

BASIC에 의한 컴퓨터 프로그램(4)

編 輯 室

BASIC에서의 반복과 배열조작

BASIC문은 행번호 순서에 따라 실행되는 것이 원칙이지만, 때에 따라서는 어느 행까지 건너 뛰고 싶든지, 어느 행으로 옮기고자 하는 경우가 생긴다.

조건에 따라서는 분기하여, 각각의 처리를 하기 위하여 프로그램의 일부분을 반복하여 실행해야 할 경우도 생긴다.

위와 같은 경우에 BASIC에서는 다음과 같은 문이 있다.

- (1) 지정한 행으로 건너 뛰는 문
 GOTO 문—— 지시한 행으로 간다.
 GOSUB 문—— 지시한 행으로 가서 명령을 수행한 후 제자리로 되돌아 온다.
- (2) 분기에 대한 문
 IF 문—— 조건식에 따라서 분기
 ON 문—— 번호에 따라서 분기
- (3) 반복에 대한 문
 FOR~ NEXT 문—— 일정한 회수의 반복
 WHILE~WEND 문—— 부정회수의 반복 등이 있다.

1. GOTO문(지정한 곳으로 가는 문)

무조건 특정한 행까지 건너 뛰거나 돌아오게 하려면 GOTO 문을 사용한다. 영어의 go to의 의미로, BASIC에서는 GO와 TO의 사이에 공백을 넣지 않는 것이 일반적이다.

GOTO문의 일반형은

GOTO 행번호

〈예〉 GOTO 200

(행번호 200의 문으로 가라는 명령이다.)

1-1. GOTO문의 응용(1) — 같은 프로그램의 반복실행 예

프로그램의 마지막에서 다시 처음으로 가도록 GOTO문을 사용하면, 정지 키(Key)를 칠 때까지는 쉬지 않고 몇번이고 반복 실행할 수 있게 된다.

프로그램 예…… GOTO 문(1)

```

10 INPUT "A=" :A
20 INPUT "B=" :B
30 C=A+B
40 D=A-B
50 PRINT "A+B=";C, "A-B=" ;D
60 GOTO 10

```

위의 프로그램이 계속 수행하는 것을 정지시키면 Break라는 글씨가 나타나면서 정지하게 된다.

1-2. GOTO문의 응용(2) — 값을 바꾸어서 반복실행 예

N의 값을 1, 2, 3……과 같이 순차적으로 바꾸어서 N^2 과 $1/N$ 을 계산하여 출력하는 예를 들어 보면 다음 프로그램과 같다.

프로그램 예……GOTO 문(2)

```

10 N=1
20 S=N^2
30 T=1/N
40 PRINT "N=";N, "S=" ;S, "T="; T
50 N=N+1
60 GOTO 20

```

위의 프로그램에서 처음 N의 값은 1을 가지고 있다가 한번 수행하면 1을 더해서 N의 값은 2로 변하여, 다시 행번호 20으로 가서 수행하게 된다. 이렇게 차차 N의 값은 수가 바뀌어서 반복할 수 있게 한다.

중간에 오류등으로 프로그램이 정지했을 때, 중간 결과를 살리면서 도중의 문에서 실행을 다시 계속 시킬 수 있는 방법은 GOTO 문을 직접 실행 형식으로 사용해서

GOTO 실행시작 행번호

라고 하면 된다.

예를 들어

GOTO 50

이라는 명령을 하면 행번호 50의 위치에서 실행이 다시 시작될 수 있다.

2. IF문(분기 문)

어떤 조건이 성립할 때만 지정한 위치에 가도록 하는 문으로 IF 문이 있는데 다음과 같이 한다.

IF 조건식 THEN 행번호

이것을 다음과 같이 쓸 수도 있다.

IF 조건식 GOTO 행번호

조건이 성립하지 않을 때의 행선을 지정할 수도 있으며, 일반형은

IF 조건식 THEN 행번호 ELSE 행번호

가 된다.

여기서 THEN 이나 ELSE, GOTO의 양쪽에는 반드시 1자 이상의 공백을 두어야 한다.

```

<예>IF A=B+C THEN 50
      IF A=B GOTO 60
      IF C>D THEN 300 ELSE 500

```

3. 조건식의 작성 방법

조건식으로는 다음과 같은 것들이 BASIC에서 사용된다.

3-1. 2개의 식을 등호 또는 부등호로 맺은 것

```

<예>A=B
      C>D
      ABS(E)<0.01

```

BASIC에서 사용되는 부등호에는 다음과 같은 것들이 있다.

- ① 같다 =
- ② 같지않다 < > 또는 > <
- ③ 작다 <
- ④ 같거나 작다 < = 또는 = <
- ⑤ 같거나 크다 > = 또는 = >
- ⑥ 크다 >

3-2. 등식 또는 부등식에 AND, OR, NOT를 붙인것

AND는 양쪽이 모두 성립하는 것

OR는 한쪽만이 성립하는 것

NOT는 부정

을 나타낸다.

<예>A=0 AND B=0

C=0 OR C=0

위의 형태를 괄호로 싸서, NOT을 붙여 AND나 OR로 연결 한 것은 괄호로 싸 부분이 먼저 실행되지만 괄호를 생략하면 다음 순서로 실행하게 된다.

NOT> AND> OR

<예>NOT A OR B AND C는 다음식과 같게 된다.

(NOT A) OR (B AND C)

여기서 AND, OR, NOT의 양쪽은 반드시 1자 이상 공백을 넣어야 하며, 괄호가 올 때는 공백을 넣지 않아도 된다.

3-3. 조건식 응용 예-2차 방정식의 근 2차 방정식

$$ax^2+bx+c=0$$

의 근을 계산하고 출력하는 프로그램

여기서 근은 실근일 때도 있고 허근일 때도 있다.

우선 먼저 2차 방정식 근의 공식을 사용한다.

먼저 판별식

$$D=b^2-4ac$$

의 값을 계산하여, 그 등호에 따라서 다음과 같이 처리한다.

$$D>0 \text{ 이면 } x=(-b \pm \sqrt{D}) / (2a)$$

$$D=0 \text{ 이면 } x=-b / (2a)$$

$$D<0 \text{ 이면 } x=-b / (2a) \pm (\sqrt{|D|} / (2a)) i$$

프로그램 예..... 2차 방정식의 근

```

10 INPUT "A=":A
20 INPUT "B=":B
30 INPUT "C=":C
40 D=B^2-4*A*C
50 IF D<0 GOTO 310
60 IF D=0 GOTO 210
110 REM
120 X1=(-B+SQR(D))/(2*A)
130 X2=(-B-SQR(D))/(2*A)
140 PRINT X1=:X1
150 PRINT X2=:X2
160 GOTO 410
210 REM
220 X=-B/(2*A)
230 PRINT "X=":X

```

```

240 GOTO 410
310 REM
320 U=-B/(2*A)
330 V=SQR(ABS(D))/(2*A)
340 PRINT "U1=":U;"+":V;"i"
350 PRINT "U2=":U;"-":V;"i"
410 END

```

위의 프로그램을 실행하면 다음과 같은 결과가 나온다.

```

run
A=? 2
B=? 10
C=? 12
X1=-2
X2=-3
OK

```

```

run
A=? 10
B=? 20
C=? 30
U1=-1+1.414214 i
U2=-1-1.414214 i
OK

```

```

run
A=? 5
B=? -10
C=? 5
X=1
OK

```

4. IF문(조건에 의한 처리문)

프로그램의 흐름에는 조건에 의하여 각각의 처리를 하고 다시 같은 과정을 처리하는 경우가 자주 생긴다.

예를 들면 조건식이 성립하면 '처리A'를 수행해야 하고 성립하지 않으면

‘처리B’를 수행하게 하는 경우일 때,
처리가 비교적 간단하므로 IF문을 쓸 수 있다.

IF 조건식 THEN 처리A ELSE 처리B

또는 2 뒷부분(ELSE 이후)을 생략한 형식

IF 조건식 THEN 처리A

이다.

여기서 처리A는 『조건식이 성립하였을 때 실행할 프로그램』, 처리B는 『조건식이 성립하지 않았을 때 실행할 프로그램』으로 실제로는

하나의 문

또는

복합문

으로 몇개의 문을 「:」으로 구하여 1행에 쓰는 형식을 취하여 실행하게 할 수 있다.

〈예〉 IF X<0 THEN Y=0 ELSE Y=X
IF A<B THEN PRINT A ELSE PRINT B
IF Z=3 THEN A=0: B=1: C=2

4-1. 조건식 응용예- 수 헤아림

학생 N명의 성적을 M1, M2, M3, ..., MN으로 읽고 60점 이하의 학생수 J를 헤아려서 출력하는 프로그램의 예의 경우

수를 헤아리는 데는 수를 기록하기 위한 변수(Counter)를 준비하여, 처음에 그곳에 0를 넣어 주고 순서적으로 1을 더해가면 된다.

프로그램 예.....수 헤아림
10 INPUT “학생수” :N
20 J=0
30 FOR I=1 TO N

40 INPUT M
50 IF M<60 THEN J=J+1
60 NEXT I
70 INPUT “60점 이하의 학생수는”:K:“명”

위의 프로그램을 수행하면 다음과 같이 된다.

run
학생수는? 6
? 75
? 55
? 35
? 76
? 82
? 57
60점 이하의 학생수는 3명
OK

5. FOR~NEXT 문(반복문)

프로그램 중에서 일정 회수의 반복을 할 때는 IF 문을 써서 일정회수가 끝났는지를 물어보는 것 보다 FOR 문을 쓰는 편이 편리하다.

FOR문은 다음과 같은 형식으로 쓴다.

FOR 제어변수명=시작값 TO 마지막 값 STEP 변화값

이것은 『제어변수의 값을 시작값에서 마지막 값까지 변화값 만큼 더하면서 NEXT까지의 과정을 반복해서 실행하라』라는 의미이며, 그 변화값이 1일 때는 생략하여

FOR 제어변수명=시작값 TO 마지막 값

이라고 써도 된다.

반복 부분의 끝에는 반드시

NEXT 제어변수명

을 써야 한다.

시작값, 마지막값, 변화값은 반드시 정수를 쓸 필요는 없다.

시작값, 마지막값, 변화값은 변수나 식을 써도 무방하다.

변화값은 음의 값을 사용할 수도 있다.

N의 값을 1에서 10까지 1씩 증가시키면서 N^2 , $1/N$, \sqrt{N} 의 값을 계산하여 출력하려는 프로그램 예는 다음과 같다.

프로그램 예..... 반복

```
10 PRINT, "N^2," "1/N," "SQR(N)"
20 FOR N=1 TO 10
30 PRINT N, N^2, 1/N, SQR(N)
40 NEXT N
```

6. 다중 반복문

FOR~NEXT에 의한 반복(loop)를 몇회라도 여러번 중복지켜 할 수 있다.

N은 1에서 10까지, M은 1에서 N까지의 모든 조합에 대해서 분수 M/N을 소수로 고쳐서 출력하려는 프로그램은 다음과 같다.

프로그램 예..... 다중 반복

```
10 FOR N=1 TO 10
20 FOR M=1 TO N
30 PRINT M: "/" :N: "=": M/N
40 NEXT M
50 PRINT
60 NEXT N
```

위의 프로그램에서 행10의 FOR 문에서 N의 값을 1에서 10까지 바꾸고 있으며, 이에 대응하는 NEXT문은 행 60에 있으므로, 행 20에서 행 50이 각 N의 값에 대해서 실행된다.

행 20의 FOR 문에서는 M의 값을 1에서 N까

지 바꾸고 있으며, 이에 대응하는 NEXT문은 행 40에 있으므로, 행 30의 문은 N, M의 모든 조합에 대하여 실행된다.

행 30에서는 우선 M의 값을 출력하고, 그 다음에는 「/」, 다음에 「N」의 값, 다음에 「=」, 마지막으로 「M/N」의 값을 출력하게 된다.

7. 배열

BASIC에서는 여러가지 배열을 취급할 수 있게 된다.

배열에서는 이름을 붙일 수 있으며, 이름을 붙이는 규칙은 변수명을 부여하는 방법과 동일하다.

배열 중의 각각 데이터는 다음과 같이 순서를 생각할 수 있다.

배열명(몇번째) 1열 배열일 때

배열명(몇번째) .행, 왼쪽에서 몇번째) ... 4차 배열일 때

예를 들어

A(7)A라는 배열의 7번째 데이터

B(3, 4).....B라는 배열의 3번째 행의 왼쪽에서 4번째 데이터

등으로 표시 할 수 있다.

이러한 배열은 인용, 대입, 계산식에 사용할 수 있으며 입력, 출력에도 사용할 수 있다.

예를 들면

인용→ K=L(7)

대입→ B(30)=175

계산식→ Z=P(2) * X+C(3)

입력→ INPUT C(31)

출력→ PRINT Y(B)

등으로 표현할 수 있으며, 그때 위치 지정에 변수나 식을 사용해도 된다.

Z(K)=X(I) + Y(J)

A(L+1) = B(2*N)

8. 배열 선언

배열을 선언하는 문은 DIM문으로 배열을 사용할 때 먼저

DIM 배열명(크기) 1열 배열일 때
DIM 배열명(크기, 크기).....4각 배열일 때

와 같은 형식으로 배열의 크기를 결정 등록하여 두어야 한다. 다만 크기가 10이하일 경우는 선언하지 않고 배열을 사용해도 가능하다.

위에서 크기라고 하는 것은 번호의 최대값으로, 컴퓨터는 그 값에 맞추어 배열의 기억장소를 할당하게 된다.

크기를 알고 있을 때에는 그 값을 정수로 쓰

면 된다.

예를 들면

DIM A(4, 5), B(7), C(123)

입력 데이터나 계산 상황에 맞춰서 크기를 정할 때는, 여기에 변수나 식을 사용할 수도 있다.

DIM A(M, N), B(M+1), C(N+M)

배열에서 그 크기의 최소값은 0이다.

DIM A(4, 5)에서 배열 A를 4행5열로 선언하여 사용한 뒤, 필요 없게 된 경우 ERASE A로 배열 선언을 해제 하여도 되고, 또는 DIM A(7, 8)과 같이 앞에서 사용한 크기와 다른 크기를 선언해도 된다.