 d보다 작을때 (d의 값은 난의 넓이보다 작아야 함)는 뒤따르는 영이 공급된다. 앞서는 영의 지우기가 소수점의 왼쪽에 있는 모든 수에 적용된다.

척도 인자의 값은 이 요소의 값이 외부 문자 표현으로 편집되기 전에 10을 P로 제공해서 자료 나열에 있는 연관된 요소의 값을 곱해준다. 소수 자리수가 영이면, 고른수

부분만이 사용된다.

출력에서, 만일 고정점 수의 값이 영보다 작으면, 음 기호가 외부 문자 표현의 앞에 붙는다: 만일 이것이 영보다 크면 빈자가 나타난다. 그런고로, 난 넓이 지정은 기호와 소수점이 있는 경우 소수점을 제상해야 한다. 만일 난 넓이가 유의 숫자나 기호가 떨어져 나가게 지정되면, SIZE 조건이 일어날 것이다

주: 출력에서, F 서식으로 편집된 자료는 마지막 숫자 자리에서 사사오입되며, 잘리지 않는다.

#### (6) LINE 서식 항목

LINE 서식 항목은:

LINE (줄-수)

줄 자리 서식 항목은 다음의 자료 항목이 인쇄될 PRINT 화일의 지면에서 특정 줄을 지정한다.

일반규정:

1. "줄 수" ("Line number")는 (때로는 W로 표시되기도 함) 256보다 작은 무기호 십진 고른 상수이어야 한다.
2. LINE 서식 항목은 앞으로 올 줄이 현재 지면의 지정된 줄이 되도록 빈줄이 끼워짐을 지정한다.
3. 만일 지정된 줄이 현재 지면에서 이미 지나갔거나 또는 지정된 줄이 세워진 한계를 넘으면, END PAGE 조건이 일어난다.
4. 만일 "줄 수"가 영이면, 이는 일로 대행된다.

5. LINE 서식 항목은 자료 나열이 다하기 전에 감지되지 않으면 효과가 없다.

(7) PAGE 서식 항목

PAGE 서식 항목은 :

PAGE

지면넘기기 서식 항목은 새로운 지면이 설정됨을 지정한다.

일반규칙 :

1. 새 지면의 설정은 다음의 인쇄가 첫 줄에 될 것임을 암시한다.
2. PAGE 서식 항목은 이것이 자료 나열이 다하기 전에 감지되지 않으면 효과가 없다.

(8) R 서식 항목

R 서식 항목 :

R (문-명찰-지시자) (R (Statement-Label-Designator))

이 외판 서식 항목은 (Remote Format item) 외판 서식 항목을 FORMAT 문에 있는 서식 항목으로 대치함을 말한다.

일반규칙 :

1. "문 명찰 지시자"는 FORMAT 문의 문 명찰을 이의 값으로 가지고 있는 명찰 상수나 요소 명찰 변수이다.

FORMAT 문은 서식 항목을 1대치할 서식 나열을 가진다.

“문 명찰 지시자”는 첨자를 가지지 못한다.

2. R 서식 항목과 지정된 FORMAT 문은 동일 블럭에 내적  
이어야 한다.

3. FORMAT 문은 R 서식 항목을 갖지 못한다.

#### (9) SKIP 서식 항목

SKIP 서식 항목은 :

SKIP [ ( 다음 줄 또는 기록마디의 상대적 - 자리 ) ]

( SKIP [ ( Relative-Position-of-next-Line or Record ) ] )

주 : 비 - PRINT 화일에 대해서는, 각 기록마디가 줄을 나타낸다.

전너기 서식 항목은 PRINT나 비 - PRINT 화일 모두에 사용될  
수 있다. GET와 PUT 문에서 이는 새로운 줄 ( PRINT 화일에서 )  
또는 새로운 기록마디 ( 비 - PRINT 화일에서 )가 앞으로 올 줄 또  
는 기록마디가 됨을 지정한다.

일반규칙 :

1. “ 다음 줄 또는 기록마디의 상대적 자리 ” ( 때로는 W로 표  
시되기도 함 )는 PRINT 화일에서 0부터 3까지, 비 - PRINT  
화일에서 1부터 3까지의 무기호 십진 고른 상수이어야 한  
다. 만일 이것이 없으면, 1로 대행된다.

2. 새 줄 또는 기록마디는 지금의 줄 또는 기록마디를 지나는  
줄 또는 기록마디를 지정한다.

3. 만일  $W$ 가  $1 \leq$ 이면,  $W - 1$ 의 빈줄 또는 기록마디가 출력인 경우 삽입되고 입력인 경우 건너편다.
4. 만일 SKIP ( $\emptyset$ )이 PRINT 화일에 지정되면, 이의 효과는 줄 띄기가 없이 이송장치 복귀가 된다. 이미 기록된 문자는 새로운 문자의 인쇄에 의해 중첩될 것이다. 보기로, 밑줄치기 ( $\underline{\quad}$ ,  $\underline{\quad}$ 와 같이)가 가능하다.
5. 만일 SKIP 서식 항목이 줄이나 기록마디의 끝에 지정되지 않으면, SKIP (1)이 수행된다. 이 말은, 한 줄 간다는 것이다.
6. 만일 PRINT 화일에서 지정된 줄이 설정된 한계 너머 있으면, END PAGE 조건에 일어난다.
7. SKIP 서식 항목은 이것이 자료 나열이 다하기 전에 감지되지 않으면 효과가 없다.

(10) X 서식 항목

X 서식 항목은:

X (난 - 넓이)

자리뛰기 서식 항목은 자료 흐름에 있는 자료 항목의 상대적 자리뛰기를 관리한다. 이는 PRINT 화일에만 한한 것이 아니다.

일반규칙:

1. "난 넓이" (때로는  $W$ 로 표시되기도 함)는 256 보다 작은 무기호 십진 고른 상수이어야 한다. 이는 현재 자리로부터 다음 난 까지 사이에 빈 문자 수를 지정한다.

- 2 . 입력에서 , 지정된 문자 수는 건너뛰며 전송되지 않는다.
- 3 . 출력에서 , 지정된 수의 빈자가 흐름에 삽입된다.
- 4 . 자리뛰기 서식 항목은 이것이 자료가 다하기 전에 감지되지 않으면 효과가 없다.

## 제 6 장 : 자료 변환 (資料 變換 Data Conversion )

이 장은 산수 변환 (Arithmetic Conversion) 과 자료 형식 변환 (Conversion of data Types) 을 위한 규칙을 망라한다. 개개 형식 변환 (Type Conversion) 은 독립된 제목 아래 열거되었다. 본문에 추가해서, 일곱개 표 (Table) 가 수록되어 있다.

- 표 6-1 은 산수 변환 결과의 정도 (精度 Precision) 을 산출하는 규칙을 말해준다.
- 표 6-2 는 산수를 비트-줄로 변환하는 결과의 길이 (Length) 를 찾는 데 사용될 수 있다.
- 표 6-3 은 고른수 올리기 (Ceiling) (CEIL) 에 사용될 수 있다.
- 표 6-4 부터 6-7 은 산수 연산 식에서의 변환을 설명해 주고, 지정된 연산자와 두 연산항에 유래되는 결과의 속성을 보여준다.

### 제 1 절 산수 변환 (算數 變換 Arithmetic conversion)

산수 변환에 관한 규칙은 값이 하나의 산수 표현에서 다른 것으로 전환 (轉換) 되는 방법을 말한다. 이는 전환 (Transformation) 의 결과로 값이 바뀌질 수 있다. 보기로, 십진 고정점수로 정확히 표현될 수 있는 0.2 는 이진수로 정확히 표현되지 못한다. 값에 있어서 그런 변화의 범위는 목적과 (目的 Target) 원시

(原始 Source)의 정도에 좌우된다. 식의 값내기에서, 목적의 정도는 원시 속성으로부터 유도된다. 일어날 수 있는 오류를 추정하고 이해하기 위해서는, 정도 규칙을 이해해야 한다; 이 규칙은 또 실제상 어떤 재량을 남겨두므로, 변환이 진행되는 방법에 관한 얼마간의 지식을 갖는 것이 도움이 될 것이다.

산수 연산의 연산항을 위한 일반적 변환 규칙은 그림 6-1에 보인다.

틀리는 연산항	변환 전	변환 후
자료 형식 (data type)	수치 문자 (numeric character)	규약 산수 (coded arithmetic)
기수 (base)	DECIMAL	BINARY
척도 (scale)	FIXED	FLOAT

그림 6-1 산수 연산에서의 연산항 변환

(1) 부동점수 변환 (floating-point conversion)

조직/360 실무에서, 십진과 이진의 부동점수는 조직/360에서 사용되는 내부 십육진수 형으로 유지된다. 만일 지정된 정도가 6의 십진수보다 크거나 21의 이진수보다 크면, 이 수는 긴 부동점수 형 (지수를 가진 14 십육진 자리수)으로 유지된다. 만일 정도가 6의 십진수보다 작거나 같거나, 21의 이진수보다 작거나

같으면, 이 수는 짧은 부동점수 형으로 유지된다 (지수를 가진 6 십육진 자리수).

이진수와 십진수 간(間)에 실제의 변환이 부동점수인 때는 일어나지 않는다. 단지 정도 변화만이 긴 것에서 짧은 것으로, 이는 자르기에 의해 이루어진다. 짧은 것에서 긴 것으로, 이는 영으로 늘리기에 의해 이루어진다. 된다. 그러나 선언된 부동점수 자료의 정도와 기수(基数 base)는 원시로부터 결정되는 중간결과 의 형의 속성과 더불어 목적 속성(target attribute)의 계산에 영향을 미친다.

## (2) 정도 변환(精度 變換 Precision Conversion)

정도 변환은 만일 지정된 목적 속성이 원시 정도와 다르면 일어난다. 특별히, 원시와 목적이 서로 다른 기수이면 항상 정도 변화가 일어난다. 부동점수에서 고정점수로 변환할 때는 또한 실제 상의 정도 변화가 있다, 이는 부동점수가 표현되는 특이한 방법 때문이다. 정도 변화는 자르거나 영으로 채우기에 의해 수행된다. 부동점수(floating-point number)는 오른쪽이 영으로써 늘어남으로 인해 짧은 정도(short Precision)에서 긴 정도(long Precision)로 변환되고, 오른쪽에서 잘림으로써 그 반대의 변환이 된다.

고정점수(fixed-point number)는 십진 또는 이진소수점을 중심으로 유지되며 왼쪽 또는 오른쪽에서 잘리며, 왼쪽 또는 오른쪽에서 영으로 채워진다. 고정점 이진 변수의 이진 소수점은 가장 오른쪽 이진 숫자 뒤에 있는 것으로 항상 대항되므로, 그런 변수에 대입된 고정점 이진수 값은 오른쪽에서 늘어나지는 않는다; 물론

왼쪽에서는 늘어나기 (extension) 가 일어난다, 또 오른쪽에서는 자르기가 일어날 수 있다.

오른쪽에 있는 유의 숫자의 상실 (喪失 loss) 에 관해 지적해주는 것은 없다. 왼쪽에 있는 숫자의 상실은 만일 SIZE 조건이 가능하게 되어 있다면 검사될 것이다. 조직/360 실무에서, 이진 고정점 수는 선언된 넓이가 어떠한 31 비트 (bit) 의 낱말 (word) 로 기억된다. 십진 수는 항상 홀수개의 숫자로 기억된다, 이는 조직/360 에서 짜인 십진수 (Packed decimal) 로 유지되기 때문이다.

### (3) 기수 변환 (基数 變換 base conversion)

기수에서의 변화는 고른수 아닌 고정점 수의 값에만 통상으로 영향을 미칠 것이다. 어떤 십진 소수 (小数 fraction) 는 이진수로 정확히 표시되지 못한다, 고로 어떤 오류가 자르기에 의해 일어날 것이다. 어떤 이진 소수는 변환 규칙에 의해 자동으로 생성된 것보다 더 많은 십진 숫자를 정확한 표현을 위해서 요구할 것이다, 또 이것은 자르기에 의해 유래하는 오류가 될 수도 있다.

이진 고정점 수의 범위는 십진 고정점 수의 범위보다 작으므로, 십진수를 이진수로 변환함에 있어 유의 숫자가 왼쪽에서 상실될 수도 있다. 이는 SIZE 조건을 가져올 것이다, 그러나 이 조건이 SIZE 접두어에 의해 명시로 가능하게 되지 않으면 중단이 일어나지 않을 것이다.

상수의 일차적 표시법은 십진수이며, 그러므로 대부분의 상수는 십진수로 써진다. 상수의 정도는 이것이 써지는 방법에 따라 유도된다. 고정점 이진수로 변환될 소수 아닌 상수를 쓸 때는

조심을 해야 한다.

제 2 절 자료의 형식 변환 (資料의 形式 變換 data type conversion)

기본적 자료 형식 변환이 그림 6-2에 있다.

~에서 / ~으로	규약 산수	수치 문자	문자 줄	비트
규약 산수		y	X	y
수치 문자	Y		Y <sup>1</sup>	y
문자 줄	X	X		y <sup>2</sup>
비트 줄	Y	y	Y <sup>3</sup>	

그림 6-2. 기본적 자료 형식 변환

y - 변환은 자동으로 일어남

X - 안되는 변환

1 - 수치 문자 자료는 문자 줄로 //해석//된다

2 - 오직 문자 0과 1만이 줄에 있을 경우

3 - 비트 값 0과 1은 문자 0과 1로 됨

(1) 규약 산수에서 수치 문자로 (coded arithmetic to numeric character)

수치 문자로 변환될 규약 산수 자료는 필요하면 수치 문자 항목의 PICTURE 속성에 의해 결정되는 척도 (scale) 와 정도를 가진 십진수 값으로 변환된다.

(2) 수치 문자에서 규약 산수로 ( numeric character to coded arithmetic )

규약 산수로 변환될 수치 문자 자료는 먼저 해당 PICTURE 속성에 의해 지정된 대로의 척도와 정도의 십진 항목으로서 해석된다. 이 항목이 다음에 규약 산수 목적의 기수, 척도, 정도로 변환된다.

(3) 수치 문자에서 문자-줄로 ( numeric character to character-string )

수치 문자 자료 항목은 문자 줄로서 해석된다. 문자 줄의 길이는 해당 PICTURE 속성으로 기술된 수치 문자 자료 항목의 길이와 같다. (즉 양 길이가 같다.)

(4) 문자-줄에서 비트-줄로 ( character - string to bit-string )

원시 줄에 있는 문자 1은 목적 줄에서 비트 1이 되고, 0은 0으로 된다. 원시 줄에 있는 0과 1 이외의 어떤 문자도 CONVERSION 조건을 일으킬 것이다.

만일 원시 줄이 목적보다 길으면, 우선 모든 원시 문자가 (여분의 문자도 함께) 변환되고, 이때 0 또는 1 이외의 문자는 CONVERSION 조건을 일으킬 것이다. 어떤 여분의 문자가 오른쪽

에서 잘리는 것은 변환 후에 되는 것이다. 만일 목적이 원시보다 길으면, 목적은 오른쪽에 영으로 채워진다.

(5) 비트-줄에서 문자-줄로 (bit-string to character-string)

비트 0은 문자 0으로 되고, 비트 1은 문자 1이 된다. 생성된 문자 줄은, 이는 원시 비트 줄과 같은 길이이다. 목적이 대입된다.

만일 원시 비트 줄이 목적 문자 줄보다 짧으면, 목적의 나머지는 빈자로 채워진다.

보기가 다음에 있다.

원 시	값	결 과
'1011' B	CHAR(4)	'1011'
'10101' B	CHAR(10)	'10101bbbb'
'0001' B	CHAR(1)	'0'

CONVERSION 조건은 비트에서 문자로의 변환에서는 일어날 수 없다; 그렇지만, 비트 줄에서 변환되어 생성된 문자 줄은 만일 빈자가 삽입되었다면 재변환될 때는 변환 오류가 될 수 있다.

(6) 규약 산수에서 비트-줄로 (coded arithmetic to bit-string)

산수의 절대값이 먼저 이진 고른수로 변환된다. 이 고른수의 정도는 표 6-2에서 보인 바대로 비트-줄 목적의 길이와 같다.

이 고른수는 음양 기호를 빼고 비트 줄로서 처리된다. 이 중간

결과 줄이 목적에 대입된다. 몇개의 보기가 표 6-3에 있다.

(7) 비트-줄에서 규약 산수로 (bit-string to coded arithmetic)

비트 줄은 실무에서 정의된 최대 정도를 가진 무기호 이진 고른수로 해석된다. D-편성자에서, 이는 31 비트이다. 줄은 31 비트 보다 크지 않아야 한다. 만일 줄이 31 비트보다 짧으면, 영이 왼쪽에 삽입된다. 비트-줄을 산수로 변환한 결과는 항상 양수이다. 채우기는 왼쪽이고, 오른쪽이 아님을 유의하라.

(8) 수치 문자에서 비트-줄로 (numeric character to bit-string)

수치 문자 난은 먼저 규약 산수로 변환되고 다음에 앞에 나온 규칙 ((6)번)에 따라 비트 줄로 변환된다.

(9) 비트-줄에서 수치 문자로

비트 줄이 먼저 규약 산수로 변환되고 다음에 앞에 나온 규칙 ((1)번)에 따라 수치 문자로 변환된다.

### 제 3절 고른수 올리기 값의 표 (table of ceiling values)

표 6-3은 변환에서 정도와 길이를 결정하는데 사용되는 고른수로 올린 값을 계산함에 있어 프로그래머를 돕기 위한 것이다. 이는

1 부터 15 까지의 값을 3.32 로 곱한 결과의 값을 올리기 (ceiling) 한 것과 1 부터 56 까지의 값을 3.32 로 나눈 결과의 올리기 한 것을 보여준다.

제 4 절 산수 연산 결과에 관한 표 ( tables for  
results of arithmetic operation )

표 6-4 부터 6-7 까지는 지정된 연산자와 두 연산항의 속성에 기초해서 산수 연산 결과의 속성을 보여준다. 이들 표에서 목적 속성 ( target precision ) 은 실무상 정해진 최대치를 넘지 못한다, 이는 FIXED DECIMAL 에서 15, FIXED BINARY 에서 31, FLOAT DECIMAL 에서 16, FLOAT BINARY 에서 53 이다.

원시속성	원시값	중간결과	목적속성	결과
FIXED BINARY (10)	15	0000001111	BIT(10)	0000001111
FIXED BINARY (1)	1	1	BIT(1)	1
FIXED DECIMAL(1)	1	0001	BIT(1)	0
FIXED BINARY (3)	+3	011	BIT(3)	011
FIXED DECIMAL(2,1)	1.1	0001	BIT(4)	0001
FLOAT BINARY (4)	1.25	0001	BIT(5)	00010

그림 6-3 산수를 비트-출로 바꾸는 보기

표 6-1 산수변환의 정도

원시속성	목적속성	목적정도
DECIMAL FIXED(P, q)	DECIMAL FLOAT	P
DECIMAL FIXED(P, q)	BINARY FIXED	$1+P*3.32, q*3.32$ (주3를 보라)
DECIMAL FIXED(P, q)	BINARY FLOAT	$P*3.32$
DECIMAL FLOAT(P)	BINARY FLOAT	$P*3.32$
BINARY FIXED(P, q)	BINARY FLOAT	P
BINARY FIXED(P, q)	DECIMAL FIXED	$1+P/3.32, q/3.32$ (주4를 보라)
BINARY FIXED(P, q)	DECIMAL FLOAT	$P/3.32$
BINARY FIXED(P)	DECIMAL FLOAT	$P/3.32$

주(註):

1.  $P*3.32$  와  $P/3.32$  의 경우에서, 결과의 CEIL이 취해진다:  
취한 값은 결과와 같거나 바로 큰 고른 수이다.

2. 목적정도는 사용상 정의된 최대치를 넘지 못한다, FIXED DECIMAL에서 15, FIXED BINARY에서 31, FLOAT DECIMAL에서 16, FLOAT BINARY에서 53.

3. q가 음수이면, 다음 식이 적용된다 :  
 $( \text{MIN}(\text{CEIL}(P * 3.32) + 1, 31), \text{CEIL}(\text{ABS}(q) * 3.32) * \text{SIGN}(q) )$ .

4. q가 음수이면, 다음식이 적용된다 :  
 $( \text{CEIL}(P / 3.32) + 1, \text{CEIL}(\text{ABS}(q) / 3.32 * \text{SIGN}(q)) )$

표 6-2 변환된 비트줄의 길이 (규약산수를 비트줄로)

원시속성	목적길이
DECIMAL FIXED(P,q)	$(P - q) * 3.32$
DECIMAL FLOAT(P)	$P * 3.32$
BINARY FIXED(P,q)	$P - q$
BINARY FLOAT(P)	$P$

주(註) :  $P * 3.32$  와  $(P - q) * 3.32$  의 경우에서 결과의 CEIL이 취해진다. 또, D-원성자에서, 목적길이는 1과 31 사이에 있어야 한다. 경계 포함.

표 6-3. 3.32로 곱해 지거나 나누어진 값의 올리기

X	CEIL ( $x / 3.32$ )	Y	CEIL ( $y / 3.32$ )
1	4	1 - 3	1
2	7	4 - 6	2
3	10	7 - 9	3
4	14	10 - 13	4
5	17	14 - 16	5
6	20	17 - 19	6
7	24	20 - 23	7
8	27	24 - 26	8
9	30	27 - 29	9
10	34	30 - 33	10
11	37	34 - 36	11
12	40	37 - 39	12
13	44	40 - 43	13
14	47	44 - 46	14
15	51	47 - 49	15
		50 - 53	16
		54 - 56	17

		제 1	
		DECIMAL FIXED( $p_1, q_1$ )	DECIMAL FLOAT( $p_1$ )
계 산 항	DECIMAL FIXED ( $p_2, q_2$ )	DECIMAL FIXED( $p, q$ ) $P = 1 + \text{MAX}(p_1 - q_1, p_2 - q_2)$ $+ \text{MAX}(q_1, q_2)$ $q = \text{MAX}(q_1, q_2)$	DECIMAL FLOAT( $P$ ) $P = \text{MAX}(P_1, P_2)$
	DECIMAL FLOAT ( $P_2$ )	DECIMAL FLOAT( $P$ ) $P = \text{MAX}(P_1, P_2)$	DECIMAL FLOAT( $P$ ) $P = \text{MAX}(P_1, P_2)$
	BINARY FIXED ( $p_2, q_2$ )	BINARY FIXED( $p, q$ ) $P = 1 + \text{MAX}(r - s, p_2 - q_2)$ $+ \text{MAX}(s, q_2)$ $q = \text{MAX}(s, q_2)$ 여기서 $r = 1 + P_1 * 3.32$ $s = q_1 * 3.32$	BINARY FLOAT( $P$ ) $P = \text{MAX}(r, P_2)$ 여기서 $r = P_1 * 3.32$
	BINARY FLOAT ( $P_2$ )	BINARY FLOAT( $P$ ) $P = \text{MAX}(r, q_2)$ 여기서 $r = P_1 * 3.32$	BINARY FLOAT( $P$ ) $P = \text{MAX}(r, P_2)$ 여기서 $r = P_1 * 3.32$

연산 결과의 속성

연	산	항
BINARY FIXED( $p_1, q_1$ )		BINARY FLOAT( $p_1$ )
BINARY FIXED( $p, q$ ) $P = 1 + \text{MAX}(p_1 - q_1, r - s)$ $\quad + \text{MAX}(q_1, s)$ $q = \text{MAX}(q_1, s)$ 여기서 $r = 1 + P_2 * 3.32$ $s = q_2 * 3.32$		BINARY FLOAT( $P$ ) $P = \text{MAX}(P_1, r)$ 여기서 $r = P_2 * 3.32$
BINARY FLOAT( $P$ ) $P = \text{MAX}(p_1, r)$ 여기서 $r = p_2 * 3.32$		BINARY FLOAT( $P$ ) $P = \text{MAX}(p_1, r)$ 여기서 $r = P_2 * 3.32$
BINARY FIXED( $p, q$ ) $P = 1 + \text{MAX}(p_1 - q_1, p_2 - q_2)$ $\quad + \text{MAX}(q_1, q_2)$ $q = \text{MAX}(q_1, q_2)$		BINARY FLOAT( $P$ ) $P = \text{MAX}(P_1, P_2)$
BINARY FLOAT( $P$ ) $P = \text{MAX}(P_1, P_2)$		BINARY FLOAT( $P$ ) $P = \text{MAX}(P_1, P_2)$

		제 1	
		DECIMAL FIXED( $P_1, q_1$ )	DECIMAL FLOAT( $P_1$ )
제 2 연 산 항	DECIMAL FIXED ( $P_2, q_2$ )	DECIMAL FIXED( $P, q$ ) $P = P_1 + P_2 + 1$ $q = q_1 + q_2$	DECIMAL FLOAT( $P$ ) $P = \text{MAX}(P_1, P_2)$
	DECIMAL FLOAT ( $P_2$ )	DECIMAL FLOAT( $P$ ) $P = \text{MAX}(P_1, P_2)$	DECIMAL FLOAT( $P$ ) $P = \text{MAX}(P_1, P_2)$
	BINARY FIXED ( $P_2, q_2$ )	BINARY FIXED( $P, q$ ) $P = r + P_2 + 1$ $q = s + q_2$ 여기서 $r = 1 + P_1 * 3.32$ $s = q_1 * 3.32$	BINARY FLOAT( $P$ ) $P = \text{MAX}(r, P_2)$ 여기서 $r = P_1 * 3.32$
	BINARY FLOAT ( $P_2$ )	BINARY FLOAT( $P$ ) $P = \text{MAX}(r, P_2)$ 여기서 $r = P_1 * 3.32$	BINARY FLOAT( $P$ ) $P = \text{MAX}(r, P_2)$ 여기서 $r = P_1 * 3.32$

산 결과의 속성

연산	항
BINARY FIXED( $P_1, q_1$ )	BINARY FLOAT ( $P_1$ )
BINARY FIXED( $P, q$ ) $P = P_1 + r + 1$ $q = q_1 + s$ 여기서 $r = 1 + P_2 * 3.32$ $s = q_2 * 3.32$	BINARY FLOAT( $P$ ) $P = \text{MAX}(P_1, r)$ 여기서 $r = P_2 * 3.32$
BINARY FLOAT( $P$ ) $P = \text{MAX}(P_1, r)$ 여기서 $r = P_2 * 3.32$	BINARY FLOAT( $P$ ) $P = \text{MAX}(P_1, r)$ 여기서 $r = P_2 * 3.32$
BINARY FIXED( $P, q$ ) $P = P_1 + P_2 + 1$ $q = q_1 + q_2$	BINARY FLOAT( $P$ ) $P = \text{MAX}(P_1, P_2)$
BINARY FLOAT( $P$ ) $P = \text{MAX}(P_1, P_2)$	BINARY FLOAT( $P$ ) $P = \text{MAX}(P_1, P_2)$

표 6-6 나 누 지 연 산

		계 1	
		DECIMAL FIXED( $P_1, q_1$ )	DECIMAL FLOAT( $P_1$ )
계 연 산 항	DECIMAL FIXED ( $P_2, q_2$ )	DECIMAL FIXED( $P, q$ ) $P = 15$ $q = 15 - ((P_1 - q_1) + q_2)$	DECIMAL FLOAT( $P$ ) $P = \text{MAX}(P_1, P_2)$
	DECIMAL FLOAT ( $P_2$ )	DECIMAL FLOAT( $P$ ) $P = \text{MAX}(P_1, P_2)$	DECIMAL FLOAT( $P$ ) $P = \text{MAX}(P_1, P_2)$
	BINARY FIXED ( $P_2, q_2$ )	BINARY FIXED( $P$ ) $P = 31$ $q = 31 - ((r - s) + q_2)$ 여기서 $r = 1 + P_1 * 3.32$ $s = q_1 * 3.32$	BINARY FLOAT( $P$ ) $P = \text{MAX}(r, P_2)$ 여기서 $r = P_1 * 3.32$
	BINARY FLOAT ( $P_2$ )	BINARY FLOAT( $P$ ) $P = \text{MAX}(r, P_2)$ 여기서 $r = P_1 * 3.32$	BINARY FLOAT( $P$ ) $P = \text{MAX}(r, P_2)$ 여기서 $r = P_1 * 3.32$

결 과 의 속 성

연 산 항	
BINARY FIXED( $P_1, q_1$ )	BINARY FLOAT( $P_1$ )
BINARY FIXED( $P, q$ ) $P = 31$ $q = 31 - ((P_1 - q_1) + S)$ 여기서 $S = q_2 * 3.32$	BINARY FLOAT( $P$ ) $P = \text{MAX}(P_1, r)$ 여기서 $r = P_2 * 3.32$
BINARY FLOAT( $P$ ) $P = \text{MAX}(P_1, r)$ 여기서 $r = P_2 * 3.32$	BINARY FLOAT( $P$ ) $P = \text{MAX}(P_1, r)$ 여기서 $r = P_2 * 3.32$
BINARY FIXED( $P, q$ ) $P = 31$ $q = 31 - ((P_1 - q_1) + q_2)$	BINARY FLOAT( $P$ ) $P = \text{MAX}(P_1, P_2)$
BINARY FLOAT( $P$ ) $P = \text{MAX}(P_1, P_2)$	BINARY FLOAT( $P$ ) $P = \text{MAX}(P_1, P_2)$

표 6-7 지수식 연산 결과의 수성

경우	제 1 연산 항	제 2 연산 항	결과 의 목적 속성
경우(1)	FIXED DECIMAL( $P_1, q_1$ )	값 $n$ 인 부호없는 고른 상수	FIXED DECIMAL( $P, q$ ) [준비된 $P = (P_1 + 1) * n - 1$ $q = q_1 * n$ ]
경우(2)	FIXED BINARY( $P_1, q_1$ )	값 $n$ 인 부호없는 고른 상수	FIXED BINARY( $P, q$ ) [준비된 $P = (P_1 + 1) * n - 1$ $q = q_1 * n$ ]
경우(3)	FIXED DECIMAL( $P_1, q_1$ ) 또는 FLOAT DECIMAL( $P_1$ )	FIXED DECIMAL( $P_2, q_2$ ) 또는 FLOAT DECIMAL( $P_2$ )	FLOAT DECIMAL( $P$ ) [경우(1) 제외]
경우(4)	FIXED BINARY( $P_1, q_1$ ) 또는 FLOAT BINARY( $P_1$ )	FIXED DECIMAL( $P_2, q_2$ ) 또는 FLOAT DECIMAL( $P_2$ )	FLOAT BINARY( $P$ ) [경우(2) 제외]
경우(5)	FIXED DECIMAL( $P_1, q_1$ ) 또는 FLOAT DECIMAL( $P_1$ )	FIXED BINARY( $P_2, q_2$ ) 또는 FLOAT BINARY( $P_2$ )	$P = \text{MAX}(P_1, \text{CEIL}(3.32 * P_2))$ FLOAT BINARY( $P$ ) [경우(1) 제외]
경우(6)	FIXED BINARY( $P_1, q_1$ ) 또는 FLOAT BINARY( $P_1$ )	FIXED BINARY( $P_2, q_2$ ) 또는 FLOAT BINARY( $P_2$ )	$P = \text{MAX}(\text{CEIL}(3.32 * P_1), P_2)$ FLOAT BINARY( $P$ ) [경우(2) 제외]

## 제 7 장 : 내조립 함수와 유사 변수

(内組立 函数와 類似 變数 built-in function and pseudo-variables)

모든 내조립 함수와 유사 변수는 이 장에서 논술된다. 이 장의 대체적 구성은 다음과 같다 :

### 1. 계산 내조립 함수 (computational built-in function 計算)

- a. 줄-취급 내조립 함수 (줄-取扱 string-handling built-in functions)
- b. 산수 내조립 함수 (算数 arithmetic built-in functions)
- c. 수학 내조립 함수 (数学 mathematical built-in functions)
- d. 배열 다루기 내조립 함수 (配列 다루기 array manipulation built-in functions)

### 2. 기타 내조립 함수 (其他 miscellaneous built-in functions)

### 3. 유사 변수 (類似 變数 pseudo-variables)

계산 내조립 함수는 위에서 보인 바와 같이 줄 취급, 산수 연산 (절대 값, 자르기 등), 수학 연산 (삼각 함수, 제곱근 등), 배열 다루기 함수를 갖추고 있다. 계산 내조립 함수는 :

ABS	ADD	BINARY
CEIL	DECIMAL	DIVIDE
FIXED	FLOAT	FLOOR

MAX	MIN	MOD
MULTIPLY	PRECISION	ROUND
SING	TRUNC	

수학 :

ATAN	ATAND	ATANH
COS	COSD	COSH
ERF	ERFC	EXP
LOG	LOG <del>10</del>	LOG 2
SIN	SIND	SINH
SQRT	TAN	TAND
TANH		

배열 다루기

ALL	ANY	PROD
SUM		

기타 내조립 함수는 여러가지 일을 수행한다; 보기로 현재의 일자, 시간, 기타 내조립 함수는 :

ADDR	DATE	NULL
STRING	TIME	

유사 변수의 절은 세개의 PL/I 유사 변수 STRING, SUBSTR, UNSPEC 에 관해 간단히 논술된다. 자세한 것은 해당 내조립 함수에 있다.

모든 내조립 함수와 유사 변수는 그들이 속한 제목 아래 영자 순으로 나 타난다.

내조립 함수와 유사 변수의 인수와 결과의 속성에 관한 요약이  
"부록 다"에 있다.

## 제1 절 계산 내조립 함수

### [1] 줄 취급 내조립 함수

(string handling built-in functions)

여기서 기술되는 함수는 줄을 다루는 데 사용된다. 별명이 없  
는 한, 요소식이나 배열 이름이 인수(引數 argument)로서 사용  
될 수 있다. 인수가 배열 이름이면, 이 함수에 의해 돌아가는  
값은 인수와 동일한 차원과 한계(次元과 限界 dimension and  
bound)를 가진 배열 이름이다. (함수는 배열 인수의 각 요소에  
대해 수행된다)

중간 결과는 변환 규칙과 문자줄과 비트-줄 대입 규칙에 따라  
계산된다.

#### BIT

정의: BIT는 주어진 값을 비트-줄로 변환하여 그 결과를 불  
러낸 점으로 돌려보낸다. 이 함수는 비트-줄 변환 결과의 크기  
를 환리할 수 있게 한다.

인용(引用 reference):

'BIT(식[, 크기]) (BIT(expression[, size]))

인수(引數 arguments): 인수 "식"은 비트-줄로 변환될 재  
료를 나타낸다; 이 인수는 비트-줄, 문자-줄, 산수 요소 식 또  
는 배열 이름이 될 수 있다. 인수 "크기"는, 만약 지정되면

결과(結果)의 길이를 제공하는 십진 고른 상수이어야 한다. 만일 '크기'가 지정되지 않으면, 이는 제 6 장, '자료 변환'에서 주어진 형식 변환 규칙에 따라 결정된다. 만일 '식'이 배열 이름이면, '크기'는 개개 요소에 적용된다.

결과(結果 result): 이 함수에 의해 돌아가는 값은 비트-줄로 변환된 식이다. 이 비트-줄의 길이는 앞서 얘기한 것처럼 '크기'에 의해 결정된다.

BOOL

정의(定義 definition): BOOL은 두개의 주어진 비트-줄에 주어진 불 연산(boolean operation)의 결과의 비트 표현인 비트-줄을 만들어 낸다.

인용:

BOOL(X, Y, W)

인수: 인수 'X'와 'Y'는 'W'로 지정된 불 연산이 수행될 두개의 비트-줄을 나타낸다; 이들 인수는 비트-줄, 문자-줄, 산수 요소 식 또는 배열 이름이 될 수 있다. 만일 'x'와 'y'가 비트-줄이 아니면, 이들은 비트-줄로 변환된다. 만일 'x'와 'y'가 길이가 틀리면, 짧은 쪽이 긴 줄의 길이에 맞추기 위해 오른쪽이 영으로 늘어난다.

인수 'w'는 불 연산을 나타낸다; 이 인수는 비트-줄, 문자-줄 산수 요소 식, 또는 배열 이름이 될 수 있다.

이는 길이 4의 비트-줄로 변환되며 n이 0 또는 1인  $n_1, n_2, n_3, n_4$ 로 정한다. 이 경우 16개의 가능한 비트-조합이 있으며 그런고로 16가지의 불 연산이 있다. 'x'와 'y'에 작용키 위해, 'w'는 길이 4의 비트-줄로 필요하면 함수가 불

불러내지기 전에 바뀐다.

만일 둘 이상의 인수가 배열이면, 배열은 동일 한계를 가져야 한다.

결과: 이 함수에 의해 돌아오는 값은 'x'와 'y'와 길이가 같은 비트-줄 z이다. 개개 z의 비트는 다음과 같이 'x'와 'y'의 대응 비트에 따라 행해지는 불 연산에 의해 결정된다; z의 i번째 비트는 다음에 있는 불 표에 있는 것 처럼 'x'와 'y'의 i번째 비트의 결합에 따라서  $n_1, n_2, n_3$  또는 n의 값으로 정해진다.

$x_i$	$y_i$	$w_i$
0	0	$n_1$
0	1	$n_2$
1	0	$n_3$
1	1	n

보기 :

우선 불표를 다음과 같이 정해 놓았다고 하자.

$x_i$	$y_i$	$w_i$
0	0	0
0	1	1
1	0	1
1	1	0

여기서 주의할 것은  $x_i$ 와  $y_i$ 의 순서는 앞의 불 표의 결합 순서를 따르고 그 순서대로의 연산 결과를 지정해 주어야 한다.

여기서의 불 연산은 '0110'이 된다.

이 정의에 따르면 0.0이면 결과가 0, 0.1이면 결과는 1, 1.0이면 1, 1.1이면 0이다.

보기 :

A BIT(5), B BIT(5) INIT ('11011'B), C BIT(5) INIT ('01100'B)라 하자.

내  
2

A = BOOL(B,C, '0110'B);

이 연산은 다음과 같이 풀 수 있다.

	(1)	(2)	(3)	(4)	(5)
B	1	1	0	1	1
C	0	1	1	0	0
	1	0	1	1	1

B의 첫 비트는 1, C의 첫 비트는 0이므로 정의한 연산에 의해 1.0은 1이므로 결과는 1이다. B의 두번째 비트는 1, C의 대응하는 것은 1이므로 1.1은 0이므로 결과는 0이다.

이하 같은 규칙으로 전체의 결과는 10111이므로:

A → 10111 B가 들어 간다.

동일 연산을 가지고 B와 C의 위치를 바꾸어 보자.

A = BOOL(C,B, '0110'B);

	(1)	(2)	(3)	(4)	(5)
C	0	1	1	0	0
B	1	1	0	1	1
	1	0	1	1	1

결과는 같다. A → 10111 B. 이것은 연산을 그런 결과를 받게 만들었기 때문이다. 다음의 보기를 보라.

보기 :

불 연산을 다음과 같이 정한다. 즉  $w \rightarrow 1101'B$ . 그러면 표는 다음과 같다.

$x_i$	$y_i$	$w_i$
0	0	1
0	1	1
1	0	0
1	1	1

앞에서 선언된 변수를 그대로 사용한다.

$A = \text{BOOL}(B, C, 1101'B) ; A = \text{BOOL}(C, B, 1101'B) ;$

$B \quad 1 \quad 1 \quad 0 \quad 1 \quad 1 \quad C \quad 0 \quad 1 \quad 1 \quad 0 \quad 0$

$C \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad B \quad 1 \quad 1 \quad 0 \quad 1 \quad 1$

$0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 1$

$A \rightarrow 01100'B \quad A \rightarrow 11011'B$

CHAR

정의: CHAR는 주어진 값을 문자-줄로 바꾸어 불러내는 점으로 그 결과를 돌려 보낸다. 이 함수는 문자-줄 변환 결과의 크기를 관리할 수 있게 한다.

인용:

CHAR ( 식 [, 크기] )

인수: 인수 '식'은 문자-줄로 변환될 재료를 나타낸다; 이 인수는 비트-줄, 문자-줄, 수차문자 요소식 또는 배열 이름이 될 수 있다. 인수 '크기'는, 이것이 지정되었을 때, 결과의 길이를 말해 주는 십진 곱론수 상수이어야 한다.

만일 '크기'가 지정되지 않으면, 이는 제6장, '자료 변환'에서 주어진 형식 변환 규칙에 따라 결정된다. 만일 '식'이 배열 이름이면, '크기'는 배열의 개개 요소를 인용한다.

결과: 이 함수에 의해 돌아가는 값은 문자-줄로 변환된 '식'이다. 이 문자-줄의 길이는 '크기'에 의해 결정된다.

#### HIGH

정의: HIGH는 대조 순서에 있는 최고 문자로써 주어진 길이의 문자-줄을 만든다; 이 말은, 만들어진 줄에 있는 개개 문자는 대조 순서에 있는 최고 문자이다 (제2장을 보라)

인용:

#### HIGH(*i*)

인수: 인수 '*i*'는 형성될 줄의 길이를 지정하는 무기호 십진 고른수 상수이어야 한다. 조직/360 실무에서, 이 문자는 십육진 FF로 기억된다.

결과: 이 함수에 의해 돌아가는 값은 길이 '*i*'의 문자-줄이며; 줄에 있는 문자-줄은 대조 순서에 있는 최고 문자이다.

#### INDEX

정의: INDEX는 지정된 줄에서 지정된 비트 또는 문자-줄 형상(形状 configuration)을 찾는다. 만일 이 형상이 발견되면, 그 줄 안에서의 형상이 시작하는 자리가 불러내는 점으로 돌아온다.

인용:

#### INDEX (줄, 형상) (INDEX (string, config))

인수: 두개 인수가 지정되어야 한다. 첫째 인수, '줄'은 탐색 장소를 나타낸다; 둘째 인수, '형상'은 줄에서 찾아야 할

의  
3

비트 또는 문자-줄 형상을 나타낸다. 이들 인수는 비트-줄, 문자-줄, 이진 규약 산수, 또는 수치 문자 요소 식 또는 배열 이름이어야 한다.

만일 두개의 인수가 모두 비트-줄이 아니거나, 또는 만일 한개의 인수만이 비트-줄이면, 두개의 인수는 가능하다면 문자-줄로 변환된다. 만일 두개 인수가 모두 비트-줄이면, 변환이 일어나지 않는다. 이진 규약 산수는 비트-줄로 변환되고 수치 문자 인수는 위 변환이 수행되기 전에 문자-줄로 변환된다. 만일 두개의 인수가 배열이면, 배열은 동일한 한계를 가져야 한다.

결과: 이 함수에 의해 돌아가는 값은 태만 정도(15)인 이진 고른수이다.

이 이진 고른수는 다음의 하나이다:

1. '형상'이 발견된 '줄'안의 위치(이는 문자 또는 비트 자리). 만일 둘 이상의 '형상'이 '줄'에 존재하면, 발견된 첫번째 위치(왼쪽에 오른쪽으로 감지됨)가 돌아간다.
2. 만일 '형상'이 '줄'에 존재하지 않으면, 이 값은 0.

보기: 만일 ASTRING이 다음을 포함한 문자-줄이라면;

```
QXZNAMEA,X,FERST,2,SECOND
```

```
I = INDEX(STRING, 'X');
```

은 불러내는 점으로 이진 값 열을(10<sub>10</sub>) 돌려보낼 것이다. 이 이진 값은 ASTRING 안에서 형상 'X'의 자리를 나타낸다. 그렇지만, 만일 문이 다음과 같다고 하면:

```
I = INDEX(STRING, X) 0,
```

들의 이진수 값이 불러내는 점으로 돌아갈 것이다. 이 값은 ASTRING 안에서 첫번째 나타난 X의 위치이다.

## LOW

정의 : LOW는 대조 순서에 있는(对照 順序 collating sequence) 최저 문자로 지정된 길이의 문자-줄을 형성한다; 이것은, 형성된 줄의 개개 문자는 대조 순서에서 최저 문자가 된다는 말이다. (제2장을 보라)

인용 :

LOW(*i*)

인수 : 인수 '*i*'는 형성될 줄의 길이를 지정하는 무기호 십진 고른수 상수이어야 한다.

결과 : 이 함수에 의해 돌아가는 값은 길이 '*i*'인 문자-줄이며; 줄에 있는 개개 문자는 대조 순서상 최저 문자이다.

조직 / 360 실무에서, 이 문자는 십육진수(十六進數 hexadecimal) 00이다.

## REPEAT

정의 : REPEAT는 주어진 줄의 값을 취하여 이 자체에 추가해서 지정된 횟수(回数)만큼 이어 붙여 만든 줄 값으로 된 새로운 줄을 형성한다.

인용 :

REPEAT(줄, *i*)

인수 : 인수 '줄'은 이것으로부터 새로운 줄이 형성될 문자 또는 비트-줄을 나타낸다; 이 인수는 이진 규약 산수, 비트-줄, 문자-줄 또는 수치 문자 요소식 또는 배열 이름이 될 수 있다.

만일 비트나 문자-줄이 아닌 인수가 지정되면, 이는 함수가 불러내지기 전에 비트 또는 문자-줄로 변환된다.

인수 '*i*'는 십진 고른수 상수이어야 한다.

이는 '줄'이 이것에 추가해서 이어 붙어질 횟수를 나타낸다;  
'i'는 0보다 커야 한다.

결과: 이 이 함수에 의해 돌아가는 값은 '줄'이 추가해서  
'i'회 이어 붙여진 값이다. 바꿔 말하면, 돌아가는 값은 '줄'  
의 값이  $i+1$  회로 된 줄이다.

보기:  $BSTR \longrightarrow 101/B$ 이면:

$A = REPEAT(BSTR, 6);$

은 A에 '101101101101101101101/B'를 보낼 것이다. 즉 '101'  
B가 7번 반복으로 이어 붙여진 것이다.

SUBSTR

정의: SUBSTR는 주어진 줄로부터 정해진 길이의 버금줄(sub-  
string)을 뽑아서 불러낸 점으로 보내준다. (SUBSTR는 또한  
유사 변수(pseudo-variable)로도 사용될 수 있다)

인용:

SUBSTR ( 줄, i, j )

인수: 인수 '줄'은 이것으로부터 버금줄이 뽑혀질 줄을 나타  
낸다; 이 인수는 이진 규약 산수, 비트-줄, 문자-줄 또는 수치  
문자 요소 식 또는 배열 이름이 될 수 있다. 만일 '줄'이  
비트 또는 문자-줄이 아니면, 함수가 불러 내지기 전에 비트 또  
는 문자-줄로 변환된다. 인수 'i'는 버금줄의 개시 점을 나  
타내며 'j'는 버금줄의 길이를 나타낸다. 인수 'i'는 고른  
수 상수로 변환될 수 있는 요소 식이어야 한다. (이는 만일 '줄'  
이 배열일 때만은 배열 이름이 될 수 있다); 'j'는 십진  
고른수 상수이어야 한다. 만일 'i'가 배열이면, 이는 '줄'과  
동일한 한계를 가져야 한다.

「줄」의 길이가 「k」라 하면, 인수 「i」와 「j」는 다음 조건을 만족해야 한다.

1. j는 k보다 작거나 같고 1보다 크거나 같아야 한다.
2. i는 k보다 작거나 같고 1보다 크거나 같아야 한다.
3. i + j - 1의 값은 k보다 작거나 같아야 한다.

그러므로, 「i」와 「j」로 지정된 버금줄은 「줄」 이내에 들어야 한다. 조건 1은 편성자에 의해 검사되나; 조건 2와 3은 검사되지 않음을 유의하라.

결과: 이 함수에 의해 돌아가는 값은 첫째 인수의 i번째 문자 또는 비트에서 시작하고 j개 문자 또는 비트에 걸치는 버금줄이다.

보기: AAA가 길이 30의 문자-줄이면:

```
ITEM = SUBSTR(AAA,7,14);
```

는 AAA의 7번째 자리부터 14개 문자를 뽑아낸 버금줄을 보낸다.

UNSPEC

정의: UNSPEC는 주어진 값의 내부 규약 표현(internal coded representation)이 되는 비트-줄을 돌려보낸다. (UNSPEC는 또 유사 변수로 사용될 수 있다.)

인용:

```
UNSPEC(x)
```

인수: 인수 「x」는 내부 규약 표현이 발견될 산수, 문자-줄 또는 지침 값(指針 값 pointer value)(요소 식 또는 배열만)이 될 수 있다; 「x」는 비트-줄이 될 수 없다.

결과: 이 함수에 의해 돌아가는 값은 「x」의 내부 규약 포

현이다. 이 표현은 비트-줄 형 (bit-string form) 이다. 이 줄의 길이는 'x'의 속성에 따라 좌우되고 조직/360에서 다음과 같이 정해진다.

1. 만일 'x'가 정도 (p, q)의 FIXED BINARY이면, 길이는 32.
2. 만일 'x'가 정도 (p, q)의 FIXED DECIMAL이면, 길이는  $8 * \text{FLOOR}((P+2)/2)$ 로 정해진다.
3. 만일 'x'가 정도 p의 FLOAT BINARY이면, 길이는
  - a. 만일 p가 21보다 작거나 같으면, 32.
  - b. 만일 p가 21보다 크면, 64.
4. 만일 'x'가 정도 p의 FLOAT DECIMAL이면, 길이는
  - a. 만일 p가 6보다 작거나 같으면, 32.
  - b. 만일 p가 6보다 크면, 64.
5. 만일 'x'가 길이 n의 문자-줄이거나, 또는 문자-줄 값이 길이 n인 수치 문자 항목이면, 길이는  $8 * n$ ; D-편성자에서, n은 8보다 크지 않아야 한다.
6. 만일 'x'가 지침이면, 길이는 32.

[2] 산수 내조립 함수 (算数 内組立 函数 arithmetic built-in function)

산수 내조립 함수에 의해 돌아가는 모든 값은 규약 산수 형이다. 이들 함수의 인수는 역시 규약 산수 형이어야 한다. 만약 인수가 규약 산수가 아니면, 함수가 불러내지기 전에, 이는 제 6 장, '자료 변환'에 기술된 규칙에 따라 규약 산수로 변환된다. 앞으로의 함수 기술에서, 인수에 관한 인용은 만일 변환이 필요하면 항상 변환된 인수를 의미함을 유의하라.

어떤 함수 기술에서는, 문구(文句) '최고 특성으로 변환된다'는 말이 사용된다; 이는 제 I 부, 제 4 장, '식'의 '산수 변환에서 자료 변환' 절에서 말한대로 특성이 혼합되어 있는 것에 관한 규칙이 대두됨을 의미한다.

보통, 산수 내조립 함수의 인수는 요소·식 또는 배열 이름이 될 수도 있다. 만일 인수가 배열 이름이면, 이 함수에 의해 돌아가는 값은 인수와 동일 차원과 한계를 가진 배열이다. (함수는 배열의 개개 요소에 작용된다)

그러므로, 보기를 들면, 배열 인수가 절대 값을 내는 함수에 쓰이면, 돌아가는 값은 배열이며, 개개 요소는 개개 요소(인수의)의 절대 값이다.

별명이 없는 한, 돌아가는 값의 기수, 척도, 정도(hase, scale, precision)는 제 6 장 '자료 변환'에 있는 바처럼 식 연산항 변환을 위한 규칙에 따라 결정된다.

이들 중 많은 함수에서, 부호 N이 사용된다. 이 부호는 값이 가질 수 있는 최대 정도를 나타낸다. 이는 조직/360 실무에서 다음과 같이 정해진다:

N은

FIXED	DECIMAL	15
FLOAT	DECIMAL	16
FIXED	BINARY	31
FLOAT	BINARY	53

ABS

정의: ABS는 주어진 수의 절대 값을 찾아서 불러낸 점으로 보내준다.

인용 :

ABS(X)

인수 : 절대값이 (絶対값 absolute value) 발견될 수는 "x"로 주어진다.

결과 : 이 함수에 의해 돌아가는 값은 "x"의 절대값이다. 기수, 척도, 정도는 "x"의 그것과 같다.

ADD

정의 : ADD는 두개의 주어진 값의 합계를 내서 불러낸 점으로 보내준다.

이 함수는 더하기 연산의 결과의 정도를 관리할 수 있게 해준다.

인용 :

ADD(x, y, p, [, q])

인수 : 인수 "x"와 "y"는 더해질 값을 나타낸다. 인수 "p"와 "q"는 결과의 정도를 지정하는 십진 고른수 상수이어야 한다; "q"는 기호가 붙을 수 있다.

결과의 척도가 고정점수이면, 4개 인수가 모두 지정되어야 한다; 만일 결과의 척도가 부동점수이면, "q"는 지정되지 않는다.

결과 : "x"와 "y"의 합계가 이 함수에 의해 돌아가는 값이다. 결과의 정도는 "p"와 "q"에 의해 결정된다; 이 정도는 함수의 실행동안 유지된다.

BINARY

정의 : BINARY는 주어진 값을 이진수로 바꿔서 불러낸 점으로 돌려보낸다.

이 함수는 이진 변환 결과의 정도를 관리할 수 있게 한다.

인용 :

$\text{BINARY}(x[, p[, q]])$

인수 : 첫번째 인수 'x'는 이진수로 변환될 값을 나타낸다. 인수 'p'와 'q'는, 만일 지정되면, 이진수 결과의 정도를 말해주는 십진 고른수 상수이어야 한다; 'q'는 기호가 붙을 수 있다. 고정점수 결과의 정도는 (p, q); 부동점수 결과의 정도는 (p)이다. 만일 'p'와 'q'가 모두 생략되면, 결과의 정도는 제 6 장 '자료 변환'에 있는 기수 변환 (base conversion)에 관한 규칙에 따라 결정된다. 'q'는 부동점수 인수에서는 생략되어야 함을 유의하라.

결과 : 이 함수에 의해 돌아가는 값은 'x'와 등가인 이진수이다. 이 값의 척도는 'x'의 그것과 같다. 정도는 'p'와 'q'에 의해 주어진다.

CEIL

정의 : CEIL은 주어진 값보다 크거나 같은 것중 가장 작은 고른수를 결정하여 불러낸 점으로 보내준다.

인용 :

$\text{CEIL}(x)$

인수 : 인수는 'x'

결과 : 이 함수에 의해 돌아오는 값은 'x'보다 크거나 같은 고른수 중에서 제일 작은 고른수 (integer)이다. 기수, 척도 정도는 한가지 예외를 빼고는 'x'의 그것과 같다; 만일 'x'가 정도 (p, q)의 고정점수이면, 결과의 정도는 다음과 같이 정해진다.

$(\text{MIN}(N, \text{MAX}(p-q+1, 1)), \emptyset)$

## DECIMAL

정의: DECIMAL은 주어진 값을 십진수로 변환시켜 불러낸 점으로 보내준다. 이 함수는 십진수로 변환한 결과의 정도를 관리할 수 있게 한다.

인용:

DECIMAL ( x [, p [, q] ] )

인수: 첫째 인수 'x'는 십진수로 변환될 값이다. 인수 'p'와 'q'는, 지정되면, 십진 결과의 정도를 제공하는 십진 고른수 상수이어야 한다; 'q'는 기호가 붙을 수 있다. 고정점수 결과의 정도는 (p, q); 부동점수 결과의 정도는 (p)이다. 만일 'p'와 'q'가 모두 생략되면, 결과의 정도는 제6장, '자료 변환'에 있는 기수 변환에서 주어진 규칙에 따라 결정된다. 부동점수 인수인 경우는 'q'가 생략되어야 함을 유의하라.

결과: 이 함수에 의해 돌아오는 값은 인수 'x'와 등가인 십진수이다; 이의 정도는 'p'와 'q'에 의해 주어진다.

## DIVIDE

정의: DIVIDE는 주어진 값을 주어진 다른 값으로 나누어 불러낸 점으로 돌려보낸다. 이 함수는 나누기 연산 결과의 정도를 관리할 수 있게 한다.

인용:

DIVIDE ( x, y, p [, q] )

인수: 인수 'x'는 나누어지는 수이며 인수 'y'는 나누는 수이다. 인수 'p'와 'q'는 ('q'는 기호가 붙을 수 있다) 결과의 정도를 지정하는 십진 고른수 상수이어야 한다. 만일 결과가 고정점수이면, 4개 인수 전부가 지정되어야 한다; 만일 결

과가 부동점 수이면, "q"는 지정되어서는 안된다.

결과: 이 함수에 의해 돌아가는 값은 "x"를 "y"로 나눈 몫이다. 결과의 정도는 "p"와 "q"에 의해 결정된다; 이 정도는 함수의 실행 동안 유지된다.

#### FIXED

정의: FIXED는 주어진 값을 고정점 척도(fixed-point scale)로 바꿔 불러낸 점으로 보내준다. 이 함수는 고정점수 변환 결과의 정도를 관리할 수 있게 해준다.

인용

FIXED(x [, p [, q]])

인수: 첫번째 인수 "x"는 고정점수로 변환될 값을 나타낸다. 인수 "p"와 "q"가 지정되면 결과의 정도(p, q)를 제공하는 십진 고른수 상수이어야 한다. ("q"는 기호가 붙을 수 있다) 조직/360 실무에서, 만일 "p"와 "q"가 생략되면, "p"는 이진수 "x"에 대해 15로 대행되고 십진수 "x"에 대해 5로 대행된다; "q"는 0으로 대행된다.

결과: 이 함수에 의해 돌아가는 값은 인수 "x"와 등가인 고정점수이며; 이의 정도는 (p, q)이다.

#### FLOAT

정의: FLOAT는 주어진 값을 부동점수로 바꿔서 불러낸 점으로 보내준다. 이는 부동점수 변환 결과의 정도를 관리할 수 있게 한다.

인용:

FLOAT(x [, p])

인수: 첫째 인수 "x"는 부동점수로 변환될 값을 나타낸다.

둘째 인수 'p'는, 이것이 지정되는 경우, 결과의 정도를 주는 십진 고른수 상수이어야 한다. 조직/360에서, 만일 'p'가 생략되면, 이는 이진수 'x'에 대해 21, 십진수 'x'에 대해 6으로 대행된다.

결과: 이 함수에 의해 돌아오는 값은 'x'와 등가인 부동점수이다; 이것의 정도는 'p'이다.

#### FLOOR

정의: FLOOR는 주어진 값보다 크지 않은 최대 고른수를 결정하여 그 고른수를 불러낸 점으로 보내준다.

인용:

FLOOR(x)

인수: 인수는 'x'이다.

결과: 이 함수에 의해 돌아가는 값은 'x'를 넘지 않는 최대 고른수이다.

이 값의 기수, 척도 정도는 한가지 예의를 빼고는 'x'의 그것과 같다: 만일 'x'가 정도(p, q)의 고정점수 값이면, 결과의 정도는:

$\text{MIN}(N, \text{MAX}(p - q + 1, 1), 0)$

#### MAX

정의: MAX는 둘 이상의 주어진 식의 조(組)로부터 최고값의 식을 선별하여 그 값을 불러낸 점으로 보내준다.

인용:

$\text{MAX}(x_1, x_2, \dots, x_n)$

인수: 둘 이상의 인수가 주어져야 한다.

결과: MAX에 의해 돌아가는 값은 최대값을 가진 인수의 값이

다. 돌아가는 값은 지정된 모든 인수 중에서 최고 특성에 부합되게 변환된다. 만일 인수가 고정점 수이고 다음의 정도를 가진다면,

$$(P_1, q_1), (P_2, q_2), \dots, (P_n, q_n)$$

그러면 결과의 정도는 다음과 같다.

$$(\text{MIN}(N, \text{MAX}(P_1 - q_1, \dots, P_n - q_n) + \text{MAX}(q_1, \dots, q_n)), \text{MAX}(q_1, \dots, q_n))$$

MIN

정의: MIN은 둘 이상의 주어진 식의 조 중에서 최저값의 식을 선별하여 그 값을 불러낸 점으로 보내준다.

인용:

$$\text{MIN}(x_1, x_2, \dots, x_n)$$

인수: 둘 이상의 인수가 주어져야 한다.

결과: MIN에 의해 돌아가는 값은 최저값의 인수이다. 돌아가는 값은 지정된 모든 인수 중에서 최고 특성에 부합되게 변환된다. 만일 인수가 고정점수이고 다음의 정도를 가지면,

$$(P_1, q_1), (P_2, q_2), \dots, (P_n, q_n)$$

그러면 그 결과의 정도는 다음과 같다.

$$(\text{MIN}(N, \text{MAX}(P_1 - q_1, \dots, P_n - q_n) + \text{MAX}(P_1, \dots, P_n)), \text{MAX}(q_1, \dots, q_n))$$

MOD

정의: MOD는 하나의 수를 다른 하나로 나눈 결과의 나머지를 선별하여 불러낸 점으로 보내준다.

인용:

$$\text{MOD}(x_1, x_2)$$

인수 : 두개의 인수가 주어져야 한다. 함수가 불러내지기 전에, 각 인수의 기수, 척도는 제 6 장, '자료 변환'에 있는 식 연산항 변환 규칙에 따라 변환된다.

결과 : MOD 에 의해 돌아가는 값은  $x_1$  을  $x_2$  로 나누어 고른 수 몫 (integer quotient) 을 산출하고 나머지의 양수이다. 만일 결과가 부동점수이면, 이것의 정도는 이 인수 중 높은 쪽의 정도가 된다. (이 말은,  $p = \text{MAX}(p_1, p_2)$ ); 만일 결과가 고정점수이면, 이것의 정도는 다음과 같이 정해진다 :

$$(\text{MIN}(N, p_2 - q_2 + \text{MAX}(q_1, q_2), \text{MAX}(q_1, q_2)))$$

여기서 :

( $p_1, q_1$ ) 과 ( $p_2, q_2$ ) 는 각각 ' $x_1$ ' 과 ' $x_2$ ' 의 정도이다. 기수와 척도는 변환된 인수의 그것과 같다.

#### MULTIPLY

정의 : MULTIPLY 는 두개의 주어진 값의 곱 (product 積) 을 찾아서 불러낸 점으로 보내준다. 이는 곱하기 연산 결과의 정도를 관리할 수 있게 한다.

인용 :

MULTIPLY ( $x_1, x_2, p [, q]$ )

인수 : 인수  $x_1$  과  $x_2$  는 곱해진 값을 나타낸다. 인수 ' $p$ ' 와 ' $q$ ' 는 (' $q$ ' 는 기호가 붙을 수 있다) 결과의 정도를 지정하는 십진 고른수 상수이어야 한다.

만일 결과가 고정점 값이면, 세개 인수 모두가 지정되어야 한다; 만일 결과가 부동점 수이면, ' $q$ ' 는 지정되어서는 안된다.

결과 : ' $x_1$ ' 과 ' $x_2$ ' 의 곱은 이 함수에 의해 돌아가는 값이다. 결과의 정도는 ' $p$ ' 와 ' $q$ ' 에 의해 결정된다; 이 정도

는 함수의 실행동안 유지된다.

#### PRECISION

정의: PRECISION은 주어진 값을 지정된 정도로 변환시켜 불러낸 점으로 보내준다. 이는 왼쪽 자르기를 위해 사용되지 못한다.

인용:

PRECISION (  $x$ , p [, q ] )

인수: 첫째 인수 ' $x$ '는 지정된 정도로 변환될 값을 나타낸다. 인수 ' $p$ '와 ' $q$ '는 (' $q$ '는 기호가 붙을 수 있다) 결과의 정도를 지정하는 십진 고른수 상수이어야 한다. 만일 ' $x$ '가 고정점수 값이면, ' $p$ '와 ' $q$ '는 지정되어야 한다; 그렇지만, 만일 ' $q$ '가 없으면, PL/I 편성자는 경고 선언문을 발하고 영의 값으로 대행된다. 만일 ' $x$ '가 부동점수이면, 단지 ' $p$ '만이 지정되어야 한다.

결과: 이 함수에 의해 돌아가는 값은 지정된 정도로 변환된 ' $x$ '의 값이다. 기수와 척도는 ' $x$ '의 그것과 같다.

#### ROUND

정의: ROUND는 지정된 숫자에서 주어진 값을 사사오입하여 (四捨五入) 불러낸 점으로 보내준다.

인용:

ROUND ( 식, n )

인수: 첫째 인수 ' $식$ '은 규약 산수 또는 수치 문자이어야 한다. 이는 사사오입될 값을 나타내는 요소 식 또는 배열 이름이다; 둘째 인수 ' $n$ '은 ' $식$ '의 값이 사사오입될 숫자 자리를 지정하는 우기호 십진 고른수 상수이어야 한다. ' $n$ '은

식'의 십진(또는 이진) 소수점 오른쪽에 있는 숫자의 수보다 작아야 한다.

결과: 만일 '식'이 고정점수이면, ROUND는 십진(또는 이진) 소수점 오른쪽에 있는 n번째 자리에서 반올림된 '식'의 값이다. 결과의 기수와 척도는 '식'의 기수와 척도와 같다; 결과의 정도는:

$$(\text{MIN}((p+1), N), q)$$

만일 '식'이 부동점수 식이면, 둘째 인수는 무시되고, 식' 값의 내부 부동점수 표현의 가장 오른쪽 비트가 0이면 1로되고; 만일 가장 오른쪽 비트가 1이면, 식' 값은 불변이다. 돌아가는 값의 기수, 척도, 정도는 '식' 값의 그것과 같다.

음수의 반올림은 그 수의 절대값이 반올림됨을 유의하라.

보기: 만일 x가 값 36.24976을 가진 정도(7,5)의 고정점 십진 변수이고, y와 z가 정도(6,4)의 고정점수 십진 변수라 하면, 그리고 다음 문을 실행했다고 하면,

$$y = \text{ROUND}(x, 3);$$

$$z = \text{ROUND}(x, 4);$$

y의 값은 36.2500이고 z의 값은 36.2498이다.

SIGN

정의: SIGN은 값이 양수, 음수 또는 영인가를 결정하여, 불려낸 점으로 보내준다.

인용:

SIGN(x)

인수: 인수는 'x'이다.

결과: 이 함수는 다음 규칙에 따라 태단 정도(15)인 고정점

이진 값을 보내준다.

1. 만일 'x'가 영보다 크면, 돌아가는 값은 1.
2. 만일 'x'가 영이면 돌아가는 값은 영.
3. 만일 'x'가 영보다 작으면, 돌아가는 값은 -1.

#### TRUNC

정의: TRUNC는 다음과 같이 주어진 값을 고른수로 자른다:

첫째 이는 주어진 값이 양수, 음수, 영인가를 결정한다. 만일 값이 음수이면, TRUNC는 그 값보다 큰 제일 작은 고른수를 보내준다; 만일 값이 양수 또는 영이면, TRUNC는 이 값을 넘지 않는 제일 큰 고른수를 보내준다.

인용:

#### TRUNC(x)

인수: 인수는 'x'이다.

결과: 만일 'x'가 영보다 작으면, TRUNC에 의해 돌아가는 값은 CEIL(x)이다. 만일 'x'가 영보다 크거나 같으면, TRUNC에 의해 돌아가는 값은 FLOOR(x)이다. 어느 경우에도, 결과의 기수, 척도는 'x'의 그것과 같다. 만일 'x'가 부동점수이면, 정도는 그대로이다. 만일 'x'가 정도(p, q)의 고정점수이면, 결과의 정도는:

$$(\text{MIN}(N, \text{MAX}(p - q + 1, 1)), 0)$$

### [3] 수학 내조립 함수 (数学 内組立 函数 mathematical built-in functions)

수학 내조립 함수에 쓰는 모든 인수는 규약 산수 형(coded arithmetic form)과 부동점수 척도(floating-point scale)

이어야 한다. 이 규칙에 맞지 않은 어떤 인수도 함수가 불러내지기 전에 제 6 장, "자료 변환"에서 논술된 규칙에 따라서 규약 산수와 부동점수로 변환된다. 그렇지만, 다음의 함수 기술에서, 만일 변환이 필요하다면, 인수에 관한 인용은 항상 변환된 인수를 의미한다.

보통, 수학 내조립 함수에 쓰는 인수는 요소 식 또는 배열 이름이 될 수 있다. 만일 인수가 배열이면, 이 함수에 의해, 돌아가는 값은 인수와 동일한 차원과 한계를 가진 배열이다 (함수는 배열의 개개 요소에 수행된다).

그래서, 보기로, 여현 (餘弦 cosine) 함수  $\cos$ 에 보내진 배열 인수는 배열 인수의 결과가 된다. 개개 요소는 배열 인수에 있는 대응 요소의 여현이다.

모든 수학 내조립 함수는 규약 산수 부동점수 값을 보내준다. 이들 값의 기수와 정도는 언제나 인수의 그것과 같다.

이절 끝에 있는 그림 7-1은 수학 내조립 함수에 관해 신속한 참조를 제공한다.

#### ATAN

정의: ATAN은 주어진 값의 역정접 (逆正接 arctangent)을 찾아 호도 (弧度 radian)로 표시된 결과를 점으로 보내준다.

인용:

ATAN ( x [, y ] )

인수: 인수 "x"는 항상 지정되어야 한다; 인수 "y"는 선택이다. 만일 "y"가 생략되면, "x"는 역정접이 찾아질 값을 나타낸다.

만일 "y"가 지정되면, 역정접이 계산될 값은 식  $x/y$ 로 취

해진다. 이 경우, 'x'와 'y' 모두 동시에 영이 될수 없다.

결과: 'x'가 하나만 지정되면, ATAN에 의해 돌아가는 값은 'x'의 역정점이며, 'x'는 호도법 (radian)으로 표시된다,

여기서:

$$-\pi/2 < \text{ATAN}(x) < \pi/2$$

만일 'x'와 'y'가 둘다 지정되면, 이 함수에 의해 돌아가갈 수 있는 값은 다음과 같이 정해진다.:

1.  $y > 0$ 와 어떤  $x$ 에 대해, 이 값은 호도로 된 역정점 ( $x/y$ )이다.
2. 만일  $x > 0$ 이고  $y = 0$ 이면, 이 값은 ( $\pi/2$ ) 호도이다.
3. 만일  $x \geq 0$ 이고  $y < 0$ 이면, 이 값은 ( $\pi + \text{역정점}(x/y)$ ) 호도이다.
4. 만일  $x < 0$ 이고  $y = 0$ 이면, 이 값은 ( $-\pi/2$ ) 호도이다.
5. 만일  $x < 0$ 이고  $y < 0$ 이면, 이 값은 ( $-\pi + \text{역정점}(x/y)$ ) 호도이다.

#### ATAND

정의: ATAND는 주어진 값의 역정점을 산출하여 도(度 degree)로 표시된 값을 불러낸 점으로 보내준다.

인용:

ATAND ( x [, y ] )

인수: y가 생략되면, 'x'가 역정점이 산출될 값이 된다.

만일 'y'가 지정되면, 역정점이 산출될 값은 식  $x/y$ 로 표현된다; 이 경우, 'x'와 'y'는 동시에 영이 되지 못한다.

결과: 만일 'y'가 생략되면, 이 함수에 의해 돌아가는 값은 호도로 표시된 'x'만의 역정점이다. 여기서:

$$-90 < \text{ATAND}(x) < 90$$

만일  $y$ 가 지정되면, 이 함수에 의해 돌아가는 값은 값이 도 (度) 로 표시되고 호도로 표시되지 않은 점을 계하고는  $\text{ATAN}(x,y)$  이다. ( " ATAN " 을 보라 ) ; 이 말은, 돌아가는 값은 다음과 같이 정해진다는 것이다 :

$$\text{ATAND}(x,y) = ( 180 / \pi ) * \text{ATAN}(x,y)$$

내  
7

ATANH

정의 : ATANH는 주어진 값의 역 쌍곡 정접을 찾아 불러낸 점 으로 보내준다.

인용 :

ATANH(x)

인수 : 역 쌍곡 정접 ( 逆 双曲 正接 inverse hyperbolic tangent ) 이 찾아질 값은 " x " 에 의해 표현된다. " x " 의 절대값은 1 보다 작아야 한다 ; 이는, 이것이 만일  $AB(x) \geq 1$  이면 오류가 된다는 말이다.

결과 이 함수에 의해 돌아가는 값은 " x " 의 역 쌍곡 정접이 다.

COS

정의 : COS는 주어진 값의 여현 ( 餘弦 cosine ) 을 찾아, 이는 호도로 표시된 " x " 이다. 그 결과를 불러낸 점으로 보내준다.

인용 :

COS(x)

인수 : 여현이 찾아질 값은 " x " 로 주어진다. 이 값은 호도 로 표시되어야 한다.

결과 : 이 함수에 의해 돌아가는 값은 " x " 의 여현이다.

COSD

정의 : COSD는 도로 표시된 주어진 값의 여현을 찾아 그 결과를 불러낸 점으로 보내준다.

인용 :

COSD(x)

인수 : 여현이 찾아질 값은 'x'로 주어진다 ; 이 값은 도 (度 degree)로 표시되어야 한다.

결과 : 이 함수에 의해 돌아가는 값은 'x'의 여현이다.

COSH

정의 : COSH는 주어진 값의 쌍곡 여현 (双曲 餘弦 hyperbolic cosine)을 찾아 그 값을 불러낸 점으로 보내준다.

인용 :

COSH(x)

인수 : 쌍곡 여현이 찾아질 값은 'x'로 주어진다.

결과 : 이 함수에 의해 돌아가는 값은 'x'의 쌍곡 여현이다.

ERF

정의 : ERF 주어진 값의 오차 함수 값 (誤差 函數 값 error function)을 찾아 그 결과를 불러낸 점으로 보내준다.

인용 :

ERF(x)

인수 : 오차 함수값이 찾아질 값은 'x'로 표현된다.

결과 : 이 함수에 의해 돌아가는 값은 다음과 같이 정의된다.

$$ERF(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

ERFC

정의 : ERFC는 주어진 값에 대한 오차 함수값 (ERF)의 보수

외  
8

를 (補數 complement) 찾아 그 결과를 불러낸 점으로 보내준다.

인용:

$$\text{ERFC}(x)$$

인수: 인수 'x'는 오차 함수값의 보수가 찾아질 값을 나타낸다.

결과: 이 함수에 의해 돌아가는 값은 다음과 같이 정의된다:

$$\text{ERFC}(x) = 1 - \text{ERF}(x)$$

EXP

정의: EXP는 e (자연 대수의 밑)를 주어진 지수로 제공하여 그 결과를 불러낸 점으로 보내준다.

인용:

$$\text{EXP}(x)$$

인수: 인수 'x'는 e가 제공될 지수를 지정한다.

결과: 이 함수에 의해 돌아가는 값은 e에 'x'를 제공한 것이다.

LOG

정의: LOG는 주어진 값의 자연 대수 (밑이 e)를 찾아 그것을 불러낸 점으로 보내준다.

인용:

$$\text{LOG}(x)$$

인수: 인수 'x'는 자연 대수가 찾아질 값이다; 이는 영보다 작거나 같으면 안된다.

결과: 이 함수에 의해 돌아가는 값은 'x'의 자연 대수 (natural logarithm 自然对数)이다.

### LOG<sub>10</sub>

정의 : LOG<sub>10</sub>은 주어진 값의 상용 대수 (常用 对数 common logarithm) (이는 밑 (base)이 10)를 불러낸 점으로 보내준다.

인용 :

### LOG<sub>10</sub>(x)

인수 : 인수 'x'는 상용대수가 찾아질 값을 나타낸다; 이 값은 영보다 커야 한다.

결과 : 이 함수에 의해 돌아가는 값은 'x'의 상용대수이다.

### LOG<sub>2</sub>

정의 : LOG<sub>2</sub>는 주어진 값의 이 대수 (二 对数 binary logarithm) (이는 밑이 2이다)를 찾아 불러낸 값으로 보내준다.

인용 :

### LOG<sub>2</sub>(x)

인수 : 인수 'x'는 이대수가 찾아질 값이다; 이는 영보다 커야 한다.

결과 : 이 함수에 의해 돌아가는 값은 'x'의 이대수이다.

### SIN

정의 : SIN은 호도로 표시된 주어진 값의 정현 (正弦 sine)을 찾아, 불러낸 점으로 보내준다.

인용 :

### SIN(x)

인수 : 인수 'x'는 정현이 찾아질 값이다; 이는 호도로 표시되어야 한다.

결과 : 이 함수에 의해 돌아가는 값은 "x"의 정현이다.

SIND

정의 : SIND는 도로 표시된 주어진 값의 정현을 찾아, 불러낸 점으로 보내준다.

인용 :

SIND(x)

인수 : 인수 "x"는 정현이 찾아질 값이다; "x"는 도(度)로 표시되어야 한다.

결과 : 이 함수에 의해 돌아가는 값은 "x"의 정현이다.

SINH

정의 : SINH는 주어진 값의 쌍곡 정현(双曲 正弦 hyperbolic sine)을 찾아 그 결과를 불러낸 점으로 보내준다.

인용 :

SINH(x)

인수 : 인수 "x"는 쌍곡 정현이 찾아질 값이다.

결과 : 이 함수에 의해 돌아가는 값은 "x"의 쌍곡 정현이다.

SQRT

정의 : SQRT는 주어진 값의 제곱근(平方根)을 찾아 불러낸 점으로 보내준다.

인용 :

SQRT(x)

인수 : 인수 "x"는 제곱근이 찾아질 값이다; 이는 영보다 작으면 안된다.

결과 : 이 함수에 의해 돌아가는 값은 "x"의 제곱근이다.

TAN

정의 : TAN은 호도로 표시된 주어진 값의 장접(正接 tangent)

을 찾아, 불러낸 점으로 보내준다.

인수:

$TAN(x)$

인수: 인수  $x$ 는 정점이 찾아질 값이다;  $x$ 는 호도로 표시되어야 한다.

결과: 이 함수에 의해 돌아가는 값은  $x$ 의 정점이다.

TAND

정의: TAND는 호도로 표시된 주어진 값의 정점을 찾아, 그 결과를 불러낸 점으로 보내준다.

인용:

$TAND(x)$

인수: 인수  $x$ 는 정점이 찾아질 값을 나타낸다;  $x$ 는 호도로 표시되어야 한다.

결과: 이 함수에 의해 돌아가는 값은  $x$ 의 정점이다.

TANH

정의: TANH는 주어진 값의 쌍곡 정점 (双曲 正接 hyperbolic tangent) 을 찾아 그 결과를 불러낸 점으로 보내준다.

인용:  $TANH(x)$

$TANH(x)$

인수: 인수  $x$ 는 쌍곡 정점이 찾아질 값을 나타낸다.

결과: 이 함수에 의해 돌아가는 값은  $x$ 의 쌍곡 정점이다.

수학 함수의 요약

그림 7-1은 수학 내조립 함수를 요약한다. 이를 이용함에

있어, 독자는 다음 사항에 주의해야 한다.

외  
9

1. 모든 인수는 규약 산수이며 부동점 척도이어야 하거나, 또는 그런 수로 바뀌질 수 있어야 한다.
2. 개개 함수에 의해 돌아가는 값은 언제나 부동점 수이다.
3. 오류 조건 ( error condition ) 은 PL/I 언어에서 정의된 것과 같다.  
 D - 편성자에 의해 점검되는 자세한 오류 조건은 IBM 조직 / 360 원반과 테이프 운영 조직, PL/I 프로그래머 안내서 ( IBM System / 360 Disk and Tape Operating systems, PL/I Programmer's Guide, Order No. Gc24-9005 ) 에서 볼 수 있다.

함수 인용	돌아가는 값	오류 조건
ATAN(x)	호도의 역정점(x) $-(\pi/2) < \text{ATAN}(x) < \pi/2$	
ATAN(x,y)	함수 설명을 보라	x = 0 이고 y = 0 이면 오류
ATAND(x)	도의 역정점(x) $-90 < \text{ATAND}(x) < 90$	
ATAND(x,y)	함수 설명을 보라	x = 0 이고 y = 0 이면 오류
ATANH(x)	$\tanh^{-1}(x)$	$\text{ABS}(x) \geq 1$ 이면 오류
COS(x)	cosine(x)	
x 는 호도		

함수 인용	돌아가는 값	오류 조건
COSD(x) x는 도	cosine(x)	
COSH(x)	cosh(x)	
ERF(x)	$\frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$	
ERFC(x)	1 - ERF(x)	
EXP(x)	e를 x제곱한 값	
LOG(x)	$\log_e(x)$	x = ≤ 0 이면 오류
LOG10(x)	$\log_{10}(x)$	x = ≤ 0 이면 오류
LOG2(x)	$\log_2(x)$	x = ≤ 0 이면 오류
SIN(x) x는 호도	sine(x)	
SIND(x)	sine(x)	
X는 도		
SINH(x)	sinh(x)	
SOPT(x)	$\sqrt{x}$	x < 0 이면 오류
TAN(x) x는 호도	tangent(x)	
TAND(x)	tangent(x)	
X는 도		
TANH(x)	tanh(x)	

[ 4 ] 배열 다루기 내조립 함수 (配列 array manipulation built-in functions)

여기에 기술되는 내조립 함수는 배열 다루기를 위해 사용될

수 있다. 모든 이들 함수는 배열 이름 인수를 요하고 단일 요소 값을 보내준다. 이들 함수는 요소 값을 보내주므로, 이들 중 어느것에 대한 인용도 요소 식으로 생각되어야 함을 유의하라.

#### ALL

정의 : ALL 은 주어진 비트-줄 배열의 모든 비트를 검사하여 그 결과를 요소 비트-줄 형으로 불러낸 점으로 보내준다. 요소 비트-줄은 주어진 배열 요소의 비트들이 모두 일(1)인지 아닌가를 가리킨다.

인용 :

ALL(x)

인수 : 인수 'x' 는 비트-줄의 배열이다. 만일 요소가 비트가 아니면, 이들은 비트로 변환된다.

결과 : 이 함수에 의해 돌아가는 값은 'x' 에 있는 요소 길이와 같은 하나의 비트-줄이다. ('x' 의 모든 요소는 물론 같은 길이를 가져야 한다), 그리고 이의 비트 값은 다음 규칙에 따라 결정된다.

만일 'x' 에 있는 모든 요소의  $i$  번째 비트가 1 이면, 결과의  $i$  번째 비트는 1 ; 아니면 결과의  $i$  번째 비트는 0 이다.

#### ANY

정의 : ANY 는 주어진 비트-줄 배열의 비트들을 검사하여 그 결과를 비트-줄 요소 형으로 불러낸 점으로 보내준다.

이 요소 비트-줄은 주어진 배열 요소의 대응 비트 중의 하나가 적어도 1 인가 아닌가를 가리킨다.

인용 :

### ANY(x)

인수 : 인수 'x'는 비트-줄의 배열이다. 만일 요소가 비트-줄이 아니면, 이는 비트-줄로 변환된다.

결과 : 이 함수에 의해 돌아가는 값은 'x'에 있는 요소와 길이가 같은 문자-줄이다. 그리고 이 비트 값은 다음 규칙에 따라 결정된다.

만일 'x'에 있는 어떤 요소의  $i$ 번째 비트가 1이면, 결과의  $i$ 번째 비트가 1이고; 아니면 결과의  $i$ 번째 비트는 0이다.

### PROD

정의 : PROD는 주어진 배열의 모든 요소의 곱(積 product)을 찾아 그 곱을 불러낸 점으로 보내준다.

인용 :

### PROD(x)

인수 : 인수 'x'는 규약 산수 부동점수 요소의 배열이어야 한다. 만일 그렇지 않으면, 개개 요소는 바로 앞의 곱에 곱해지기 전에 규약 산수 부동점수로 변환된다.

결과 : 이 함수에 의해 돌아가는 값은 'x'에 있는 모든 요소의 곱이다.

결과의 척도는 부동점이고, 반면 기수와 정도는 'x'의 변환된 요소의 그것과 같다.

### SUM

정의 : SUM은 주어진 배열의 모든 요소의 합(合 sum)을 찾아 불러낸 점으로 보내준다.

인용 :

### SUM(x)

인수: 인수 'x'는 규약 산수 부동점수 요소의 배열이어야 한다. 만일 그렇지 않으면, 개개 요소는 앞의 합계에 더해지기 전에 규약 산수 부동점수로 변환된다.

결과: 이 함수에 의해 돌아가는 값은 'x'에 있는 모든 요소의 합이다.

결과와 척도는 부동점이고, 반면 기수와 정도는 인수의 변환된 요소의 그것과 같다.

## 제 2 절 기타의 내조립 함수(其他의 內組立 函數)

이 절에서 기술되는 함수는 서로간에 또는 다른 종류간에 공통점이 적다.

어떤 것은 인수를 요하고 어떤 것은 요하지 않는다.

### ADDR

정의: ADDR는 주어진 변수가 할당되어 있는 위치를 찾아 불러낸 값을 지침 값(指針 값 pointer value)을 보내준다.

이 지침 값은 변수가 할당되어 있는 위치를 가리킨다.

인용:

ADDR(x)

인수: 인수 'x'는 위치가 찾아질 변수이다. 이는 요소 변수, 배열, 구조체, 배열의 요소, 구조체의 요소가 될 수 있다. 이는 어느 형식의 자료도 어떤 종류의 기억소 종류가 될 수 있다.

결과: ADDR는 'x'가 할당되어 있는 위치와 동일한 지침 값을 보내준다. 'x'가 매개변수(媒介變數 parameter)이면,

돌아가는 값은 대응하는 인수 (가 인수 또는 다른 것) 의 것이다. 만일 'x' 가 기초된 변수 (based variable) 이면, 돌아가는 값은 'x' 에 선언된 지칭 변수로부터 결정된다; 만일 이 지칭 변수가 값을 갖지 않으면, ADDR에 의해 돌아가는 값은 부정 (不定) 이다.

DATE

정의 : DATE는 현재의 날짜를 불러낸 점으로 보내준다.

인용 :

DATE

인수 : 없음.

결과 : 이 함수에 의해 돌아가는 값은 yyymmdd로 표시되는 길이 여섯의 문자-줄이다. 여기서 :

yy는 현재의 연 (年)

mm은 현재의 달 (月)

dd는 현재의 일 (日)

보기 : 만일 현재의 일자가 1968.2.29일이면, 다음의 실행은

X = DATE ;

문자-줄 '680229'를 불러낸 점으로 보내주게 할 것이다.

주 : DATE 내조립 함수는 BUILTIN 속성으로 명시 선언되어야 한다.

NULL

정의 : NULL은 빈 지칭 값 (null pointer value) 을 불러낸 점으로 보내준다.

인용 :

NULL

인수 : 없음.

결과 : 이 함수에 의해 돌아가는 값은 빈 지침 값이다. D-변성자에서, 빈 지침 값은 특별한 지시하는 것 (indicator) 으로 사용될 수 있는 무효한 주소 (x 00FEFFFF) 이다.

주 : NULL 내조립 함수는 BUILTIN 속성으로 명시 선언되어야 한다.

#### STRING

정의 : STRING은 UNALIGNED 속성을 가진 주어진 구조체로부터 문자-줄을 형성하여 불러낸 점으로 보내준다. (이는 또한 유사 변수로 사용될 수 있다)

인용 :

STRING (구조체-이름) :

인수 : 인수 '구조체-이름'은 UNALIGNED 속성을 가진 구조체 이름이어야 한다. 이 구조체는 (structure-構造体) 문자-줄과/또는 수치문자 자료 (numeric character data) 만으로 내용을 다루어야 한다.

결과 : 이 함수에 의해 돌아가는 값은 '구조체-이름'에 있는 모든 요소를 잇댄 것으로 되는 문자-줄이다.

주 : 만일 잇댄 결과가 길이에서 225 자 (byte) 가 넘을 때는, 이 결과는 다음의 길이의 줄이 된다 :

MOD (구조체 이름의 길이, 255)

#### TIME

정의 : TIME은 현재의 시각 (時刻) 을 불러낸 점으로 보내준다.

인용 :

TIME

인수: 없음.

결과: 이 함수에 의해 돌아가는 값은 hhmmsssttt로 되는 길이 아홉의 문자-줄이다.

여기서:

hh는 현재 일의 시(時)

mm은 분

ss는 초

ttt는 천분의 일 초수이다.

보기: 만일 현재의 시각이 오후 4시, 23분, 19초, 80밀리초 (milliseconds)이면, TIME을 인용한 것은 문자-줄 '162319080'을 불러낸 점으로 보내줄 것이다.

주: TIME 내조립 함수는 BUILTIN 속성으로 명시 선언되어야 한다.

### 제3절 유사-변수 (類似變數 pseudo-variables)

일반적으로, 유사-변수란 다른 변수가 변수를 받아들이기 위해 나타날 수 있는 곳은 어디든지 나타날 수 있는 특정한 내조립 함수이다. 간단히 말해서, 이들은 받아들이는 난으로서 사용될 수 있는 내조립 함수이다. 유사-변수는 대입 문에서 등호(等號)의 왼쪽에 또는 GET 문의 자료 나열(資料 羅列 data list)에 나타날 수 있다. 이는 다른 곳에는 나타날 수 없다.

유사-변수는 품어질 수 없다. 보기로, 다음 문은 틀린다:

```
UNSPEC ( SUBSTR ( A , 1 , 2 ) ) = '00' B ;
```

세 개의 유사-변수가 있다. STRING, SUBSTR, UNSPEC. 이들은 내조립 함수와 같으므로, 개개 유사-변수에 관한 짚막한 기술만이 여기에 주어질 것이다; 자세한 것에 대해서는 해당 내조립 함수가 연구되어야 한다.

STRING

인용:

STRING (구조체 이름)

설명: 인수 '구조체 이름'은 문자-줄 그리고/또는 수치 문자 자료만으로 내용을 이루고 있는 구조체 이름이어야 한다. 원시 줄은 (source string) 모든 요소가 대입되거나, 또는 대입되는 줄이 나머지가 없을 때까지 '구조체 이름'의 요소에 일부분씩 대입된다.

SUBSTR

인용:

SUBSTR (줄,  $i$ ,  $j$ )

설명: SUBSTR에 대입되는 값은 내조립 함수 SUBSTR에서 정의된 바처럼 (한가지 예외는 SUBSTR 유사-변수에서 '구조체 이름'은 요소 변수이어야 한다) 문자-줄 또는 비트-줄 변수 '구조체 이름'의 버금줄 (substring)에 대입된다.

UNSPEC

인용:

UNSPEC(v)

설명: 문자 'v'는 산수, 문자-줄 또는 지침 형식의 요소 변수를 나타낸다.

이는 비트-줄 변수가 되지 못한다. UNSPEC에 대입되는 값은

값이 내지고, 비트-줄로 변환되고 (이것의 길이는 "v"의 속성에  
관계된다. -- UNSPEC 내조립 함수를 보라), 그런 다음에 "v"  
의 형식 변환없이 "v"에 대입된다.

## 제 8 장 : ON - 조건 ( ON - 條件 ON - conditions )

ON - 조건이란 FL/I 에서 ON 문을 수단으로 하여 지정될 수 있는 예외 조건 ( 例外條件 exceptional condition ) 을 말한다.

만일 조건이 가능하게 되어 있으면, 그 조건의 출현은 중단을 가져온다. 계속해서 그 중단은 그 조건을 위한 현재의 행위 지정 ( 行為指定 action Specification ) 의 실행 ( 実行 execution ) 을 초래 ( 招來 ) 한다. 만일 그 조건에 대한 ON 문이 효과가 없으면, 현재의 행위 지정은 그 조건에 대한 표준조직 행위 ( 標準組織行為 standard system action ) 가 된다.

만일 그 조건에 대한 ON 문이 효과가 있으면, 현재의 행위 지정은 SYSTEM 이거나, 이 경우 그 조건에 대해서 표준 조직 행위가 취해진다, 또는 ON - 단위 ( ON - 單位 ON - Unit ) 이다. 이 경우 프로그래머가 그 조건에 대해서 취해질 그 자신의 행위를 제공한다 ( 이는 빈 문 ( Null Statement 또는 GO TO 문이다 ).

만일 조건이 가능하지 않으면 ( 이는 불능인 경우 ) 그리고 그 조건이 출현하면, 중단이 일어나지 않고 오류가 초래될 것이다.

어떤 조건은 이것들이 조건 전치어에 ( 條件 前置語 Condition Prefix ) 의해 명시적 불능으로 되지 않는 한 언제나 가능하게 ( Enabled 可能하게 ) 된다; 다른것 ( 이는 SIZE ) 은 이것이 조건 전치어에 의해 명시로 ( 明示로 ) 가능하게 되지 않는 한 언제나 불능하게 ( 不能하게 Disabled ) 된다; 또 그 외는 언제나 가능하게 되고 불능하게 될 수 없다.

조건전치어에 의해 명시적으로 불능하게 되지 않는 한 언제나 가능한 것은 :

CONVERSION

FIXEDOVERFLOW

OVERFLOW

UNDERFLOW

ZERODIVIDE

위 조건의 개개는 NO를 앞에 붙여 지정한 조건 전치어에 의해 불가능하게 될 수 있다. 그래서 조건 전치어에서 다음의 하나는 해당 조건을 불가능하게 할 것이다 :

NOCONVERSION

NOFIXEDOVERFLOW

NOOVERFLOW

NOUNDERFLOW

NOZERODIVIDE

이런 조건 전치어는 해당 조건을 이 전치어 (NO가 붙은)의 범위 (範圍 scope)에 걸쳐 불가능하게 만든다. 이 조건은 이 범위 밖에서는 가능하게 남는다 (제1부 제1장 "예외조건 취급과 프로그램 검토"를 조건 전치어의 범위에 관한 논술을 위해 보라).

거꾸로, 이것이 조건 전치어에 의해 가능하게 되지 않는 한 언제나 불가능한 조건은 SIZE 조건 전치어에 이 조건이 나타남은 이 조건을 전치어의 범위에 걸쳐 가능하게 만든다; 이 조건은 이 범위 밖에서는 불가능하게 남는다.

더욱이, NOSIZE라 지정하는 조건 전치어는 이 전치어의 범위에 걸쳐 SIZE 조건을 불가능하게 할 것이다.

다른 모든 조건은 항상 가능하고 프로그램 기간동안 남아있다.

이들 조건들은 :

ENDFILE

ENDPAGE

ERROR

KEY

RECORD

TRANSMIT

## 제 1 절 이 장의 구성내용

이 장은 각 조건을 논리적 부류로 묶고, 그 부류 안에서 영자 순으로 나타낸다. 일반적으로 다음 지식이 각 조건에 주어진다 ;

1. 일반형식 (一般形式 general format) -- 이것은 조건 이름과 그 외의 것으로 내용을 이룰 때만 주어진다.
2. 설명 (說明) -- 조건이 일어날 수 있는 환경을 포함한 조건의 논술. 가능하게 되어 있는 조건은 언제나 SIGNAL 문에 의해 제기 (提起) 됨을 유의하라 ; 이 사실은 설명에 들어 있지 않다.
3. 결과 (結果) -- 조건을 출현시키는 원인이 되는 연산 결과 이는 조건이 불능할 때도 이것이 가능할 때와 같이 적용된다. 어떤 경우는 결과는 부정이다 (不定) ;  
이런 결과가 예측되지 않는다.  
이것은 이런 경우마다 말이 있을 것이다.
4. 표준 조직 행위 -- 중단이 일어나고 프로그래머가 이 중단을 취급하기 위한 ON-단위를 지정하지 않았을 때 조직 (system)에 의해 취해지는 행위
5. 상태 (狀態 Status) -- 프로그램의 시작에서 조건의 가능, 불능 상태와 어떻게 조건이 불능하게 (만일 될 수 있다면) 또는 가능하게 될 수 있는가의 지적
6. 정상 복귀 (正常復歸) (Normal return) -- 빈 문의 ON-단위의 결과로써 통제 (統制 Control)가 돌아가는 점. GO TO 문 ON-단위는 비정상 On-단위 끝내기이다. 만일 조건이 SIGNAL 문에 의해 일어났으면, 정상복귀는 언제나

SIGNAL 문 바로 다음 문임을 유의하라. 조건 ENDFILE, KEY, CONVERSION 은 이들과 연관된 빈 문 ON - 단위를 가질 수 없고, 그러므로, 정상 복귀는 이들 조건들에 대해 이루어질 수 없음을 유의하라.

조건들은 다음과 같이 부류로 묶어졌다.

1. 계산 조건 (計算條件 Computational conditions) -- 자료 취급 식의 값내기, 계산에 연관된 조건들. 이들은 :

CONVERSION	FIXEDOVERFLOW
OVERFLOW	SIZE
UNDERFLOW	ZERODIVIDE

2. 입력/출력 조건 (入力/出力 input / output) -- 자료 전송 (data transmission 資料転送) 에 연관된 조건들. 이들은 :

ENDFILE	ENDPAGE
KEY	RECORD
TRANSMIT	

3. 조직 행위 조건 (組織行為 條件 system action condition) -- 한개 조건이 나타난 후에 취해지는 표준 조직 행위를 확장하는 설비를 준비하는 조건 (이는 ERROR) 줄로 구성된 문자 줄 자료에 시도될 때 일어난다. 이런 시도는 내부에서나 또는 입력/출력 연산 동안 행해질 수 있다. 보기로 이 조건은 비트 줄로 변환될 문자 줄에 0 또는 1 이 아닌 문자가 존재할 때 또는 산수로 해석될

수 없는 문자가 STREAM 전송 연산에서 산수 변수에 들어가려 할 때 일어난다.

## 제 2 절 계산 조건

### CONVERSION 조건

설 명 : CONVERSION 조건은 변환 (變換 Conversion) 이 수행될 변환에 무효한 문자 모든 문자-출 자료의 변환은 왼쪽에서 오른쪽 순으로 한 문자씩 옮겨지며 조건은 첫번째 부적한 문자에 일어난다.

그런 문자가 감지되면, 중단이 일어나고 (물론 CONVERSION 이 불가능하지 않다고 가정한 경우) 그 조건을 위한 행위지정이 실행된다.

결 과 : CONVERSION 이 나타날 때 전체 결과 난의 내용은 부정 (不定) 이다.

표준 조직 행위 : ON - 단위가 없으면, 조직은 전언문 (傳言文 Message) 을 인쇄하고 ERROR 조건을 일으킨다.

상 태 : CONVERSION 은 프로그램 전반에 걸쳐 가능하다. 단 NOGONVERSION 을 지정한 조건 전치어의 범위 안에서는 제외된다.

정상 복귀 : 빈 ON - 단위는 이 조건에 지정되지 못한다.

### FIXEDOVERFLOW 조건

설 명 : FIXEDOVERFLOW 조건은 고정점 산수 연산 결과의 길이

가 N을 넘을 때 나타난다. 조직 / 360 실무에서, N은 십진 고정점 값에 대해 15이고 이진 고정점 값에 31이다. 이 조건은 또 비산수 연산(非算數 演算)에서 일어날 수 있다.

결 과 : 무효한 산수 연산의 결과는 알 수 없다.

표준 조직 행위 : ON - 단위가 없으면, 조직은 전언문을 인쇄하고 ERROR 조건을 일으킨다.

상 태 : FIXEDOVERFLOW는 프로그램 전반에 가능하며, NOFIXE-DOVERFLOW를 지정한 조건 전치어의 범위 안에서는 제외된다.

정상 복귀 : 만일 빈 ON - 단위 (null on - unit)가 이 조건에 지정되면, 통제는 중단점 바로 다음의 점으로 돌아간다.

#### OVERFLOW 조건

설 명 : OVERFLOW 조건은 부동점 수의 크기가 허용된 최대값을 넘을 때 나타난다. (조직 / 360 실무에서, 부동점수 또는 중간 결과의 크기가 대략  $10^{75}$  또는  $2^{252}$ 를 넘지 않아야 한다.

결 과 : 이런 부적당한 부동점수의 값은 정해지지 않는다.

표준 조직 행위 : ON - 단위가 없으면 조직은 전언문을 인쇄하고 ERROR 조건을 일으킨다.

상 태 : OVERFLOW 는 프로그램 전반에 가능하며, NOVERFLOW 를  
지정한 조건 전처어의 범위 안에서는 제외된다.

정상 복귀 : 만일 빈 ON - 단위가 이 조건에 지정되면, 통제는  
중단점 바로 다음 점으로 돌아간다.

#### SIZE 조건

설 명 : SIZE 조건은 고순위 (高順位, high - order) (이는 가  
장 왼쪽) 의 영 아닌 이진수 또는 십진수 숫자가 대입 연산에  
서 상실 (喪失) 될 때 (대입은, 변수에 대입 또는 중간 결과에  
대입) 나타난다. 이런 상실은 서로 다른 자료 형식, 서로 다른  
기수, 서로 다른 척도, 서로 다른 정도를 포함하는 변환으로부터  
초래된다.

SIZE 조건은 기본점에서 FIXEDOVERFLOW 조건과 다르다, 이것  
은, FIXEDOVERFLOW 는 산출된 고정점수 값의 크기가 N (허용최  
대값) 을 넘을 때 나타나고, 반면에 SIZE 는 자료 항목에 대입될  
값의 크기가 선언된 (또는 태만) 자료 항목의 크기를 넘을 때  
일어난다. SIZE 는 FIXEDOVERFLOW 가 그 값의 산출에서 일어  
나고 아니고에 관계 없이 값의 대입에서 일어날 수 있다.

결 과 : 윗러는 크기의 값을 받는 자료 항목의 내용은 부정 (不  
定) 이다.

표준 조직 행위 : ON - 단위가 없으면, 조직은 전언문을 인쇄하고  
ERROR 조건을 일으킨다.

상 태 : SIZE는 NOSIZE 조건 전치어의 범위 안과 프로그램 전반의 어디서나 불능하며, SIZE를 지정한 조건 전치어의 범위 안에서는 가능하다.

정상 복귀 : 만일 빈 ON-단위가 이 조건에 지정되었으면, 통제는 중단점 바로 다음의 점으로 간다.

#### UNDERFLOW 조건

설 명 : UNDERFLOW 조건은 부동점수의 절대 크기가 허용된 최소값보다 작을 때 나타난다 (조직 / 360 실무에서, 부동점수의 절대치 크기가 대략  $10^{-78}$  또는  $2^{-260}$  보다 작으면 안된다). UNDERFLOW 조건은 같은 수가 배질 때는 일어나지 않는다. (이는 흔히 유의 오류 (significance error) 라 불리운다).

D-편성자에서 식  $X**(-y)$ 는  $X**Y$ 의 역수를 취해 값이 내지기 때문에, OVERFLOW 조건이 UNDERFLOW 조건 대신 일어난다.

결 과 : ON-단위가 없거나 빈 ON-단위가 있으면 무효한 부동점 값은 0으로 되고, 아니면 결과는 부정이다.

표준 조직 행위 : ON-단위가 없으면, 조직은 전언문을 인쇄하고 중단이 일어난 점으로부터 계속한다.

상 태 : UNDERFLOW는 프로그램 전반에 가능하며, NOUNDERFLOW를 지정한 조건 전치어의 범위 안에서는 제외된다.

정상 복귀 : 만일 BIN ON - 단위가 이 조건에 지정되면, 통제는 중단점 바로 다음 점으로 돌아간다.

#### ZERODIVIDE 조건

설명 : ZERODIVIDE 조건은 영을 가지고 나누려는 시도가 있을 때 일어난다. 이 조건은 부동점과 고정점수 나누기에서 일어난다.

주 : 고정점 산수에서,  $PL/I$  은 기계의 FIXEDPOINT DIVIDE 조건이 일어날 때는 언제나 ZERODIVIDE 조건을 일으킴을 유의하라

결과 : 영에 의한 나누기의 결과는 부정이다.

표준 조직 행위 : ON - 단위가 없으면, 조직은 전언문을 인쇄하고 ERROR 조건을 일으킨다.

상태 : ZERODIVIDE 조건은 프로그램 전반에 가능하며, NOZERODIVIDE 를 지정한 조건 전치어의 범위 안에서는 제외이다.

정상 복귀 : 만일 BIN ON - 단위가 이 조건에 지정되면, 통제는 중단점 바로 다음점으로 돌아간다.

#### 제 3 절 입력/출력 조건 ( 入力/出力 条件 input/output conditions )

입력/출력 조건은 언제나 가능하고 조건 전치어로 나타나지 못한다 ; 이들은 ON, SIGNAL, REVERT 문에만 지정될 수 있다.

## ENDFILE 조건

일반 형식 : ENDFILE (파일-이름)

설명 : ENDFILE 조건은 GET, READ, REWRITE 연산 중에 일어날 수 있다 ; 이것은 GET 또는 READ 문에서 호명된 파일의 파일 구분자 (파일 区分子 File delimiter) (이는 파일의 끝임을 나타내는 것)가 읽혀지거나 또는 INDEXED 파일을 인용하는 READ나 REWRITE 문에 있는 열쇠 (Key)가 파일에 있는 최고 열쇠보다 높을 때 야기된다. 계속한 GET나 READ 문은 파일 구분자를 지나서 읽을 것이다.

ENDFILE 조건은 표준 조직 파일 SYSIN에 대해서 일어나지 못한다.

ENDFILE이 일어난 뒤에는, 파일은 닫혀야 한다.

표준 조직 행위 : ON - 단위가 없으면, 조직은 전언문을 인쇄하고 ERROR 조건을 일으킨다.

상태 : ENDFILE 조건은 언제나 가능하게 되어있다 ; 이는 불가능하게 되지 못한다.

정상 복귀 : 빈 ON - 단위는 이 조건에 지정되지 못한다.

## ENDPAGE 조건

일반 형식 : ENDPAGE (파일-이름)

"파일 이름"은 PRINT 속성을 가진 파일이름이어야 한다.

설명 : ENDPAGE 조건은 PUT문이 현재의 지면을 (紙面 Page)

위해 지정된 한계를 지나서 새로운 줄을 시작하려고 할 때 일어난다. 이 한계는 OPEN 문에서 PAGESIZE 선택항에 의해 지정 될 수 있다. 만일 PAGESIZE가 지정되지 않으면, 설치시 정해진 조직 한계가 적용된다. 이 한계를 넘으려 하는 시도는 자료 전송시에 될 수 있고 (PUT 문에서 지정된 어떤 서식 항목에 포함되어), LINE 또는 SKIP 선택항에 의해서도 될 수 있다. ENDPAGE 는 매 지면에 한번만 일어난다. ENDPAGE가 일어날 때, 현재의 줄 수 (line number) 는 PAGESIZE 선택항에 의해 (또는 태만에 의해) 지정된 것보다 하나 크다; 이는 또 동일 지면에 계속 인쇄할 수 있다. ENDPAGE가 일어난 뒤에, 새로운 지면이 다음 방법의 하나로써 시작될 수 있다:

1. PAGE 선택항 또는 PAGE 서식 항목의 실행
2. 현재의 줄 수보다 작거나 같은 줄 수를 지정한 LINE 선택항 또는 LINE 서식 항목

이들 중에 하나가 일어날 때 새로운 지면이 PAGE 선택항이 실행될 때와 같은 식으로 시작된다. 이는 ENDPAGE 조건이 일어나지 않으며 현재의 줄은 1로 세워진다.

만일 ENDPAGE가 자료 전송 중에 일어나고, 빈 ON - 단위로부터 돌아오면, 자료는 현재의 줄에 기록된다. 만일 ENDPAGE가 LINE 또는 SKIP 선택항으로부터 일어나고, 빈 ON - 단위로부터 돌아오면, LINE 이나 SKIP 에 의해 지정된 행위는 무시된다.

ENDPAGE 조건은 표준 조직 화일 SYSLSST 에서는 일어날 수 없다.

표준 조직 행위: ON - 단위가 없으면, 조직은 새로운 지면을 시작한다.

상 태 : ENDFPAGE 는 언제나 가능하고 ; 불능할 수 없다.

정상 복귀 : 이 조건에 대한 ON - 단위의 실행에 따라, FUT 문의 실행이 위에서 말한 바 대로 계속한다.

신호된 ( Signalled ) ENDFPAGE 조건은 화일에 효과가 없다.

#### KEY 조건

일반 형식 : KEY ( 화일 - 이름 )

설 명 : KEY 조건은 열쇠있는 기록마디에 관한 연산 중에만 일어날 수 있다. 이는 다음 어느 경우에도 일어난다.

1. 열쇠있는 기록마디가 READ 또는 REWRITE 문에서 발견되지 않는다. 이 경우, 자료가 읽혀들어갈 변수의 내용은 예측할 수 없다.
2. WRITE 문으로 중복된 열쇠를 추가하려 할 때 ( REGIONAL (3) 화일에서는 아님 )
3. 열쇠가 정확치 못하게 지정되었다.
4. 열쇠있는 기록마디를 추가할 자리가 없다.

표준 조직 행위 : ON - 단위가 없으면 조직은 전언문을 인쇄하고 ERROR 조건을 일으킨다.

상 태 : KEY 는 언제나 가능하다 ; 이는 불능하게 되지 못한다.

정상 복귀 : 빈 ON - 단위가 이 조건에 지정될 수 없다.

## RECORD' 조건

일반 형식 : RECORD (파일-이름)

설 명 : RECORD 조건은 READ, WRITE, REWRITE 또는 LOCATE

연산 동안에만 일어날 수 있다. 이는 다음의 하나로 일어난다;

1. 기록마디의 크기가 변수의 크기보다 크다 (F, V, F 형식에서).

2. 기록마디의 크기가 변수의 크기보다 작다 (F 형식에서).

만일 기록마디의 크기가 변수의 크기보다 크면, 기록마디에있는 여분 (餘分)은 입력에서 상실되고 출력에서 예상할 수 없다. 만일 기록마디의 크기가 변수의 크기보다 작으면, 변수에 있는 여분은 출력에서 전송되지 않고 입력에서 변함 없다.

고정 블러크된 기록마디에서, RECORD 조건은 블러크에 있는 기록마디 수 만큼 일어난다.

표준 조직 행위 : ON - 단위가 없으면, 조직은 전언문을 인쇄하고 ERROR 조건을 일으킨다.

상 태 : RECORD 는 언제나 가능하다 ; 이는 불가능하지 못한다.

정상 복귀 : 빈 ON - 단위의 실행에 따라, 실행은 RECORD 가 일어난 바로 다음 문으로 계속한다.

## TRANSMIT 조건

일반 형식 : TRANSMIT (파일-이름)

설 명 : TRANSMIT 조건은 어느 입력/출력 연산 동안에도 일어날

수 있다. 이는 불변의 전송 오류에 의해 일어나며, 그런 결과로 전송되는 어떤 자료도 잠재적으로 부정확하다. 입력 동안, 이 조건은 잠재적으로 부정확한 자료 항목이나 기록마디의 대입 후에 일어난다.

출력 동안, 이 조건은 잠재적으로 부정확한 자료 항목이나 기록마디의 전송이 시도된 뒤에 일어난다.

추가해서 TRANSMIT 조건은 만일 기록마디의 길이가 완충역 길이 ( Buffer length ) 와 틀리면 STREAM INPUT 화일에 일어난다.

표준 조직 행위 : ON - 단위가 없으면, 조직은 전언문을 인쇄하고 ERROR 조건을 일으킨다.

상 태 : TRANSMIT 는 언제나 가능하고, 불가능하게 되지 못한다.

정상 복귀 : 빈 ON - 단위의 실행에 따라서, 처리는 STREAM 입력/출력에서 다음 자료 항목으로, 또는 RECORD 입력/출력에서 다음 문으로 계속한다.

#### 제 4 절 조직 행위 조건 ( 組織 行為 条件 system action conditions )

##### ERROR 조건

설 명 : ERROR 조건은 다음 환경 아래서 일어난다.

1. "오류 전언문을 인쇄하고 ERROR 조건을 일으킨다" 는 행위가 있을 ON - 조건에 대한 표준 조직 행위의 결과로써

2. 프로그램 실행 동안 일어난 오류의 결과로써 ( 여기에는  
ON - 조건이 없다. )

표준 조직 행위 : ON - 단위가 없으면, D - 편성자는 전언문을 인  
쇄하고 통제를 운영 조직 통제 프로그램으로 보내준다.

상 태 : ERROR 는 언제나 가능하다 ; 이는 불가능하게 되지 못한다.

정상 복귀 : 빈 ON - 단위의 실행에 따라서 통제가 운영 조직  
통제 프로그램 ( operating system control program ) 으로 돌  
아간다.

## 제 9 장 속성 (屬性 attributes)

PL/I 프로그램에 나타나는 이름은 여러 가지 의미 중의 하나를 가질 수 있다.

보기를 들면, 이는 산수 자료를 인용하는 변수가 되기도 하고; 화일 이름; 문자 줄을 인용하는 변수; 문 명찰(文名札 Statement label); 문 명찰을 인용하는 변수가 되기도 한다. 이름이 표현하는 성질, 또는 특성(보기로, 산수 변수에 의해 표현되는 산수적 특성)과 또 한편 이름 자체의 성질(범위(Scope), 기억소 종류(Storage class)들과 같은 것)은 함께 이름과 연관될 수 있는 속성의 조(組 Set)를 구성한다.

속성(屬性 attribute)은 편성자(Compiler)에게 DECLARE 문에서 지정되는 포식어(標識語 identifier)에 고유한 의미를 부여하게 한다. 보기로, 변수가 산수자료 속성이면, 기수(基数 base), 척도(尺度 scale), 정도(精度 precision) 속성이 이 이름과 연관되어야 한다. 연관된 속성은 DECLARE 문에서 지정된 것이거나 태만(怠慢 default)에 의해 대행(代行)된 것이다.

이 장은 여러 가지 속성에 대해 논술 한다. 속성은 기능(機能)에 따라 모아지고 영자순(英字順)으로, 자세한 논술이 개개 속성에 대한 규칙(規則), 태만(default), 형식(形式 format)을 보이면서 뒤따른다.

### 제 1 절 속성의 지정 (屬性的 指定 specification of attributes)

DECLARE 문에서 지정된 속성은 빈자(blank)로 떨어진다.

차원 (次元 dimension), 길이 (length), FILE, 정도 속성 지정을 제외하고는, 이들은 어떤 순서로 나타나도 된다. 차원 속성 지정은 배열 이름의 바로 다음에 와야 한다; 길이와 정도 속성 지정은 그들의 연관된 속성 중에 하나를 뒤따라야 한다; FILE 속성은 화일 이름 선언(宣言 declaration)에서 가장 먼저 나타나야 한다. 쉼표 (comma)가 이것이 DECLARE문에 있는 마지막 이름이 아닌 한 특정 이름을 위한 마지막 속성 지정 (또는 이름 자체, 이는 만일 이름에 속성이 지정 안된 경우) 뒤에 와야 한다.

속성을 인자로 만들기 (因子로 만들기 factoring of attributes) 여러개 이름에 공통인 속성은 선언에서 많은 포식어에 대해 동일 속성을 반복 지정하는 것을 피하기 위해 인자로 만들 수 있다. 인자로 하는 것은 이름들을 괄호에 싸고, 적용될 속성의 조를 이 뒤에 오게 함으로써 목적을 달성한다. 인자로 된 모든 속성은 모든 이름에 적용된다. 인자로 된 속성은 어떤 이름에 대해서도 무시될 수 없다. 그러나 나열 (羅列 list)에 있는 어떤 이름도 인자로 된 속성과 모순이 없으면 다른 속성을 받을 수 있다.

D-편성자에서, 인자로 만들기는 9 수준까지 품어질 수 있다. 그런 품기에 대한 설명은 아래의 네번째 보기를 보라.

괄호에 싸인 나열에 있는 이름은 쉼표로 떨어진다.

주: 구조체 수준 번호 (structure level numbers)는 또한 인자로 될 수 있다, 그러나 인자로 된 수준 번호는 괄호에 싸인 나열 앞에 와야 한다.

보기 :

```
DECLARE ( A, B, C, D ) BINARY FIXED ( 31 ) ;  
DECLARE ( E DECIMAL ( 6,5 ), F CHARACTER(10) ) STATIC ;  
DECLARE 1 A, 2 ( B, C, D ) BINARY FIXED(15) . . . ;  
DECLARE ((A,B) FIXED(10), C FLOAT(5)) EXTERNAL;
```

수준 번호는 구조체의 요소에만 인자로 될 수 있다; 이는, 만일 인자되기가 구조체 선언에 나타나면, 해당 항목들이 구조체 요소로서 인식된다.

보기 : 다음의 선언에서

```
DCL 1 A, 2 ( B, C, D ), 3 ( E, F, G ); B, C, D, E, F, G는  
구조체 A의 요소로 대항될 것이다, 그래서 논리 수준 ( 論理  
水準 logical level ) 이 2로 부여될 것이다.
```

다음의 구조체로 하기 위해서

```
DCL 1 A, 2 C, 2 D, 3 E, 3 F, 3 G; D는 인자로 된 괄호에  
서 빠져나와야 한다.
```

## 제 2 절 자료 속성 ( 資料 屬性 data attributes )

### [ 1 ] 문제 자료 ( 問題資料 problem data )

문제 자료를 위한 속성은 산수와 줄 변수를 기술하기 위해 사용된다. 산수 변수는 자료 항목의 기수, 척도, 정도를 지정하는 속성을 가진다. 줄 변수는 변수가 문자 줄 또는 비트 줄을 표현하는 가를 지정하는 속성과 유지 될 길이를 지정하는 속성을 가진다.

산수 자료 속성은 ( arithmetic data ) :

BINARY | DECIMAL

FIXED | FLOAT

( 정도 )

PICTURE

줄 자료 속성은 ( string data ) :

BIT | CHARACTER

( 길이 )

PICTURE

다른 속성들이 자료 변수에 선언될 수 있다. INITIAL 속성은 변수에 주어질 초기값 ( 初期값 ) 을 지정한다.

DEFINED 속성은 자료 항목이 다른 자료에 할당되어 있는 동일 기억역을 차지 함을 지정한다.

기억소 종류와 범위 속성이 또한 자료에 제공될 수 있다.

ALIGNED 와 UNALIGNED 속성은 기억소에 있는 자료 요소의 자리잡기를 지정한다.

배열 변수 ( 配列変數 array variables ) 에 대해서, 차원 속성은 배열의 차원 수와 한계 ( 限界, bound ) 를 지정한다.

## [ 2 ] 프로그램 통제 자료 ( 統制資料 program control data )

프로그램 통제 자료에 대한 속성은 연관된 이름이 프로그래머에 의해 그의 프로그램 실행을 통제하기 위하여 사용됨을 지정한다.

프로그램 통제 속성은 LABEL 과 POINTER 이다.

제 3 절 입구 이름 속성 (入口 이름属性 Entry name attributes)

입구 이름 속성은 입구 이름으로서 선언되는 이름을 알려주며 그 입구점 (entry point) 의 설비를 기술한다. 보기로, 속성 BUILTIN은 선언의 범위 안에서 해당 이름을 인용하는 것은 동일 이름에 대한 내조립 함수 또는 유사 변수로서의 인용으로 해석됨을 지정한다.

입구 이름 속성은:

ENTRY  
RETURNS  
BUILTIN

제 4 절 파일 서술 속성 (파일 叙述 属性 file description attributes)

파일 서술 속성은 파일 이름으로서의 포식어를 설정하고 그 파일에 관한 특성을 서술한다, 보기로, 파일에 있는 차료가 어떻게 전송되는가, 파일의 기록마디는 완충역을 지날 것인가. 만일 동일 파일 이름이 둘 이상의 외부 수속 (external procedure) 에서 선언되면 이 선언은 모순되지 않아야 한다. 파일 이름 매개변수 (媒介変数) 를 빼고, 파일 이름은 언제나 EXTERNAL 속성을 가져야 한다; 이는 명시로나 태만에 의해서나; 파일 이름 매개변수는 범위 속성을 갖지 못한다.

파일 서술 속성은:

FILE  
 STREAM | RECORD  
 PRINT  
 INPUT | OUTPUT | UPDATE  
 SEQUENTIAL | DIRECT  
 BUFFERED | UNBUFFERED  
 BACKWARDS  
 ENVIRONMENT (선택항 나열)  
 KEY ED

#### 제 5 절 범위 속성 (範圍屬性 scope attributes)

범위 속성은 하나의 이름이 다른 외부 수속에 알려지는 가의 여부를 지정한다.

범위 속성은 EXTERNAL 과 INTERNAL이다.

이름의 범위에 관해서는, 제 1 부, 제 7 장, "이름의 인지"를 보라. 프로그램의 동일 포식어에 대한 모든 외부 선언은 동일 이름에 관한 선언으로서 연결된다. 이 이름의 범위는 이 포식어를 위한 모든 외부 선언의 범위를 합한 것이다.

동일 포식어에 관한 모든 외부 선언(外部 宣言 external declarations)에서, 선언된 속성은 일치되어야 한다. 그것은 모든 선언이 한개의 이름을 가지기 때문이다. 보기로, 만일 포식어 ID가 한 수속(Procedure)에서 EXTERNAL 화일 이름으로서 선언되고 동일 프로그램의 다른 수속에서 EXTERNAL 입구 이름으로서 선언 된다면 오류가 될 것이다.

INTERNAL 속성은 선언된 이름이 선언이 블록에 포함된 블록에 포함된 블록을 제외하고는 다른 어떤 블록에도 알려질 수 없음을 지정한다. 이는 화일 이름에는 지정되지 못한다.

동일 포식어가 한개의 블록에서 받은 속성이 다른 블록에서 받은 속성과 일치하는가에 관계 없이 두개 이상의 블록에서 INTERNAL 속성을 받을 수 있다. 그것은 편성자가 그런 선언을 서로 다른 이름을 인용하는 것으로 고려하기 때문이다.

#### 제 6절 기억소 종류 속성 (記憶所 種類屬性 storage class attributes)

기억소 종류 속성은 자료 변수를 위한 기억소의 형식을 지정한다.

기억소 종류 속성은:

STATIC

AUTOMATIC

BASED (지침 - 변수)

#### 제 7절 속성의 열거

다음은 영자순으로, 속성에 관해 상세히 설명한다. 택일 속성 (択一 屬性 alternative attributes) 은 함께 논술된다.

(1) ALIGNED와 UNALIGNED (자료 속성)

ALIGNED와 UNALIGNED 속성은 접근 속력 또는 기억소 경제성 등에 영향을 주기 위해 기억소의 자료 요소 자리잡기를 지정한다. 이들은 요소, 배열, 구조체에 지정될 수 있다. 그러나 소구조체 (minor-structure)에는 안된다.

ALIGNED는 조직/360 실무에서 자료 요소가 자료 형식의 요구에 따라 기억소 경제에 할당됨을 지정한다.

조직/360 실무에서 UNALIGNED 속성은 자료 요소가 앞서는 자료 요소에 잇대어서 기억되고, 낱말 또는 두낱말 (word or double word) 항목은 길이 4 또는 8의 문자 줄과 비슷한 방법으로 다음번 가능한 자경계 (byte boundary)에 배치됨을 지정한다.

주: D-수준 편성자는 조직/360에서 자할당, 이는 문자 종류 자료, 만 허용하는 자료에만 UNALIGNED를 뒷받침해주므로, UNALIGNED 또는 ALIGNED의 지정은 어떤 경우에 효과가 없거나 틀린다. 그러나, 전 PL/I (全 full PL/I) 언어와 비교해 보기 위해 가끔 명시 선언된 ALIGNED에 의해 태만 속성 UNALIGNED를 무효로 하는 것도 필요하다.

일반 형식: ALIGNED | UNALIGNED

일반 규칙:

1. 이들이 원래 요소 속성이지만, ALIGNED와 UNALIGNED는 어떤 배열이나 구조체에도 적용된다. 그러나 소구조체에는 안된다. 그래서 속성은 ALIGNED나 UNALIGNED 속성으로 명시 선언되지 않은 모든 포함되어 있는 요소에 적용된다.
2. 포함되어 있는 배열이나 소구조체에 관한 속성의 적용은 이

를 포함하는 구조체에 의해서 포함되어 있는 집합체 요소에 달리 적용 되려하는 ALIGNED나 UNALIGNED 속성 보다 우선한다.

3. 문자종류 자료를 가진 중첩 정의 (overlay defining) 에서, 정의된 항목과 기초 항목의 중첩 부분은 다같이 정렬되지 않아야 한다. (즉 undigned 팀) 그 밖의 모든 형식의 정의에서, 등가인 항목은 다같이 ALIGNED이거나 다같이 UNALIGNED이어야 한다.

4. 수속 블럭내기에 있는 인수 (引數) 의 ALIGNED와 UNALIGNED 속성은 대응하는 매개변수의 속성과 같아야 한다.

5. 만일 기초원 변수가 다른 변수를 생성키 위해 인용되면, 양쪽의 ALIGNED와 UNALIGNED 속성은 같아야 한다.

6. ALIGNED와 UNALIGNED에 관한 태만 대행 (怠慢代行) 은 요소 단위로 적용된다.

7. D-편성자에서, 비트 줄을 포함하고 있는 구조체는 속성 ALIGNED를 가져야 한다. 이는 태만 속성 UNALIGNED는 이 경우 허용 안되기 때문이다.

태만 대행 (怠慢代行 default assumptions) :

1. 태만은 요소 수준에 적용된다. 문자 줄 자료와 수치 문자 자료에 관한 태만은 UNALIGNED이다; 다른 모든 형식의 자료에 관해서는 태만은 ALIGNED이다.

UNALIGNED 속성은 문자-줄 자료와 수치 문자 자료에만 지정될 수 있다. ALIGNED 속성은 비트-줄에 관해서 배열이나 구조체에 들어 있는 형 (形 form) 으로 명시 선언되어야

한다. 후자의 경우 대구조체 (major structure) 에 대해

ALIGNED 속성을 지정하는 것으로 족하다.

2. 모든 연산자와 내조립 함수에서 ALIGNED와 UNALIGNED에 관한 태만이 결과의 요소에 적용된다.

3. 상수 (常数 constants) 는 ALIGNED 또는 UNALIGNED에 관해 태만을 취한다.

## (2) AUTOMATIC, STATIC, BASED (기억소 종류 속성)

기억소 종류 속성 (storage class attributes) 는 자료 변수에 사용되는 기억소 할당 (storage allocation) 의 형식 (type) 을 지정하기에 사용된다.

AUTOMATIC 는 기억소가 기억소 선언이 내적인 블럭에 들어감에 따라 할당됨을 지정한다. 이 기억소는 이 블럭에서 나올 때 해제 된다.

STATIC 는 기억소가 프로그램이 실행 될 때 할당되고 프로그램 실행이 끝날 때까지 해제되지 않음을 지정한다.

BASED (지침-변수) 속성은 기억소에서 다른 자리에 이용될 수 있는 자료의 기술이 되는 변수를 지정한다.

일반 형식:

STATIC | AUTOMATIC | BASED (지침-변수)

일반 규칙

AUTOMATIC 과 BASED 변수는 INTERNAL 범위만 가진다.

STATIC 변수는 INTERNAL 이나 EXTERNAL 범위를 가질 수

있다.

2. 기억소 종류 속성은 입구 이름, 화일 이름, 구조체의 요소, DEFINED자료, 매개변수에 지정되지 못한다.
3. 구조체 변수에서, 기억소 종류 속성은 대구조체에만 받을 수 있다. 그러면 속성은 구조체의 첫 요소에 적용된다.
4. 다음 규칙은 기초된 변수의 용법을 지배한다.
  - a. 지침 변수(指針變數)는 POINTER 속성으로 명시 선언되어야 한다. 지침 변수는 첨자 없는 요소 변수이어야 하고 구조체의 요소도 아니어야 한다; 이는 BASED 속성을 갖지 못한다.
  - b. 기초된 변수를 인용할 때, 대행된 자료 속성은 기초된 변수와 같다. 반면에 연관된 지침 변수는 자료의 위치를 가르킨다.
  - c. 기초된 변수(based variable)는 이미 있는 자료를 가르키고 기술하거나 LOCATE 문을 사용해서 완충역(buffer)에 있는 기억소를 취득하기에 사용된다.
  - d. 속성 EXTERNAL은 기초된 변수 선언에 나타나지 못한다. 그러나 기초된 변수는 EXTERNAL 지침 변수와 더불어 쓸 수 있다.

태만 대행:

1. 기억소 종류 속성이 지정되지 않고 범위가 INTERNAL이면, AUTOMATIC가 대행된다.
2. 만일 기억소 종류 속성이 지정되지 않고 범위가 EXTERNAL이면, STATIC가 대행된다.

3. 만일 기억소 종류와 범위 속성이 하나도 지정되지 않으면, AUTOMATIC가 대행된다.

### (3) BACKWARDS (화일 서술 속성)

BACKWARDS 속성은 자기 테이프에 있는 SEQUENTIAL INPUT 화일의 기록마디가 역순으로, 즉 마지막부터 처음 기록마디 순으로 접근됨을 지정한다.

일반 형식:

BACKWARDS

일반 규칙

1. BACKWARDS 속성은 RECORD 화일에만 적용된다; 그래서, STREAM 속성과는 모순이다.
2. BACKWARDS 속성은 테이프 (tape) 화일에만 적용된다.
3. BACKWARDS 속성은 가변 길이 기록 마디에는 지정 못된다.

### (4) BASED (기억소 종류 속성)

AUTOMATIC 를 보라

### (5) BINARY 와 DECIMAL (산수 자료 속성)

BINARY 와 DECIMAL 속성은 이진수 (二進數) 또는 십진수 (十進數 decimal)로서 산수 변수에 의해 표현되는 자료 항목의 기수 (基数 base) (또는 진법 (進法)) 를 지정한다.

일반 형식 :

BINARY | DECIMAL

일반 규칙 :

BINARY 또는 DECIMAL 속성은 PICTURE 속성에 함께 지정될 수 없다.

태만 대행 :

선언되지 않은 포식어 ( 또는 차원 , UNALIGNED , ALIGNED , ENTRY , RETURNS , 기억소 종류 , 범위 속성 중의 하나 이상으로만 선언된 포식어 ) 는 첫째 문자에 쫓아 배당된 속성을 가지고 산수 변수로서 대행된다. I 부터 N 까지의 문자로 시작하는 포식어에 대해 , 태만 속성은 FIXED BINARY (15) 이다. 다른 영문자로 시작하는 포식어에 대해 , 태만 속성은 FLOAT DECIMAL(6)이다. 만일 FIXED나 FLOAT가 지정되면 , DECIMAL이 대행된다. 만일 DECIMAL이나 BINARY가 선언되면 , FLOAT가 대행된다. 태만 정도는 조직 / 360 실무에서 정의된 것과 같다.

보기 :

```
DECLARE A BINARY, B DECIMAL ;
```

A에 대한 태만은 FLOAT(21) ; B에 관한 태만은 FLOAT(6)이다.

(6) BIT 와 CHARACTER ( 줄 속성 )

BIT 와 CHARACTER 속성은 줄 변수를 지정하기 위해 사용된다. BIT 속성은 비트 줄을 지정한다. CHARACTER 속성은 문자 줄을 지정한다. 줄에 관한 길이 속성이 또한 지정되어야 한다.

일반 형식:

BIT ( 길이 )

CHARACTER ( 길이 )

일반 규칙:

1. 길이 속성은 선언된 줄의 길이를 지정한다. 이는 십진 고  
른수 상수이어야 하고, 기호가 없으며 영보다 커야 한다.  
최대 길이 지정은 문자 줄에서 255이고 비트 줄에서 64  
이다.
2. 길이 속성은 사이에 낀 빈자가 있거나 없이 CHARACTER  
또는 BIT 바로 뒤에 와야 한다.
3. BIT와 CHARACTER 속성은 PICTURE 속성과 동시에 지정되지  
못한다.
4. PICTURE 속성은 문자-줄 변수를 선언하는데 CHARACTER 대  
신 사용될 수 있다. ( PICTURE 속성을 보라 )
5. 모든 줄 속성은 PICTURE 속성이 사용되지 않는 한 명시  
선언되어야 한다. 줄 자료에 관한 태만은 없다.
6. BIT 줄은 UNALIGNED 속성을 갖지 못한다.

(7) BUFFERED와 UNBUFFERED ( 화일 서술 속성 )

BUFFERED 속성은 외부 기억소로 또는 외부 기억소로부터 전송동  
안 SEQUENTIAL RECORD의 개개 기록마디가 기초된 변수를 이용해  
서 주소가 정해질 수 있는 중간 기억소인 완충역을 지나야 함을  
지정한다.

UNBUFFERED 속성은 그런 기록마디가 완충역을 지나지 않음을

지정한다. 별도의 완충역이 UNBUFFERED 화일에 관해 D-편성자에 의해 사용되지 않는다.

일반 규칙:

BUFFERED | UNBUFFERED

일반 규칙:

1. BUFFERED와 UNBUFFERED 속성은 연속의 SEQUENTIAL RECORD 화일에 지정될 수 있다; BUFFERED 속성은 INDEXED SEQUENTIAL 화일에 지정될 수 있다. STREAM이나 DIRECT 속성을 가진 화일은 이들 속성을 갖지 못한다.
2. UNBUFFERED 속성은 가변 길이 또는 블러크된 기록마디에 지정되지 못한다.
3. UNBUFFERED 화일은 자기 테이프나 직접 접근 장치(2321 제외)와 연관된 화일에만 지정될 수 있다.
4. UNBUFFERED 속성은 테이프 자국(tape mark)이 앞에 없는 INPUT 화일에 지정되면 안된다. 이 형식의 화일에서 BUFFERED 속성이 지정되어야 한다.

태만 대행:

태만은 BUFFERED이다.

#### (8) BUILTIN (입구 속성 入口)

BUILTIN 속성은 선언 범위 안에 있는 연관된 이름에 대한 어떤 인용도 동일 이름에 대한 내조립 함수 또는 유사 변수에 관한 인용으로서 해석됨을 지정한다.

일반 규칙:

### BUILTIN

일반 규칙:

1. BUILTIN은 동일 포식어가 다른 뜻을 가지게끔 선언되어 있는 다른 블럭에 포함되어 있는 블럭에서 내조립 함수 또는 유사 변수로 인용하기 위해서 사용된다.
2. 만일 BUILTIN속성이 입구 이름을 위해 선언되면, 입구 이름은 다른 속성을 가질 수 없다.
3. BUILTIN 속성은 매개변수에 선언될 수 없다.
4. BUILTIN 속성은 (1) CALL문에 또는 인수 나열을 가진 함수 인용에 나타나거나 (2) ENTRY 속성으로 명시 선언되지 않은 입구 이름에 대해서는 문맥상 (文脈上 contextually) 선언된다.

#### (9) CHARACTER (줄 속성)

BIT를 보라.

#### (10) DECIMAL (산수 자료 속성)

BINARY를 보라.

#### (11) DEFINED (자료 속성)

DEFINED 속성은 선언되는 변수가 다른 자료에 배당되어 있는 동일 기억소의 일부 또는 전부를 나타냄을 지정한다. DEFINED 속성은 요소, 배열, 대구조체 변수에 선언될 수 있다.

일반 형식:

DEFINED 기초 - 포식어

「기초 포식어」는 이의 기억소가 선언되는 변수에 의해 표현되는 또 하나의 변수이다.

정의 하는 규칙:

1. INITIAL, 기억소 종류, 범위 속성은 정의된 항목에 지정되지 못한다.
2. 기초는 EXTERNAL 속성을 가질 수 있으나, 정의된 항목을 언제나 INTERNAL 속성을 가지고 다른 범위 속성으로 선언될 수 없음을 유의하라.  
만일 기초가 외적이면, 이의 이름은 이것이 외적으로 선언된 모든 블럭에 알려질 것이다. 그러나 정의된 항목은 그렇지 않을 것이다. 그렇지만, 정의된 항목의 값은 만일 기초 항목의 값이 어느 블럭에서 바뀌면 바뀔 것이다.
3. 기초 포식어는 정의된 항목이 선언된 블럭 안에 언제나 알려져야 한다. 기초 포식어는 DEFINED 속성을 갖지 못한다. 이는 기초된 변수가 될 수 없다. 그리고 이는 매개 변수가 되지 못한다.
4. 기초 포식어는 소구조체 또는 구조체의 요소가 될 수 없다. 두 형식의 정의가 있다. 대응 정의와(対応定義 correspondence defining) 중첩 정의(重疊定義 overlay defining) 만일 정의된 항목과 기초 포식어가 모두 동일 차원 수를 가진 배열이면, 대응 정의가 효과적이다. 다른 경우는, 중첩 정의가 효과적이다.

대응 정의

대응 정의는 정의된 항목의 요소에 관한 인용은 기초 포식어의

대응하는 요소에 관한 인용으로 해석됨을 뜻한다.

대응하는 배열은 동일한 차원수와 한계수를 가져야 한다. 기초 표식어의 요소와 정의된 항목의 요소는 동일한 서술을 가져야 한다.

중첩 정의

중첩 정의는 정의된 항목 (defined item) 이 기초에 할당된 기억소의 일부 또는 전부를 차지함을 뜻한다. 이 방법에서, 두 변수중의 한 값을 바꾸는 것은 다른 것의 값에 영향을 줄 것이다. 중첩 정의는 그림 9-1에 보인 항목간에 허용된다.

중첩 정의에 관한 규칙:

정의된 항목의 영역은 기초의 영역보다 크지 못한다. 영역은 자료의 부분 길이를 합하므로써 산출된다. 배열의 모든 개개 요소도 포함해서.

정의된 항목	기초 표식어
규약 산수 요소 변수	동일 기수, 척도, 정도의 첨자 없는 규약 산수 요소 변수
요소 명찰 변수	첨자 없는 요소 명찰 변수
요소 지침 변수	첨자 없는 요소 지침 변수
문자 종료 * 변수	문자 종료 * 자료
구조체	이의 구성이 만나는 짝의 항목들이 규약 산수, 명찰, 지침 요소 변수들의. 중첩 정의에 유효한 것인 같은 구조체 이 요소들은 또한 동일 길이의 줄 또는 수치문자 자료 항목들

정의된 항목	기 초 포 식 어
	<p>이 될 수 있다.</p> <p>* 문자 종류는 다음으로 된다:</p> <p>a. 정렬 안된 ( unaligned ) 수치 문자 자료</p> <p>b. 정렬 안된 문자 줄</p> <p>c. 항목 a , b , d로 된 구조체</p> <p>d. 항목 a , b로 된 배열</p>

그림 9 - 1 중첩 정의에 관한 허용 항목

(12) 차원 (次元 dimension) (배열 속성)

차원 속성은 배열의 차원수와 개개 차원의 한계 (限界 bound) 를 지정한다.

일반 형식:

( 한계 [ , 한계 [ , 한계 ] ] )

일반 규칙:

1. 한계 수는 배열의 차원 수를 지정한다. 일반 형식에서 보인 것처럼, 최대 차원 수는 D-평성자에서 삼(3)이다.
2. 개개 한계는 영보다 큰 무기호 십진 고른수 상수이어야 한다. 이 수는 해당 차원의 상한을 지정한다. 하한은 항상 1이다. 그러므로, 이 수는 또한 해당 차원의 크기를 지정한다. 보기로, 만일 한계가 8이면, 차원의 크기는 1, 2, 3, ..... , 8이다.

3. 차원 속성은 배열 이름 바로 뒤에 와야 한다. 사이에  
빈자는 선택적이다.

### (13) DIRECT 와 SEQUENTIAL (화일 서술 속성)

DIRECT 와 SEQUENTIAL 속성은 RECORD 화일에 있는 기록마디가  
접근되는 방법을 지정한다. SEQUENTIAL 은 기록마디가 자료 집  
합에 있는 물리적 순서에 따라 접근됨을 지정한다. DIRECT 는  
화일의 기록마디가 열쇠 (key) 를 이용하여 접근됨을 지정한다.  
직접 화일 (direct file) 의 개개 기록마디는 이와 연관된 열쇠  
를 가져야 한다.

SEQUENTIAL 과 DIRECT 는 화일의 현재 용법만을 지정한다 ; 이  
들은 화일과 연관된 자료 집합의 물리적 성질을 지정하는 것이  
아니다.

일반 형식:

SEQUENTIAL | DIRECT

일반 규칙:

1. DIRECT 화일은 또 DIRECT 에 의해 암시되는 KEYED 속성을  
가져야 한다. SEQUENTIAL 화일은 만일 SEQUENTIAL 속성이  
INDEXED 구성의 화일과 연관된 경우만 KEYED 속성을 가질  
수 있다.
2. DIRECT 와 SEQUENTIAL 속성은 STREAM 속성과 연관되어 지  
정될 수 없다.

태만 대행:

태만은 RECORD 화일에 대해 SEQUENTIAL 이다.

#### (4) ENTRY 속성

ENTRY 속성은 선언되는 포식어가 입구 이름임을 지정한다.

일반 형식: ENTRY

일반 규칙:

1. ENTRY 속성은 어떤 곳에서 선언되고 입구 이름이 인용될 때 (인수 나열에서와 같이) 그 블럭에 알려지지 않은 어떤 입구 이름에 대해서도 지정되어야 한다.

ENTRY 속성은 만일 블럭 안에서 이 입구 이름이 ENTRY를 암시하는 RETURNS 속성을 가지고 선언되면 지정될 필요가 없다. ENTRY와 BUILTIN은 모순되는 속성이며 양자가 동시에 한개 포식어에 지정되지 못한다.

2. ENTRY 속성은 매개변수가 되는 입구 이름에 대해서는 암시되거나 명시 선언되어야 한다.
3. ENTRY 속성은 이름이 내적인 블럭 안에서만 INTERNAL 입구 이름에 선언될 수 있다. ENTRY 속성을 가지고 선언된 내부 수속은 동일 선언에서 INTERNAL 속성을 받아야 한다.

태만 대행:

ENTRY 속성은 규칙 1에서 기술한 것처럼, 암시에 의해 대행될 수 있다. PROCEDURE 문이나 ENTRY 문의 명찰로서 이름이 나타나는 것은 입구 이름으로서 그 포식어의 명시 선언이 된다.

#### (5) ENVIRONMENT (화일 서술 속성)

ENVIRONMENT 속성은 PL/I 언어의 일부가 아닌 여러가지 화일

특성을 지정하는 실무상 정의된 속성이다.

일반 형식 :

ENVIRONMENT ( 선택항 - 나열 )

선택항 - 나열은 PL/I의 개개 실무를 위해 개별적으로 정의된다. D-편성자에서 이는 다음과 같다.

ENVIRONMENT ( [ COSECUTIVE  
REGIONAL ( { 1 | 3 } )  
INDEXED ] )

F ( 블러크크기 [ , 기록마디크기 ] )

V ( 최대블러크 크기 )

U ( 최대블러크 크기 )

[ BUFFERS ( { 1 | 2 } ) ]

MEDIUM ( 논리 - 장치 - 이름 , 물리 - 장치 - 형식 )

[ CTLASA | CTL 36 Ø ]

[ LEAVE ]

[ NOTAPEMK ]

[ NOLABEL ]

[ VERIFY ]

[ KEYLENGTH ( 십진 - 고른수 - 상수 ) ]

[ EXTENTNUMBER ( 십진 - 고른수 - 상수 ) ]

[ INDEXMULTIPLE ]

[ HIGHINDEX ( { 2311 | 2314 } ) ]

[ CONTROLTRACKS ( 십진 - 고른수 - 상수 ) ]

[ KEYLOC ( 십진 - 고른수 - 상수 ) ]

[ INDEXAREA ( 십진 - 고른수 - 상수 ) ]

[ ADDBUFF ( 십진 - 고른수 - 상수 ) ]

#### 일반 규칙

1. 개개 파일 선언은 연관된 ENVIRONMENT 속성을 가져야 한다.
2. 선택항들은 하나 이상의 빈자로 떨어져야 한다.
3. CONSECUTIVE 선택항은 ( n + 1 ) 번째, 어떤 파일의, 기록마디는 그 파일의 n 번째 기록마디 뒤에 위치함을 말한다. CONSECUTIVE 선택항이 위임하는 I/O 장치의 보기는 카드 독취기나 인쇄기이다.

CONSECUTIVE 도, REGIONAL 도 INDEXED 선택항도 지정되지 않으면, CONSECUTIVE 선택항이 태만에 의해 배당된다.

4. REGIONAL 선택항은 기억 매체에 있는 기록마디의 물리적 위치가 열쇠에 의해 지정됨을 암시한다. 열쇠는 프로그래머에 의해 지정되며 기록마디에 접근하는 유일한 수단을 이룬다. REGIONAL 선택항은 직접 접근 파일 ( direct access file )에만 허용된다.

REGIONAL (1) 은 기록마디가 파일에 있는 첫번째 기록마디에 관련된 상대 위치에 의해 인용되는 파일에 사용된다. 상대 기록마디 번호 ( relative record number ) 는 KEY 또는 KEYFROM 선택항에서 지정된다.

REGIONAL (3) 은 기록마디가 파일에 있는 첫번째 궤도 ( track ) 에 상대되는 이 기록 마디를 포함하고 있는 위치와 이 기록마디와 연관된 열쇠에 의해 인용되는 파일에 사용된다.

열쇠 ( key ) 와 상대 궤도 번호 ( relative track number )

( 이는 열쇠의 일 부분이 됨 ) 가 모두 선택항 KEY 또는 KEYFROM 에서 지정된다.

5. INDEXED 선택항은 열쇠있는 기록마디로 된 색인 순차 자료 집합 ( 索引 順次 資料 集合 indexed sequential data set ) 을 기술하며 , 이들 기록마디는 몇개 수준의 색인 ( indexed ) 에 의해 위치가 정해질 수 있다.
6. F, V, U 선택항 ( option 選擇項 ) 은 물리 기록마디 ( 物理 記錄마디 physical record ) 를 기술하기 위해 사용된다. F는 고정길이 기록마디 ( fixed length record ) 를 지정하고 , V는 가변 길이 기록마디를 ( variable length records ) 지정하고 U는 부정 길이 ( undefined length ) 의 기록마디를 지정한다.

고정 - 길이 기록마디는 블럭크 크기 지정을 요구한다. 기록마디 크기 지정은 선택적이다. 블럭크 크기와 기록마디 크기는 무기호 십진 고른수 상수로 지정된다. 블럭크 크기를 기록마디 크기로 나눈 몫은 고른수 이어야 한다. 고정 - 길이 블럭크된 기록마디는 만일 블럭크 크기와 기록마디 크기가 동시에 지정되면 만들어진다. 블럭크 인수 ( blocking factor 블럭크 因数 ) 는 기록마디 크기로 블럭크 크기를 나눈 것이다. 만일 블럭크 크기만 지정되면 , 기록마디 크기는 블럭크 크기와 같은 것으로 되고 , 화일은 블럭크 안된 것으로 간주된다.

만일 고정 - 길이 블럭크된 기록마디가 READ SET 나 LOCATE 문에 의해 전송되면 , 이 기록마디 크기는 8로 나누어져야 한다. 그렇지만 , LOCATE 문과 SET 선택항을 가진 READ

문은 INDEXED 자료 집합에 사용되지 못한다.

기록마디/블록 길이는 적어도 1자(字 byte) (자기 테이프 화일에서는 적어도 18자)가 되어야 하고 32,767자를 넘지 못한다. 장치 형식과 해당 최대 블록 길이는 다음과 같다.

2540	( PRINT 속성 없음 )	80
2540	( CTLASA, CTL 360 )	81
1442	( PRINT 속성 없음 )	80
1442	( CTLASA, CTL 360 )	81
2520	( PRINT 속성 없음 )	80
2520	( CTLASA, CTL 360 )	81
2501	( PRINT 속성 없음 )	80
1403	} ( PRINT 속성 또는 CTLASA 또는 CTL 360 )	133
3211		
1403	} ( PRINT 속성 없음 )	132
3211		
1404	( PRINT 속성 또는 CTLASA 또는 CTL 360 )	133
1404	( PRINT 속성 없음 )	132
1443	( PRINT 속성 또는 CTLASA 또는 CTL360 )	145
1443	( PRINT 속성 없음 )	144
1445	( PRINT 속성 또는 CTLASA 또는 CTL360 )	114
1445	( PRINT 속성 없음 )	113
2400	( PRINT 속성 없음 )	32,767

2420	}		
3420			
2400	}	( PRINT 속성 )	145
2420			
3420	}		
2311		( 열쇠 없음 , PRINT 속성 없음 )	3,625
2311	}	( PRINT 속성 )	145
2311		( 열쇠 포함 )	3,605
2314	}	( 열쇠 없음 , PRINT 속성 없음 )	7,294
1319			
1314	}	( PRINT 속성 )	145
1319			
2314	}	( 열쇠 포함 )	7,249
1319			
2321	}	( 열쇠 없음 , PRINT 속성 없음 )	2,000
2321		( PRINT 속성 )	145
2321	}	( 열쇠 포함 )	1,984

블러크된 기록마디인 INDEXED 화일에서, 열쇠는 개개 기록마디의 자료 부분에 포함되며 그래서 기록마디 크기는 열쇠의 길이를 포함 해야 한다.

만일 넘치기 계도 ( overflow tracks ) 가 블러크된 기록마디의 INDEXED 화일에 사용되면 ( DIRECT UPDATE 화일에 WRITE 하는 경우 ), 열자가 순차 ( 順次 sequence ) 연결 난 ( link field 連結欄 ) 을 위해 필요하다. 이는 ( 기록마디크기 + 열쇠길이 + 10 ) 이

열쇠 가진 기록마디에서 궤도 용량 (軌道 容量 track capacity) 을 넘지 않아야 함을 뜻한다. 그래서, 만일 프로그래머가 INDEXED 화일을 만들기 위해 선택항 OFLTRACKS( $n$ ), ( $n > 0$ ), 를 사용하면, 그는 최대 기록마디 크기를 산출함에 있어서 위 공식에 주의를 기울여야 한다.

V 선택항으로 지정된 최대 블럭 크기는 블럭과 기록마디를 위해 관리 낱말 (管理 낱말 control word) 을 포함해야 한다. 그래서, READ, SET, LOCATE 문으로써 전송되는 기록마디의 기록마디 크기는 8로 나눈뒤에 나머지가 4가 되어야 한다.

V와 U와 기록마디 크기 선택항을 가진 F 선택항은 자기 테이프 화일과 원반 화일 (圓盤 화일 disk file) 에만 허용된다.

주: STREAM 화일에서, 고정-길이 블럭 안된 기록마디에만 허용된다.

7. BUFFERS( $n$ ), 여기서  $n$ 은 1 또는 2, 는 사용되는 완충역 수를 지정하기 위해 사용된다. BUFFERS 선택항은 BUFFERED도 UNBUFFERED속성도 STREAM화일은 숨은 완충역을 가지므로 허용되지 않으나 STREAM 화일에 사용될 수도 있다. BUFFERS 선택항은 또 BUFFERED RECORD 화일에 사용될 수 있다. UNBUFFERED 속성은 BUFFERS 선택항을 배제한다. 태만은 BUFFERS(1)이다. BUFFERED 속성은 INDEXED SEQUENTIAL 화일에 사용될 수 있지만, 선택항 BUFFERS( $n$ )은 지정되어서는 안된다.

8. MEDIUM 선택항은 논리 장치 이름과 선언된 화일을 위한 장치 형식을 지정하기에 사용된다. 논리 장치 이름은

SYSXXX 꼴을 가진다. 여기서 XXX는 다음이 될 수 있다.

- a. IPT — 조직 입력 장치
- b. LST — 인쇄에 사용되는 조직 출력장치
- c. PCH — 조직 출력 장치 (카드 천공)
- d. 000부터 222 — 논리 장치 SYS000부터 SYS222

장치 - 형식 (device type) 지정은 사용되는 장치 번호를 말한다. 보기로, 만일 IBM 1442N1 카드 독취/천공기가 사용되면, 선택항은 1442로 써질 것이다. 그림 9-2는 어떻게 개개 장치 형식이 지정되는 가를 보인다.

그림 9-2에 열거된 장치 형식은 그림 9-3에 보인 것처럼 논리 장치 이름 SYSIPT, SYSLST, SYSPCH에 배당될 수 있다.

편성 때에 MEDIUM 선택항에서 지정된 것과는 독립된 장치 형식이 어떤 조건 아래서 논리 장치 SYSIPT, SYSLST, SYSPCH에 사용될 수 있다. ASSIGN 문으로, 사용자는 프로그램 실행 동안 사용될 실제의 장치 형식을 지정할 수 있다. 독립 장치를 위한 과정과 표는 만일 다음의 요구가 맞으면 D-수준 편성자에 의해 자동으로 생성된다:

- a. 출력 파일은 STREAM과 PRINT 속성 (SYSLST) 또는 CTLASA 선택항을 가진 RECORD 속성 (SYSLST와 SYSPCH) 을 가져야 한다. 개개 기록마디의 첫 문자는 이들 경우에 통제 문자 (control character) 로 해석된다. 입력 파일은 (SYSIPT) STREAM이나 RECORD가 될 수 있다.
- b. 기록마디는 블러크 안되며 고정 길이이어야 한다. 기록마디 크기 (n) 은 다음과 같아야 한다:

$n \leq 80$  SYSIPT에

$n \leq 121$  SYSLST에

$n \leq 81$  SYSPCH에

만일 화일이 DASD에 등재되어 있다면, 80, 121, 81이  
SYSIPT, SYSLST, SYSPCH에 각각의 값이 되어야 한다.

SYSIPT에 배당된 테이프 화일에서, RECORD 조건이 만  
일 물리 기록마디크기가 ENVIRONMENT 속성에서 주어진 블러  
크 크기를 넘으면 일어날 것이다.

카드 화일에서, RECORD 조건은 일어나지 않을 것이다,  
그러나 개개 READ 명령은 ENVIRONMEUT 속성에서 주어  
진 길이에 관계 없이 새 카드를 읽어들이게 할 것이다.

c. BACKWARDS 속성과 NOTAPEMK 선택항은 지정되지 못한  
다.

만일 이 요구가 맞으면, 해당 논리 장치 (그림 9-3을  
보라) 에 제공된 어떤 장치 형식도 (재편성의 필요 없이  
배당된다.

장 치 형 식	번	호	장 치 - 형 식 지 정
카드 독취천공 장치	IBM	2540	( 독취기 ) 2540
	IBM	2540	( 천공기 ) 2540
	IBM	1442	N1 1442
	IBM	1442	N2 1442
	IBM	2520	B1 2520
	IBM	2520	B2 2520
	IBM	2520	B3 2520
	IBM	2501	2501
인 쇄 기	IBM	1403	1403
	IBM	1404	1404
	IBM	1443	1443
	IBM	1445	1445
	IBM	3211	1403
자기 테이프 구동 장치	IBM	2400	( 9 - 궤도 ) 2400
	IBM	2400	( 7 - 궤도 ) 2400
	IBM	2420	2400
	IBM	3420	2400
D A S D	IBM	2311	2311
	IBM	2314	2314
	IBM	2321	2321
	IBM	2319	<b>2314</b>

그림 9-2 장치 형식과 해당 지정

논리 장치 이름	장 치 형 식
SYSIPT	IBM 2540 (독취기)
	IBM 1442N1
	IBM 2501
	IBM 2520B1
	IBM 2400 (7 - 또는 9 - 웨도)
	IBM 2311
	IBM 2314
	IBM 1403
SYSLST	IBM 1404
	IBM 1443
	IBM 2400 (7 - 또는 9 - 웨도)
	IBM 2311
	IBM 2314
	IBM 2540 (천공기)
SYSPCH	IBM 1442N1
	IBM 1442N2
	IBM 2520B1
	IBM 2520B2
	IBM 2520B3
	IBM 2400 (7 - 또는 9 - 웨도)
	IBM 2311
	IBM 2314

그림 9-3 SYSIPT, SYSLST, SYSPCH에 연관된 장치 형식

9. CTLASA 와 CTL360 선택항은 기록마디의 첫째 문자가 각각 ASA 또는 조직/360 통제 문자로 해석되는 RECORD 인쇄기 또는 천공기 화일에 사용되는 ENVIRONMENT 속성의 두 개의 선택항이다. (이들은 서로 배제성(排除性)이다). 기록마디 변수의 첫 문자를 정확한 통제 문자로 하는 것은 프로그래머의 책임이다.

CTLASA 로 사용될 수 있는 문자 부호(Code)는 그림 9-4에 그것들의 해석과 함께 열거되어 있다.

CTL360으로 사용될 수 있는 8-비트 부호가 9-5와 9-6에 그것들의 기능과 함께 열거되어 있다.

만일 통제 문자가 그림 9-5 또는 9-6에 열거되어 있는 것이 아니면 결과는 알 수 없다.

부 호	해 석
(빈자)	인쇄 전 한 줄 띄기
∅	인쇄 전 두 줄 띄기
-	인쇄 전 세 줄 띄기
+	줄 안 땀
1	인쇄 전 경로(經路 Channel) 1로 건너기
2	인쇄 전 경로 2로 건너기
3	인쇄 전 경로 3으로 건너기
4	인쇄 전 경로 4로 건너기
5	인쇄 전 경로 5로 건너기
6	인쇄 전 경로 6으로 건너기
7	인쇄 전 경로 7로 건너기

부 호	해 석
8	인쇄 전 경로 8로 건너기
9	인쇄 전 경로 9로 건너기
A	인쇄 전 경로 10으로 건너기
B	인쇄 전 경로 11로 건너기
C	인쇄 전 경로 12로 건너기
V	적소 (積所 Stacker) 1을 선택
W	적소 2를 선택

그림 9 - 4 CTLASA에 사용될 수 있는 문자 부호

8-비트 부호	천 공 조 합	기 능
1442의 적소 선택		
10000001	12,0,1	적소 1로 선별
11000001	12,1	적소 2로 선별
2540의 주머니 선택		
00000001	12,9,1	주머니 1로 선별
01000001	12,0,9,1	주머니 2로 선별
10000001	12,0,1	주머니 3으로 선별
2520의 적소 선택		
00000001	12,9,1	적소 1로 선별
01000001	12,0,9,1	적소 2로 선별

그림 9 - 5. CTL360으로 적소 선택에 사용될 수 있는 비트 조합

8-비트 부호	천공 조합	기능
00000001	12,9,1	쓴다 (줄 띄기 없음)
00001001	12,9,8,1	쓰고 1줄 띄기
00010001	11,9,1	쓰고 2줄 띄기
00011001	11,9,8,1	쓰고 3줄 띄기
10001001	12,0,9	쓰고 경로 1로 건너기
10010001	12,11,1	쓰고 경로 2로 건너기
10011001	12,11,9	쓰고 경로 3으로 건너기
10100001	11,0,1	쓰고 경로 4로 건너기
10101001	11,0,9	쓰고 경로 5로 건너기
10110001	12,11,0,1	쓰고 경로 6으로 건너기
10111001	12,11,0,9	쓰고 경로 7로 건너기
11000001	12,1	쓰고 경로 8로 건너기
11001001	12,9	쓰고 경로 9로 건너기
11010001	11,1	쓰고 경로 10으로 건너기
11011001	11,9	쓰고 경로 11로 건너기
11100001	11,0,9,1	쓰고 경로 12로 건너기
00001011	12,9,8,3	끝 1줄 띄기
00010011	11,9,3	끝 2줄 띄기
00011011	11,9,8,3	끝 3줄 띄기
10001011	12,0,8,3	끝 경로 1로 건너기
10010011	12,11,3	끝 경로 2로 건너기
10011011	12,11,8,3	끝 경로 3으로 건너기
10100011	11,0,3	끝 경로 4로 건너기

8-비트 부호	천공 조합	기능
10101011	11,0,8,3	곧 경로 5로 건너기
10110011	12,11,0,3	곧 경로 6으로 건너기
10111011	12,11,0,8,3	곧 경로 7로 건너기
11000011	12,3	곧 경로 8로 건너기
11001011	12,0,9,8,3	곧 경로 9로 건너기
11010011	11,3	곧 경로 10으로 건너기
11011011	12,11,9,8,3	곧 경로 11로 건너기
11100011	0,3	곧 경로 12로 건너기
00000011	12,9,3	연산 없음

그림 9-6 OTL36 Ø으로 인쇄기 통제에 사용될 수 있는 비트 조합

10. LEAVE 선택항은 뒤감기 연산이 화일을 열고 닫을 때에 이루어지지 않는 것을 지정하기에 사용된다. 이는 화일을 정한 자리에 놓기 위해서 BACKWARDS 속성을 가진 화일에는 꼭 주어져야 한다.
11. 테이프 화일에 대한 NOTAPEMK는 명찰 없는 화일에 있는 자료 기록 마디의 앞에 앞서는 테이프자국 (Leading Tape-mark)이 써지는 것을 그만두게 한다. NOTAPEMK는 테이프 OUTPUT 화일에서 지정된, NOLABEL과 더불어 사용될 수도 있다. 이 선택항은 UNBUFFERED 화일에는 허용 안 된다.
12. NOLABEL 선택항은 화일 명찰이 자기 테이프 화일에 대해

처리되지 않음을 지정하기에 사용된다.

만일 NOLABEL 선택항이 출력 화일에 지정되면, 테이프 자국이 NOLABEL, 선택항에 부과해서 NOTAPEMK 선택항이 ENVIRONMENT 속성에서 지정되지 않는 한 테이프에 있는 첫번째 기록마다로서 자동으로 켜진다.

0 비표준 (Non-Standard) 명찰과 추가 자용차 명찰은 처리되지 않는다.

13. VERIFY 선택항은 읽기검사 (Read - Check) 가 매 기록 연산 뒤에 수행됨을, 지정하기에 사용된다. 이 선택항은 직접 접근 장치에만 허용된다.

14. KEYLENGTH 선택항은 (REGIONAL (3)과 INDEXED 화일에만 허용된다.) 입력과 출력 연산에서 열쇠의 길이를 지정함에 사용된다.

REGIONAL (3)의 열쇠길이는, 이는 8보다 크고 256보다 작아야 한다. 기록마다 길이에 포함해서는 안된다.

15. EXTENTNUMBER 선택항은 REGIONAL 과 INDEXED 화일에서 사용되는 구역 수 (区域數 Number of Extents) 를 지정하기에 사용된다.

REGIONAL 화일에서, EXTENTNUMBER(n)은 선택적이다. 만일 지정되면, n은 0보다 크고 256보다 작아야 한다. 태만 값은 3이다.

INDEXED 화일에서, EXTENTNUMBER(n)은 지정되어야 한다.

n의 값은 모든 자료 구역을 포함해야 한다. 주 색인

(Master Index 主索引)과 원통 색인 (圓筒索引

Cylinder Index) 구역, 그리고 독립된 넘치기 구역 (Indep-

endent Overflow Extent) 주와 원통 색인 구역은 개개 색인이 별개의 EXTENT 카드를 요하지만 하나의 구역으로 서 계산된다. 그래서, 지정될 수 있는 최소 수는 둘이다: 기본 자료 영역을 위한 구역과 원통 색인을 위한 구역. n에 관한 최대값은 255이다.

16. INDEXMULTIPLE 선택항은 INDEXED 화일에서 주 색인이 이 화일을 위해서 설정될 (또는 설정된) 것임을 지정하기에 사용된다.

17. HIGHINDEX 선택항은 INDEXED 화일에서 고수준 (高水準 High-Level) 색인 또는 색인들이 MEDIUM 선택항에서 지정된 것과 틀리는 장치 형식에 등재되어 있는 장치의 형식 (2311 또는 2314 또는 2321)을 지정하기에 사용된다.

18. OFLTRACKS (n) 선택항은 INDEXED OUTPUT와 UPDATE 화일에서 추가되는 기록마디를 위해서 개개 원통에 확보될 궤도 수를 지정하기에 사용된다. 지정된 수는 다음 한계 안에 있어야 한다.

$0 \leq n \leq 8$  2311 화일

$0 \leq n \leq 18$  다른 모든 화일

만일 OFLTRACKS (n) 선택항이 생략되면, 태만 값은  $n = 0$ 이다. 만일 새 기록마디가 INDEXED 화일에 삽입될 예정이고 독립된 넘치기 영역이 쓸 자리가 없으면, 선택항 OFLTRACKS (n)이  $n \neq 0$ 으로 지정되어야 한다; 그렇지 않으면 KEY 조건이 일어날 것이다.

입력 화일에서, 이 선택항은 뜻이 없으며, 무시된다.

19. KEYLOC (n) 선택항은 기록마디 안에 있는 열쇠 난의 고 순위 자리 (High - Order Position) 를 지정하기에 사용된다. 블럭 안된 기록마디에서, KEYLOC (n)은 선택적이다; 블럭된 기록 마디에서, KEYLOC (n)은 지정되어야 한다. 이는 다음 한계 안에 있어야 한다.

$$1 \leq n \leq \text{기록마디크기} - \text{열쇠길이} + 1$$

태만 값 n은 1이다. (블럭된 기록마디에만)

20. INDEXAREA (n) 선택항은 기억소-원통-색인 선택항이 INDEXED DIRECT 화일에 사용됨을 지정하기에 사용된다. n에 관한 값 (원통 색인에 확보되는 자 수)은 다음 한계 안에 있어야 한다:

$$3 \text{ 미 (열쇠길이} + 6) \leq n < 32 \text{ K}$$

한번에 기억소에 있을 수 있는 원통 색인 등재 수 (N)은 다음과 같이 산출될 수 있다.

$$N = n / (\text{열쇠길이} + 6) - 4$$

21. ADDBUFF (n) 선택항은, DIRECT UPDATE 화일에서, 새 기록마디가 내부에 더해지고 직접-접근 장치에 더해지지 않음을 지정하기에 사용된다. n의 값은 (UPDATE 영역에 확보되는 자 수) 다음 한계 안이어야 한다.

$$74 + \text{블러크크기} + \text{열쇠길이} \leq n < 32 \text{ K}$$

한번에 기억소 안에 읽혀들어올 수 있는 블럭 수 (N)는 다음과 같이 계산될 수 있다.

$$N = (n - 24) / (50 + \text{열쇠길이} + \text{블러크크기})$$

주: N을 한개 케도에 있는 블럭의 수보다 크게 한다는 것은 의미가 없다.

태만 대행 :

CONSECUTIVE 자료 집합 구성이 달리 기재가 없는 한 대행된다. 테이프 타래는 LEAVE 선택항이 지정되지 않으면 되감긴다. 만약 BUFFERS (n) 선택항이 지정되지 않으면, 한개의 완충역이 할당된다.

#### (16) EXTERNAL 과 INTERNAL (범위 속성)

EXTERNAL 과 INTERNAL 속성은 이름의 범위를 지정한다.

INTERNAL 은 이름이 선언한 블럭과 이것에 포함되어 있는 블럭에만 알려질 수 있음을 지정한다. EXTERNAL 은 이름이 동일 이름에 관한 외부 선언을 포함한 다른 블럭에 알려질 수 있음을 지정한다.

일반 형식 :

EXTERNAL | INTERNAL

일반 규칙 :

1. 모든 화일 이름은 외부적 (외적 External) 이어야 한다. 이들은 내부적 (내적 Internal) 으로 선언되지 못한다.
2. 모든 외부 이름은 D-편성자에서 최대 길이 여섯 문자로 제한된다.

태만 대행 :

INTERNAL 은 내부 수속의 입구 이름과 모든 기억소 종류의 변

수에 대항된다. EXTERNAL 은 화일 이름과 외부 수속의 입구 이름에 대항된다.

### (17) FILE 속성

화일 속성은 선언된 포식어가 화일 이름임을 지정한다.

일반 형식 : FILE

일반 규칙 :

화일 속성은 이것이 화일 기술 속성에 의해 암시되므로 생략되어도 된다.

### (18) FIXED와 FLOAT (산수 자료 속성)

FIXED와 FLOAT 속성은 선언되는 산수 변수의 척도 (尺度 Scale) 를 지정한다.

FIXED는 변수가 고정점 자료 항목을 표현함을 지정한다.

FLOAT는 변수가 부동점 자료 항목을 표현함을 지정한다.

일반 형식 : FIXED | FLOAT

일반 규칙 :

FIXED와 FLOAT 속성은 PICTURE 속성에 지정되지 않는다.

대항

선언되지 않은 포식어 (또는 차원, UNALIGNED, ALIGNED, 기억소 종류, 범위 속성 중의 하나 이상으로만 선언된 포식어)는 첫째 문자에 따라 해당된 속성을 가진 산수 변수로 대행된다. 문자 I 부터 N까지로 시작하는 포식어에 관해, 태만 속성은 FIXED BINARY(15)이다. 다른 문자로 시작하는 포식어는, 태만 속성은 FLOAT DECIMAL(6)이다. 만일 BINARY나 DECIMAL이 지정되면, FLOAT가 대행된다. 만일 FIXED나 FLOAT가 지정되면, DECIMAL이 대행된다. 태만 정도는 조직/360 실무에서 정한 것과 같다.

#### (19) FLOAT (산수 자료 속성)

FIXED를 보라

#### (20) INITIAL (자료 속성)

INITIAL 속성은 기억소가 이것에 할당될 때 자료에 대입되는 초기 상수 값을 지정한다.

일반 형식 :

INITIAL (항목 [ , 항목 ] . . . )

일반 규칙 :

1. 하나의 요소 변수에는 하나의 상수 값만이 요구된다. 둘 이상의 값을 배열에 줄 수 있다.
2. 배열에 지정된 상수 값은 행대순 (行大順 Row-Major Order)

- 으로(마지막 첨자가 빨리 변함) 배열의 연속한 요소에 대입된다. 보기  
 DECLARE A ( 2.2 ) INITIAL ( 1.2.3.4 ) ; 는 다음 결과  
 가 된다 ; 할당시에 A ( 1.1 )는 1로, A ( 1.2 )는 2로,  
 A ( 2.1 )은 3으로, A ( 2.2 )는 4로 된다.
3. 만일 배열에 너무 많은 상수가 지정되면, 여분의 것은 무시  
 된다. 만일 부족하게 지정되면, 배열의 나머지는 초기값을  
 갖지 않는다.
  4. 나열의 각 항목은 상수거나 중복 지정이어야 한다.
  5. 중복 지정 (重複指定 Iteration specification) 은 다음  
 일반 줄의 하나를 갖는다 :  
 (중복-인수) 상수  
 (중복-인수) (항목[, 항목] . . . .)
  6. #중복 인수 (Iteration Factor) # 는 십진 고른수 상수이  
 며 1보다 크거나 같아야 한다.
  7. 만일 괄호에 싸인 십진 고른수 상수가 한개만 줄 초기 값  
 의 앞에 오면, 이는 이 줄의 반복 인수 (反復 因素 Repetition  
 Factor)로서 해석된다. 만일 두개가 나타나면,  
 첫째 것은 초기치 중복 인수가 되고 (Iteration Factor),  
 둘째 것은 줄 반복 인수로 된다.  
 보기로, 다음을 생각하라 :  
 ((2) 'A' )는 ( 'A A' )와 같다.  
 ((2) (1) 'A' )는 ( 'A' , 'A' )와 같다.
  8. 중복은 뚫어질 수 있다. 뚫어진 중복의 최대 깊이는 괄(3)  
 이다.
  9. 명찰 변수를 위한 초기 값으로 주어진 명찰 상수는 명찰

변수 선언이 나타난 블럭 안에 알려져야 한다.

10. INITIAL 속성은 다음에는 허용되지 않는다 :

입구 이름

파일 이름

DEFINED 자료

구조체

매개변수

기초된 변수

지침 변수

STATIC 명칭 변수

주 : INITIAL 속성은 구조체의 요소적 이름에 주어질 수 있다.

보기 :

```
DECLARE SWITCH INITIAL ( '1' B ) BIT ( 1 ) ,  
MAXVAL INITIAL ( 99 ) , MINVAL INITIAL ( -99 ) ,  
A ( 100 , 10 ) INITIAL ( 1920 ) 0 , ( 20 ) ( ( 3 ) 5.9 ) ;
```

마지막 보기는 다음 결과가 된다 :

A의 처음 920개 요소는 0으로 되고, 나머지 80개 요소는 5.5.5.9의 순서로 20번 반복으로 내용을 이룬다.

(21) INPUT, OUTPUT, UPDATE (파일 서술 속성)

INPUT, OUTPUT, UPDATE 속성은 파일의 기능을 말해준다.

INPUT는 자료가 자료 집합에서 프로그램으로 전송됨을 지정한다.

OUTPUT는 자료가 프로그램에서 자료 집합으로 전송됨을 지정한다.

다. 즉 새로이 자료 집합이 만들어진다. UPDATE는 자료가 어떤 방향으로 전송됨을 지정한다 ; 이 말은, 화일이 입력과 출력 화일 양쪽이라는 것이다.

일반 형식 :

INPUT | OUTPUT | UPDATE

일반 규칙 :

1. INPUT 속성을 가진 화일은 PRINT 속성을 갖지 못한다.
2. OUTPUT 속성을 가진 화일은 BACKWARDS 속성을 갖지 못한다.
3. UPDATE 속성을 가진 화일은 STREAM, BACKWARDS, 또는 PRINT 속성을 갖지 못한다. SEQUENTIAL 화일을 위한 UPDATE 선언은 제자리-갱신 방식 (Update - in - Place Mode 제자리-更新 方式)을 가르킨다. 새 기록마디의 추가는 불가능하다. 그런 화일에 접근하기 위해서는, 연속한 문은 READ 다음에 REWRITE 이어야 한다.
4. 화일이 PRINT 속성으로 선언되지 않는한, 이 경우 OUTPUT가 암시된다. 앞의 속성 중의 하나가 각 화일에 주어져야 한다.
5. 이들 속성은 UNBUFFERED 화일의 경우에는 제외하고, 이 경우에 INPUT나 OUTPUT가 OPEN 문에서 지정될 수 있다. DECLARE 문에서 지정되어야 한다.
6. OUTPUT는 INDEXED DIRECT 화일에는 지정되지 못한다.

태만 대행 :

PRINT 속성은 OUTPUT를 암시한다.

(22) INTERNAL (범위 속성)

EXTERNAL 을 보라

(23) KEYED (파일 서술 속성)

KEYED 속성은 파일에 있는 개개 기록마디는 이와 연관된 열쇠를 가지고, 문 선택항 KEY와/또는 KEYFROM이 파일에 있는 기록마디에 접근하기 위해서 사용될 수 있음을 지정한다.

일반 형식 : KEYED

일반 규칙 :

1. KEYED 파일은 만일 이것이 INDEXED SEQUENTIAL 구성인 경우만 순차로 읽혀들여질 수 있다.
2. KEYED 속성은 DASD 장치에 등재되어 있는 자료 집합과 연관된 파일에만 지정될 수 있다.

태만 속성 :

DIRECT 속성은 KEYED를 암시한다.

(24) LABEL (프로그램 통제 자료 속성)

LABEL 속성은 선언되는 포식어가 명찰 변수이며 값으로서 문  
문 명찰을 가짐을 지정한다.

일반 형식 : LABEL

일반 규칙 :

1. 이 변수는 이 변수의 범위 안에 알려진 어떤 문 명찰도  
값으로 가질 수 있다.
2. 만일 변수가 매개변수이면, 이것의 값은 인수로서 보내진 어  
떤 문 명찰 변수나 변수될 수 있다.
3. 입구 이름은 명찰 변수의 값이 되지 못한다.
4. INITIAL 속성은 STATIC 명찰 변수에 지정되지 못한다.

(25) 길이 (Length) (출 속성)

BIT를 보라.

(26) OUTPUT (화일 서술 속성)

INPUT를 보라.

(27) PICTURE (자료 속성)

PICTURE 속성은 문자-출과 수치 문자 자료의 내부와 외부  
형식을 정의하고 자료의 편집을 지정하기 위하여 사용된다.

수치 문자 자료 (Numeric Character Data)는 산수 값을 가지나 내부에서 문자 꼴로 기억되는 자료이다. 수치 문자 자료는 산수 연산이 수행될 수 있기전에 규약 산수로 변환되어야 한다.

모양 문자 (模樣 文字 Picture Character)는 제 4 장 "모양 지정 문자"에 설명되어 있다.

일반 형식 :

PICTURE { '문자-모양-지정'  
'수치-모양-지정'

"모양 지정"은, 이는 문자나 숫자 중의 하나, 반 따옴표 (single Apostrophes) (형식에 보인 바와 같이)에 싸여 있는 모양 문자 줄로 이루어진다. 낱말의 모양 문자는 반복 인수가 앞에 올 수도 있다. 이것은 괄호에 싸여 있으며 영보다 큰 무기호 십진 고른수 상수인  $n$ 이며, 문자의  $n$ 번 반복을 가르킨다. 지정에 있는 모양 문자는 난으로 (欄으로 Field) 나누어지며, 이 중의 어떤 것은 소난 (小欄 Subfield)으로 내용이 이루어진다.

일반 규칙 :

1. "문자 모양 지정"은 문자-줄 자료 항목을 기술하기 위해 사용된다.
  - o 모양 문자  $X$ 만이 사용된다. 이것은 당해 자리가 어떤 문자로도 이루어질 수 있음을 가르킨다. 적어도 하나의  $X$ 는 지정되어야 한다. 문자 모양 지정은 소난이 포함 안된 단일의 난이다.

보기 :

DECLARE ORDER# PICTURE '(13)X' ;

이 선언은 ORDER#의 값이 길이 13의 문자 줄이 됨을 지정한다. 보기로, 문자 줄 'GF342-63-0024'는 이 기술에 부합될 것이다.

편집과 지우기 (Editing and Supression) 문자는 문자 모양 지정에 허용되지 않는다. 개개 모양 지정 문자는 자료 항목에서 실 문자 (実 文字)를 표현해야 한다.

2. #수치 모양 지정#은 이것의 사용법에 따라 산수 값 또는 문자-줄 값을 표현하는 문자 항목을 기술하기에 사용된다. 수치 모양 지정은 하나 이상의 난으로 이루어질 수 있고, 이 난의 어떤 것은 소난으로 나누어질 수 있다. 단일(單一)의 난은 고정점 수 또는 부동점 수의 가수(假數 Mantissa)를 기술하는데 사용된다. 이것들은 두개의 소난으로 나누어질 수도 있다. 하나는 고른수 부분을 기술하고, 다른 것은 소수 부분을 기술한다. 부동점 수에서 둘째 난은 지수를 기술하기 위해 요구된다; 이것은 소난으로 나누어지지 못한다. 네개의 기본 모양 문자가 수치 모양 지정에 사용될 수 있다:

9. 십진 숫자를 가르킴.

V. 지정된 소수점의 자리를 가르킴. 이것은 자료 항목의 문자-줄 값에 있는 실 문자를 지정하는 것이 아니다. 이것은 모양 지정의 소난 끝을 가르킨다.

K. 부동점수 자료 항목에서, 지수(指數 Exponent)가 K다음의 연관된 모양 문자 자리에서 시작한다고 가정됨을 가르킨다. 이 K는 지정의 두개 난을 구분한다.

ㅉ. 부동점수 자료 항목에서, 해당 자리가 지수의 시작을 가르키기 위해서 문자 표로 되어 있음을 가르킨다. ㅉ는 또한 두개 난을 구분한다.

이들 문자에 추가해서, 영 지우기 문자, 편집 문자, 음양 기호 문자가 편집을 지시하기 위해 수치 모양 지정에 포함될 수도 있다.

○ 편집 문자는 수치 문자 자료 항목의 산수 값의 일부는 아니다. 그러나 이것들은 이것의 문자-줄 값의 일부가 된다. 개개 수치 모양 지정은 적어도 한개의 숫자 지정자(數字 指定字 Digit Specifier)가 있어야 한다. 반복 인수가 수치 모양 지정에서 허용된다.

3. 수치 문자 자료 항목은 십진 기수만을 가진다. 이것의 척도와 정도는 모양 문자에 의해 지정된다. PICTURE 속성은 기수, 척도, 정도 속성과 결합해서 지정되지 못한다.

4. 다음의 문항들은 여러가지 산수 자료 형식을 위해 모양 문자의 결합을 가르킨다.

a. 십진 고정점수 항목은 다음의 일반 꼴로 서술된다:

PICTURE '[9]...[V] [9]...'

기호, 편집, 영지우기 모양 문자가 고정점수 지정에 포함될 수 있다. V는 지정에서 한번보다 많이 나타날 수 없다. 그리고 이것은 점 또는 쉼표를 삽입시키는 소수점 또는 쉼표(Decimal Point or Comma)와 결합해서 사용되어도 된다. 만일 V가 포함되지 않으면, 소수점은 가장 오른쪽 수자의 오른쪽에 있는 것으로 대행된다. 하나의 음양 기호(Sign)만이 난에 포함될 수 있다. 지정은 적어도

한개의 숫자 자리를 포함해야 한다.

보기 :

```
DECLARE A PICTURE '999 V 99' ;
```

이 지정은 다섯 숫자의 수치 문자 항목을 서술하며, 이것의 두 숫자는 소수 숫자로 처리된다.

b. 십진 부동점수 항목은 다음 일반 꼴에 의해 서술된다 :

```
PICTURE '[9] . . . [V] [9] . . . { E|K} 9 [9]'
```

첫째 난과 둘째 난 모두 적어도 하나의 숫자 자리를 가지고 있어야 한다. 지수난은 두 숫자를 초과하지 못한다. 이것은 조직 / 360 실무에서 십진 부동점 수의 지수 난에서 두개 숫자만을 허용하기 때문이다. 만일 산수 자료 항목이 이 서술된 변수에 대입되면, 지수 난은 허용된 숫자 지정 문자를 둘다 가져야 한다. 그렇지 않으면 지수난의 둘째 숫자가 상실될 것이고 SIZE 조건이 일어날 것이다.

기호, 편집, 영 지우기 모양 문자가 부동점수 지정에 들어갈 수 있다. 한개의 음양 기호 지사만이 각 난에 허용된다. 오직 한개의 V만이 허용되고, 이것은 첫째 난에만 나타날 수 있다. 고정점수 지정에서와 같이, V는 소수점 편집 문자와 결합해서 (.V 또는 V.과 같이) 나타나도 된다. X, T, I, R, CR, DB와 영국돈 모양 문자는 허용 안된다.

5. 수치 문자의 변수의 정도는 실제적 또는 조건적 숫자 자리의 수에 좌우된다. 숫자 자리는 다음 문자로 지정될 수 있다.

q. 이는 실제적 숫자 문자이다.

$\left. \begin{matrix} Z \\ * \\ T \\ I \\ R \end{matrix} \right\}$  이것은 영 지우기를 지정하는 조건적 숫자 문자이다.  
 $\left. \begin{matrix} S \\ + \\ - \end{matrix} \right\}$  이것은 중복천공 (Over Punch)을 지정하는 숫자 문자이다.  
 $\left. \begin{matrix} \text{ } \\ \text{ } \\ \text{ } \end{matrix} \right\}$  이것은 조건적 숫자 부동 문자 (Drifting Characters)이다.

위의 개개 문자는 부동 줄에서 첫번째 조건적 숫자 부동 문자를 빼고는 숫자 자리를 지정한다. 단독으로 사용된 조건 숫자 부동 (浮動 Drifting) 문자는 숫자 자리를 지정하는 것이 아니다. 고정 점수의 정도는 (p,q)이다. 여기서 P는 모양 지정에 있는 숫자 자리 수이며 q는 V뒤에 오는 숫자 자리 수이다. 부동점수의 정도는 (P)이다. 여기서 P는 E나 K앞에 오는 숫자 자리 수이다. 지시된 정적 (靜的 Static) 편집 문자나 삽입 문자는 정도의 지정에 들어가는 것이 아니다. 이들은 그러나 만약 자료 항목이 내부적으로 문자 줄에 대입된다면 문자 수에 계산되어야 한다.

6. 영국돈 (英國 돈 Sterling) 자료 항목을 표현하는 변수는 각각 파운드 (Pound), 실링 (Shilling), 펜스 (Pence) 단이 되는 세계의 단으로 되어 있는 수치 모양 지정을 사용해서 지정될 수 있다. 펜스 단은 두개의 소단으로 나누어질 수도 있다. 이렇게 기술된 자료는 세계 단의 각각에 해당하는 세계의 계속한 수의 문자 형식으로 기억된다. 만일 어떤 산수 연산이 이 변수에 지정되면, 이것의 값은 펜스로 값을 표현하는 규약 고정 점 십진수로 변환된다. 영국돈 모양 지정은 다음 꼴을 갖는다:

PICTURE , G [ 편집 - 문자 - 1 ] . . .

M 파운드 - 난

M [ 분리자 - 1 ] . . .

실링 - 난

M [ 분리자 - 2 ] . . .

펜스 - 난

[ 편집 - 문자 - 2 ] . . .

이들 난에 사용될 수 있는 모양 지정 문자, 편집 문자, 분리자들은 제 4 장, # 모양 지정 문자 # 에 논술되어 있다. 영국돈 자료 항목의 정도 ( p·q ) 는 다음과 같이 정해진다.

q = 펜스 난에 있는 소수 숫자 수.

p = 3 + q + 파운드 난에 있는 실제적과 조건적 숫자 자리수.

#### (28) POINTER ( 프로그램 통제 자료 속성 )

POINTER 속성은 선언되는 포식어가 지침 변수 ( 指針 變數 Pointer Variable ) 이며 어떤 기억소 종류에 존재하는 자료를 가르키기 위해 사용될 수 있음을 지정한다.

일반 형식 : POINTER

일반 규칙 :

1. POINTER 속성은 DECLARE 문을 통해서만 포식어에 부여될 수 있다. 그래서, 지침 변수는 POINTER 속성으로 명시 선언되어야 한다.
2. 지침 변수의 값은 두가지 방법으로 설정될 수 있다.
  - a. 지침 대입에 의해서.

- b. READ나 LOCATE 문에 있는 SET 절 (節 Clause)로서.
3. 지침 자료는 산수 식의 연산학으로 나타나지 못하며, 지침 자료와 다른 자료 형식 간에 변환이 이루어지지 못한다.
  4. 지침 자료에 직접 적용될 수 있는 연산자는 비교 연산자 =과 ]=뿐이다.
  5. 지침 자료는 STREAM 전송을 통해 읽히거나 쓰이지 못한다.
  6. 지침 변수는 BASED 속성을 갖지 못한다. 그러므로, 지침 변수는 BASED 속성을 가진 구조체의 요소가 되지 못한다.

(29)정도 (精度 Precision) (산수 자료 속성)

정도 속성은 변수의 값에 대해서 유지될 유의 숫자의 최소 수를 지정하기 위해 사용된다. 그리고 고정점 십진 변수에서 소수점을 지정하기 위해 사용된다. 정도 속성은 이진수와 십진수에 다같이 적용된다.

일반 형식 :

(숫자의 -수 [ , 척도 -인자 ] )

“숫자의 수”와 “척도 인자”는 무기호 십진 곱셈수 상수이다. “숫자의 수”는 영이 되지 못한다. 정도 속성 지정은 흔히 간단히 (p,q)로 표현된다. 여기서 p는 “숫자의 수”를 q는 “척도 인자”를 나타낸다.

일반 규칙 :

1. 정도 속성은 사이에 2개의 빈자가 있거나 없이 동일 인수로된 수준에 기수나 척도 (DECIMAL 또는 BINARY나, FIXED 또

는 FLOAT)속성 바로 뒤에 와야 한다.

2. #숫자의 수#는 변수에 대입되는 자료 항목을 위해 유지될 숫자의 수를 지정한다. 척도 인자(尺度因子 Scale Factor)는 소수 숫자의 수를 지정한다. 실제로 소수점이 나타나는 것이 아니다;이것의 위치가 가정될 뿐이다.
3. #척도 인자#는 소수점의 오른쪽 숫자의 수를 나타내는 십진 고른수 상수이다. 이는 십진 고정점 변수와만 사용될 수 있다. 이진 고정점 변수는 다만 고른수만을 표현하고 그런 고로 언제나 영의 가정된 척도 인자를 갖는다.
4. 척도 인자가 십진 고정점 자료에 지정되지 않으면, 이는 영으로 대행된다;이 말은,변수가 고른수를 표현한다는 것이다.
5. 척도 인자는 숫자의 수보다 클 수 있다. 그런 척도 인자는 언제나 소수(小数)를 지정한다. 가장 오른쪽 숫자에서 지정된 숫자의 수 만큼 왼쪽에 소수점이 있는 것이 된다. 사이에 낀 영이 가정되어 있으나,이들은 기억되지 않고;단지 지정된 숫자의 수만 실제로 기억된다.
6. 정도 속성은 PICTURE 속성과 결합해서 지정되지 못한다.
7. 조직/360 실무에서 허용된 숫자 수의 최대값은 십진 고정점수 자료에서 15, 이진 고정점수 자료에서 31, 십진 부동점수 자료에서 16, 이진 부동점수 자료에서 53이다. D-편성자에서 척도 인자는 15보다 크지 못하다.

태만 대행:

D-편성자에서 태만은 다음과 같다:

(5,0) DECIMAL FIXED

(15) BINARY FIXED

(6) DECIMAL FLOAT

(21) BINARY FLOAT

(30) PRINT (화일 서술 속성)

PRINT 속성은 화일의 자료가 마지막에 인쇄될 것임을 지정한다. PAGE, LINE, SKIP 선택항, PUT 문의, 과 OPEN 문의 PAGESIZE 선택항은 PRINT 속성을 가진 화일과만 사용될 수 있다. 이들 선택항은 제 10장, "문"에 설명되어 있다.

일반 형식 :

PRINT

일반 규칙 :

1. PRINT 속성은 OUTPUT와 STREAM 속성을 암시한다.
2. PRINT 속성은 개개 기록마디의 첫번의 자료 글자를 ASA 인쇄 통제 문자로 확보시킨다. 어떤 기록마디의 길이 지정도 이 통제 문자를 계상하여 1더하기 인쇄줄의 길이가 되어야 한다. 이들 통제 문자는 PAGE, SKIP, LINE 서식 항목 또는 선택항에 의해 세워진다.

대만 대행 :

만일 FILE이나 STRING 지정이 PUT 문에 나타나지 않으면, 표준 출력 화일이 대행된다.

### (31) RECORD와 STREAM (파일 서술 속성)

RECORD와 STREAM 속성은 파일 사용될 자료 전송의 종류를 지정한다. STREAM은 파일의 자료가 흐름(Stream)에서 변수로 또는 식에서 흐름으로 대입되는 문자열로 된 자료 항목의 연속적 흐름이라 여겨짐을 가르킨다. RECORD는 파일이 물리적 독립 기록마디의 집합체로 이루어지고, 이 기록마디들의 각각은 어떤 풀이든 하나 이상의 자료 항목으로 이루어짐을 가르킨다. 개개 기록마디는 직접 변수로, 변수에서 또는 직접 완충역으로, 완충역에서 일체(一體)로서 전송된다.

일반 형식 :

RECORD | STREAM

일반 규칙 :

1. STREAM 속성을 가진 파일은 OPEN, CLOSE, GET, PUT 문에서만 지정될 수 있다.
2. RECORD 속성을 가진 파일은 OPEN, CLOSE, READ, WRITE, REWRITE, LOCATE 문에서만 지정될 수 있다.
3. STREAM 속성을 가진 파일은 다음 속성의 어느 것도 갖지 못한다. : UPDATE, DIRECT, SEQUENTIAL, BACKWARDS, BUFFERED, UNBUFFERED, KEYED.
4. RECORD 속성을 가진 파일은 PRINT 속성을 갖지 못한다.

태만 태행 :

태만은 STREAM.

(32) RETURNS (입구 이름 속성)

RETURNS 속성은 이 선언의 범위 안에서 함수 인용에 사용되는 입구 이름에 DECLARE 문에서 지정된다. 이것은 입구 이름이 함수로서 불러내질 때 돌아가는 함수 값의 속성을 기술하기에 사용된다.

일반 형식 :

```
RETURNS (속성 . . .)
```

이것은 다음과 같은 식으로 사용된다 :

```
DECLARE 입구-이름 [ ENTRY ]  
      RETURNS (속성 . . . ) ;
```

일반 규칙 :

1. RETURNS 속성은 ENTRY 속성을 암시한다. 그러므로, ENTRY는 생략될 수 있다.
2. RETURNS 뒤의 괄호에 싸인 속성은 빈자로 떨어진다. 이들 속성은 입구 이름이 앞에 붙은 PROCEDURE 나 ENTRY 문에서 지정된 (또는 태만에 의해 대행된) 속성과 부합되어야 한다. 만일 돌아가는 실 값의 속성이 RETURNS 속성에 선언된 그것들과 부합하지 않아도, 변환이 수행되지 않는다.
3. 단지 산수, 줄, PICTURE, 또는 POINTER 속성만이 지정될 수 있다.
4. RETURNS 속성으로 선언된 내부 수속은 동일 선언에서 또한 INTERNAL 속성을 받아야 한다. 내부 함수에서, RETURNS 속성은 함수 수속과 동일한 블럭에 내적인 DECLARE 문에

서 서만 지정될 수 있다.

5. 입구 이름에 태만 속성이 적용되지 않는한, 함수에 관한 어떤 불려내기도 입구 이름을 위한 RETURNS 속성 지정의 범위 안에 나타나야 한다.

태만 대행 :

만일 RETURNS 속성이 함수 인용의 범위 안에 지정되지 않으면, 돌아가는 값에 대해 대행된 태만은 만일 입구 이름이 I부터 N까지의 문자로 시작하면 FIXED BINARY (15)이고; 아니면 FLOAT DECIMAL(6)이다. 태만 정도는 조직/360 실무에서 정한 것과 같다.

(33) SEQUENTIAL (화일 서술 속성)

DIRECT를 보라

(34) STATIC (기억소 종류 속성)

AUTOMATIC를 보라.

(35) STREAM (화일 서술 속성)

RECORD를 보라.

(36) UNALIGNED (자료 속성)

ALIGNED를 보라.

(37) UPDATE (화일 서술 속성)

INPUT를 보라.

## 제 10 장 : 문 (文, Statement)

이 장은 영자순으로 PL/I 문을 소개한다. 대부분의 문에서 다음 지식이 제공된다.

1. 기능 : 뜻에 관한 간략한 설명과 문의 용법 :
2. 일반형식 : 문의 구문 (構文)
3. 구문규칙 : 일반형식에 모순되지 않는 구문의 규칙
4. 일반규칙 : 문의 용법을 지배하는 규칙과 이것의 PL/I에서의 의미

### (1) 대입 문 (代入 文, Assignment Statement)

기능 (機能, Function) :

대입 문은 식의 값을 내서 요소·배열·구조체에 값을 대입된다.

일반형식 (一般形式, General Formats) :

대입 문은 다섯의 일반형식의 형 (型, Type) 이 있다. 이 형은 그림 10-1 에서 보인 바와 같다.

형 1 . 요소 대입

$$\left\{ \begin{array}{l} \text{요소 - 변수} \\ \text{유사 - 변수} \end{array} \right\} = \text{요소 - 식} ;$$

형 2 . 배열 대입

$$\text{배열 - 변수} = \left\{ \begin{array}{l} \text{배열 - 식} \\ \text{요소 - 식} \end{array} \right\} ;$$

형 3 . 구조체 대입

$$\text{구조체 - 변수} = \left\{ \begin{array}{l} \text{구조체 - 식} \\ \text{요소 - 식} \end{array} \right\};$$

형 4 . 문 명찰 대입 ( 2 가지 꼴 )

$$a . \text{요소 - 명찰 - 변수} = \left\{ \begin{array}{l} \text{명찰 - 상수} \\ \text{요소 - 명찰 - 변수} \end{array} \right\};$$

$$b . \text{명찰 - 배열} = \left\{ \begin{array}{l} \text{명찰 - 상수} \\ \text{요소 - 명찰 - 변수} \\ \text{명찰 - 배열} \end{array} \right\};$$

형 5 . 지침 대입 ( 2 가지 꼴 )

$$a . \text{요소 - 지침 - 변수} = \text{요소 - 지침 - 식};$$

$$b . \text{지침 - 배열} = \left\{ \begin{array}{l} \text{요소 - 지침 - 식} \\ \text{지침 - 배열} \end{array} \right\};$$

그림 10-1. 대입 문형

구문 규칙 ( 構文 規則, Syntax Rules ) :

1. 형 1 에서 받아 들이는 난에 있는 변수 ( 이는 등호의 왼쪽 ) 는 자료 형식이 산수나 줄인 단일 요소를 표현해야 한다.

2. 형 2 에서 받아 들이는 난에 있는 변수는 산수나 줄 요소의 배열을 표현해야 한다. 만일 요소 식이 등호의 오른쪽에 나타나면, 식의 값은 받아 들이는 난에 있는 배열의 개개 요소에 대입된다.

만일 배열 식이 등호의 오른쪽에 나타나면, 받아 들이는 난에 있는 배열과 식에 있는 모든 배열 연산항은 동일한 차원 수와 같은 한계를 가져야 한다.

3. 형 3에서, 받아 들이는 난에 있는 변수는 구조체를 표현해야 하고 구조체의 개개 요소는 산수나 줄 요소이어야 한다. (지침과 명찰 요소가 또한 허용된다. 그러나 이는 특별한 경우이다; 규칙 3을 보라.)

만일 요소 식이 등호의 오른쪽에 나타나면, 식의 값은 받아 들이는 난에 있는 구조체의 매 요소에 대입된다.

만일 구조체 식이 등호의 오른쪽에 나타나면, 양쪽에 있는 구조체들의 상대적 구조가 같아야 한다.

4. 형 4, 항목 b에서, 만일 명찰 상수나 요소 명찰 변수가 오른쪽에 나타나면, 상수나 변수의 값은 받아 들이는 난에 있는 명찰 배열의 매 요소에 대입된다.

만일 명찰 배열이 오른쪽에 나타나면, 차원 수와 개개 차원의 한계가 받아 들이는 난에 있는 명찰 배열의 그것들과 같아야 한다.

5. 형 5에서, "요소 지침 식"은 요소 지침 변수이거나 요소 지침 값을 보내주는 함수 인용이어야 한다.

형 5, 항목 b에서, 만일 요소 지침 식이 등호의 오른쪽에 나타나면, 식의 값이 받아 들이는 난에 있는 지침 배열의 매 요소에 대입된다.

또 항목 b에서, 만일 지침 배열이 오른쪽에 나타나면, 차원과 한계는 받아 들이는 난에 있는 지침 배열의 그것과 같아야 한다.

일반규칙 :

1. 대입 문은 다음과 같이 값이 내진다.
  - a. 형 1, 4, 5에서, 첨자 있는 것이나 유사-변수로 받아 들이는 난에 나타나는 어떤 식도 왼쪽에서 오른쪽으로 값이 내진다. 등호의 오른쪽에 있는 식은 값이 내지고 이것의 값이 받아 들이는 난에 있는 변수에 대입된다.
  - b. 형 2, 3에서, 대입문은 관계되는 배열이나 구조체의 해당 요소를 가지고 있는 요소 대입 문의 연속으로 취급된다. 배열에서, 요소는 행대순으로 대입되고; 구조체에서, 요소는 이들이 선언된 순서로 대입된다. 뒷 자리에 관한 값 내기의 결과는 배열이나 구조체에서 앞 자리의 값내기와 대입에 의해 영향을 받을지도 모를을 유의하라(다음에 오는 보기 1을 보라).
  - c. 필요할 때는, 오른쪽에 있는 식의 값은 제6장, "자료변환"에서 주어진 규칙에 따라서 받아 "들어는 난에 있는 변수의 특성으로 변환된다.
2. 받아 들이는 난에 있는 변수가 줄이거나 유사-변수 UNSPEC 일 때, 오른쪽에 있는 식은 일반규칙 1에서처럼 값이 내지고, 대입은 가장 왼쪽 비트나 문자 자리에서 시작해서 왼쪽에서 오른쪽으로 수행된다. 다음이 또한 적용될 수도 있다.
  - a. 만일 식의 값이 줄보다 길으면, 이 값은 줄의 길이에 맞게 오른쪽에서 잘린다.
  - b. 만일 식의 값이 줄보다 짧으면, 이 값은 비트 줄에서는 영으로 그리고 문자 줄에서는 빈자로 오른쪽에서 늘려진다.
3. 만일 지침이나 명찰 변수가 받아 들이는 난에 나타나는 구

구조체의 요소이면, 이것은 구조체에 있는 다른 요소와 꼭 같은 모양으로 값이 대입된다. 그렇지만, 변환이 수반되지 않고, 따라서 그런 지침이나 명찰 변수에 대입되는 값은 다른 지침이나 명찰 변수이어야 한다.

4. 형 '4, 5에 각각 보인 바와 같이 명찰 배열과 지침 배열 대입은 일반규칙 1에 있는 배열 대입에 관한 규칙을 따른다.

보기 1 :

다음 보기는 배열대입을 설명해 준다.

A → 2	4	B → 1	5
	3		7 8
	1		3 4
	4		6 3

$A = (A+B) ** 2 - A(1, 1);$

실행 뒤에, A는 다음 값을 갖는다.

7	74
93	189
9	114
93	114

A(1, 1)에 관한 새로운 값 7이 다른 모든 요소에 관한 값대기에 사용되었음을 유의하라.

보기 2 :

다음은 줄 대입을 설명해 준다.

A는 줄 'XZ/BQ',

B는 줄 'MAFY',

C는 길이 3의 줄,

D는 길이 5의 줄이라 한다.

C = A, C의 값은 'XZ/',

C = 'X', C의 값은 'Xbb',

D = B, D의 값은 'MAFYb',

D = SUBSTR(A, 2, 3) || SUBSTR(A, 2, 3),

D의 값은 'Z/BZ/'

SUBSTR(A, 2, 4) = B, A의 값은 'XMAFY'.

SUBSTR(B, 2, 2) = 'R', B의 값은 'MRby'.

보기 3 :

다음 보기는 (여기서 A, B, C는 요소변수) 요소대입을 설명해 준다.

A = A + SIN(B) + C \*\* 2;

보기 4 :

다음 보기는 구조체 대입을 설명해 준다.

```
a . DECLARE 1 X, 2 Y, 2 Z, 2 R, 3 S, 3 P, 1 A, 2 B, 2 C, 2 D,
      3 E, 3 Q ;
```

X = X + A ;

이것은 다음과 동가이다:

$X \cdot Y = X \cdot Y + A \cdot B ;$

$X \cdot Z = X \cdot Z + A \cdot C ;$

$X \cdot S = X \cdot S + A \cdot E ;$

$X \cdot P = X \cdot P + A \cdot Q ;$

b . DECLARE 1. A, 2 B, 2 C, 3 D, 3 E ;

$A = A + A \cdot B ;$

이 대입 문은 다음과 동가이다 :

$A \cdot B = A \cdot B + A \cdot B ;$

$A \cdot C = A \cdot C + A \cdot B ;$

나중 것은 다음과 동가이다.

$A \cdot D = A \cdot D + A \cdot B ;$

$A \cdot E = A \cdot E + A \cdot B ;$

보기 5 :

다음 보기는 문 명찰 대입을 설명해 준다 :

DCL P LABEL ;

P=A ;

GO TO P ; . . .

A : X=Y\*\*\*2 ;

이 일련의 문은 GO TO P가 실행될 때 통계를 A로 이행 (移行) 시킨다.

보기 6 :

다음 보기는 부동점수 자료에 대입되는 모양 서술에 의해 정의된 자료의 변환과 그 반대를 보여준다 :

DECLARE A FLOAT , B PICTURE '999V99'.

A = B ; ( B가 고정점수에서 부동점수로 변환된다 )

B = A ; ( A가 부동점수에서 고정점수로 변환된다 )

## (2) BEGIN 문

기 능 :

BEGIN 문은 시작 블럭의 첫머리가 되며 시작 블럭임을 가리킨다.

일반형식 :

BEGIN ;

일반규칙 :

BEGIN 문은 시작 블럭 ( 始作 블럭 Begin Block ) 를 구분하기 위해 END 문과 연합해서 사용된다. 시작 블럭에 관한 완전한 논술은 제 1부, 제 6 장, "블럭, 통제의 흐름, 기억소 할당"에서 찾아 볼 수 있다.

## (3) CALL 문

기 능 :

CALL 문은 수속을 불러내어 통제를 그 수속의 지정된 입구점으로 옮겨 보낸다.

일반형식 :

CALL 입구—이름 [ ( 인수 [,인수] . . . ) ] ;

구문규칙 :

1. 입구 이름은 불러 내지는 수속의 입구점 (入口點, Entry Point) 을 나타낸다.
2. 인수 (引數, Argument) 는 기초된 변수, 내조립 함수 이름, 연산 구조체식 또는 연산 배열식을 제외하고는 어떤 식도 될 수 있다.  
유효한 인수의 보기로는 소구조체 이름, 명찰 변수, 입구 이름, 지침 식, 줄 상수, 배열 이름, 화일 이름들이 있다.  
그렇지만, 만일 인수의 속성이 이것과 대응하는 매개변수와 맞지 않아도, 변환이 일어나지 않고 오류가 초래될 것이다.

일반규칙 :

불러내지는 수속에서 인수를 표현하는 매개변수와 인수의 상호관계에 관한 상세한 서술에 관해서는 제 1 부, 제 10 장, "버금과정과 함" 을 보라.

(4) CLOSE 문

기 능 :

CLOSE 문은 앞에서 여는 것에 의해 연관되었던 자료 집합으로

부터 호명된 파일을 분리한다. 이것은 또한 만일 INPUT 나 OUTPUT 와 PAGESIZE 가 만일 그 파일의 열기에서 지정되면, 지정된 파일로부터 분리한다. 그렇지만, DECLARE 문에서 그 파일에 명시로 지정된 모든 속성은 효력이 남아 있다.

일반형식 :

```
CLOSE FILE ( 파일—이름 ) [ , FILE ( 파일—이름 ) ] . . . ;
```

일반규칙 :

1. "파일 이름"은 닫혀질 파일을 가리킨다. 두개 이상의 그러한 지정이 하나의 CLOSE 문에 지정될 수 있으므로, 두개 이상의 파일이 하나의 CLOSE 문에 의해 닫혀질 수 있다.
2. 닫힌 파일은 ( INDEXED 파일을 제외하고 ) 이것이 닫혀진 후에 재개 (再開) 될 수 있다.
3. 열리지 않은 파일을 닫거나 이미 닫혀진 파일을 닫는 것은 효과가 없다.
4. 만일 CLOSE 문에 의해 닫히지 않으면, 이것은 이것이 열린 프로그램의 종료 (終了) 와 더불어 자동으로 닫힌다. 그렇지만, 중첩 사용에 대해서는 IBM 조직 / 360 원반과 테이프 운영 조직, PL/I 프로그래머 안내서, Order No. GC 24-9005 의 "중첩 Overlay" 에 있는 규칙 제 13 을 보라.

(5) DECLARE 문

기능 :

DECLARE 문은 이름의 속성을 명시 선언하는 주요한 수단이다.

일반형식 :

```
DECLARE [수준] 포식어 [속성] . . .  
[ , [수준] 포식어 [속성] . . . ] . . . ;
```

구문규칙 :

1. "수준"은 무기호 영이 아닌 십진 고른수 상수이다. 이것은 구조체 선언에만 나타날 수 있다 ;  
대구조체는 수준 1 을 가져야 한다. 빈자가 수준과 포식어 사이에 있어야 한다.
2. 보통, 속성은 이것이 적용되는 포식어 바로 뒤에 와야 한다 (일반형식에 보인 바처럼). 그렇지만, 여러 이름의 선언에 공통한 속성은 여러 포식어에 관한 동일속성의 반복 지정을 피하기 위해 인자로 될 수 있다. 인자로 하는 것은 괄호에 포함되는 선언을 둘러싸고 (비공통 속성을 포함) 이 위에 공통속성의 조(組)를 놓음으로써 성취된다. 인자로 된 수준번호의 경우, 수준 번호 (水準 番號 Level Number)는 괄호에 싸인 나열의 앞에 와야 한다 (빈자가 인자로 된 수준 번호와 왼쪽 괄호 사이에 요구되지는 않는다). 인자로 만들기는 8 의 수준까지 품어질 수 있다. 인자로 만들기의 보기에 관해서는, 제 9 장 "속성"에 있는 "속성을

인자로 만들기 "를 보라.

일반규칙 :

1. 대구조체 포식어나 구조체에 포함되어 있지 않은 포식어는 특정 블럭 안에 있는 DECLARE 문에서만 지정될 수 있다. 그 포식에 관해 명시로 주어진 모든 속성은 그 DECLARE 문에서 함께 선언되어야 한다. (그렇지만, FILE 속성을 가진 어떤 포식어는 할 수 있다면 OPEN 문에서 INPUT 또는 OUTPUT 속성을 받을 수 있다. 이 장에 있는 "OPEN 문"과 제I부, 제8장 "입력과 출력"을 보라).
2. 독립된 블럭과 편성에서 (Compilation) 외부 이름의 속성은 맞아야 한다.
3. 명찰이 DECLARE 문에 붙어도 된다 (그렇지만, 이런 명찰은 주석 (Comment) 으로 취급되며, 따라서 의미가 없다). 조건 찬치어가 DECLARE 문에 붙지 못한다.
4. DECLARE 문의 갯수는 제한받지 않는다. 그러나 한개 DECLARE 문안에 있는 선언의 갯수에는 제한받는다 (이 갯수는 편성자가 진행되는 해당된 기억소 크기에 좌우된다. 이 점은 긴 구조체에 중요한 의미를 갖는다).

(6) DISPLAY 문

기 능 :

DISPLAY 문은 기계 운전원 (運轉員, Operator) 에게 전언문 (傳言文, Message) 을 표명 (表明) 시킨다. 선택항이 운전원에게 응

당을 위해 허용된다.

일반형식 :

DISPLAY(요소-식) [ REPLY ( 문자-줄-요소-변수 ) ] ;

구문규칙 :

" 문자-줄 요소 변수 "는 유사-변수가 되지 못한다.

일반규칙 :

1. DISPLAY 문의 실행은 요소 식의 값을 내서, 필요하다면, 문자 줄로 변환된다. 이 문자 줄이 운전원에게 표명될 전언문이다. D-편성자에서, 이것은 80 문자 길이를 초과하지 못한다.
2. 만일 REPLY 선택항이 지정되면, 기계 운전원은 선택항에 있는 문자-줄 요소 변수에 대입될 전언문으로 응답할 것이다. D-편성자는 응답의 길이를 제한하지 않는다 ; 그렇지만, 이 문자 줄 변수는 다른 어떤 문자 줄과 같이 255 문자를 넘지 못한다.
3. 만일 REPLY 선택항이 지정 안되면, 실행은 DISPLAY 문의 실행 뒤에 중단 없이 계속한다.
4. 만일 REPLY 선택항이 지정되면, 프로그램의 실행은 운전원의 응답이 끝날 때까지 중지된다.

## (7) DO 문

### 기능 :

DO 문은 DO-모임의 머리가 되며 모임 안에 있는 문의 실행을 반복 지정하기에 사용될 수 있다.

### 일반형식 :

DO 문에 관한 세가지 형식의 형이 그림 10-2에서 보여준다.

### 구문규칙 :

1. 세가지 형에서, DO 문은 DO-모임을 마감하는 END 문과 연합해서 사용된다. 다만 형 1은 모임 안에 있는 문의 반복 실행을 하지 않는다.
2. 형 3에서, "변수"는 단일 요소를 표현해야 한다; 이는 첨자가 붙지 못한다. 산수 변수가 보통으로 사용되나, 만일 앞으로 올 일반규칙에서 전개한 것이 유효한 PL/I 프로그램의 결과를 조태하면 명찰, 지침, 줄 변수가 허용된다. 그렇지만 만일 "변수"가 산수나 비트 줄이 아니면, "식 2"와 "식 3"은 생략되어야 한다.
3. 지정에 있는 개개 식은 요소 식이어야 한다.
4. 만일 "By 식3"이 "지정"에서 생략되고 만일 "To 식 2"가 나타나면, "식 3"은 1로 대행된다.
5. 만일 "To 식 2"가 "지정"에서 생략되면, 반복 실행은 WHILE 절(節, Clause)이나 모임 앞의 어떤 문에 의해서 끝나질 때까지 계속한다.

6. 만일 "To 식 2"와 "By 식 3"이 지정에서 모두 생략되면, 이것은 "식 1"의 값을 가진 통계 "변수"를 가지고 모임의 한번 실행을 암시한다. 이것은 만일 "WHILE 식 4"가 있더라도 같다.

형 1 . DO ;

형 2 . DO WHILE ( 요소-식 ) ;

형 3 . DO 변수=지정 [ , 지정 ] . . . ;

여기서 "지정"은 다음 꼴을 가짐 :

$$\text{식 1} \left\{ \begin{array}{l} \text{TO 식 2 [ BY 식 3 ]} \\ \text{BY 식 3 [ TO 식 2 ]} \end{array} \right\} [ \text{WHILE ( 식 4 ) } ]$$

그림 10-2 DO 문의 일반형식

일반규칙 :

1. 형 1에서, DO 문은 단지 D-모임 ( Do-Group )의 개시를 구분해 줄 뿐이다 ; 이것은 반복 실행을 제공하지 않는다.
2. 형 2에서, DO 문은 DO-모임의 개시를 구분하며 다음에 의해 정의된 반복 실행을 준비한다 :

LABEL : DO WHILE ( 식 ) ;

문 - 1

. . .

문 - n

END ;

NEXT : 문

/\* STATEMENT FOLLOWING THE DO GROUP \*/

위의 것은 다음 전개와 정확히 동가이다.

LABEL : IF ( 식 ) THEN ;

ELSE GO TO NEXT ;

문 - 1

.

.

문 - n

GO TO LABEL ;

NEXT : 문

/\* STATEMENT FOLLOWING THE DO GROUP \*/

3. 형 3 에서, DO 문은 DO - 모임의 개시를 구분하며 다음과 같이 정의된 바처럼 통제된 반복 실행을 준비한다.

LABEL : DO 변수 = 식 1 TO 식 2

BY 식 3 WHILE ( 식 4 ) ;

문 - 1

.

.

.

문 - n

LABEL : END ;

NEXT : 문

위의 것은 다음 전개와 정확히 동가이다.

```

LABEL : e1 =식 1 ;
        e2 =식 2 ;
        e3 =식 3 ;
        V = e1 ;
LABEL2 : IF ( e3 >= 0 ) ε ( V > e2 ) | ( e3 < 0 ) ε ( V < e2 )
        THEN GO TO NEXT ;
        IF ( 식 4 ) THEN ;
        ELSE GO TO NEXT ;
문 - 1
.
.
.
문 - n
LABELY : V = V + e3 ;
        GO TO LABEL2 ;
NEXT : 문

```

위 전개 (展開) 에서  $e_1, e_2, e_3$  는 각각 "식 1", "식 2", "식 3" 의 속성을 가진 편성자가 만든 작업 영역이다;  $V$  는 "변수" 와 같다.

a. 앞의 전개는 다만 한개의 "지정" 결과만을 보인다. 만일 DO 순이 둘 이상의 "지정" 을 포함하면, NEXT 명찰이 붙은 문은 다음에 오는 "지정" 을 위한 전개에서 첫 번째 문이 된다. 둘째번 전개는 여러가지 면에서 첫 번째 전개와 흡사하다. 그래서, 만일 둘째번 "지정" 이 DO

문에 나타나면 둘째번 전개는 다음과 같을 것이다.

```
NEXT : e5=식5 ;  
.  
.  
.  
V=e5 ;  
LABEL3 : IF . . . THEN GO TO NEXT X ;  
IF (식8) THEN ;  
ELSE GO TO NEXT X ;  
문-1  
.  
.  
.  
. . .  
문-m  
LABEL 4 : V=V+e7 ;  
GO TO LABEL3 ;  
NEXT X: 문
```

문 1 부터 m 은 프로그램에서 실제로 중복하지 않음을 유의하라.

- b . 만일 WHILE 절이 생략되면 , 전개에서 문-1 바로 앞의 IF 문은 생략된다.
- c . 만일 " TO 식2 "가 생략되면 , 문 e2 =식2 와 LABEL 2 가 붙은 IF 문은 생략된다.
- d . " TO 식2 "와 " BY 식3 "이 둘다 생략되면 , 문 GO TO

LABEL 2 와 함께 e2, e3 을 포함한 모든 문이 생략된다.

4. 형 2 와 형 3 은 WHILE 절은 문 실행의 매 반복 전에, 연관된 요소 식은 값이 내지고, 그리고 만일 필요하다면, 비트 줄로 변환됨을 지정한다. 만일 결과의 줄에서 어떤 비트가 1 이면, DO-모임의 문들이 실행된다. 만일 모든 비트가 0 이면, 그러면 형 2 에서 DO-모임의 실행이 끝나고, 형 3 에서 WHILE 절을 포함한 "지정"과 연관된 실행만 끝난다; 만일 존재하면, 다음 "지정"에 관한 반복실행이 시작한다.
5. "지정"에서, "식 1"은 통제 "변수"의 초기값을 표현한다; "식 3"은 모임에 있는 문들의 매 반복 뒤에 통제 변수에 더해질 증분(增分, Increment)을 표현한다; "식 2"는 통제 "변수"의 마지막 값을 표현한다. DO 모임에 있는 문의 실행은 통제 "변수"의 값이 "식 1"과 "식 2"에 의해 정해진 범위가 넘자마자 "지정"에 대한 것을 끝낸다. 마지막 "지정"에 관한 실행이 끝날 때, 통제는 DO-모임 다음의 문으로 간다.

주: 통제 변수는 끝내는 값으로서 지정된 값을 넘는 값을 포함될 수 있어야 한다:

```
DECLARE X PICTURE '99' ;
```

```
DO X = 0 TO 99 ;
```

```
•
```

```
•
```

```
END ;
```

```
DO X = 99 TO 0 BY -1 ;
```

END ;

이들 모임은 절대로 끝나지 않는다. 이는 X가 끝나는 값 100 또는 -1을 각각 포함할 수 없기 때문이다.

6. 통제는 만일 DO - 모임이 형 1에서의 DO 문에 의해서 구분될 때만 DO - 모임 밖에서부터 DO - 모임 안으로 이행할 수도 있다; 이 말은, 만일 반복실행이 지정되지 않을 경우이다. 그런 결과로 반복 DO - 모임은 ENTRY 문을 포함하지 못한다.

#### (8) END 문

기능 :

END 문은 블럭 (Block) 와 모임 (Group) 을 끝낸다.

일반형식 :

END [ 문—명찰—상수 ] ;

일반규칙 :

1. END 문은 언제나 해당하는 END 문이 없는 가장 앞에 오는 DO, BEGIN, PROCEDURE 문에 의해 시작되는 모임이나 블럭을 끝낸다. 그래서 만일 문 명찰 상수가 END 뒤에 붙으면, 이것은 DO, BEGIN 또는 PROCEDURE 문의 명찰이어야 한다.

만일 END가 둘 이상의 명찰이 부착된 DO나 BEGIN 문과 대응하면, END 뒤에 붙은 명찰은 열쇠말 DO나 BEGIN 바로 앞에 오는 것이라야 한다.

2. 만일 통제가 수속에 관한 END 문에 닿으면, 이것은 RETURN 문처럼 취급된다.

### (9) ENTRY 문

기능 :

ENTRY 문은 수속의 이차 입구점 (二次 入口點, Secondary Entry Point) 을 지정한다.

일반형식 :

입구 이름 : ENTRY [ ( 매개변수  
[ , 매개변수 ] . . . ) ] [ 속성 ] . . . ;

구문규칙 :

1. ENTRY 문에 지정될 수 있는 유일한 속성은 산수, 줄, PICTURE, 지침 속성이다. 지정된 속성은 이것이 이 입구에서 함수 인용으로 불러 내질 때 수속에 의해 돌아오는 값의 특성을 결정한다.
2. 조건 전치어는 ENTRY 문에 지정되지 못한다.
3. ENTRY 문은 이것에 부착된 한개의 입구 이름만을 가질 수

있다.

4. 12 개를 초과하는 매개변수가 한개 ENTRY 문에 나타나지 못한다. ( 한개 수속에 있는 매개변수의 전합계는 12 개를 넘지 못한다 )

일반규칙 :

1. 이차 입구점의 매개변수와 이 입구점에 보내진 인수 사이에 설정된 관계는 일차 입구점 매개변수와 인수사이에 설정된 그것과 같다. 이 사항에 관한 완전한 논술에 관해서는 제 I 부, 제 10 장 "버금과정과 함수" 를 보라.
2. 규칙 1에서 설명된 바처럼, ENTRY 문에 지정된 속성은 이것이 이 입구점에서 함수 인용으로 불러 내질 때 수속에 의해 돌아가는 값의 특성을 결정한다. 이 수속에 의해 돌아오는 값은 ( 이것은 RETURN 문에 있는 식의 값이다 ), 만일 필요하면, 지정된 속성에 맞게 변환된다. 만일 속성이 입구점에 지정 안되면, 태만 속성이 입구점을 불러내기에 사용되는 입구 이름의 첫째 글자에 따라 제공된다. 이차 입구점의 속성은 ( 태만에 의하건 다른 방법에 의하건 ) 만일 수속이 함수 인용으로 사용되면 일차 입구점의 그것과 정확히 같아야 한다.
3. ENTRY 문은 이차 입구점을 정의한 수속에 내적이어야 한다. 이는 이 수속에 포함되어 있는 어떤 블럭에도 내적

이 아니다 ; 또 반복실행을 지정하는 DO 모임 안에 들어가지 못한다.

4. 이차 입구점의 매개변수는 이 블럭 안에서 명시 선언되어야 한다 (이 말은, PROCEDURE 문의 매개변수 나열 안에 있거나 DECLARE 문에 있거나 또는 양자 다거나).

5. D-편성자에서, 외부 이름의 최대길이는 여섯자이다. 그런대로, 외부수속에 내적인 ENTRY 문의 이름은 여섯자보다 크면 안된다.

#### (10) FORMAT 문

기능 :

FORMAT 문은 전송될 자료의 서식을 관리하는 편집-지시 전송 문에 의해 사용될 수 있음을 지정한다.

일반형식 :

명찰 : [ 명찰 : ] . . . FORMAT ( 서식-나열 ) ;

구문규칙 :

1. "서식나열"은 제 I 부, 제 8 장, "입력과 출력"에서 서술된 바처럼 편집-지시 전송과 더불어 서식 나열을 지배하는 규칙에 따라 지정되어야 한다.

2. 적어도 한개의 "명찰"이 FORMAT 문에 지정되어야 한다. 보통 이 명찰 중의 하나가 (또는 이 명찰 중의 하나를 가

진 명찰 변수) 외판 서식 항목 ( Remote Format Item ) 에  
서 지정된 문 명찰 지시자 ( 指添子 ) 이다.

일반규칙 :

- 1 . GET 나 PUT 문은 편집-지시 ( Edit-Directed ) 자료 지정의  
서식 나열에서 외판 서식 항목 R 을 가질 수 있다 . R 에  
의해 표현된 서식 나열의 부분은 R 로 지정된 문 명찰에  
의해 FORMAT 문으로 공급되어야 한다 . R 서식항목은  
FORMAT 문의 서식 나열에 나타나지 못한다 .
- 2 . 외판 서식항목과 FORMAT 문은 동일 블럭에 내부적이어서  
한다 .
- 3 . 단일 조건 전치어가 FORMAT 문과 연관되면 , 이것은 FORMAT  
문을 인용하는 GET 나 PUT 문과 연관된 조건 전치어와 같  
아야 한다 .

(11) GET 문

기능 :

GET 문은 다음 방법의 하나로 사용될 수 있는 STREAM 전송  
문이다 :

- 1 . 이것은 외부의 원천에서 ( 이는 , 자료집합에서 ) 하나 이상의  
내부의 받아 들이는 난 ( 이는 , 하나 이상의 변수 ) 으로 , 자료  
를 대입케 한다 .
- 2 . 이것은 내부 원천에서 ( 이것은 , 문자-줄 변수에서 ) 하나 이  
상의 내부의 받아 들이는 난 ( 이것은 , 하나 이상의 변수 ) 으  
로 자료를 대입케 한다 .

일반형식 :

```
FILE (파일-이름)
GET(                ) 자료-지정 ;
STRING (문자-줄-번호)
```

구문규칙 :

1. "자료지정"은 제 I부, 제 8 장, "입력과 출력"에서 서술된 바와 같다.
2. "자료지정"은 FILE이나 STRING 선택항 뒤에 와야 한다. 만일 지정되면.
3. "문자 줄 번호"는 "자료지정"에 있는 변수에 대입되는 값을 준비하고 있는 문자 줄을 인용하는 것이다.
4. "파일이름"은 "자료지정"에 있는 변수에 대입될 값을 준비할 자료집합과 연관되어 있는 (암시 열기에 의하건 명시 열기 (Opening)에 의하건) 파일의 이름이다. 이것은 STREAM 과 INPUT 속성을 가져야 한다.
5. 만일 FILE도 STRING도 나타나지 않으면, 표준 조직 입력 화일이 대행된다.

일반규칙 :

1. 만일 FILE 선택항이 안열린 화일을 인용하면, 화일은 암시로 열린다.
2. 만일 STRING 선택항이 지정되면, 내부 GET 연산이 언제나 지정된 줄의 처음에서 시작한다. 만일 이 줄에 있는 문자의 수가 "자료지정"에 있는 변수에 요구되는 총 문자의 수보다 작으면, ERROR 조건이 일어난다. "자료지정"에 있는 변수는

문자 줄이 아니어도 됨을 유의하라; 내부 대입은 흐름에서 내부 기억소로 전송하는 것과 같다. 단지 다른 점은 "문자 줄 변수"가 입력 흐름으로 간주될 수 있다는 것 뿐이다.

## (12) GO TO 문

기능:

GO TO 문은 통제를 지정된 명찰에 의해 정해지는 문으로 이행시킨다 (移行).

일반형식:

```
GO TO 명찰-상수  
GOTO 요소-명찰-변수
```

일반규칙:

1. 만일 "요소명찰변수"가 지정되면, 이 명찰 변수의 값은 통제가 옮겨갈 문을 결정한다. 명찰 변수는 GO TO 문의 실행마다 틀리는 값을 가질 수도 있으므로, 통제가 언제나 동일 문으로 가지는 않는다.
2. GO TO 문은 비활동 (非活動) 블럭으로 통제를 보내지 못한다.
3. GO TO 문은 만일 DO-모임이 반복 실행을 지정하면 DO-모임의 밖에서 DO-모임 안에 있는 문으로 통제를 옮겨주지 못한다. 이것은 GO TO 문이 DO-모임의 안으로부터 불러내진 수속을 끝내지 않는 한 또는 GO TO가 DO-모임

안으로부터 통제를 부여받은 ON-단위가 아닌 한 그렇다.

4. 만일 GO TO 문이 한 블록의 안에서 그 블록 안에 포함되어 있지 않는 점으로 통제를 옮기면, 이 블록은 끝난다. 또, 만일 이행점(移行點)이 끝나게 될 블록을 직접 기동시키지 않는 블록 안에 들어 있으면, 활동순서상의 모든 중간 블록도 또한 끝난다(제 I 부, 제 6 장, "블록, 통제의 흐름, 기억소 할당"을 보라). 하나 이상의 블록이 GO TO 문에 의해 끝나칠 때, 조건들은 원상복귀되고 자동 변수(Automatic Variable)들은 마치 블록이 보통의 방법으로 끝나했을 때처럼 해방된다.
5. GO TO 문이 함수로서 불러내진 수속 밖으로 통제를 옮기면, 해당 함수 인용을 포함한 식의 값내기는 중지된다.
6. 만일 GO TO 문이 ON-단위이면, 지정된 명찰이 첨자가 붙지 않아야 한다.

### (13) IF 문

능 :

IF 문은 지정된 식의 값을 시험하여 그 시험의 결과에 따라서 실행의 흐름을 통제한다.

일반형식 :

IF 요소-식 THEN 단위-1 [ ELSE 단위-2 ]

구문규칙 :

1. 각 "단위"는 단일 문 ( DO, END, PROCEDURE, BEGIN, DECLARE, FORMAT, ENTRY는 제외 ), DO-모임, 또는 시작 블럭 중의 하나이다.
2. IF 문 자체는 세미콜론 ( ; ) 으로 끝나지 않는다 ; 그렇지만 개개 "단위"는 세미콜론으로 끝나야 한다.
3. 각 "단위"는 명찰이 붙거나 조건 전치어를 가져도 된다. ( 단일 "단위"에 명찰이 붙으면, 통제가 IF 문 밖에서부터 이것으로 옮겨질 수 있다.

일반규칙 :

1. 만일 필요하다면, 요소식은 값이 내리고 비트 줄로 변환된다. ELSE절이 ( 이것은, ELSE와 이 뒤에 오는 "단위" ) 지정되면, 다음이 일어난다 :  
만일 줄에 있는 어떤 비트가 1이면, "단위 - 1"이 실행되고, 통제는 IF 다음의 문으로 간다. 만일 줄에 있는 모든 비트가 값 0을 가지면, "단위 - 1"이 지나치고 "단위 - 2"가 실행된다. 그 뒤에 통제는 다음 문으로 간다. ELSE절이 지정 안되면, 다음이 일어난다 :  
만일 줄에 있는 어떤 비트가 1이면, "단위 - 1"이 실행된다. 그리고 통제는 IF 문 다음의 문으로 간다. 만일 모든 비트가 0이면, "단위 - 1"이 실행되지 않고 통제는 다음 문으로 간다.  
개개 "단위"는 통제를 옮기는 문 ( 보기로, GO TO ) 을 가질 수 있다 ; 그러면, IF 문의 정상 순서는 무시될 수 있다.

2. IF 문은 품어질 수 있다 ; 이 말은, "단위" 중의 하나 또는 모두가 자체가 IF 문이 될 수 있다는 것이다. 개개 ELSE는 동일 블록 또는 DO-모임에서 가장 안쪽의 짝없는 IF와 연관되므로, 빈문(Null Statement)인 ELSE가 원하는 통제의 순서를 지정하기 위해 필요할 때도 있다.
3. IF 문 자체에 붙은 조건 전치어는 "요소 식"에만 적용된다.
4. 산수 항목의 변환은 (보기로, IF X=Y THEN . . . 에서) 산수 변환에 관한 정도 규칙을 따른다 (제 6 장 : 자료 변환을 보라)

#### (14) LOCATE 문

기능 :

LOCATE 문은 BUFFERED 속성을 가진 연속 출력 화일에만 사용될 수 있는 RECORD 전송 문이다. 이것은 기초된 변수에 관한 기록마디의 생성을 허용하기 위해 출력 완충역에 있는 기초된 변수에 기억소를 할당한다. 기록마디는 완충역 안에 있는 기초된 변수에 값을 대입함으로써 만들어진다. 기록마디는 다음에 오는 WRITE, LOCATE, CLOSE 문 (암시 닫기 연산이전) 이 지정된 화일에 실행되기 바로 전까지 외부 매체로 전송되지 않는다.

일반형식 :

```
LOCATE 기초된-변수 FILE (화일-이름)
      SET (지침-변수) ;
```

구문규칙 :

- 1 . FILE 과 SET 지정은 일반형식에서 보인 순서로 나타나야 한다.
- 2 . 기초된 변수는 소구조체나 구조체의 요소가 아닌 첨자 안 붙은 기초된 변수이어야 한다.
- 3 . 지침 변수는 첨자가 붙었거나 안붙은 요소, 지침 변수여야 한다.
- 4 . 화일 이름은 최후에는 기록마디를 받아 들일 자료집합과 연관되어 있는 ( 열기에 의해 ) 화일의 이름이다.

일반규칙 :

- 1 . 기초된 변수는 가변-길이 기록마디에서 기록마디의 길이를 결정하기 위해 사용된다. LOCATE 문이 실행될 때, SET 지정에 있는 지침변수는 기초된 변수가 할당되는 완충역에서 위치를 가리키게 세워진다.
- 2 . 기초된 변수에 의해 가리켜진 기록마디는 그 화일에 관한 다음의 WRITE, LOCATE 또는 CLOSE 연산 ( 암시건·명시건 ) 바로 전에 완충역에서 출력화일로 써진다. 블러크된 기록마디에서, 기록마디는 전체의 블러크가 다될 때까지 써지지 않는다. 기초된 변수에 의해 가리켜진 기록마디의 길이는 고정-길이 기록마디에서 8 로 나누어져야 하고, 가변-길이 기록마디에서 8 로 나눈 뒤에 나머지 4 가 되어야 한다.
- 3 . FILE 지정은 이미 열린 화일을 인용해야 한다.

(15) 빈문 ( NULL 文 )

기 능 :

빈 문은 행위가 없고 순차적 문의 실행을 변경하지 않는다.

일반형식 :

[ 명찰 : ] . . . ;

(16) ON 문

기 능 :

ON 문은 중단이 지정된 예외 조건 때문에 초래될 때 취해질 행위 ( 프로그래머가 정한 또는 표준 조직 행위 ) 를 지정한다.

일반형식 :

ON 조건 { SYSTEM ; | ON - 단위 }

구문규칙 :

1. 조건은 제 8 장, " ON - 조건 " 에서 서술된 것 중의 하나이어야 한다.
2. " ON - 단위 " 는 중단이 지정된 " 조건 " 의 출현 때문에 초래될 때 취해질 프로그래머가 정한 행위를 표현한다. 이것은 단일의 명찰 안 붙은 GO TO 나 GOTO 문이다.
3. " ON - 단위 " 자체는 세미콜론을 요하므로, 일반형식에서 " ON 단위 " 에 세미콜론을 보이지 않았다. 그렇지만, 말 SYSTEM 은 세미콜론이 뒤따라야 한다.

일반규칙 :

1. ON 문은 지정된 조건에 일어난 중단을 취급하는 방법을 결정한다. 중단이 표준 조직방식으로 취급되는가 또는 프로그래머가 공급한 방법으로 취급되는가는 다음과 같이 ON - 문에 있는 행위지정에 의해 결정된다 :

a. 만일 행위지정이 SYSTEM이면, 표준조직행위가 취해진다. 표준조직 행위는 모든 조건에 동일하지 않다. 그러나 대부분의 조건에서 조직은 단순히 전언문을 인식하고 ERROR 조건을 일으킨다. 제 8 장, "ON - 조건"은 각 조건에 관한 표준 조직 행위를 제공한다 (표준 조직 행위는 언제나 만일 중단이 일어나고 그 조건에 관한 ON 문이 효력이 없으면 취해진다).

b. 만일 행위 지정이 "ON - 단위"이면, 프로그래머는 자신이 공급한 중단취급 행위, 다시 말해서 ON - 단위 자체에 있는 문으로 정의된 행위를 갖는다.

ON - 단위는 ON 문이 실행될 때는 실행되지 않는다; 이것은 지정된 조건의 출현으로 중단이 초래될 때만 실행된다 (또는 만일 SIGNAL 문에 의해 일어난 조건에 의해 중단이 초래되면).

2. 주어진 블럭에서 ON 문의 실행에 의해서 설정된 행위 지정 (이는, "ON - 단위"나 SYSTEM)은 이 블럭의 전체와 이 블럭에 의해 시작되는 어떤 활동순서에 있는 모든 블럭 전반에 유효하게 남아 있다.

이는 다음과 같이 다른 ON 문이나 REVERT 문의 실행

에 의해 무효가 되지 않는 한 그렇다.

- a. 만일 나중의 ON 문이 앞의 ON 문과 동일한 조건을 지정하고 그리고 이 나중의 ON 문이 앞의 ON 문을 포함하고 있는 블럭에 의해 시작되는 활동순서 안에 놓인 블럭에서 실행되면, 앞의 ON 문의 행위지정은 임시로 보류되거나 정지된다. 이것은 REVERT 문의 실행에 의하거나, 또는 나중의 ON 문을 포함하고 있는 블럭이 끝마침에 의해 회복된다.
- b. 만일 나중의 ON 문과 앞의 ON 문이 동일 블럭의 동일 블럭내기에 내적이면, 앞의 ON 문의 효력이 완전히 무효로 된다.

3. GO TO 문 ON-단위의 명칭은 그 ON-단위를 위한 ON 문이 실행되는 블럭 안에 알려져야 한다. (ON 문은 문의 흐름이 감지될 때 실행된다; 반면, 그 조건에 관한 행위지정은 연관된 중단이 출현할 때만 실행됨을 기억하라. 또 GO TO에 관한 규칙 6을 보라)
4. 입력/출력조건외의 화일이름은 조건을 지정한 ON 문이 내적인 수속이나 시작 블럭 안에 알려져야 한다.
5. 실행중에 일어난 조건은 만일 이 것이 일어난 점에서 그 조건이 가능하게 되어 있는 경우만 중단이 초래된다.
  - a. SIZE 조건은 태만에 의해 불가능하게 된다. 다른 모든 조건은 태만에 의해 가능하게 된다.
  - b. OVERFLOW, FIXEDOVERFLOW, UNDERFLOW, ZERODIVIDE, CONVERSION, SIZE의 가능하게 하기와 불가능하게 하기는 조건 전치어에 의해 관리될 수 있다.

## (17) OPEN 문

### 기능

OPEN 문은 파일 이름을 자료 집합과 연관시키는 것에 의해 파일을 연다.

이것은 또 만일 완전한 조의 속성이 열릴 파일에 선언되지 않았으면 그 파일에 관한 속성의 지정을 보완할 수 있다.

### 일반 형식

```
OPEN FILE (파일-이름)      선택항-모임  
      [,FILE (파일-이름)   선택항-모임]...;
```

여기서 "선택항-모임"은 다음과 같다:

```
[ INPUT | OUTPUT ]
```

```
[ PAGESIZE (요소-식) ]
```

### 구문 규칙

1. INPUT 또는 OUTPUT 선택항은 UNBUFFERED 파일에 관한 OPEN 문에만 지정될 수 있다. 만일 이것이 OPEN 문에 지정되지 않으면, 해당하는 INPUT 또는 OUTPUT 속성이 그 파일에 관한 DECLARE에서 지정되어야 한다. INPUT 또는 OUTPUT는 OPEN과 DECLARE 문에서 동시에 지정되지 못한다.
2. FILE 지정은 제일 먼저 나타나야 한다.

3. "파일 이름"은 자료 집합과 연관된 파일의 이름이다.  
여러개의 파일이 한 개 OPEN 문에 의해 열려질 수 있다.

#### 일반 규칙

1. 이 열린 파일을 여는 것은 파일에 영향을 주지 않는다.  
이런 경우, "선택항 도입"에 있는 어떤 식도 값이 내진다. 그러나 이것들은 사용되지 않는다.
2. PAGE 선택항은 STREAM과 PRINT 속성을 가진 파일에 관해서만 지정될 수 있다. 요소 식은 값이 내지고 고른수로 변환된다. 이것은 한 지면(紙面)에 대한 최대 줄(line) 수를 나타낸다. 이 고른수는 영보다 크고 256보다 작아야 한다. PRINT 파일에 대한 지속적인 전송 동안, 새로운 지면이 PUT 문에서 PAGE 서식 항목에 의해서나 선택항에 의해 시작될 수 있다. D-편성자에서, 만일 PAGESIZE가 지정되지 않으면, 태만이 설치시에 지정된 조직 한계에 의해 정해진다.
3. PRINT 파일이 열려질 때, 새로운 지면이 시작된다.

#### (18) PROCEDURE 문

##### 기능

PROCEDURE 문은 다음의 기능을 가진다.

- 0 이것은 수속의 머리를 이룬다.

- 이것은 수속에 관한 일차 입구점을 정한다.
- 이것은 만일 있다면 일차 입구점에 관한 매개변수를 지정한다.
- 이것은 수속이 가질 수 있는 정해진 특성의 특성을 지정할 수 있다.
- 이것은 수속이 일차 입구점에서 함수로써 불러내질 때 수속에 의해 돌아가는 값의 속성을 지정한다.

#### 일반 형식

주. 수속에서 (Main Procedure) :

[ (조건-이름들) : ] 입구-이름 :

PROCEDURE OPTIONS (선택항-나열) ;

여기서, D-편성자에서 "선택항-나열"은 다음과 같이 정해진다.

```
MAIN [,ONSYSL0G]
```

버금과정에서 (subroutine)

[ (조건-이름들) : ] 입구-이름 :

PROCEDURE [ (매개변수 [ , 매개변수 ] ... ) ] ;

함수에서 (function)

[ (조건-이름들) : ] 입구-이름 :

PROCEDURE [ (매개변수 [ , 매개변수 ] ... ) ]

[ RETURNS (자료 - 속성) ] ;

[ 자료 - 속성 ]

#### 구문 규칙

1. "자료 속성은 수속이 일차 입구점에서 함수로서 불러내질 때 이 수속에 의해 돌아가는 값의 속성을 표현한다. 산수, 줄, PICTURE, LABEL, POINTER 속성만이 허용된다.

2. OPTIONS 는 특정 수속 지정이다.

이것과 "자료 속성"은 어떤 순서로도 나타날 수 있고 빈자로 떨어질 수 있다. OPTIONS 는 프로그램에서 한개의 외부 수속에만 지정될 수 있고 지정되어야 한다.

3. 한개의 입구 이름만이 PROCEDURE 문에 나타나야 한다.

4. 한개 수속에 지정될 수 있는 서로 다른 매개변수의 최대수 (ENTRY 문에 지정된 것도 합해서)는 12를 초과하지 못한다.

#### 일반 규칙

1. 수속이 불러내질 때, 하나의 관계가 수속에 보내진 인수와 불러내진 수속에서 이들 인수를 표현하는 매개변수 사이에 설정된다.

이 논제는 제 1부, 제 10 장, "버금과정과 함수"에 논술되어 있다.

2. OPTIONS 지정은 외부 수속에만 사용될 수 있다. MAIN 선택항은 이 수속이 초기 수속 (initial procedure)이며

프로그램의 실행에서 첫번 단계로서 운영조직 (operating system)에 의해 불러내질 것임을 지정한다. Onsyslog 선택항은 on 조건으로부터 연유된 행위에서 초래되는 모든 출력이 조직 일지 (組織 日誌 system log) 기록될 것임을 지정한다. 다른 선택항들은 허용 안된다. 만일 둘다 지정되면, MAIN이 먼저 나타나야 한다.

OPTIONS 속성으로 선언된 수속은 프로그램 동안 활동으로 남아 있고 그런고로 다른 수속에 의해 불러내지지 못한다. D-편성자에서, 오직 한개의 외부 수속만이 OPTIONS(MAIN) 지정을 가져야 한다.

3. "자료 속성"은 수속이 일차 입구점에서 함수 인용으로 불러내질 때 수속에 의해 돌아가는 값의 속성을 지정한다. 불러내지는 수속의 RETURN 문에서 지정된 값은 이것이 불러낸 수속으로 돌아가기 전에 이들 속성에 맞게 변환된다.

만일 "자료 속성"이 지정되지 않으면, 해당 속성이 공극된다. 그런 경우, 입구점의 이름이 수속이 불러내지는 입구 이름) 태만 기수, 척도, 정도를 결정하기에 사용된다.

4. 외부 수속의 입구 이름은 하나의 외부 이름이고 이것은 D-편성자에서 최대 길이 여섯으로 제한된다.

(19) PUT 문

기 능

PUT 문은 다음 방법의 하나로 사용될 수 있는 STREAM 전송 문이다.

1. 이것은 하나 이상의 내부 기억소에 있는 값을 외부 매체에 있는 자료 집합으로 전송케 한다.

이에 덧붙여, 이는 PRINT 화일의 서식을 관리한다.

2. 이것은 하나 이상의 내부 기억소에 있는 값을 내부의 받아들이는 난으로 매입시킨다 (문자-줄 변수로 표현된 난).

#### 일반 형식

STRING 선택 항과 함께 :

PUT STRING (문자-줄-변수)

자료-지정 ;

FILE 선택 항과 함께 :

PUT [ FILE (화일-이름) ]

PAGE [ LINE (요소-식) ]

SKIP [ (요소-식) ]

LINE (요소-식)

자료-지정 ;

PAGE [ LINE (요소-식) ] 자료-지정 ;

SKIP [ (요소-식) ] 자료-지정 ;

LINE (요소-식) 자료-지정 ;

#### 구문 규칙

1. 만일 FILE도 STRING 선택항도 나타나지 않으면, 표준 조직 출력 화일이 대행된다.
2. FILE 선택항은 외부 매체에 있는 자료 집합으로 전송을 지정한다. 이 선택항에 있는 화일 이름은 값을 받아들이는 자료 집합과 연관되어 있는(암시에 의하건 명시 열기에 의하건) 화일의 이름이다. 이 화일은 OUTPUT와 STREAM 속성을 가져야 한다.
3. STRING 선택항은 내부 기억소에서( "자료 지정"에 있는 변수 또는 식으로 표현되는) 문자 줄로( "문자 줄 변수"로 표현되는) 전송을 지정한다. "문자 줄 변수"는 유사 변수가 되지 못한다.
4. "자료 -지정" 선택항은 제 1부, 제 8장, "입력과 출력"에서 논술한 바와 같다.
5. 만일 FILE이나 STRING 선택항이 나타나면, 이것은 첫 번째 선택항이어야 한다. 만일 자료-지정이 나타나면, 이것은 마지막 선택항이어야 한다. PAGE, LINE, SKIP 또는 "자료 지정" 중의 하나가 나타나야 한다.

#### 일반 규칙

1. 만일 FILE 선택항이 지정되고, "화일 이름"이 열리지 않은 화일을 인용하면, 이 화일은 명시로 열린다.
2. 만일 STRING 선택항이 지정되면, PUT 연산은 적당한 변환이 일어난 뒤에 줄의 처음에(이 말은, 가장 왼쪽 문

자 자리) 값을 대입하는 것으로 시작한다. 빈자와 구분자가 보통 삽입된다. 만일 줄이 자료를 수용할 정도로 충분히 길지 않으면, ERROR 조건이 일어난다. "자료 지정"에 있는 변수는 반드시 문자 줄이 아니어도 됨을 유의하라; 내부 대입은 내부 기억소에서 흐름으로 전송되는 것과 동일하다. 단지 차이점은 "문자-줄 변수"가 출력 흐름과 대치되는 것이다.

3. 선택항 PAGE, SKIP, LINE을 PRINT 화일에만 부여될 수 있다. 만일 지정되면, 이것은 "자료 지정"에서 정해진 값의 전송이 일어나기 전에 효력을 발생한다. PAGE와 LINE이 PUT 문에 지정되면, PAGE가 LINE에 앞서 효력을 발생한다.

4. PAGE 선택항은 새로운 지면을 자료 집합 안에서 정하게 한다.

만일 "자료 집합"이 처음 소개되면, 값의 전송은 새 지면에서 시작한다. 새로운 지면은 줄 1을 암시한다.

5. SKIP 선택항은 그 자료 집합을 위해 새로운 줄을 정하게 한다.

"요소 식"은, 만일 나타나면, 0과 3 사이의 고른수 W로 변환된다. 만일 W가 0보다 크면, W-1 번 줄이 만들어지고, 새로운 줄은 W 더하기 지나온 줄 값이다.

만일 W가 영이면, 이 효과는 현재의 줄이 고정되어 남아 있는 이송장치 복귀를 말한다; 문자는 먼저 쓰여진 위에

의  
3  
L

중복 인쇄될 것이다. 만일 "요소 식"이 나타나지 않으면, W는 1로 대행된다. 만일 W보다 작은 줄 수가 현 지면에 남았으면 (여기서 현 지면의 줄 수는 OPEN 문의 PAGESIZE 선택항에 의해서나 태만에 의해서 결정된다), ENDPAGE 조건이 일어난다.

6. LINE 선택항은 자료로 집합에서 새로운 줄(line)을 정하게 한다.

"요소 식"은 고른수 W로 변환된다. 새로운 줄은 W와 같게 정해진다. 그리고 빈 줄이 먼저의 줄 사이에 삽입된다. 그렇지만, 만일 W가 먼저 줄과 같거나 작으면, 또는 만일 W가 현 지면의 줄 수보다 크면, ENDPAGE 조건이 일어난다 (OPEN 문에 있는 PAGESIZE 선택항을 보라) 만일 W가 영과 같거나 작으면, 이것은 1로 대행된다.

(20) READ 문

기 능

READ 문은 INPUT나 UPDATE 파일로부터 내부 기억소에 있는 변수로 기록마디를 전송시키는 RECORD 전송 문이다.

일반 형식

```
READ FILE (파일-이름)
  { SET (지침-변수) }
```

$$\left\{ \text{INTO (변수)} \left[ \left\{ \begin{array}{l} \text{KEY (요소-식)} \\ \text{KEYTO (변수)} \end{array} \right\} \right] \right\}$$

#### 구문 규칙

1. FILE 지정은 먼저 나타나야 한다.
  - o INTO나 SET가 지정되어야 한다.
2. "파일-이름"은 기록마디가 읽혀지는 파일의 이름이다. 이 파일은 RECORD 속성을 가져야 하며 또 INPUT나 UPDATE 속성 중에 하나를 가져야 한다.
3. INTO 선택항의 변수는 기록마디가 읽혀 들어갈 변수이다. 이것은 구조체에 들어있지 않은 첨자 안 붙은 변수이어야 한다. 이것은 명찰변수나 매개변수가 되지 못하며 DEFINED 속성을 갖지 못한다.
4. Key나 KEYTO는 만일 INTO가 지정되는 경우만 지정될 수 있다.

#### 일반 규칙

1. FILE 지정에 있는 파일은 이미 열려 있어야 한다.
2. KEY 선택항은 만일 파일이 DIRECT 속성을 가지고 있으면 나타나야 한다. "요소 식"은 어떤 기록마디가 읽혀질 것인가를 결정하는 열쇠이다. (열쇠의 논술에 관해서는 제1부, 제8장, "입력과 출력"을 보라.)  
KEY 선택항은 SEQUENTIAL과 KEYED 속성을 가진 파일

에 나타날 열 있다. 이 경우, 화일은 지정된 열쇠를 가진 기록마디에 자리잡게 된다. 그런다음, 기록마디는 KEY 선택항 없는 READ 문을 사용하여 그 점에서부터 순차로 읽힐 수 있다. 조직/360 실무에서, 자료 집합은 INDEXED 구성을 가져야 한다.

만일 INDEXED SEQUENTIAL 화일에 관한 READ 문의 KEY 선택항에서 지정된 열쇠가 자료 집합에서 발견되지 않으면, 화일은 KEY 조건이 일어나기에 앞서 자료 집합에 있는 다음 열쇠에 자리잡게 된다.

3. SET 선택항은 UNBUFFERED나 DIRECT 속성 또는 ENVIRONMENT 속성의 INDEXED 선택항을 가진 화일에는 지정되지 못한다. 이 선택항은 기록마디가 완충역으로 읽혀 들어오고 "지침 변수"가 이 완충역 안에 있는 기록마디의 위치를 가르키도록 세워짐을 지정한다. 기록마디에 관한 서술은 지침 변수와 연관된 기초된 변수에 의해 결정된다. 지침 변수의 값은 다음에 오는 READ 문이 실행되거나 또는 화일 닫힐 때까지 유용하다.

4. KEYTO 선택항은 만일 화일이 SEQUENTIAL과 INDEXED 속성을 가질 때만 부여될 수 있다. 이는 읽혀질 기록마디의 열쇠가 문자 줄 대입에 관한 규칙에 따라서 "문자 줄 변수"에 대입됨을 지정한다. D-수준 편성자에서, 대입된 값은 기록된 열쇠(recorded key)이다. 이 "문자-줄 변수"는 기록된 열쇠와 같은 길이를 가져야 한다.

주: KEYTO 선택항은 수치 모양 지정으로 선언된 변수를 지정하지 못한다.

## (21) RETURN 문

기 능

RETURN 문은 RETURN 문이 내적인 수속의 실행을 끝내고, 통제를 불러낸 수속으로 보내준다. 이는 또 불러낸 수속으로 하나의 값을 보내줄 수 있다.

일반 형식

```
RETURN [ ( 요소 - 식 ) ] ;
```

일반 규칙

1. 만일 "요소 식"이 지정되지 않으면, RETURN 문은 함수로서 불러내지지 않은 수속을 끝낼 수 있을 뿐이다.

이런 문이 실행될 때, 통제는 불러낸 수속에 있는 논리상 불러낸 점 다음의 점으로 돌아간다. 만일 RETURN 문이 초기 수속에서 실행되면, 프로그램의 실행은 끝난다.

2. 만일 "요소 식"이 지정되면, 이 문에 의해 끝나는 수속은 함수 수속이어야 한다. 그런 문이 실행될 때, 통제는 불러내는 수속의 불러낸 수속으로 돌려보내진다; 이 점으로 보내지는 값은 "요소 식"의 값이다. 만일 이 값이 끝나는 수속을 위해 지정되는 명시 또는 태만 속성

에, 부합하지 않으면, 이 값은 이것이 보내지기 전에 이들 속성으로 변환된다.

## (22) REVERT 문

### 기능

REVERT 문은 만일 현재의 행위 지정이 REVERT 문이 실행되는 블럭과 동일한 블럭내기 안에서 실행되는 ON 문의 결과인 경우만 지정된 조건에 관한 현재 행위 지정의 효력을 말소한다. 이럴 때, REVERT 문이 포함된 블럭이 블럭내질 때 지정된 조건에 관해서 효력이 있는 행위 지정이 재설정되고 다시 효력을 발휘한다.

### 일반 형식

REVERT 조건 ;

### 구문 규칙

“조건”은 제 8 장, “ON - 조건”에서 서술된 것 중의 어떤 것이라도 된다.

### 일반 규칙

REVERT 문의 실행은 만일 (1) 동일 조건을 지정하고 동일 블럭에 내적인 ON 문이 이 블럭이 기동된 뒤에 실행되고 (2) 다른 똑같은 REVERT 문의 실행이 중간에 들어있지 않을

때만 앞에서 서술한바와 같은 효력을 가진다. 만일 이 두 조건의 하나라도 맞지 않으면, REVERT 문은 빈 문으로 취급된다.

### (23) REWRITE 문

기 능

REWRITE 문은 갱신(更新 update) 화일에만 사용될 수 있다. 이것은 자료 집합에 있는 기존 기록마디(既存)를 바꿔놓는다.

일반 형식

```
REWRITE FILE (파일-이름) [ FROM (변수)
      KEY (요소-식) ] ;
```

구문 규칙

1. FILE 지정은 먼저 나타나야 한다.
  - o KEY는 FROM 없이 지정되지 못한다.
2. "파일 이름"은 다시 써질 기록마디를 포함하고 있는 파일의 이름이다. 이 파일은 UPDATE 속성을 가져야 한다.
3. "변수"는 지정된 파일에 있는 기존의 기록마디를 바꿔놓을 기록마디를 표현한다. 이것은 첨자 안붙은 변수이어야 한다; 이것은 구조체 안에 포함되지 못한다; 이것은 매개변수가 되지 못한다; 그리고 이것은 DEFINED 속성을

가질 수 없다.

#### 일반 규칙

1. 이름이 FILE 지정에 나타난 파일은 이미 열려 있어야 한다.
2. KEY 선택항은 만일 파일이 DIRECT 속성을 가지면 나타날 수 있다; 이것은 다른 경우는 나타나지 못한다. 요소-식은 문자 줄로 변환된다. 이 문자 줄은 어느 기록마디가 다시 써질 것인가를 결정하는 원시 열쇠(原始 열쇠 source key)이다.
3. FROM 선택항은 DIRECT 속성 또는 INDEXED 선택항 또는 SEQUENTIAL 과 UNBUFFERED 속성 둘다 가진 것중의 하나를 가진 UPDATE 파일에는 반드시 지정되어야 한다.
4. FROM 선택항은 CONSECUTIVE 와 BUFFERED 속성을 가진 갱신 파일에서만 생략될 수 있다. 이런 경우, 다시 써질 기록마디는 완충역(buffer)에 있는 기록마디이다. 그래서 이 기록마디는 마지막 읽힌 기록마디이어야 하며 이것은 SET 선택항을 가진 READ 문에 의해 읽혀졌어야 한다. (기록마디는 완충역에 있는 기록마디에 대입이 되었던 아니던 갱신될 것이다.) 만일 이것이 INTO 선택항을 가진 READ 문에 의해 읽혔다면, 이 기록마디는 변경 없이 다시 써질 것이다.

내  
4  
L

## (24) SIGNAL 문

### 기 능

SIGNAL 문은 중단의 출현을 가장한다.

이것은 연관된 조건에 관한 현재 행위 지정을 시험하기에 사용되어도 좋다.

### 일반 형식

SIGNAL 조건

### 구문 규칙

“조건”은 제 8 장, “ON - 조건”에서 서술된 것 중의 하나이다.

### 일반 규칙

1. SIGNAL 문이 실행될 때, 이것은 마치 지정된 조건이 실제로 일어난 것과 같다. 순차적 실행이 중단되고 통제가 지정된 조건을 위한 현재의 on-단위로 옮겨간다. 만일 ON - 단위가 빈 문이면, 통제는 SIGNAL 문 바로 다음의 문으로 정상적으로 돌아온다.
2. 만일 지정된 조건이 불능하면, 중단이 일어나지 않고, SIGNAL 문은 빈 문과 같이 된다.
3. 만일 지정된 조건에 관한 현재의 ON - 단위가 없으면, 그

조건에 관한 표준 조직 행위가 수행된다.

신호된 ENDPAGE 조건에 관해서, 표준 조직 행위는 프로그래밍의 실행에 효력을 갖지 못한다.

## 25) STOP 문

기능

STOP 문은 이것이 실행되는 프로그램을 즉시 끝나게 한다.

일반 형식

STOP

STOP

## (26) WRITE 문

기능

WRITE 문은 내부 기억소에 있는 변수에서 OUTPUT나

UPDATE 파일로 기록마디를 전송하는 RECORD 전송 문이다.

일반 형식

WRITE FILE (파일-이름) FROM (변수)

[ KEYFROM (요소-식) ] ;

구문 규칙

1. FILE 지정은 먼저 나타나야 한다.

2. //파일 이름//은 기록마디가 써질 파일을 지정한다. 이 파일은 OUTPUT 속성 또는 속성 DIRECT 와 UPDATE 중의 하나를 가진 RECORD 파일이어야 한다.
3. FROM 지정에 있는 //변수//는 써질 기록마디를 내용으로 가진다.  
이것은 첨자 안 붙은 변수이어야 한다; 이것은 구조체 변수에 포함되지 못한다; 이것은 매개변수가 되지 못한다; 그리고 이것은 DEFINED 속성을 갖지 못한다.
4. KEYFROM 선택항은 DIRECT 파일에 꼭 지정되어야 한다. 이는 또 INDEXED SEQUENTIAL 파일에 지정될 수 있다. 그러나 다른 어떤 파일에도 지정되어서는 안된다.

#### 일반 규칙

1. 파일은 이미 열려있어야 한다.
2. 만일 KEYFROM 선택항이 지정되면, //요소 식//은 기록마디가 써질 자료 집합에 있는 상대 위치를 지정하는 원시 열쇠이다. (원시 열쇠에 관해서는 제1부, 제8장, //입력과 출력//을 보라. )  
REGIONAL (3)와 INDEXED 자료 집합에서, D - 수준 편성자와 더불어, KEYFROM 은 또한 길이가 KEYLENGTH 선택항에 의해 결정되는 기록된 열쇠를 지정한다.

附 錄



부 록 나: 입력/출력문 서식

파일의 형식		RECORD																						
		STREAM			SEQUENTIAL						DIRECT													
		INPUT	OUTPUT, NOT PRINT	OUTPUT PRINT	CONSECUTIVE BUFFERED	CONSECUTIVE UNBUFFERED		INDEX SEQ	REGIONAL (1)		REGIONAL (3)		INDEXED DIRECT											
유효한 입력/출력문 서식과 사용되는 ON - 조건		INPUT	OUTPUT, NOT PRINT	OUTPUT PRINT	INPUT	OUTPUT	UPDATE	선연결 INPUT	선연결 OUTPUT	선연결 UPDATE	후로, 선연결 INPUT	선연결 INPUT/OUTPUT, INPUT	선연결 INPUT/OUTPUT, OUTPUT	INPUT	OUTPUT	UPDATE	INPUT	OUTPUT	UPDATE	INPUT	OUTPUT	UPDATE	INPUT	UPDATE
OPEN	FILE(파일 이름) FILE(파일 이름) INPUT FILE(파일 이름) OUTPUT FILE(파일 이름) PAGESIZE(n)	o	o	o	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M
CLOSE	FILE(파일 이름)	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
GET *	FILE(파일 이름) EDIT(자료) (서식) [(자료) (서식)] ... FILE(파일 이름) LIST(자료)	o																						
PUT *	FILE(파일 이름) EDIT(자료) (서식) [(자료) (서식)] ... FILE(파일 이름) LIST(자료) FILE(파일 이름) PAGE LINE(n)   FILE(파일 이름)(PAGE LINE(n)  SKIP(n) ) FILE(파일 이름)(PAGE LINE(n)  SKIP(n) )EDIT(자료) (서식) (자료) (서식) FILE(파일 이름)(PAGE LINE(n)  SKIP(n) ) LIST(자료)		o	o																				
READ	FILE(파일 이름) INTO(변수) FILE(파일 이름) SET(저장) FILE(파일 이름) INTO(변수) KEY(식) FILE(파일 이름) INTO(변수) KEYTO(변수)				o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
REWRITE	FILE(파일 이름) FILE(파일 이름) FROM(변수) FILE(파일 이름) FROM(변수) KEY(식)				o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
LOCATE	변수 FILE(파일 이름) SET(저장)				o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
WRITE	FILE(파일 이름) FROM(변수) FILE(파일 이름) FROM(변수) KEY FROM(식)				o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
일어날 수 있는 조건	CONVERSION SIZE ENDFILE(파일 이름) ENDPAGE(파일 이름) KEY(파일 이름) RECORD(파일 이름) TRANSMIT(파일 이름)	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o

사용기호: M=이 문외 사용은 임의이다.  
 O=입력/출력문에서:이 문 서식의 사용은 선택적이다.  
 ON조건에서 :이 조건이 일어날지도 모른다.  
 \* = GET/PUT STRING은 입력/출력문이 아님을 유의하라.

부록 다 : 내조립 함수와 유사-변수의 인수와 결과에  
 관한 요약 (arguments and results of built-in  
 functions and Pseudo-Variables -summary)

충분한 교찰이 내조립 함수와 유사-변수의 인수 나열에 관해 부  
 여되지 않았으므로, 각개 함수 (유사-변수)에 관한 인수와 결과에  
 대한 속성이 여기서 요약된다.

ABS ( X )

X : 요소 식, 배열 이름 ;

산수\*, 비트 줄

결과 : 단일 요소, 배열 ;

규약 산수\*\*

ADD ( X , Y , P [ , q ] )

X, Y : 요소 식, 배열 이름 ;

산수\*, 비트 줄

P : 십진 고른수 상수

q : [ 기호 붙은 ] 십진 고른수 상수

결과 : 단일 요소, 배열 ;

규약 산수\*\*

ADDR ( X )

XX : 요소 변수 이름, 배열 이름, 구조체 이름, 배열이나 구조체  
 의 요소 이름 ;

어떤 자료 형식도 ; 정적, 자동적, 외부, 매개변수, 기초된,  
 정의된

결과 : 지침 값

ALL ( X )

X : 배열 이름 ;  
비트 줄, 문자 줄, 산수\*

결과 : 단일 요소 ; 비트 줄

ANY ( X )

X : 배열 이름 ; 비트 줄, 문자 줄, 산수\*

결과 : 단일 요소 ; 비트 줄

ATAN ( X [ , Y ] )

X, Y : 요소 식, 배열 이름 ; 산수\*, 비트 줄

결과 : 단일 요소, 배열 ; 이진 또는 십진 부동점 수

ATAND ( X [ , Y ] )

X, Y : 요소 식, 배열 이름 ; 산수\*, 비트 줄

결과 : 단일 요소, 배열 ; 이진 또는 십진 부동점 수

ATANH ( X )

X : 요소 식, 배열 이름 ; 산수\*, 비트 줄

결과 : 단일 요소, 배열 ; 이진 또는 십진 부동점 수

BINARY ( X [ , P [ , q ] ] )

X : 요소 식, 배열 이름 ; 산수\*, 비트 줄

P : 십진 고른수 상수

q : [ 기호불은 ] 십진 고른수 상수

결과 : 단일 요소, 배열 ; 이진수

BIT ( 식 [ , 크기 ] )

식 : 요소 식, 배열 이름 : 비트 줄, 문자 줄, 산수\*

크기 : 십진 고른수 상수

결과 : 단일 요소, 배열 : 비트 줄

BOOL ( X, Y, W )

X, Y, W : 요소 식, 배열 이름 : 비트 줄, 문자 줄, 산수\*

결과 : 단일 요소, 배열 : 비트 줄

CEIL ( X )

X : 요소 식, 배열 이름 : 산수\*

결과 : 단일 요소, 배열 : 규약 산수\*\*

CHAR ( 식 [ , 크기 ] )

식 : 요소 식, 배열 이름 : 비트 줄, 문자 줄, 수치 문자

크기 : 십진 고른수 상수

결과 : 단일 요소, 배열 : 문자 줄

COS ( X )

X : 요소 식, 배열 이름 : 산수\*, 비트 줄

결과 : 단일 요소, 배열 : 이진 또는 십진 부동점수

COSD ( X )

X : 요소 식, 배열 이름 : 산수\*, 비트 줄

결과 : 단일 요소, 배열 : 이진 또는 십진 부동점수

COSH ( X )

X : 요소 식, 배열 이름 : 산수\*, 비트 줄

결과 : 단일 요소, 배열 이름 : 이진 또는 십진 부동점수

DATE

결과 : 단일 요소 : 문자 줄

DECIMAL ( X [ , P [ , q ] ] )

X : 요소 식, 배열 이름; 산수\*, 비트 줄

P : 십진 고른수 상수

q : [ 무기호 ] 십진 고른수 상수

결과 : 단일 요소, 배열; 십진 고정점수나 부동점수

DIVIDE ( X , Y , P [ , q ] )

X, Y : 요소 식, 배열 이름; 산수\*, 비트 줄

P : 십진 고른수 상수

q : [ 무기호 ] 십진 고른수 상수

결과 : 단일 요소, 배열; 규약 산수\*\*

ERF ( X )

X : 요소 식, 배열 이름; 산수\*, 비트 줄

결과 : 단일 요소, 배열; 이진 또는 십진 부동점수

ERFC ( X )

X : 요소 식, 배열 이름; 산수\*, 비트 줄

결과 : 단일 요소, 배열; 이진 또는 십진 부동점수

EXP ( X )

X : 요소 식, 배열 이름; 산수\*, 비트 줄

결과 : 단일 요소, 배열; 이진 또는 십진 부동점수

FIXED ( X [ , P [ , q ] ] )

X : 요소 식, 배열 이름; 산수\*, 비트 줄

P : 십진 고른수 상수

q : [ 무기호 ] 십진 고른수 상수

결과 : 단일 요소, 배열; 이진 또는 십진 고정점수

FLOAT ( X [ , p ] )

X : 요소 식 , 배열 이름 : 산수 \* , 비트 줄

P : 십진 고른수 상수

결과 : 단일 요소 , 배열 : 이진 또는 십진 부동점수

FLOOR ( X )

X : 요소 식 , 배열 이름 : 산수 \*

결과 : 단일 요소 , 배열 : 규약 산수 \* \*

HIGH ( i )

i : 무기호 십진 고른수 상수

결과 : 문자 줄

INDEX ( 줄 , 외형 )

줄 : 요소 식 , 배열 이름 : 비트 줄 , 문자 줄 , 이진수 , 수치

문자

외형 : 줄과 같음

결과 : 단일 요소 , 배열 , 이진 고른수

LOG ( X )

X : 요소 식 , 배열 이름 : 산수 \* , 비트 줄

결과 : 단일 요소 , 배열 : 이진 또는 십진 부동점수

LOG~~X~~ ( X )

X : 요소 식 , 배열 : 이름 : 산수 \* , 비트 줄

결과 : 단일 요소 , 배열 : 이진 또는 십진 부동점수

LOG 2 ( X )

X : 요소 식 , 배열 이름 : 산수 \* , 비트 줄

결과 : 단일 요소 , 배열 : 이진 또는 십진 부동점수

LOW ( i )

i : 무기호 십진 고른수 상수

결과 : 문자 줄

MAX ( X<sub>1</sub>, X<sub>2</sub> [, X<sub>3</sub> ····, X<sub>n</sub>] )

X : 요소 식, 배열 이름; 산수\*

결과 : 단일 요소, 배열; 규약 산수\*\*

MIN ( X<sub>1</sub>, X<sub>2</sub> [, X<sub>3</sub> ····, X<sub>n</sub>] )

X : 요소 식, 배열 이름; 산수\*

결과 : 단일 요소, 배열; 규약 산수\*\*

MOD ( X<sub>1</sub>, X<sub>2</sub> )

X : 요소 식, 배열 이름; 산수\*

결과 : 단일 요소, 배열; 규약 산수\*\*

MULTIPLY ( X<sub>1</sub>, X<sub>2</sub>, P [, q] )

X : 요소 식, 배열 이름; 산수\*, 비트 줄

P : 십진 고른수 상수

q : [무기호] 십진 고른수 상수

결과 : 단일 요소, 배열; 규약 산수\*\*

NULL

결과 : 지침 값

PRECISION ( X<sub>1</sub>, P [, q] )

X : 요소 식, 배열 이름; 산수\*, 비트 줄

P : 십진 고른수 상수

q : [무기호] 십진 고른수 상수

결과 : 단일 요소, 배열; 규약 산수\*\*

**PROD ( X )**

X : 배열 이름; 비트 줄, 산수\*

결과: 단일 요소; 이진 또는 십진 부동점수

**REPEAT ( 줄, i )**

줄: 요소 식, 배열 이름; 비트 줄, 문자 줄, 이진수, 수치  
문자

i: 십진 고른수 상수

결과: 단일 요소, 배열; 비트 줄, 문자 줄

**ROUND ( 식, n )**

식: 요소 식, 배열 이름; 산수\*

n: 무기호 십진 고른수 상수

결과: 단일 요소, 배열; 규약 산수\*\*

**SIGN ( X )**

X: 요소 식, 배열 이름; 산수\*

결과: 단일 요소, 배열; 고정점 이진수

**SIN ( X )**

X: 요소 식, 배열 이름; 산수\*, 비트 줄

결과: 단일 요소, 배열; 이진 또는 십진 부동점수

**SIND ( X )**

X: 요소 식, 배열 이름; 산수\*, 비트 줄

결과: 단일 요소, 배열; 이진 또는 십진 부동점수

**SINH ( X )**

X: 요소 식, 배열 이름; 산수\*, 비트 줄

결과: 단일 요소, 배열; 이진 또는 십진 부동점수

SQRT ( X )

X : 요소 식, 배열 이름; 산수\*, 비트 줄

결과: 단일 요소, 배열; 이진 또는 십진 부동점수

STRING ( 줄이름 ) ( 내조립 함수 )

줄이름: 구조체 이름 (정렬 안됨); 문자 줄, 수치 문자

결과: 단일 요소; 문자 줄

STRING ( 줄이름 ) ( 유사 변수 )

줄이름: 구조체 이름 (정렬 안됨); 문자 줄, 수치 문자

SUBSTR ( 줄, i, j ) ( 내조립 함수 )

줄: 요소 식, 배열 이름; 비트 줄, 문자 줄, 이진수, 수치 문자

i : 요소 식, 배열 이름; 산수\*

j : 십진 고른수 상수

결과: 단일 요소, 배열; 비트 줄, 문자 줄

SUBSTR ( 줄, i, j ) ( 유사 변수 )

줄: 요소 변수 이름; 비트 줄, 문자 줄

i : 요소 식; 산수\*

j : 십진 고른수 상수

SUM ( X )

X : 배열 이름; 비트 줄, 산수\*

결과: 단일 요소; 이진 또는 십진 부동점수

TAN ( X )

X : 요소 식, 배열 이름; 산수\*, 비트 줄

결과: 단일 요소, 배열; 이진 또는 십진 부동점수

TAND ( X )

X : 요소 식, 배열 이름; 산수\*, 비트 줄

결과: 단일 요소, 배열; 이진 또는 십진 부동점수

TANH ( X )

X : 요소 식, 배열 이름; 산수\*, 비트 줄

결과: 단일 요소, 배열; 이진 또는 십진 부동점수

TIME

결과: 단일 요소; 문자 줄

TRUNC ( X )

X : 요소 식, 배열 이름; 산수\*

결과: 단일 요소, 배열; 규약 산수\*\*

UNSPEC ( X ) (내조립 함수)

X : 요소 식, 배열 이름; 문자 줄, 산수\*, 지침

결과: 단일 요소, 배열; 비트 줄

UNSPEC ( V ) (유사 변수)

V : 요소 변수 이름; 문자 줄, 산수\*, 지침

\* 산수는 다음을 포함한다: 이진수, 십진수, 고정점수, 부동점수, 수치 문자

\*\* 규약 산수는 다음을 포함한다: 이진수, 십진수, 고정점수, 부동점수

## 부록 라 : F-편성자와의 일치성 (一致性)

DOS/TOS PL/I D-편성자는 IBM 조직/360 운영 조직 (IBM system/360 operating system) 관리하에 운영되는 PL/I F-편성자와 양립한다 (兩立). 보통, D-편성자에 관해 써진 PL/I 원시 프로그램은 F-편성자 아래서 편성되고 실행될 때 동일한 결과를 생산한다. 그렇지만, 편성자는 지금도 발전하고 있으므로, 이 책에서 서술된 D-편성자에 관한 판과 IBM 조직/360 PL/I 참고서 (PL/I reference manual) 판간에 어떤 불일치성이 존재한다. 이들 불일치성 (不一致性)이 여기서 논술될 것이다.

1. PL/I 언어에서 정의된 어떤 오류 조건은 D-편성자에 의해서는 검사되지 않으나 F-편성자에 의해서는 검사된다. 보기로, D-편성자는 반복 DO-모임 안으로 옮겨가는 것에 대해 검사를 하지 않는다 ;

그러므로, 프로그래머는 실행시에 예기치 못한 결과를 얻을 것이다. 그렇지만, F-편성자는 이전 오류 조건을 검사하고 만일 이런 일이 있으면 진단을 내릴 것이다.

2. 만일 F 또는 E 서식 항목으로 관리되는 출력 동안 SIZE 조건이 나타나면, 오류가 되는 값은 D-편성자에 의해 별표 (\*)로 된 난으로 전송된다, 반면에 F-편성자는 잘려진 값을 전송한다. (이것은 SIZE의 가능 여부에 상관 없다.)

3. 만일 F 서식 항목의 관리 하에서 출력으로 전송되는 값의 크기가 일 ( 1 ) 보다 작거나, 또는 만일 E 서식 항목하에 전송되는 값의 가수 ( 假數 ) 가 영이면, F -편성자는 소수점 앞에 한 개의 앞서는 영을 놓고; D -편성자는 안 놓는다.  
보기로, D -편성자에 의해 -.500 으로 전송되는 값은 F -편성자에 의해 -0.500 으로 전송될 것이다.
4. F -편성자는 D -편성자에는 유효하나 F -편성자에는 무효한 어떤 ENVIRONMENT 속성의 선택항에 대해서도 경고 진단을 발하고, 이를 무시한다. 이런 선택항은 F -편성자에서 DD 문에서 지정되어야 한다.
5. 열쇠말 SYSIN 과 SYSPRINT 는 D -편성자에서 뜻이 없다.  
그렇지만, 이것들은 F -편성자에서는 뜻을 가지며, 고로 사용에 주의를 해야 한다. 이 책의 제 1 부, 제 8 장의 " 표준 화일 " 은 이 제목에 대한 완전한 논술이 있다.
6. F -편성자 아래서 D -수준 프로그램이 시행될 때, D -편성자에서 사용되지 않는 열쇠말은, 보기로 REAL, COMPLEX, PT, 문제가 될지도 모른다. 보기로, 단일 REAL 이 D -수준 프로그램에서 하나의 외부 수속 이름이면, 이 프로그램이 F -편성자 아래서 시행되기 전에 이 수속의 이름은 바뀌어야 한다.  
안그러면, REAL 에 관한 함수 인용은 이 이름의 내조립 함수에 관한 인용으로 처리될 것이다.
7. D -편성자 하에서, DO -문에 있는 TO 와 BY 식의 값내기

순서는 먼저 "식 2"의 값이 내지고 다음에 "식 3"의 값 내기로 처리된다. 반면 F-편성자는 이것들이 나타난 순서로 식의 값을 낸다. 만일 이들 식들의 값내기 동안 함수가 불러내지고 이 함수가 다른 식에 있는 변수를 변화시킬 경우에는 상이한 결과를 가져올 수 있다.

8. F-편성자 아래서, 수치 문자 자료 항목의 문자 값은 모든 숫자 자리(고른수와 소수)가 지워질 때는 제일 오른쪽 숫자 자리에 부동 문자 놓여질 것이다. 물론 모든 숫자자리가 그 부동 문자로 되어 있는 경우이다. D-편성자 아래서, 이 부동 문자는 나타나지 않는다; 문자 값은 전체가 빈자로 된다.
9. F-편성자에서, DISPLAY 문의 REPLY 선택항에 있는 응답의 길이는 126 문자를 넘지 못한다; D-편성자에서, 이 길이는 제한되지 않는다.
10. D-편성자는 IF-THEN절을 한개 문으로서 취급하는 반면에 F-편성자는 이를 둘로 취급한다.
11. D-수준 편성자 아래서, 만일 READ가 사용자에게 의해 일어나지 않으면 REWRITE에 앞서 READ가 일어난다. 반면에 F-편성자 아래서 오류 선언문이 발해진다.
12. F-수준 편성자에서, 블럭된 기록마디를 가진 INDEXED DIRECT UPDATE 화일에서, 개개 기록마디는 다시 써져야 한다. 이 규칙은 D-수준 편성자에는 적용되지 않는다.

13. D-편성자 아래서, 만일 READ KEY가 SEQUENTIAL INDEXED 화일에 제거되고, 그 열쇠를 가진 기록마디가 없을 때, ON KEY 조건이 일어나고, 화일은 다음의 높은 기록마디에 자리잡는다. F-편성자에서, ON KEY 조건이 또한 일어나 OS 자료 관리는 첫째 기록마디에 화일을 다시 자리잡게 한다.

14. D-편성자는 BIT 줄을 위한 태만을 처리하지 않는다, 이 태만은 UNALIGNED이다, 그러나 ALIGNED로 고정시켰다.

F-편성자는 UNALIGNED를 처리한다.

15. D-편성자 아래서, WRITE KEYFROM 문에서, 열쇠는 KEYFROM에 의해 지정된 변수에서 기록마디로 자동으로 옮겨간다. 그렇지만, F-편성자에서, 만일 열쇠 변수가 기록마디에 있는 지정된 자리의 내용과 같지 않으면, 오류 조건이 일어난다. 그러므로 기록마디로 열쇠 변수를 옮기는 것은 F-수준 프로그래머의 책임이다.

16. 가변 길이 기록마디는 D-편성자에서 ENVIRONMENT 속성에서 V로(최대 블럭 크기) 지정된다. 블럭으로 하기는 실제의 기록마디 크기에 따라 이루어진다. F-편성자에서, V(최대 블럭 크기)는 블럭 안된 기록마디를 가리킨다.

17. 만일 INDEX 내조립 함수의 인수가 둘다 이진 규약 산수이면, 또는 하나가 이진 규약 산수이고 다른 것이 비트 줄이면, D-편성자는 이 함수를 불러내기에 앞서 이들을 정확히 비트 줄로 변환한다. 그렇지만, F-편성자는 이 인수들이

문자 줄로 변환된 결과를 가지고 직접 함수를 불러낸다,  
이것은 상이한 결과가 될 것이다.

18. 만일 PRECISION 내조립 함수의 첫째 인수가 FIXED 이면, 그러면 세번째 인수는 반드시 지정되어야 한다. 그렇지만, PL/I D-편성자는 만일 이 경우에 세번째 인수가 나타나지 않으면 편성시에 오류 전언문이 주어지고, 값은 영으로 대행된다. F-편성자는 끝마치는 오류 전언문을 발한다.

19. 구조체 선언에서, 1이 아닌 수준 번호 앞에 있는 DECLARE 열쇠말은 D-편성자에 의해서는 오류로서 인식되지 않는다.

20. 프로그램을 최적으로 하기 위해, D-편성자는 대입 동안에 일어날지도 모르는 가능한 중단을 고려함이 없이 상수의 대입을 제거할 수도 있다.

21. 상이한 차원의 배열 대입은 D-편성자에 의해서는 문제되지 않는다. F-편성자는 편성을 끝낸다.

22. 비트-줄 비교의 결과는 만일 줄 길이가 한자의 고른수 배가 아니면 D와 F-편성자에 따라 다를 수도 있다.

23. 255보다 큰 길이 속성의 인수를 가진 STRING 내조립 함수의 사용에서, D-편성자는 MOD 256의 길이를 가진 줄을 보내준다. 동일한 경우 F-편성자는 오른쪽을 자른다.

24. D-편성자는 ENVIRONMENT의 선택항 사이에 구분하는 빈자가 생략되었을 때 진단을 주지 않는다.  
F-편성자는 S-수준 진단을 내린다.

25. 함수의 PROCEDURE 문은 편성자에 따라 상이한 형식을 가질 수도 있다 (불일치성).

26. 만일 E-서식 항목으로 읽힌 입력 난이 빈자로만 되어있으면, D-편성자는 양의 영으로 대항한다. F-편성자는 CO-VERSION 조건을 일으킨다.

부록 마 : 용어의 정의 (用語의 定義)

이 장은 이 책에서 사용된 대부분의 용어에 관한 정의를 서술한다.

접근 (接近 access) : 자료를 인용하거나 자료를 끄집어내는 여러가지 행위.

행위 지정 (行為 指定 action specification): ON 문에서, on - 단위나 한개의 열쇠 말 SYSTEM, 이것은 호명된 조건이 일어나기 때문에 중단이 초래될 때 취해질 행위를 지정한다.

기동 (起動 activation) : 블럭 실행의 설립. 수속 블럭은 이것이 입구점에서 블럭내질 때 기동된다; 시작 블럭은 이것이 정상 순서 흐름에서 만날 때 기동된다.

활동적 (活動的 active) : 블럭이 기동되고 나서 마치기 전까지의 상태.

부가 속성 (附加 屬性 additive attribute) : 태만이 없고, 만일 요구된다면, 언제나 명시로 서술되어야 하는 화일 속성.

주소 (住所 address) : 자료 항목이 기억될 수 있는 특정의 기억소 위치.

할당된 변수 (割当된 變數 allocated variable) : 기억소가 연관된 변수.

할당 (割当 allocation) : 변수와 기억소의 연관.

영문자 (또는 영자 英文字 alphabetic character) : A 부터 Z와 확장 영문자 #, \$, @의 문자.

영수문자 (英數文字 alphanumeric character) : 영문자 또는 숫자.

택일 속성 (扨一 屬性 alternative attribute) : 선택할 수 있는 둘 이상의 모임에서 고를 수 있는 화일 속성. 만약 아무것도 지정되지 않으면, 태만이 대형 된다.

인수 (引數 argument) : 수속 인용의 일부가 되어 불러내지는 수속에 보내지는 식, 화일 이름, 문 명찰 상수 또는 변수, 또는 입구 이름. 이것은 내조립 함수나 기초된 변수가 되지 못한다.

산수 변환 (算數 變換 arithmetic conversion) : 하나의 산수 표현법에서 다른 산수 표현법으로 값이 전환되는 것.

산수 자료 (算數 資料 arithmetic data) : 기수, 척도, 정도의 특성을 가진 자료. 이것은 규약 산수 자료와 수치 문자 자료를 포함한다.

산수 연산자 (算數 演算子 arithmetic operation) : 전치 연산자 +, -, 또는 삽입 연산자 +, -, \*, /, \*\*.

배열 (配列 array) : 모두 동일 속성을 가지며, 이름이 불고 나란히 놓여있는 자료 요소의 집합체. 배열은 차원을 가지며, 첨자에 의해 구별되는 요소들을 가진다.

대입 (代入 assignment) : 변수에 값을 주기.

속성 (屬性 attribute) : 이름이 나타날 수 있는 자료 항목 또는 화일의 특성을 설명키 위해 이름 또는 식과 연관된 설명적 성질.

자동적 기억소 (自動的 記憶所 automatic storage) : 블럭크의 기동시에 할당되며 그 블럭크의 마침에 따라 해제되는 기억소.

기수 (基数 base) : 산수 값이 그 방법으로 표현되는 수 체계. PL/I에서, 기수는 이진수와 십진수이다.

기초된 변수 (基礎된 變數 based variable) : BASED (지침-

내  
1  
P

변수) 속성 지정을 가지고 선언되는 변수. 지침 변수는 이 서술을 기억소의 할당과 연관시킨다.

시작 블럭 (始作 블럭 begin block) : 이름의 범위를 구분하고 정상 순서 문 흐름에 의해 기동되는, BEGIN 문으로 시작되며 END 문으로 끝나는 문의 집합체. 이것이 블럭에서 선언된 자동적 기억소의 할당과 해방을 관리한다.

이진수 (二進數 binary) : 수 2에 기초하는 수 체계.

비트 (bit) : 이진 숫자, 0 또는 1.

비트 줄 (bit string) : 하나 이상의 비트의 줄.

비트-줄 연산자 (bit-string operator) : 연산자 NOT (아니다 (not)), AND (그리고 (and)), OR (또는 (or)) 중의 어떤것.

블럭 (block) : 시작 블럭 또는 수속 블럭.

한계 (限界 bound) : 배열의 차원에 있는 상한. 하한은 언제나 1로 된다.

완충역 (緩衝域 buffer) : 입력 동안에 기록마디가 읽혀 들어가는 출력 동안에 기록마디가 그곳으로부터 나와 써지는, 입력/출력 연산에 사용되는 중간 임시 영역.

내조립 함수 (內組立 函數 built-in function) : PL/I 이 정의한 함수 중의 하나.

부르기 (call) : CALL 문을 써서 버금 과정을 불러내는 것.

문자-줄 (文字-줄 character-string) : 자료 문자 집합에서 뽑은 하나 이상의 문자를 내용으로 하는 줄.

규약 산수 자료 (또는 부호화 산수 자료) (規約 (또는 符号化) 算數 資料 coded arithmetic data) : 특성이 기수, 척도,

정도 속성으로 주어지는 산수 자료.

대조 순서 (또는 대조 순위 对照 順序 (또는 順位) collating sequence) : 분류 (分類 sorting) 와 합병 (合併 merging) 이 그것에 기초하는 비트 표현법의 문자의 상대적 차례.

주 (또는 주석) (註 (注) 또는 註釈 comment) : /\*가 앞 이 되고 \*/로 끝나치며 한개의 빈자처럼 취급되는, 기록을 위해 사용되는 문자의 줄.

비교 연산자 (比較 演算子 comparison operation) : 연산자 <, <=, =, >=, >, >.

편성시 (編成時 compile time) : 원시 프로그램이 목적 프 로그램 (object module) 으로 번역되는 동안의 시간.

편성자 (編成者 compiler) : 원시 프로그램을 목적 프 로그램 으로 바꾸는 번역자.

복문 (또는 복합 문 複文 (複合 文) compound statement) : 또 다른 문을 내포하고 있는 문. IF와 ON이 복문의 전부이다.

잇기 (concatenation) : 길이가 두개 줄의 길이의 합계와 같이 되는 한개 줄을 형성하도록, 지시된 순서에 있는 두개 줄을 이어주는 연산. 이것은 연산자 ||로 지정된다.

조건 이름 (條件 이름 condition name) : 프로그램의 실행 동안에 일어날 수도 있는 예외 조건을 표현하는 언어 열쇠말 (language keyword).

조건 전치어 (條件 前置語 condition prefix) : 콜론 (:) 에 의해 문에 붙어있는 하나 이상의 조건 이름의 괄호에 싸인 나열 (list). 이것은 만일 지정된 조건의 하나가 이 전치어의 범 위 안에서 나타날 때 프로그램이 중단될 것인가 아닌가를 결정한

의  
2  
P

다. 나열에 있는 조건 이름들은 쉼표 (,) 로 떨어져야 한다.

상수 (常数 constant) : 이름을 갖지 않는 산수 또는 줄 자료 항목; 문 명찰.

~에 포함된 (~에 包含된 contained in) : 그 블럭의 입구 이름을 빼 블럭의 원문의 전부. (BEGIN 문의 명찰은 그 문으로 정의 된 시작 블럭에 포함되지 않는다.)

문맥상 선언 (文脈上 宣言 contextual declaration) : 포식어가 나타난 문맥에 따라서 포식어에 속성을 연관시킴. 포식어와 문맥상으로 연관되는 유일한 속성은 BUILT IN 속성이다.

변환 (變換 conversion) : 한개 표현법에서 다른 것으로 값을 전환함.

자료 (資料 data) : 정보 또는 값의 표현.

자료 문자 집합 (資料 文字 集合 data character set) : 바이트 배치형상이 사용중의 계산기에 인지되는 모든 문자.

자료 항목 (資料 項目 data item) : 자료의 한개 단위; 이것은 "요소"와 동의 (同義) 이다.

자료 나열 (資料 羅列 data list) : 입력 동안 자료 항목이 대입되며, 출력 동안 이것에서 자료 항목이 써지는 기억소 영역을 표시하는, STREAM 입력/출력 지정에서 사용되는 식의 나열. (입력에서, 이 나열은 단지 변수만으로 이루어진다.)

자료 집합 (資料 集合 data set) : 프로그램에 외부적 자료의 집합체.

자료 지정 (資料 指定 data specification) : 전송의 방식을 (LIST 또는 EDIT) 지정하며 자료 나열을 가지고 있으며, 편집-지기 전송에서는 서식 나열을 가지고 있는 흐름-지향 자료 전송 문의 일부분.

- 십진수 (十進數 decimal) : 값 10에 기초하는 수 체계.
- 선언 (宣言 declaration) : 포식어에 명시로, 문맥상으로, 또는 암시로 속성을 연관시킴.
- 태만 (怠慢 default) : 포식어가 둘 이상의 택일적 속성의 하나를 가지게끔 선언되지 않았을 때 대행되어 택일되는.
- 구분자 (区分子 delimiter) : 포식어와 상수, 또는 문과 다른 문을 분리하기에 사용되는 유효한 특수 문자 또는 특수 문자의 결합체.
- 차원수 (次元數 dimensionality) : 배열과 연관되어 있는 한계 지정의 수. 이것은 셋 (3)을 넘지 못한다.
- 불능한 (또는 불능하게 되어 있는 不能한 disabled) : 특정 조건의 출현이 프로그램 종단을 가져오지 않는 상태.
- Do - 모임 (Do-group) : DO 문이 머리에 오고 이에 대응하는 END 문으로 닫히는 일련의 문.
- 가인수 (假引數 dummy argument) : 프로그래머가 배당한 이름이 없는 인수들 위해 편성자가 배당한 변수.
- 편집 - 지시 전송 (編輯 - 指示 転送 edit-directed transmission) : STREAM 전송; 자료 나열과 서식 나열이 모두 지정된다.
- 요소 (要素 element) : 구조체나 배열과 같은 자료 항목의 집합체에 반대 개념인 단일 자료 항목. (때로는 "스칼라 항목 (scalar item)"이라 함.)
- 요소 변수 (要素 變數 element Variable) : 어느 때 어떤 점에서도 단일 값만을 표현할 수 있는 변수.
- 가능한 (또는 가능하게 되어있는 可能한 enabled) : 특정

내  
2  
P

조건의 출현이 프로그램 중단을 가져오는 상태.

입구 이름 (入口 이름 entry point) : PROCEDURE 나 ENTRY 문의 명칭.

입구점 (입구 점 入口点 entry point) : 입구 이름을 인용함으로써 불러내질 수 있는 수속의 한 점. (일차 입구점과 이차 입구 점을 보라.)

종막 (終幕 epilogue) : 블럭이 끝나갈 때에 출현하는 처리과정들.

예외 조건 (例外 条件 exceptional condition) : 넘치기 오류와 같은 예상하지 못한 국면의 출현, 이것은 프로그램 중단을 초래할 수 있다. 또는 화일의 끝과 같이 예상되는 국면의 출현, 이것은 예상 못하는 시각에 출현한다.

명시 선언 (明示 宣言 explicit declaration) : DECLARE 에서, 명칭로서 출현한, 또는 매개변수 나열에서 출현한 표식어에 속성을 배당함.

(부동점수 상수의) 지수 (浮動点数 常數의) 指数 exponent (of floating-point) : 부동점수의 밑 (底 base) 이 제곱해짐을 지정하는 십진 고른수 상수

식 (式 expression) : 값의 표현; 보기로, 홀로 또는 연산자와 결합해서 나타나는 변수와 상수, 그리고 함수 인용. 용어 "식" 은 요소 식, 배열 식, 구조체 식을 포괄한다.

외부 선언 (또는 외부적 또는 외적 선언 外部 宣言 external declaration) : 표식어에 대해 EXTERNAL 속성의 명시 선언. 이러한 표식어는 이런 선언이 존재하는 다른 모든 블럭에 알려진다.

외부 이름 (또는 외부적 또는 외적 이름 外部 이름 external name) : EXTERNAL 속성을 가진 이름.

외부 수속 (외부적 또는 외적 外部 手續 external procedure) : 다른 어떤 수속에도 포함되지 않는 수속.

난 (欄 field) (자료 흐름에 있는) : 몇개 문자로 되는 넓이가 단일 자료 또는 자리 띄기 서식 항목에 의해 정해진 자료 흐름의 일부분.

(모양 지정의 (模樣 指定의 of picture specification)

난 (欄 field) : 문자-줄 모양 지정 또는 수치 문자 모양 지정의 일부분 (또는 전부). 만일 한개 지정에서 둘 이상의 난이 나타나면, 이것은 부동점수 자료에서 K 또는 P 지수 문자에 의해 양분되며 또는 영구된 자료에서 M 난-분리자에 의해 분할된다. 고정점수 지정에서는 다만 한개 난만이 나타날 수 있다.

화일 (file) : 프로그램 안에서, 자료 집합의 상징적 표현.

화일 이름 (file name) : 프로그램 안에서 자료 집합 인용에 사용되는 상징적 이름.

서식 항목 (書式 項目 format item) : 편집-지시 전송에서 흐름에 있는 자료 항목의 표현방식을 기술하기 위하여 또는 인쇄되는 지면의 양식을 관리하기 위해 사용되는 지정.

서식 나열 (書式 羅列 format list) : 편집-지시 자료 지정에서 요구되는 서식 항목의 나열.

함수 (函數 function) : 함수 인용에서 이것의 입구 이름의 하나가 나타나므로써 불러내지는 수속.

함수 인용 (函數 引用 function reference) : 통상<sup>an</sup> 함수 나열과 연합해서, 식에서 입구 이름이 나타나는 것.

모임 (group) : Do 모임.

표식어 (表識語 identifier) : 주석이나 상수에 포함되지 않으며, 구분자가 앞 뒤에 있으며 첫째 문자가 영문자인 영수문자와 절단 문자의 줄.

의 암시 선언 (暗示 宣言 implicit declaration) : 명시 또는  
3 문맥상으로 선언됨이 없이 변수로서 사용되는 표식에 속성을  
연관시킴 ; 다만 속성이 표식어의 첫째 문자에 따라서 제공된다.

P 비활동 블럭 (또는 비활동적, 불활동, 불활동적 블럭 inactive block) : 기동되지 않았거나 끝나져진 수속 또는 시작 블럭.

삽입 연산자 (또는 이항 (二項) 연산자 挿入 演算子 infix operator) : 두개 연산항 사이의 연산을 정의한 연산자.

초기 수속 (初期 手續 initial procedure) : PROCEDURE 문이 OPTIONS (MAIN) 속성을 가진 외부 수속. 모든 PL/I 프로그램은 한개의 초기 수속을 가져야 한다. 이것은 프로그램에서 첫째 단계로서 자동으로 불러내진다.

입력/출력 (入力/出力 input/output) : 외부 매체와 내부 기억소 간의 자료 이전.

내부 블럭 (또는 내부적, 내적 内部 블럭 internal block) : 다른 블럭에 포함되어 있는 블럭.

내부 이름 (또는 내부적, 또는 내적 内部 이름 internal name) : INTERNAL 속성을 가진 표식어.

내부 수속 (또는 내부적, 내적 内部 手續 internal procedure) : 다른 블럭에 포함되어 있는 수속.

~에 내부적 (또는 내적 内部的 internal to) : 다른 블럭

크에 포함되어 있는 원문 (原文) 을 빼고서 한 블럭에 포함되어 있는 원문의 전부. 그래서 내부 블럭의 원문은 (이것의 입구 이름을 빼고서) 포함한 블럭에 내적이 아니다.

주 : 한 블럭의 입구 이름은 그 블럭에 포함되지 않는다.

중단 (中斷 interrupt) : 가능한 조건의 출현 때문에 정상 프로그램 활동이 정지되는 것.

블러 내다 (invoke) : 입구점의 하나에서 수속을 기동시키는 것.

블러 내진 수속 (블러 내진 手續 invoked procedure) : 입구점의 하나에서 기동된 수속.

블러 낸 블럭 (invoking block) : 다른 블럭을 기동시키는 문을 포함하는 블럭.

중복 (또는 반복) 인수 (重復 因数 iteration factor) : 다음을 지정하는 상수

- (1) 주어진 서식 항목이나 서식 항목의 나열이 서식 나열에서 연속해서 사용될 횟수
- (2) 주어진 상수로 초기값이 되는 배열의 연속한 요소 수

열쇠 (key) : 원시 열쇠와 기록된 열쇠를 보라.

열쇠말 (keyword) : 언어의 일부가 되는 표식어이며, 특정한 문맥에서 사용되면 이것은 편성자에 대해 특별한 뜻을 가진다.

알려지는 (알려진) (known) : 표식어의 범위를 가르켜주기에 사용되는 용어. 보기로, 표식어는 이것이 선언된 블럭에 언제나 알려진다.

명찰 상수 (名札 常数 label constant) : 문 명찰과 동의어.

명찰 전치어 (名札 前置語 label prefix) : 콜론 (:) 으로 문에 붙은 괄호에 싸이지 않은 표식어.

앞서는 영 (leading zero) : 산수의 값에서 무의 (無意한) 영; 수의 첫번째 유의 숫자 (1부터 9)의 왼쪽에 있는 모든 영.

수준 번호 (水準 番号 level number) : 구조체에서 이름의 계급을 지정하는 무기호 (또는 무부호) 십진 고른수 상수. 이것은 이름의 왼쪽에 나타나며 빈자로 구분된다.

나열-지시 전송 (羅列-指示 転送 list-directed transmission) : STREAM 전송; 흐름에 있는 자료는 빈자나 쉼표로 떨어져서 상수로서 나타난다.

대 구조체 (大構造体 major structure) : 이름이 수준 번호 1을 가지고 선언된 구조체.

소 구조체 (小構造体 minor structure) : 이름이 1보다 큰 수준 번호를 가지고 선언된 구조체.

중복 선언 (重複 宣言 multiple declaration) : 상이한 수식 없이 동일 블럭에 내부적 동일 표식어의 둘 이상의 선언, 또는 단일 프로그램 안에서 상이한 자료 속성을 가진 동일 표식어에 관한 둘 이상의 EXTERNAL 선언.

이름 (name) : 선언된 표식어.

품기 (nesting) :

- (1) 다른 블럭 안에서 블럭이 출현하는 것.
- (2) 다른 모임 안에서 모임이 출현하는 것.

(3) THEN 절이나 ELSE 절에 IF 문이 출현하는 것.

(4) 다른 함수 인용의 인수로서 함수 인용이 출현하는 것.

수치 문자 자료 (數值 文字 資料 numeric character data) :  
문자 꼴로 기억되는 모양에 의해 서술된 산수 자료. 이것은 산수 값과 문자-줄 값을 다 함께 갖는다. 이 모양은 X 모양 지정 문자를 내용으로 가져서는 안된다.

on-단위 (on-單位 on-unit) : 포함된 ON 문에서 호명된 ON-조건의 출현에 따라 실행될 행위.

연산자 (演算子 operator) : 수행될 연산을 지정하는 부호. 산수 연산자, 비트-줄 연산자, 비교 연산자, 그리고 잇기를 보라.

선택항 (選擇項 option) : 문의 실행을 조절하기 위해 프로그래머에 의해 사용될 수 있는 문에 있는 지정.

짜인 십진수 (짜인 十進數 packed decimal) : 고정점 십진수 자료의 조직/360 내부 표현법.

매개변수 (媒介變數 parameter) : 그 수속으로 보내진 인수를 나타내기에 사용되는 1 불러내진 수속에 있는 이름.

모양 (模樣 picture) : 내용과 수치 문자의 속성과 문자-줄 자료를 서술하는 문자 단위 지정.

불러낸 점 (불러낸 點 point of invocation) : 불러내지는 수속에 대한 수속 인용이 나타나는 불러낸 (불러내는) 블록에 있는 점.

지침 변수 (指針 變數 pointer variable) : 기초된 변수를 인용할 때에 사용될 수 있게 기억소를 가리켜주는 변수.

정도 (精度 precision) : 허용된 총 숫자의 수와, 고정점 변수에서, 십진 (또는 이진) 소수점의 가정된 위치로써 표시되는

산수 변수에 관한 값의 크기.

전치어 (前置語 prefix) : 문의 앞에서 콜론 (:) 으로 이어진 명찰이나 괄호에 싸인 조건 이름의 나열.

전치 연산자 (前置 演算子 prefix operator) : 한개 연산항과 관계되며 그 연산항의 앞에 오는 연산자. 전치 연산자는  $\uparrow, +, -$ .

일차 입구점 (一次 入口点 primary entry point) : PROCEDURE 문에서 명명된 입구점.

문제 자료 (問題 資料 problem data) : PL/I 프로그램에 의해 처리되는 줄 또는 산수 자료.

수속 (手續 procedure) : PROCEDURE 문이 머리가 되고 END 문으로 마감하는 문의 집합체, 이것은 프로그램의 구역을 정의하며 이름의 범위를 구분하며 이것의 이름을 인용하는 것에 의해 기동된다. 이것은 이 블럭 안에서 선언된 자동적 기억소의 할당과 해방을 관리한다.

수속 인용 (手續 引用 procedure reference) : 함수 또는 버금과정 인용.

프로그램 (program) : 하나 이상의 외부 수속의 집합, 이 중의 하나는 PROCEDURE 문에서 OPTIONS (MAIN) 지정을 가져야 한다.

프로그램 통제 자료 (프로그램 統制 資料 program control data) : 프로그램의 실행에 영향을 주기 위해 PL/I 프로그램에서 사용되는 자료. 명찰 자료와 지침 자료가 프로그램 통제 자료의 형식이다.

서막 (序幕 prologue) : 블럭의 기동시에 출현하는 처리 과정.

유사-변수 (類似-變數 pseudo-variable) : 받아들이는 난으로 사용될 수 있는 내조립 함수 이름의 하나. STRING, SUBSTR, UNSPEC 만이 사용될 수 있다.

수식된 이름 (修飾된 이름 qualified name) : 마침표 (·) 로 이어진 연속한 구조체 구성 이름, 구조체의 성분을 고유하게 나타내기 위한 것이다.

받아들이는 난 (받아들이는欄 receiving field) : 값이 대입될 수 있는 모든 난.

기록마디 (記錄마디 record) : RECORD 입력 또는 출력 연산의 전송 단위.

기록된 열쇠 (記錄된 열쇠 recorded key) : 바로 뒤따르는 자료 기록마디를 증명하기 위해서 직접-접근 권 (卷 volume) 에 기록된 문자 줄.

반복 인수 (反復 因素 repetition factor) : 문자 줄 상수의 간략한 표현으로서 줄 앞에 오는 괄호에 싸인 무기호 십진 고른수 상수. 이 반복 인수는 실제의 상수를 만들기 위해 여러 번의 출현을 지정 한다. 모양 지정에서, 반복 인수는 한개 모양 문자의 반복을 지정 한다.

반복 지정 (反復 指定 repetitive specification) : 자료 항목 나열을 전송하기 위해 계획된 횟수를 지정하는 자료 나열의 한 요소, 통상 배열과 연합해서 사용된다.

원시 열쇠 (原始 열쇠 source key) : 직접 접근 자료 집합에 있는 특정 기록마디를 증명하기 위해 RECORD 전송 문에서 인용되는 문자 줄 또는 수치 문자 자료 항목. 원시 열쇠는 기록마디를 명확하게 증명하기 위해 기록된 열쇠와 비교될, 또는 기

록된 열쇠로서 써질, 줄이다.

원시 프로그램 (原始 프로그램 source program) : 편성자에 입력으로 제공되는 프로그램.

표준 파일 (標準 파일 standard file) : GET 나 PUT 문에서 FILE 이나 STRING 선택항이 결여되었을 때 편성자에 의해 대행되는 파일.

문 (文 statement) : 프로그램의 일부분을 구분하기 위해, 프로그램에서 사용되는 자료를 기술하기 위해, 또는 취해질 행위를 지정하기에 사용되는 PL/I 프로그램의 기본 요소.

문 명찰 (文 名札 statement label) : PROCEDURE 또는 ENTRY 문을 빼 다른 어떤 문의 앞에 붙은 실증하는 이름.

문 명찰 변수 (文 名札 変数 statement label variable) : LABEL 속성을 가지고 선언되고 그래서 문 명찰을 이것의 값으로 대입할 수 있는 변수.

정적 기억소 (靜的 記憶所 static storage) : 프로그램의 실행이 시작하기 앞서 할당되며 프로그램 동안 할당된 채로 남아 있는 기억소.

흐름 (stream) : 문자 끝로 된 자료 항목의 연속한 줄로서 표현되는 외부 매체로 또는 매체에서 전송되는 자료.

줄 (string) : 단일 자료 항목으로 간주되는 문자 또는 비트의 연결된 것.

구조체 (構造体 structure) : 여러가지 속성을 가질 수 있는 자료 항목의 집합체를 인용하는 이름들의 계급성을 띤 집합.

소난 (小欄 subfield) : 고른수 아닌 고정점수 자료 항목을 기술하는 모양 지정 문자 난의 고른수 기술 부분 또는 소수 기

술 부분. 소난은 모양 문자 V로 분할된다.

버금과정 (버금 過程 subroutine) : CALL 문에 의해 불러 내지는 수속. 버금과정은 불러낸 블럭으로 값을 보내지 못한다. 그러나 이는 불러낸 수속 안에 알려진 변수의 값을 바꿀 수 있다.

첨자 (添字 subscript) : 배열의 차원 안에서의 위치를 지정하는 요소 식.

마침 (termination) : 블럭 실행의 중지, 그리고 RETURN 또는 END 문의 방법으로 기동시킨 블럭으로 통제의 복귀, 또는 Go To 문의 방법으로 기동시킨 블럭 또는 다른 어떤 활동적 블럭으로 통제의 이행 (移行). 초기 수속에 있는 RETURN 또는 END 문을 통해서 운영 조직으로 통제의 복귀나 어떤 블럭에 있는 STOP 을 통한 운영 조직으로 통제의 복귀도 프로그램의 마침 결과가 된다. 종막을 보라.

변수 (變數 variable) : 자료를 표현하는 이름. 이것의 속성은 변하지 않고 남아있다. 그러나 이는 상이한 시기에 상이한 값을 표현할 수 있다. 변수는 세가지 범주로 떨어진다: 요소, 배열, 구조체 변수. 변수는 첨자가 붙고 / 또는 수식될 수 있다.

부록바 :

PL / I 프로그램의 예

1. HOUS8.. PROC OPTIONS(MAIN),

2. DCL

PROT FILE OUTPUT PRINT STREAM

ENV (CONSECUTIVE MEDIUM(SYSLST,1403) F(133)),

TAPA FILE INPUT RECORD SEQUENTIAL

ENV (CONSECUTIVE MEDIUM ( SY50.05,2400) F(2640,88)),

3. DCL

1 TA,

2 RAI

CHAR(2),

/\* REST

2

\*/

2 DOA

CHAR(2),

/\* DO

4

\*/

2 RA2

CHAR(57),

/\* REST

61

\*/

2 HSA

PI C'9',

/\* HOUSE

62

\*/

2 RA3

CHAR(1),

/\* YEAR

63

\*/

2 YA

CHAR(2),

/\* YEAR

65

\*/

2 RA4	CHAR(2),	/* REST	67	*/
2 WA	PIC'9',	/* WALL	68	*/
2 ROA	PIC'9',	/* ROOF	69	*/
2 RA5	CHAR(3),	/* REST	72	*/
2 HAS	FIXED DEC(9,4),	/* NUMBER	77	*/
2 RA6	CHAR(10),	/* REST	87	*/
2 AA	PIC'9',	/* AREA	88	*/
DOB	CHAR(2),			
HSB	PIC'9',			
RB3	CHAR(1),			
YB	CHAR(2),			
YC	PIC'ZZ',	DEFINED YB,		
WB	PIC'9',			
ROB	PIC'9',			
HBS	FIXED DEC(9,4),			
AB	PIC'9',			

/\* KIND OF BUILDING, KIND OF WALL AND ROOF, YEAR INCLUDING TOTAL \*/

C1(5,13,14) FIXED DEC(11,4),  
C2(5,13,14) FIXED DEC(11,4),  
C3(5,13,14) FIXED DEC(11,4),  
C4(5,13,14) FIXED DEC(11,4),  
C5(5,13,14) FIXED DEC(11,4),  
C6(5,13,14) FIXED DEC(11,4)

/\* C1 IS S1,C2 EUP,C3 MEON OF DO,C3 IS DO TOTAL ALSO \*/

/\* C4-C6 ARE NATIONAL TOTAL,C6 IS ALSO TOTAL \*/

SW CHAR(1) INIT('0'), /\* LAST \*/

PAG FIXED DEC(3) INIT(0), /\* PAGE \*/

(LAB1,LAB2) LABEL,

1 ED,

2 EDP(14) PIC'ZZZZBZZ9', /\* 14\*8=112 \*/

EDC CHAR(112) DEFINED ED, /\* 112 \*/

GR3(4) CHAR(3) INIT('SI','EUP','MEO','TOT'),

GR4(5) CHAR(5)

INIT('TOT','DETAC','ATTAC','APPAR','OTHER'),

```

GR5(2)          CHAR(3) INIT('WAL','ROO'),
GR6(13)         CHAR(5) INIT
('TOT','WOOD','CLAY','CEMEN','BRICK','CONC','OTHER',
'STRAW','SH-ZN','SLATE','TILE','SLAB','OTHER'),
GRP1 CHAR(2),
GRP2 CHAR(2),
GRP3 CHAR(3),
GRP4 CHAR(5),
GRP5 CHAR(3),
GRP6 CHAR(5),
4 ON ENDFAGE(PROT) GO TO POV,
5 ON ENDFILE(TAPA) GO TO LAT,
6 LAB2=OPN,
7 CLE1.. C1=0,
8 G2=0,
9 C3=0,
10 GO TO LAB2,

```

```

11 OPN.. OPEN FILE(TAPA),FILE(PROT),.
12      C4=0,.
13      C5=0,.
14      C6=0,.
15      LAB1=REL,.
16 POV.. PAG=PAG+1,.
17      PUT FILE(PROT) PAGE EDIT
      ('HOUSING UNITS BY SIZE,TYPE CONSTRUCTION MATERIALS','TABLE 8',PAG)
      (COLUMN(21),A,COLUMN(101),A,COLUMN(121),F(3)),.
18      PUT FILE(PROT) SKIP(2) EDIT('PERIOD OF BUILDING CONSTRUCTION')
      (COLUMN(31),A),.
19      PUT FILE(PROT) EDIT
      ('TOTAL 1970 1969 1968 1967 1966 1965 1964 1963
1962 1961 1960-51 1950-45 1944BEFO')
      (COLUMN(21),A),.
20      GO TO LAB1,.
21 REL.. READ FILE(TAPA) INTO(TA),.

```

```

22  MV1... DOB=DOA,.. /* DO /*
23  HSB=HSA,.. /* BUILDING /*
24  RB3=RA3,.. /* YEAR1 /*
25  YB=YA,.. /* YEAR /*
26  WB=WA,.. /* WALL /*
27  ROB=ROA,.. /* ROOF /*
28  HBS=HAS,.. /* RATIO /*
29  AB=AA,.. /* AREA /*
30  READ FILE(TAPA) INTO(TA),.
31  PRO1.. IF HSB LT 1 OR HSB GT 4 THEN HSB=4,.
32  IF WB LT 1 OR WB GT 6 THEN WB=6,.
33  IF ROB LT 1 OR ROB GT 6 THEN ROB=6,.
34  IF RB3 LT '1' OR RB3 GT '4' THEN RB3='4',.
35  I=HSB+18,.. /* BUILDING /*
36  J=WB+18,.. /* WALL /*
37  K=ROB+1118,.. /* ROOF /*
38  IF RB3='4' THEN,.

```

```

39     ELSE GO TO YB60,.
40     L=(YC-70) * (-1B)+10B,.      /* 70,69,...61 */
41     GO TO DOE,.
42 YB60.. IF RB3='3' THEN,.
43     ELSE GO TO YB50,.
44     L='1100B,. /* 51-60 */
45     GO TO DOE,.
46 YB50.. IF RB3='2' THEN,.
47     ELSE GO TO YB44,.
48     L='1101B,. /* 45-50 */
49     GO TO DOE,.
50 YB44.. L='1101B,. /* 44AND BEFORE */
51 DOE.. IF AB LT 1 OR AB GT 3 THEN GO TO ERPU,.
52     IF AB=1 THEN,.
53     ELSE GO TO AB2,.
54     C1(I;J,L)=C1(I,J,L)+HBS,. /* RATIO ADDING WALL */
55     C1(I,K,L)=C1(I,K,L)+HBS,. /* ROOF */

```

```

56      GO TO DOT, .
57  AB2..  IF AB=2 THEN, .
58      ELSE GO TO AB3, .
59      C2(I,J,L)=C2(I,J,L)+HBS, .      /* WALL
60      C2(I,K,L)=C2(I,K,L)+HBS, .      /* ROOF
61      GO TO DOT, .
62  AB3..
63      C3(I,J,L)=C3(I,J,L)+HBS, .
64      C3(I,K,L)=C3(I,K,L)+HBS, .
64 DOT..  IF SW= '1' OR DOB.NE DOA THEN, .
65      ELSE GO TO MVI, .
        /* ADDING BUILDING TOTAL,SI,EUP,MEON
66      DO I=2 TO 5, .
67      DO J=2 TO 7, .
68      DO L=2 TO 14, .
69      C1(I,I,L)=C1(I,I,L)+C1(I,J,L), .
70      C2(I,I,L)=C2(I,I,L)+C2(I,J,L), .

```

```

71      C3(I,I,L)=C3(I,I,L)+C3(I,J,L),.
72      END,.
73      END,.
74      END,.

```

```

/* ADDING EACH AREA TOTAL */

```

```

75      DO I=2 TO 5,.
76      DO J=1 TO 13,.
77      DO L=2 TO 14,.
78      C1(I,J,L)=C1(I,J,L)+C1(I,J,L),.
79      C2(I,J,L)=C2(I,J,L)+C2(I,J,L),.
80      C3(I,J,L)=C3(I,J,L)+C3(I,J,L),.
81      END,.
82      END,.
83      END,.

```

```

/* LEFT TOTAL */

```

```

84      DO I=1 TO 5,.
85      DO J=1 TO 13,.
86      DO L=2 TO 14,.

```

```

87      C1(I,J,I)=C1(I,J,I)+C1(I,J,L),.
88      C2(I,J,I)=C2(I,J,I)+C2(I,J,L),.
89      C3(I,J,I)=C3(I,J,I)+C3(I,J,L),.
90      END,.
91      END,.
92      END,.
/* ADDING INTO LAST */
93      C4=C4+C1,.
94      C5=C5+C2,.
95      C6=C6+C3,.
/* ROUNDING */
96      C1=C1+0.5,.
97      C2=C2+0.5,.
98      C3=C3+0.5,.
/* SI,EUP,MEON,DO TOTAL PRINTING */
99 DOK.. DO K=1 TO 4,.
100      GRP1=DOB,.

```

```

101      GRP2='DO' ,.
102      GRP3=GR3(K) ,.
103      DO I=1 TO 5 ,.
104      GRP4=GR4(I) ,.
105      DO J=1 TO 13 ,.
106      IF J=13 THEN ,.
107      ELSE GO TO NEX1 ,.
108      GRP5=GR5(J) ,.
109      GO TO NEX2 ,.
110      NEX1..IF J=1000B THEN ,.
111      ELSE GO TO NEX2 ,.
112      GRP5=GR5(2) ,.
113      NEX2..GRP6=GR6(J) ,.
114      DO L=1 TO 14 ,.
115      IF K = 18 THEN ,.
116      ELSE GO TO DOL1 ,.
117      EDP(L)=CI(I,J,L) ,.

```

\*/ LAST GI \*/

\*/ SI \*/

```

118      GO TO DOLE, .
119      DOL1.. IF K=10B THEN, .
120      ELSE GO TO DOL2, .
121      EDP(L)=C2(I, J, L), . * EUP *
122      GO TO DOLE, . * NEON AND TOTAL *
123      DOL2.. EDF(L)=C3(I, J, L), .
124      DOLE.. END DOL, .
125      LAB1=FUDO, .
126      FUDO.. PUT FILE(PROT) SKIP(2) EDIT(GRP1, GRP2, GRP3, GRP4, GRP5, GRP6, EDG)
           (A(2), A(2), A(3), A(5), A(3), A(5), COLUMN(21), A(112)), .
127      GRP1=' ', .
128      GRP2=' ', .
129      GRP3=' ', .
130      GRP4=' ', .
131      GRP5=' ', .
132      END DOJ, .
133      END DOI, .

```

134 IF DOB='11' OR DOB='21' THEN GO TO LAQ,.

135 IF K='11B' THEN,.

136 ELSE GO TO DOKL,.

137 C1=C1-0.5,.

138 C2=C2-0.5,.

139 C3=C3-0.5,.

140 C3=C1+C2+C3,.

141 C3=C3+0.5,.

303

\* THIS TIME C3 IS FOR DO TOTAL, 0.5 IS FOR ROUNDING ADJUSTING \*

142 DOKL.. END DOK,.

143 LAQ.. IF SW='11' THEN GO TO LAP,.

144 LAB2=MV1,.

145 GO TO CLE1,.

146 LAP..

\* ROUNDING

C4=C4+0.5,.

147 C5=C5+0.5,.

\*/

148

C 6 = C 6 + 0.5, .

\* SI, EUP, MEON, ALL NATION TOTAL

\* /

149 LOK. . DO K = 1 TO 4, .

150 . . . GRP 1 = 'AL', .

151 . . . GRP 2 = 'L', .

152 GLEQ. . GRP 3 = GR 3(K), .

153 DLI. . DO I = 1 TO 5, .

154 . . . GRP 4 = GR 4(I), .

155 DLJ. . DO J = 1 TO 13, .

156 . . . IF J = 1 B THEN, .

157 . . . ELSE GO TO NEL 1, .

158 . . . GRP 5 = GR 5(1), .

159 . . . GO TO NEL 2, .

160 NEL 1. . IF J = 1000 B THEN, .

161 . . . ELSE GO TO NEL 2, .

162 . . . GRP 5 = GR 5(2), .

163 NEL 2. . GRP 6 = GR 6(J), .

304

```

164 ALA.. DO L=1 TO 14,.
165     IF K=1 THEN,.
166     ELSE GO TO LOL1,.
167     EDP(L)=C4(I,J,L),. * ALL SI */
168     GO TO LOLE,.
169 LOL1.. IF K=10 THEN,.
170     ELSE GO TO LOL2,.
171     EDP(L)=C5(I,J,L),. * ALL EUP */
172     GO TO LOLE,.
173 LOL2.. EDP(L)=C6(I,J,L),. * ALL MEON AND NATONA TOTALL */
174 LOLE.. END ALA,.
175     LAB1=PULA,.
176     PULA.. PUT FILE(PROT)..SKIP(2) EDIT(GRP1,GRP2,GRP3,GRP4,GRP5,GRP6,EDC)
        (A(2),A(2),A(3),A(5),A(3),A(5),A(3),A(5),COLUMN(21),A(112)),.
177     GRP1=' ',.
178     GRP2=' ',.
179     GRP3=' ',.

```

```

180 GRP4=1,.,.
181 GRP5=1,.,.
182 END DLIJ,.,.
183 END DLI,.,.
184 IF K=11B THEN,.,.
185 ELSE GO TO LOKL,.,.
186 C4=C4-0.5,.,.
187 C5=C5-0.5,.,.
188 C6=C6-0.5,.,.
189 C6=C4+C5+C6,.,.
190 C6=C6+0.5,.,.
191 * THIS TIME C6 IS NATION TOTAL *
191 LOKL,.,. END LOK,.,.
192 GO TO ENDA,.,.
193 LAT,.,. SW=1,.,.
194 GO TO PRO,.,.
195 ERPU,.,. PUT FILE(PROF), SKIP EDIT

```

(I, J, K, L, DOB, HSB, RB3, YB, WB, ROB)

(4F(2), A(2), F(1), A(1), A(2), 2F(1), F(10,4), F(1)),.

196 ENDA.. END HOUSE8..

結 果 表

HOUSING UNITS BY SIZE, TYPE

	PERIOD OF BUILDING				
	TOTAL	1970	1969	1968	1967
31 DOSI TOT WALTOT	101 754	3 239	5 010	4 870	4 125
WOOD	26 007	313	326	310	352
CLAY	38 044	694	1 190	1 073	1 086
GEMEN	22 293	849	1 581	1 760	1 318
BRICK	12 357	1 154	1 562	1 192	1 232
CONC	2 425	190	283	505	109
OTHER	627	39	68	29	29
ROOSTRAW	12 983	49	127	137	109
SH-ZN	2 887	39	59	65	49
SLATE	11 872	596	1 032	635	896
TILE	61 799	1 717	2 468	2 900	2 368
SLAB	5 757	702	1 034	928	440
OTHER	6 456	137	290	205	264
DETACWALTOT	93 002	3 137	4 496	4 388	3 763
WOOD	23 944	313	313	303	352
CLAY	36 229	694	1 173	1 056	1 075
GEMEN	19 195	821	1 466	1 710	1 222
BRICK	11 415	1 143	1 319	1 153	987
CONC	1 603	127	156	137	98
OTHER	616	39	68	29	29
ROOSTRAW	12 862	49	127	137	98
SH-ZN	2 736	39	59	59	49
SLATE	10 330	596	850	596	645
TILE	56 010	1 671	2 316	2 854	2 306
SLAB	4 838	645	860	538	401
OTHER	6 226	137	283	205	264

& CONSTRUCTION MATERIALS

TABLE 8

CONSTRUCTION								
1966	1965	1964	1963	1962	1961	1960-51	1950-45	1944BEF
2 595	3 510	2 170	3 322	3 738	2 341	28 663	17 663	20 507
332	479	381	397	527	227	7 261	5 971	9 133
704	1 092	856	1 202	1 753	813	12 563	7 144	7 874
848	1 006	527	1 114	820	587	6 001	3 355	2 526
531	657	343	430	539	557	2 370	949	841
180	257	63	159	79	117	272	157	54
0	20	0	20	20	39	195	88	79
59	127	127	216	117	156	3 591	2 747	5 421
29	88	10	40	60	39	1 020	676	713
412	572	345	404	428	314	2 944	2 252	1 041
1 466	2 013	1 360	2 181	2 389	1 421	18 115	10 733	12 669
375	406	192	251	249	293	508	261	119
254	304	137	230	495	117	2 486	993	544
2 443	3 362	2 033	2 609	3 313	2 287	27 394	16 028	17 748
332	479	381	381	479	205	6 792	5 727	7 887
704	1 016	762	1 202	1 573	792	12 265	6 528	7 389
811	977	498	508	655	577	5 620	2 707	1 622
459	635	332	381	528	557	2 306	880	733
137	235	59	117	59	117	215	98	49
0	20	0	20	20	39	195	88	69
59	127	127	205	117	156	3 548	2 697	5 414
29	88	10	29	49	39	968	655	665
342	518	235	371	371	303	2 805	1 749	948
1 427	1 974	1 349	1 564	2 072	1 378	17 220	9 744	10 135
332	362	176	215	215	293	440	254	108
254	293	137	225	489	117	2 414	928	479