

# BASIC해설(II)

朴 贊 震  
(KIET 電算室)

〈 차 례 〉
I. 문자열과 배열
II. 산술함수

## I. 문자열과 배열

일반 계산기와 달리 Computer는 배열 및 문자열을 취급하는 능력을 가지고 있으며, 이 기능은 매우 중요하게 사용된다.

“PERSONAL COMPUTER”를 써내게 하려면 PRINT Statement를 사용하여 써낼 수 있는 것은 이미 배웠다. “PERSONAL COMPUTER”와 같은 문자의 집합을 문자열이라고 한다.

문자의 예인 문자열을 나타내는 변수를 문자열 변수라고 부르고, 보통 변수의 다음에는 \$를 붙인다. 예컨대, A\$, B\$, C\$ 등은 문자열 변수이다. 또한 알파벳이나 +, -, \*, / 등의 기호는 문자열로 취급되며, 만약 이들을 수치로 취급하면 미스가 되어 에러 메시지로 경고를 받지만 1234, 12.3, -12.3 등의 경우는 수치로 취급할 수도 있지만 문자열로서도 취급할 수 있다. 따라서 문자열 변수를 사용할 때에는 수치의 취급에 충분히 주의해야 한다.

## 1. 문자열 (STRING) 입·출력

INPUT 기호를 사용하여 수치를 읽어 들이는 방법은 이미 배웠는데, 마찬가지로 문자열을 INPUT 기호를 사용하여 읽어들이 수 있다.

INPUT 기호를 사용하여 문자열을 읽어들이게 하는 프로그램 예를 보면 다음과 같다.

```
10 INPUT A$
20 PRINT A$
30 END
] RUN
? ABCD ← 문자열 A$에 대응하는 변수
   ABCD ← 문자열 A$에 대응하는 출력
```

일반적으로 문자열을 정의할 때에는 A\$="ABCD"라고 하여 문자열 ABCD를 쿼테이션 마크 안에 넣으나 INPUT 기호로 문자열을 읽어 들이게 할 때에는 위의 프로그램 예에서와 같이 쿼테이션 마크는 필요없다. 그러면 또 하나의 INPUT 방법인 READ와 DATA 기호를 사용한 프로그램을 보면 다음과 같다.

```
10 READ A$
20 PRINT A$
30 DATA ABCD
40 END
] RUN
   ABCD
```

위 프로그램에서는 READ Statement로 문자열 변수 A\$를 읽어 들이게 하는 지시로서 A\$의 내용을 DATA Statement로 ABCD라고 주고 있다. 문자열을 INPUT 기호로 읽어 들이게 할 때와 마찬가지로 DATA 기호로 문자열을 줄 때에는 쿼테이션 마크는 필요없다.

## 2. 배열 (ARRAY)

앞에서 A\$, B\$, C\$ 등을 문자열 변수라고 설명했다. 그러나 A(1), A(2), A(3) …… 등은 첨자부 변수로서 이들을 프로그램 내에서 사용하는 경우에는 DIM

이라는 기호를 사용하여 이 변수를 기억하는 자리를 컴퓨터의 메모리 안에 확보하지 않으면 안된다. 이렇게 확보된 자리를 배열 (ARRAY)이라고 한다. 예를 들면 DIM A\$(4)라고 하면 A\$(0), A\$(1), A\$(2), A\$(3), A\$(4) 라는 식으로 5개의 자리가 확보되며 DIM A\$(4, 5, 3)은  $5 \times 6 \times 4$  개의 원소 즉, 120개의 원소를 확보하게 된다.

보통 원소들은

A\$(4, 4, 1)

A\$(0, 0, 2)

등으로 나타난다.

A\$(4)는 1개의 첨자부가 붙은 문자열 변수, 그리고 A\$(4, 5, 3)은 3개의 첨자가 붙은 문자열 변수이다.

그러면 FOR.....NEXT 루프를 사용하는 2개의 첨자부 문자열 변수가 있는 프로그램 예를 들어 보자.

< 프로그램 예 >

```

10   DIM      A$(3, 4)
20   FOR      L=1      TO      2
30   FOR      M=1      TO      4
40   READ    A$(L, M)
50   NEXT    M
60   NEXT    L
70   FOR      N=1      TO      4
80   LET      A$(3, N)=A$(1, N)+" " +A$(2, N)
90   PRINT   A$(1, N), A$(2, N), A$(3, N)
100  NEXT    N
110  DATA   BLACK, LARGE, WHITE, YELLOW
120  DATA   PIG, DOG, HEN, FOX
130  END
]    RUN
BLACK      PIG      BLACK      PIG

```

LARGE	DOG	LARGE	GOG
WHITE	HEN	WHITE	HEN
YELLOW	FOX	YELLOW	FOX

지금까지 공부한 변수를 크게 분류하면 다음 4 종류가 된다.

1. 보통변수 ..... A, B, A<sub>3</sub> 등
2. 보통 첨자부 변수 ..... A(4), B(3, 2) 등
3. 문자열 변수 ..... A\$, B\$ 등
4. 첨자부 문자열 변수 ..... A\$(3), B\$(3, 2) 등

이들 4 종의 변수를 컴퓨터는 명확하게 구별할 수 있다. 예를 들면 A, A(4), A\$, A\$(3) 을 같은 프로그램 안에서 사용해도 혼란은 일어나지 않는다.

### 3. LEN

LEN은 지정된 문자열의 길이(문자수)를 나타낸다. 여기서 문자열은 변수, 연산식, 상수가 될 수 있고 괄호를 사용해야 한다. BASIC에 따라서는 문자열, 문자수의 제한이 있다.

PAT, TRS-80, APPLE-II BASIC 등의 경우 문자열의 문자수는 0으로부터 255 문자까지 허용된다. 문자수가 0인 문자열을 Null String이라 부른다. LEN을 이용한 간단한 프로그램은 다음과 같다.

< 프로그램 예 >

```

10 LET A$="PEAR AND APPLE"
20 PRINT LEN(A$)
30 END
] RUN
13

```

또한 문자열 Data가 많을 경우 LEN 기호와 TAB 기호를 사용하여 문자열 Data를 프로그래머가 필요에 따라서 일정한 간격을 띄우고 출력시킬 수 있다.

< 프로그램 예 >

```

10 DIM A$(4)

```

```

20   K=0
30   ROR   N=1   TO   4
40   READ  A$(N)
50   LET   K=K+LEN(A$(N))+3
60   PRINT A$(N); TAB(K);
70   NEXT  N
80   DATA COW, PIG, MONKEY, HORSE
90   END
]   RUN
COW      PIG      MONKEY      HORSE

```

#### 4. LEFT\$(A\$, N)

문자열의 변수 A\$의 내용중 문자열의 좌변으로부터 n개의 문자를 새로운 문자열로 한다.

LEFT\$ 기호와 동류의 기호로 MID\$ 와 RIGHT\$ 기호가 있다.

< LEFTS\$ 의 프로그램 예 >

```

10   A=2
20   A$="ABCDEF"
30   B$=LEFT$(A$, 2)
40   C$=LEFT$(A$, A+1)
50   D$=LEFT$("ABCDE", A)
60   PRINT  A$, B$, C$, D$
70   END
]   RUN
ABCDEF      AB      ABC      AB

```

#### 5. MID\$(A\$, I, J)

주어진 A\$ 문자열에서 I번째 위치로부터 시작하여 J개의 문자를 차례로 선

택하여 보조 문자열을 만든다. 문자열 명을 비롯하여 시작점 길이를 사용할 때 반드시 괄호를 써야 한다. 또한 MID\$(A\$, I, J)의 J를 생략하여 MID\$(A\$, I)라고 하면 문자열의 I번째의 문자부터 마지막 문자까지로 취급된다. 문자열은 문자열 상수나 식도 가능하고 I와 J는 수식 또는 상수도 가능하다.

< MID\$ 프로그램 예 >

```

10 LET B$="ABCDEFGH"
20 LET C$=MID$(B$, 2,6)
30 PRINT MID$(C$, 2, 3)
] RUN
CDE

```

◦ RIGHT\$(A\$, I)

문자열의 끝에서부터 I개의 글자를 나타낸다. 문자열과 I는 괄호를 사용하여 하고 문자열은 상수나 변수이며 I는 숫자형 상수 또는 변수이다.

< RIGHT\$의 프로그램 예 >

```

10 LET B$="ABCDEFGH"
20 LET C$=RIGHT$(B$, 5)
30 LET D$=C$+RIGHT$(C$, 2)
40 PRINT B$
50 PRINT C$
60 PRINT D$
70 END
] RUN
ABCDEFGH
DEFGH
DEFGHGH

```

## 6. ASCII 코드

알파벳, 숫자, 기호 등을 컴퓨터가 기억하거나 처리할 때에는 컴퓨터가 취급

하기 쉬운 수치로 변환한다. 예컨대 A는 컴퓨터 안에서는 65(10진법)로 되어 있다. 실제로는 10진법으로 표시하는 65를 사용하지 않고 2진법으로 돼있다. 그러나 BASIC 언어로 프로그램을 작성하는 경우에 혼란을 피하기 위해 코드가 명확하게 정해져 있다. 현재 마이크로 컴퓨터 대부분은 ASCII 코드를 사용하고 있으므로 ASCII 코드만을 소개한다.

〈 10진 ASCII Code표 〉

코드 (10진법)	문자기호	코드 (10진법)	문자기호	코드 (10진법)	문자기호
0	NULL	32	SPACE	64	@
1	SOH	33	!	65	A
2	STX	34	"	66	B
3	ETX	35	#	67	C
4	EOT	36	\$	68	D
5	ENQ	37	%	69	E
6	ACK	38	&	70	F
7	BEL(벨)	39	`	71	G
8	BS	40	)	72	H
9	HT	41	(	73	I
10	LF	42	*	74	J
11	VT	43	+	75	K
12	EF	44	'	76	L
13	CR(改行)	45	-	77	M
14	SO	46	.	78	N
15	SI	47	/	79	O
16	DLE	48	0	80	P
17	DC	49	1	81	Q
18	DC	50	2	82	R
19	DC	51	3	83	S
20	DC	52	4	84	T
21	NAK	53	5	85	U
22	SYN	54	6	86	V
23	ETB	55	7	87	W
24	CAN	56	8	88	X
25	EM	57	9	89	Y
26	SUB	58	:	90	Z
27	ESCAPE	59	;	91	[
28	FS	60	<	92	/
29	GS	61	=	93	]
30	RS	62	>	94	↑
31	US	63	?	95	←

註: 96~126은 省略:(小文字알파벳 등)  
127은 DEL(消去)

## 7. ASC (문자열)

ASC 기호는 지정된 문자열을 숫자의 코드(ASCII 코드)로 변환하는 기능을 가지고 있다. 그리고 문자열 인수는 반드시 괄호를 사용해야 한다.

< ASC 프로그램 예 >

```
10   LET   A$="ABC"
20   FOR   N=1   TO   LEN(A$)
30   PRINT ASC(MID$(A$, N))
40   NEXT  N
50   END
]   RUN
65   66   67
```

## 8. CHR (수식)

Character(문자나 기호)를 ASCII 코드로 변환시키는 기호는 ASC 기호이다. 그러나 그 반대의 기능을 가진 기호는 CHR\$ 기호이다. CHR\$ 기호는 지정된 10진수 ASCII 코드를 문자열로 바꾼다. 인수는 0에서 255 사이의 숫자 또는 그 범위내의 값을 갖는 변수식이어야 한다.

< CHR\$ 프로그램 예 >

```
10   FOR   I=65   TO   90
20   PRINT I, CHR(I)
30   NEXT  I
40   END
]   RUN
65   A
66   B
67   C
    :
89   Y
90   Z
```



또한 ASCII 코드에서 128 이상은 그리스 문자 등을 정의하고 있는 일도 있으나 보통은 사용되지 않는다. 그 이유는 단말기의 대부분이 그리스 문자의 인자를 할 수 없기 때문이다. 같은 이유로 ASCII 코드 97~122의 소문자에 대해서도 인자 기능이 없는 단말기에서는 사용할 수 없다.

## 9. STR\$(식)

STR\$의 기능은 수식이나 상수를 문자열로 바꾸는 기능을 가지고 있다. 여기에서 수식이나 상수는 반드시 괄호가 사용되어야 한다. 예를 들면 STR\$(A)는 문자열 A값의 글자 표현과 같게 나타낸 것이다. 만일 A=35.2라면 STR\$(A)는 35.2라는 문자열이 된다.

< STR\$ 프로그램 예 >

```

10      A=35.2
20      B=-35.2
30      PRINT  STR$(A)
40      PRINT  STR$(B)
50      PRINT  STR$(A+B)
60      PRINT  STR$(A)+STR$(B)
70      END
]      RUN
35.2
-35.2
0
35.2-35.2

```

## 10. STRING\$(n, ("문자" 또는 숫자))

주어진 기호나 문자로 n개의 길이를 갖는 문자열을 만든다.

예를 들어 STRING(10, "\*")은 "\*\*\*\*\*"로 표시된다.

이것은 일람표나 도표를 만들 때 아주 편리하게 이용된다. 또한 인수 n은 1~

255 범위의 값을 갖는 수식이다.

### 11. VAL (문자열)

STR\$ 기능의 반대로서 문자열 인수 내의 글자를 숫자로 나타낸다. 이 결과에 대한 상수의 형태는 상수형태의 규칙에 의하여 결정되는 정수, 1단계 실수, 2단계 실수로 만들 수 있다.

예를 들어 A\$="12", B\$="34"에서 VAL(A\$+"."+B\$)는 12.34의 값이 되며 VAL(A\$+"E"+B\$)는 12E34 즉,  $12 \times 10^{34}$ 의 값이 된다.

## II. 산술함수

컴퓨터를 사용하여 여러가지 계산을 할 때 평방근이나 삼각함수 등을 하나의 기호로서 구할 수 있다면 참으로 편리할 것이다. 그래서 BASIC 언어에서는 편리하게 여러가지 계산을 할 수 있도록 몇 가지의 함수기호가 준비되어 있다.

마이크로 컴퓨터용으로 개발된 BASIC 언어에서 통상 사용할 수 있는 함수의 종류는 다음과 같다.

### < BASIC 에서 사용되는 함수의 종류 >

BASIC 記號	計算하는 값
ABC(x)	절대치 $ x $
INT(x)	x 이하에서 가장 큰 整數
RND(x)	亂數
SGN(x)	x ≥ 0 일 때 +1, x < 0 일 때 -1, x = 0 일 때 0
SQR(x)	평방근 $\sqrt{x}$
SIN(x)	sin(x)
COS(x)	cos(x)
TAN(x)	tan(x)
ATAN(x)	$\tan^{-1}(x)$
TAB(x)	PRINT 스테이트먼트를 사용, x 자리째까지 이동
EXP(x)	$e^x$ (e = 2.71828)
LOG(x)	自然對數 $\log^x$

### 1. ABS(x)

이 기호는 독립변수  $x$ 의 값을 절대값으로 나타낸다.

즉,  $ABS(x)=|x|$ 의 관계가 성립한다.

예를 들면,

$$ABS(3)=3$$

$$ABS(-5)=5$$

$$ABS(3 \cdot 8)=3.8$$

만약 A, B, C의 값이 LET Statement 등으로 주어져 있다고 하면 ABS( ) 기호의 괄호 안에는 변수나 식을 써도 괜찮다.

### 2. INT(x)

독립변수  $x$ 를  $x$ 의 값보다 크지 않은 가장 큰 정수 값으로 나타낸다. 이 경우 결과는 정수값의 범위의 제약을 받지 않는다. 이 결과는 1단계 실수 전체 숫자로서 저장된다.

예를 들면,

$$INT(1.8)=1$$

$$INT(3.2)=3$$

$$INT(-.4)=-1$$

$$INT(-2 \cdot 9)=-3$$

### 3. SGN(x)

이 함수는  $x$ 의 값이 +인가 -인가를 판별하는 기능을 가진다.

$x$ 가 +인 경우에는  $SGN(x)$ 는 +1이 되고,  $x$ 가 -인 경우에는  $SGN(-x)$ 은 -1이 된다.

예를 들면,

$$SGN(4 \cdot 7)=1$$

$$SGN(.4)=1$$

$$SGN(-4 \cdot 7)=-1$$

$$SGN(-5)=-1$$

#### 4. SQR(x)

이 함수는 독립변수  $x$ 의 제곱근( $\sqrt{x}$ )을 구하는 것이다. 즉,  $SQR(x)=\sqrt{x}$  이다. 그리고  $x$ 의 값은 항상 +의 수가 아니면 안된다. 만일, -의 수일 경우에는 컴퓨터는 Error 메시지를 보낸다.

#### 5. SIN(x)

이 함수는 독립변수  $x$ 의 sin값을 구한다. 이 함수의 사용방법에서 주의해야 할 것은  $x$ 의 값은 도가 아니라 radian으로 표시해야 한다는 점이다.

radian과 도의 관계는 다음과 같다.

$$1 \text{ 도} = \frac{\pi}{180} \text{ rad} = 3.14159/180 \text{ (rad)}$$

따라서  $\sin(60^\circ)$ 를 계산하고 싶으면  $SIN(60*3.14159/180)$ 과 같이 해야 한다.

#### 6. COS(x)

이 함수는  $\cos(x)$ 의 값을 계산한다. 이  $x$ 의 값도 radian으로 계산해야 한다는 것은  $\sin(x)$ 와 마찬가지로이다.

#### 7. TAN(x)

$\tan x$ 의 값을 계산하는 기호로서  $x$ 의 값은 radian으로 표시한다는 것은  $\sin(x)$ 함수와 마찬가지로이다.

#### 8. ATN(x)

이 함수는  $\tan^{-1}(x)$ 의 값을 계산한다. 계산 결과는 radian으로 표시되고  $-\frac{\pi}{2}$ 와  $\frac{\pi}{2}$ 사이의 값이 된다.

#### 9. TAB(x)

tab기능의 역할을 하는 함수로서 tab기능이란 타이프라이터에도 있는데 인

자하는 부분을 정해진 자리까지 움직이는 것을 의미한다. 예컨대 TAB(15) 라고 하면 인자하는 부분이 16 자리째까지 움직이고 정지하여 문자를 쓸 지령을 기다린다. 통상 TAB(x)함수는 PRINT Statement 안에서 사용된다. 보통 텔레타이프(TTY)의 인자폭은 72 자리이므로 TTY의 경우에는

TAB(0)은 좌단의 위치

TAB(72)은 우단의 위치

를 의미한다.

#### 10. EXP(x)

이 함수는  $e^x$ 를 계산한다( $e=2.71828$ ).  $x$ 의 값을 87 이상으로 하면 이 함수는 너무 커져서 오버플로우를 일으킨다. 또 BASIC 언어에 따르는 이 함수를 사용할 수 없는 경우가 있는데 그 때에는  $2.71828 \uparrow x$ 라고 하면 된다.

#### 11. LOG(x)

독립변수  $x$ 의 자연대수 즉,  $\log_e(x)$ 를 취한다. 이것은 EXP함수의 역이므로  $x = \text{LOG}(\text{EXP}(x))$ 이다. 10을 밑으로 하는 수의 대수값을 구하려면  $\log_{10}(x) = \log_e(x) / \log_e(10)$ 의 공식을 사용하면 된다. 예를 들어,  $\text{LOG}(32767) / \log(2)$ 는 그를 밑으로 하는 32767의 대수이다.

#### 12. RND(x)

이것은 주어진 인수를 사용하여 의사난수를 발생하며, 이 값은 내부적으로 만들어지며 사용자가 요구하는 특정값이 아니다. RND는 0과 1 사이의 난수를 만들기 위하여 또는 주어진 인수에 따라 0보다 큰 정수형의 난수를 발생시키기 위하여 사용된다.

RND(0)는 0과 1 사이의 값을 1단계 실수형으로 나타내며 RND( $n$ )은 1~ $n$ 까지의 정수값을 순서없이 발생한다.

예를 들면, RND(55)는 0보다 크고 56보다 적은 정수형의 난수를 발생한다.