학습목표

- 반복문의 개념에 대해 설명할 수 있다.
- while 문과 do~while 문의 개념과 사용 방법에 대해 이해할 수 있다.

EBS tip

1 반복문

① 프로그램을 작성하다 보면 같은 작업을 여러 번 반복해야 할 경우가 있다. 반복문은 이처럼 반복 작업할 문장들을 여러 번 작성하지 않고 한 번 작성한 후, 이를 반복 실행할 때 사용한다.

② 반복문에는 while 문, do~while 문, for 문 등이 있다.

2 while 문

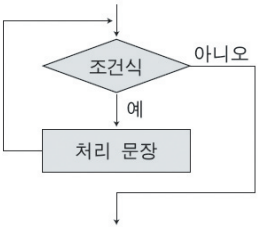
① 조건이 참인 동안 문장을 반복해서 처리할 때 사용한다.

② 기본 형식

```
while(조건식)
{
  처리 문장;
}
```

③ 조건식이 참인 동안 계속해서 [처리 문장]을 수행하게 되며 조건식의 값이 거짓이면 [처리 문장]을 수행하지 않고 반복문을 벗어나게 된다.

④ while 문의 순서도는 다음과 같다.



- 조건식을 만족하는 동안 [처리 문장]이 반복 수행된다. 조건식이 거짓이면 반복문을 종료하고 반복문을 벗어난다.

⑤ 유의할 점

- 조건식에는 반드시 괄호가 있어야 한다.
- 처리 문장 안에 while 문이 종료될 수 있도록 조건식의 결과값을 변화시키는 문장이 필요하다.
- 조건식의 종료 조건이 없는 경우 무한 루프(loop)에 빠지게 되므로 조심해야 한다. 특히 초기화되지 않은 변수가 while 문의 조건으로 사용되면 결과를 예측하기 힘든 경우도 발생한다.

⑥ 명령문 사용 예

```
a = 0, sum = 0;
while(a <= 10){
      a = a + 1;
      sum = sum + a;
}
```

◉ 무한 루프(loop)

반복문을 끝없이 계속 반복하는 현상을 말한다.

예

```
a = 5;
while(a <= 10) {
  sum = sum + a;
}
```

"a<=10"의 조건을 만족하므로 sum = sum + a; 문장을 끝없이 반복 수행하게 된다.

EBS tip

예 n값을 입력받아 1부터 n까지의 숫자 중 4의 배수를 출력하는 프로그램

프로그램 소스 코드	실행 결과

#include <stdio.h>
void main( ){
  int a, n;
  a = 1;
  printf("n값을 입력하세요.\n");
  scanf("%d", &n);
  while (a <= n){
  if (a % 4 == 0)
  printf("%3d\n", a);
  a++;
  }
}
	n값을 입력하세요.
	20
	  4
	  8
	  12
	  16
	  20

[프로그램 해설]

- scanf( ) 문을 이용하여 범위를 나타내는 변수 n에 값을 입력받는다.
- while(a <= n)에서 'a<=n' 을 만족하는 동안 반복문을 수행한다. a값은 1부터 n까지 값이 변한다. 만약 n = 20인 경우, a는 20까지 증가하여 값을 계산하여 출력한 후, 21이 되는 순간 while 문의 조건을 만족하지 않으므로 반복문을 빠져나오게 된다.
- if (a % 4 == 0)에서 a값이 4의 배수인지를 판별하여 배수이면 출력하게 된다.

예 출력할 단을 입력받아 구구단을 출력하는 프로그램

프로그램 소스 코드	실행 결과

#include <stdio.h>
void main( ){
  int a, num;
  printf("구구단 출력 프로그램\n");
  printf("출력할 단을 입력 :");
  scanf("%d", &num);
 	
  a = 1;
  while(a < 10){
  printf("%d * %d = %d\n", num, a, num*a);
  a++;
  }
}
	구구단 출력 프로그램
	출력할 단을 입력 : 6
	6 * 1 = 6
	6 * 2 = 12
	6 * 3 = 18
	6 * 4 = 24
	6 * 5 = 30
	6 * 6 = 36
	6 * 7 = 42
	6 * 8 = 48
	6 * 9 = 54

[프로그램 해설]

- scanf( ) 문을 이용하여 출력할 단을 입력받아 변수 num에 저장한다.
- a=1; → 구구단의 1부터 9까지 단에 곱할 변수 a에 1을 저장한다.
- while(a<10) → while 문을 이용하여 반복문을 수행한다. 변수 a에 저장된 값이 10보다 작으면 블록({ ... }) 안에 있는 문장을 실행하므로 num과 제어 변수 a 값을 곱셈하여 출력한 후, a의 값을 1씩 증가시킨다.
- a값은 1씩 증가한 후 10이 되면 조건을 만족하지 않으므로 반복문이 종료된다.

3 do~while 문

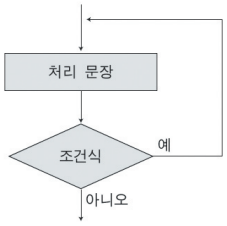
① 먼저 [처리 문장]을 수행한 후 반복 여부를 판단할 때 사용한다.

② 기본 형식

```
do {
  처리 문장;
} while(조건식);
```

③ [처리 문장]을 수행한 후 조건식의 값이 참이면 다시 [처리 문장]을 수행하고, 조건식의 값이 거짓이면 do~while 문을 빠져 나간다. do~while 문은 [처리 문장]을 적어도 한번은 실행한다는 점이 while 문과 다르다.

④ do~while 문의 순서도는 다음과 같다.
처리 문장을 수행한 후 조건식을 비교한다. 조건식을 만족하면 처리 문장을 반복 수행하고, 그렇지 않으면 반복문을 빠져 나온다.



⑤ 유의할 점
- while(조건식) 뒤에 ';' (세미콜론)이 붙는다.
- 조건식의 괄호는 생략할 수 없다.

예 음수가 입력되면 반복문을 종료하는 프로그램

프로그램 소스 코드	실행 결과

#include <stdio.h>
void main(){
  int num;
  do{
  printf("값 입력 : ");
  scanf("%d", &num);
  printf("입력된 값: %d\n", num);
  } while(num >= 0);
}
	값 입력 : 5
	입력된 값: 5
	값 입력 : 4
	입력된 값: 4
	값 입력 : 3
	입력된 값: 3
	값 입력 : -1
	입력된 값: -1

[프로그램 해설]
- scanf( ) 문을 이용하여 num값을 입력받는다.
- 입력된 값을 출력한 후 do~while 문의 조건과 비교한다.
- 0 또는 양수이면 반복문을 수행하면서 계속 값을 입력받고, 음수이면 조건을 만족하지 않으므로 반복문을 벗어나게 된다.
- 따라서 do~while 문의 경우 적어도 한 번 이상 반복문을 수행하게 된다. 왜냐하면 문장을 수행한 후 조건을 비교하기 때문이다.

EBS tip

- 일반적으로 do~while 문은 while 문과 for 문에 비해 사용 빈도가 적다.

명령문 사용 예

a = 0, sum = 0;

```
do {
  a++;
  sum = sum + a;
} while( a < 10 );
```

EBS tip

예 0이 입력될 때까지 입력받은 정수의 합을 계산하여 출력하는 프로그램

프로그램 소스 코드	실행 결과

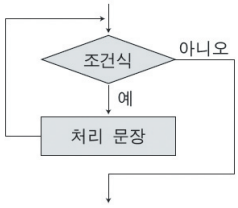
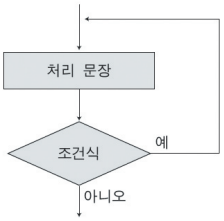
#include <stdio.h>
void main(){
  int num;
  int sum = 0;
  printf("합을 구하고자 하는 수를 입력\n");
  printf("0을 누르면 종료\n");
  do{
  scanf("%d", &num);
  sum = sum + num;
  }while(num != 0);
  printf("입력된 수의 합 = %d\n", sum);
}
	합을 구하고자 하는 수를 입력
	0을 누르면 종료
	9
	7
	5 ┐ 입력하는 수
	2 ┘
	0
	입력된 수의 합 = 23 ← 반복문을 빠져나와 누적된 값 출력

[프로그램 해설]

- 임의의 정수를 입력받아 변수 num에 저장한다.
- 합을 저장할 변수 sum을 선언하면서 0으로 초기화한다. 0으로 초기화하지 않을 경우 C 컴파일러에 따라서 임의의 값이 저장되어 있어 엉뚱한 값이 나올 수도 있으므로 합을 저장할 변수는 반드시 0으로 초기화 하는 작업이 필요하다.
- sum=sum+num; → 입력받은 정수 num을 sum에 누적하여 합을 구한다.
- while(num != 0); → 변수 num에 저장된 값이 0이 아니면 다시 반복 처리하기 위해 첫 부분인 do 문 아래로 이동한다. 변수 num에 저장된 값이 0이면 반복 작업을 끝낸다.
- 마지막 출력문인 printf( ) 문에 의해서 sum을 출력한다.

4 while 문과 do~while 문의 비교

① 두 명령문의 차이를 순서도로 나타내면 다음과 같다.

while 문	do~while 문
	
조건식을 검사하여 조건식을 만족하면 [처리 문장]을 반복 수행, 조건식을 만족하지 않으면 [처리 문장]을 수행하지 않고 벗어남.	먼저 처리 문장을 수행한 후, 조건식을 비교함. 조건식을 만족하면 [처리 문장]을 반복 수행하고, 만족하지 않으면 [처리 문장]을 수행하지 않고 벗어남.

■ 반복의 조건에서 0이 아닌 정수를 입력한 경우, 0이 아닌 모든 수를 true로 인식하기 때문에 반복의 조건은 참이 된다.

② while 문의 경우 조건식을 만족하지 않는 경우 [처리 문장]을 한 번도 실행하지 않을 수도 있다. 그러나 do~while 문의 경우 적어도 한 번은 [처리 문장]을 수행하게 된다.

EBS tip

5 while 문의 중첩

① while 문 안에 또 다른 while 문이 포함된다.
② 바깥쪽 while 문의 조건이 참인 동안 내부 while 문을 반복 수행한다.

예 while 문이 중첩된 프로그램

프로그램 소스 코드	실행 결과

#include <stdio.h>
void main( ){
  int k, m;
  k = 0, m = 0;
  while(k <= 2){
  printf("--- 현재의 k값 : %2d\n", k);
  while(m <= 3){
  printf("현재의 m값 : %2d\n", m);
  m++;
  }
  k++;
  m = 0;
  }
}
	--- 현재의 k값 : 0
	현재의 m값 : 0
	현재의 m값 : 1
	현재의 m값 : 2
	현재의 m값 : 3
	--- 현재의 k값 : 1
	현재의 m값 : 0
	현재의 m값 : 1
	현재의 m값 : 2
	현재의 m값 : 3
	--- 현재의 k값 : 2
	현재의 m값 : 0
	현재의 m값 : 1
	현재의 m값 : 2
	현재의 m값 : 3

[프로그램 해설]

- 바깥쪽 while 문에서 k = 0 인 경우, m = 0, 1, 2, 3으로 증가하면서 값을 출력한다.
- m값이 4가 된 경우 내부 while 문을 벗어나게 되며, k = 1이 되어 m = 0, 1, 2, 3으로 증가하면서 값을 출력한다.
- k값이 3이 되면 중첩 while 문을 벗어난다.

예 중첩 while 문을 이용하여 2단부터 9단까지 구구단을 출력하는 프로그램

프로그램 소스 코드	실행 결과

#include <stdio.h>
void main( ){
  int j, k = 2;
  while(k <= 9){
  j = 1;
  while(j <= 9){
  printf("%d * %d = %2d\n", k, j, k*j);
  j++;
  }
  k++;
  }
}
	2 * 1 = 2
	2 * 2 = 4
	2 * 3 = 6
	2 * 4 = 8
	2 * 5 = 10
	2 * 6 = 12
	2 * 7 = 14
	2 * 8 = 16
	2 * 9 = 18
	3 * 1 = 3
	3 * 2 = 6
	:

[프로그램 해설]

- while(k <= 9)는 2단부터 9단까지 출력하기 위한 반복문이다.
- while(j <= 9)는 각 단을 1부터 9까지 반복하면서 구구단을 출력한다.
- 예를 들어, k = 2인 경우 j는 1부터 9까지 증가하면서 반복문을 수행하여 2단의 구구단을 출력한다.
- k++은 2단을 완료하면 3단, 4단……9단까지 1씩 증가하기 위한 변수이다.

기출 모의고사

정답 및 해설 pp. 8~9

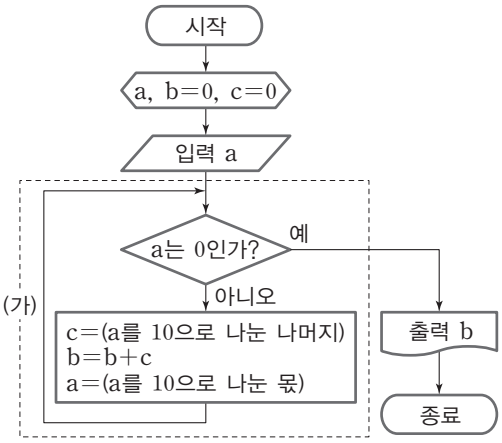
1 다음 프로그램에서 18을 입력했을 때 출력되는 값이 아닌 것은?

```
#include <stdio.h>
void main( ) {
 int n, a = 2;

 printf("정수 입력=");
 scanf("%d", &n);

 while(a < n){
  if (n % a == 0) {
      printf("%d\n", a);
  }
  a = a + 1;
 }
}
```

① 2 ② 3 ③ 6
④ 8 ⑤ 9

2011학년도 대수능

[2~3] 다음 순서도를 보고 물음에 답하시오. (단, a, b, c는 정수형 변수이다.)



2 위 순서도의 (가) 영역을 프로그램으로 구현할 때 옳은 것은?

① while(a==0){
c=a%10; b=b+c; a=a/10;
}

② while(a!=0){
c=a%10; b=b+c; a=a/10;
}

③ while(1){
c=a%10; b=b+c; a=a/10;
}

④ do{
c=a%10; b=b+c; a=a/10;
}while(0);

⑤ do{
c=a%10; b=b+c; a=a/10;
}while(a==0);

3 위 순서도에서 변수 a에 1752를 입력했을 때 출력되는 변수 b의 값으로 옳은 것은?

① 12 ② 15 ③ 21
④ 52 ⑤ 70

4 [알고리즘]은 1부터 10까지의 합을 구하는 과정이다. 점선 부분을 프로그램 코드로 나타내면?

[알고리즘]
ㄱ. 수를 저장할 두 공간 a와 s를 준비하고 각각 0을 저장한다.
ㄴ. a에 들어있는 수가 10이면 반복 작업을 끝낸다.
ㄷ. a에 들어있는 값을 1 증가시킨다.
ㄹ. s에 들어있는 값과 a에 들어있는 값을 더하여 s에 저장한다.
ㅁ. ㄴ 단계로 이동한다.

① a = a + 1;
s = s + a;
② a = 1;
s = a;
③ a = s + 1;
a = s + a;
④ a = a − 1;
s = s + a;
⑤ a = a + 1;
s + a = s;

5 [입력 조건]에 따라 프로그램을 실행하였을 때 출력 결과는?

[입력 조건] 10진수 = 4572

```
#include <stdio.h>
void main( ){
  int k, num;
  printf("10진수=");
  scanf("%d", &num);
  while(num) {
    k = num%10;
    num = num/10;
    printf("%d", k);
  }
}
```

① 2 ② 4 ③ 18
④ 2754 ⑤ 4572

6 다음 프로그램을 실행하여 출력되는 수들에 대한 설명으로 옳은 것은?

```
#include <stdio.h>
void main( ){
 int cnt;
 printf("세자리 자연수 입력\n");
 scanf("%d", &cnt);
 do{
   if (cnt%4 == 0)
        printf("%d", cnt);
   cnt=cnt-1;
 }while(cnt>99);
}
```

① 입력받은 수
② 1부터 100 사이의 4의 배수
③ 입력받은 수부터 100까지의 수
④ 입력받은 수부터 100까지의 4의 배수
⑤ 입력받은 수부터 100까지 4의 배수가 아닌 수

7 다음 프로그램에서 입력값이 12일 때 출력 결과는?

```
#include <stdio.h>
void main( ) {
  int k, m, sum;
  sum = 0;
  scanf("%d", &k);

  for(m = 1; m <= k; m++)
       if (k % m == 0)
           sum = sum + m;
  printf("%2d", sum);
}
```

① 12 ② 14 ③ 16
④ 24 ⑤ 28

8 다음 프로그램을 실행했을 때 출력되지 않는 값은?

```
#include <stdio.h>
void main( ){
     int a;
     a = 1;
     while(a < 30){
        if (a % 5 == 0)
            printf("%3d", a);
        a = a + 1;
     }
}
```

① 10 ② 15 ③ 20
④ 25 ⑤ 30

9 다음 프로그램에서 출력값의 계산식은?

```
#include <stdio.h>
void main( ){
 int a, b, k;
 a = 1;
 k = 0;
 while(a <= 5){
     b = 1;
     while(b <= a){
         k = k + b;
         b = b + 1;
     }
   a = a + 1;
 }
 printf("%2d", k);
}
```

① 1 + 2 + 3 + 4 + 5
② 1 * 2 * 3 * 4 * 5
③ 1+(2+2)+(3+3+3)+(4+4+4+4)+(5+5+5+5+5)
④ 1+(1+2)+(1+2+3)+(1+2+3+4)+(1+2+3+4+5)
⑤ 1*(1+2)*(1+2+3)*(1+2+3+4)*(1+2+3+4+5)

10 다음 프로그램의 실행 결과는?

```
#include <stdio.h>
void main( ){
  int a, k, m;
  k = 25;
  m = 4;
  a = 0;
  while(k >= m){
      k = k - m;
      a = a + 1;
  }
  printf("%d %d", k, a);
}
```

① 1 4　② 1 6　③ 2 4
④ 4 2　⑤ 6 1

11 다음 프로그램에서 입력값이 15일 때 실행 결과는?

```
#include <stdio.h>
void main( ){
  int a, b;
  printf("수 입력:");
  scanf("%d", &a);
  while( a!= 0){
     b = a % 4;
     printf("%2d", b);
     a = a / 4;
  }
}
```

① 1 2　② 2 3　③ 3 3
④ 4 2　⑤ 4 3