학습목표

- for 문에 대한 개념 및 사용법에 대해 설명할 수 있다.
- 반복문을 중단하기 위한 명령어를 사용할 수 있다.

EBS tip

1 for 문

① 반복을 시작할 초기값과 반복 조건, 증감값이 주어진 경우에 많이 사용한다.

② 기본 형식

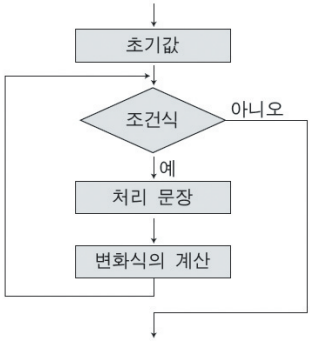
```
for(초기값; 조건식; 증감값)
{
    :
처리 문장;
    :
}
```

- 초기값 : for 문이 실행되면서 처음 한번만 수행된다. 콤마(,)를 사용하여 여러 값을 쓸 수 있다.
- 조건식 : 반복문을 수행하기 위해 검사하는 것으로 이 값이 거짓이면 for 문을 끝낸다.
- 증감값 : 제어 변수의 변경이나 갱신을 위한 것으로 콤마(,)를 사용하여 여러 식을 쓸 수 있다.

③ 실행 과정은 다음과 같다.

- 제어 변수를 초기화한다.
- 제어 변수의 조건식을 검사한다.
- 제어 조건이 참이면 처리 문장을 실행하고 제어 변수를 증감값에 따라 변화시킨다.

④ 순서도



⑤ 유의할 점

- 괄호를 생략할 수 없다. 괄호 안에 수식을 구분하는 2개의 세미콜론(;)은 반드시 있어야 한다. 괄호 안의 수식은 생략할 수 있다.
  예 for(; ;) ← 조건이 참이므로 무한히 반복
- 처리 문장이 여러 개이면 블록({ })으로 묶어 준다.

for문의 실행 과정

```
    ①     ②     ④
for(a = 1; a <= 5; a++)
   ③ sum = sum + a;
```

① 초기식을 수행한다.
② 조건식을 검사한다(조건식이 거짓이면 반복 종료).
③ 조건식이 참이면 처리할 문장을 수행한다.
④ 증감식을 수행한다.

블록으로 묶었을 때의 차이점

```
for(a = 1; a <= 5; a++)
   sum = sum + a;
   printf("%2d", sum);
```

– 출력 결과
15

```
for(a = 1; a <= 5; a++) {
   sum = sum + a;
   printf("%2d", sum);
}
```

– 출력 결과
1 3 6 10 15

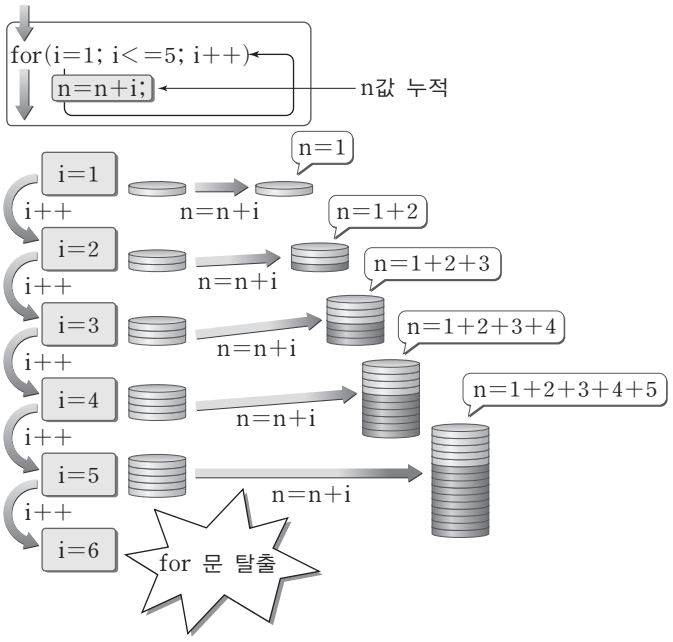
예 1부터 5까지의 합을 구하는 프로그램

프로그램 소스 코드	실행 결과

#include <stdio.h>
void main(){
  int i, n = 0;
 	
  for(i = 1; i <= 5; i++)
  n = n + i;
 	
  printf("1부터 5까지의 합:%2d\n", n);
}
	1부터 5까지의 합: 15

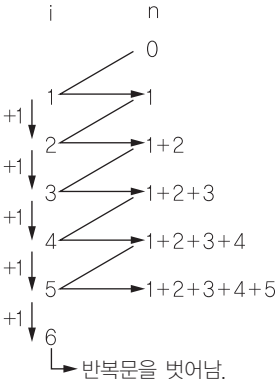
[프로그램 해설]

- i = 1로 초기화함.
- i <= 5의 조건을 검사하여 참이면 n = n + i;을 수행하게 됨.
- n = n + i;을 수행한 후 증가값에 의해 i를 1 증가시킴.
- i <= 5를 만족하는 동안 반복문을 수행하며, i 값이 6이 되면 반복문을 벗어나게 됨.
- 프로그램의 실행 과정을 살펴보면 다음과 같다.



EBS tip

◉ 값의 변화 과정



2 for 문과 while 문의 비교

```
for(초기값; 조건식; 증감값)
처리 문장;
```

⬇ while 문으로 바꾸면

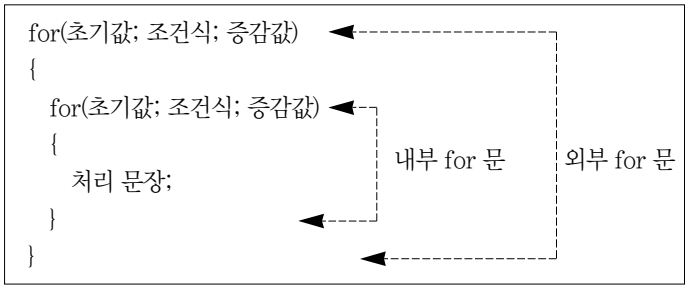
```
초기값;
while(조건식)
{
  처리 문장;
  증감식;
}
```

while 문은 보통 특정 조건을 만족하는 동안 반복해야 할 때 주로 사용되고, for 문은 특정 횟수만큼 반복 수행(예: 100번, 200번)하거나 어떤 범위의 수(예: 1부터 100까지)에 대해 같은 연산을 할 때 주로 사용된다.

3 for 문의 중첩

① for 문 내에 또다른 for 문이 중첩되어 있는 경우를 말한다.
바깥쪽 for 문의 조건을 만족하는 동안 내부 for 문을 반복 실행한다.

② 기본 형식



예 중첩된 for 문의 예제 프로그램

프로그램 소스 코드	실행 결과

#include <stdio.h>
void main( )
{
int k, m;
 	
for(k = 1; k <= 3; k++){
printf("-- 현재 k는 %d\n", k);
for(m = 1 ; m <= 2 ; m++)
printf("현재 m의 값 : %d\n", m);
}
}
	-- 현재 k는 1
	현재 m의 값 : 1
	현재 m의 값 : 2
	-- 현재 k는 2
	현재 m의 값 : 1
	현재 m의 값 : 2
	-- 현재 k는 3
	현재 m의 값 : 1
	현재 m의 값 : 2

[프로그램 해설]

• k = 1부터 시작함. k <= 3 조건을 만족하므로 반복문을 수행하게 된다.

EBS tip

◉ for 문을 while 문으로 변경

① for(a = 0; a < 3; a++) {
printf("프로그래밍");
}

② a = 0;
while(a < 3) {
printf("프로그래밍");
a++;
}

EBS tip

- m = 1에서 2까지 반복 수행하면서 m값을 출력하게 된다.
- k값을 1씩 증가시키면서 내부 반복문을 수행하게 된다.
- k값이 4가 되면 반복문을 빠져 나오게 된다.

예 역삼각형 형태의 '*' 를 출력하는 프로그램

프로그램 소스 코드	실행 결과

#include <stdio.h>
void main( ){
  int a, b, k;
  for (a = 3; a >= 1 ; a--){
  k = a * 2 – 1;
  for (b=1; b <= k ; b++)
  printf("*");
  printf("\n");
  }
}
	*****
	***
	*

[프로그램 해설]

- 바깥쪽 for 문에서 제어 변수 a가 3부터 1까지 –1씩 감소한다.
- k값은 5, 3, 1의 값을 갖게 된다.
- 안쪽 for 문에서 제어 변수 b는 1부터 k까지 변화하면서 "*"를 출력한다.

4 break 문

① for 문, while 문, do~while 문 등을 사용한 반복 문이나 switch 문으로부터 빠져 나올 때 사용한다.

② 기본 형식

```
break;
```

③ 유의할 점 : 중첩된 제어문에서 사용되었을 때는 현재 수행 중인 제어문만 빠져 나온다.

◉ 중첩된 제어문에서의 사용 예

```
for(a = 1; a <= 5; a++) {
    for(b = 1; b <= 5; b++){
        cnt++;
        if (a + b ==5) break;
    }
}
```

for 문 하나만 탈출함.

예 20 이하의 자연수에서 일정한 간격의 수를 출력하는 프로그램

프로그램 소스 코드	실행 결과

#include <stdio.h>
void main( ){
  int cnt;
 	
  for(cnt = 1; cnt <= 100 ; cnt = cnt + 5){
  if (cnt > 20) break;
  printf("%d", cnt);
  }
}
	1 6 11 16

◉ 수행 과정

cnt
1
+5 ↓
6
+5 ↓
11
+5 ↓
16
+5 ↓
21
↳ break 문에 의해 빠져 나옴.

[프로그램 해설]

- 1부터 100까지 1, 6, 11 등과 같이 5씩 증가하면서 출력하되, cnt가 20보다 크면 break 문을 만나므로 for 문을 빠져 나온다.

EBS tip

5 continue 문

① for 문, while 문, do~while 문 등의 반복문에서 일정 부분을 건너 띄고 반복 수행할 때 사용한다.

② 기본 형식

```
continue;
```

③ 유의할 점 : break처럼 밖으로 빠져 나가지 않고 프로그램의 제어가 루프의 조건식을 검사하는 부분으로 되돌아간다. 즉 for 문과 while 문은 맨 처음으로 가고 do~while 문에서는 맨 뒤로 간다.

예 1부터 10사이의 홀수를 출력하는 프로그램

프로그램 소스 코드	실행 결과

#include <stdio.h>
void main(){
int cnt;
for (cnt = 1; cnt <= 10; cnt++){
if(cnt % 2 == 0) continue;
printf("%d", cnt);
}
}
	1 3 5 7 9

[프로그램 해설]

- for 안에 있는 if 문의 조건식을 보면 cnt%2==0일 때 continue가 실행된다. 즉, cnt%2가 0일 때는 cnt가 짝수인 조건을 만족하게 되어 continue; 문이 실행된다. 따라서 printf 문에 의한 출력문은 무시된다. 실행 결과는 1부터 10까지의 숫자 중에서 홀수만 출력한다.

6 반복문의 선택

① 일반적으로 do~while 문 보다는 for 문이나 while 문을 사용한다.

② 인덱스 또는 카운터 변수를 사용하는 경우에는 for 문이 자연스럽다.

예 for(count=1; count<=200; count++)

③ 이 외의 경우에는 while 문을 사용하는 것이 편리하다.

기출 모의고사

정답 및 해설 p. 10

2010학년도 대수능

1 다음 프로그램에 대한 설명으로 옳은 것만을 〈보기〉에서 있는 대로 고른 것은?

```
#include <stdio.h>
void main( ){
 int a, b, sum=0;
  for(a=1; a<=3; a++) {
    for(b=a; b<=3; b++) {
      if (a+ b ! = 4 ) {
        sum=sum+ b;  (가)
       }
    }
  }
  printf("%d", sum);
}
```

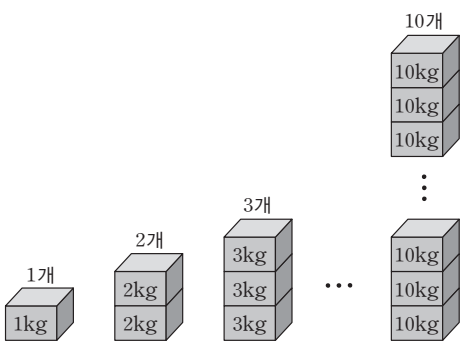
보기

ㄱ. 실행 후 출력되는 값은 9이다.
ㄴ. 점선 (가)는 프로그램 종료시까지 4번 실행된다.
ㄷ. if문(조건식)의 비교 연산은 프로그램 종료시까지 6번 실행된다.

① ㄱ ② ㄴ ③ ㄱ, ㄷ
④ ㄴ, ㄷ ⑤ ㄱ, ㄴ, ㄷ

2008년 6월 시행 평가원 모의평가

2 1kg에서부터 10kg까지의 상자를 그림과 같이 쌓았을 때, 쌓여진 상자의 총 무게를 계산하는 프로그램이다. (가)에 해당하는 명령문으로 옳은 것은?




```
#include <stdio.h>
void main( ){
  int a, b, sum=0;
  for(a=1; a<=10; a++){
    for(b=1; b<=a; b++){
          (가)
    }
}
 printf("sum = %d", sum);
}
```

① sum=a+b; ② sum=a*b;
③ sum=sum+a; ④ sum=sum+b;
⑤ sum=sum+a+b;

3 다음 프로그램의 실행 결과는?

```
#include <stdio.h>
void main( ){
  int x,y;

  for(x = 1; x <= 3; x++){
      for(y = 1; y <= 5; y++){
          if (x < y)
                printf("@");
      }
      printf("\n");
  }
}
```

① @@ @@ @@@
② @@ @@@ @@@@
③ @@@@ @@@ @@
④ @@@@ @@ @
⑤ @@@@ @@@ @

4 다음 프로그램을 실행한 후 tt에 저장되는 값을 나타내면?

```
#include <stdio.h>
void main( ){
 int cnt, tt=0;
 for(cnt=1; cnt<=99; cnt=cnt+2)
    tt=tt+cnt;
 printf("%d\n", tt);
}
```

① 1+2+3+4+ … +99
② 1+3+5+7+ … +99
③ 3+5+7+9+ … +99
④ 1+3+5+7+ … +101
⑤ 3+5+7+9+ … +101

5 다음 프로그램을 실행하여 출력되는 결과로 옳은 것은?

```
#include <stdio.h>
void main( ){
  int a, b;
  for(a = 4; a >= 1; a--){
      for(b=1; b <= a*2-1; b++)
           printf("*");
      printf("\n");
  }
}
```

① ****
***
**
*

② *
**
***
****

③ *******
 *****
  ***
   *

④
   *
  ***
 *****
****

⑤ *******
*****
***
*

6 다음 프로그램에 대한 설명으로 옳은 것을 〈보기〉에서 모두 고른 것은?

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

     a = 0, sum = 0;
     while(1){
          a = a + 1;
          sum = sum + a;
          if (sum > 10)
              break;
     }
     printf("%2d %2d", a, sum);
}
```

보기
ㄱ. while의 조건식은 참이다.
ㄴ. 변수 a, sum은 정수형 변수이다.
ㄷ. 1부터 10까지의 합을 구하여 출력하는 프로그램이다.

① ㄱ ② ㄴ ③ ㄷ
④ ㄱ, ㄴ ⑤ ㄱ, ㄷ

7 다음 프로그램의 출력 결과는?

```
#include <stdio.h>
void main( ){
  int num, cnt, k;
  cnt = 0;
  for(k = 1; k <= 5; k++ )
      for(num = k; num <= 5; num++)
        if ( (k * num) >= 7 && (k * num) <= 15 )
            cnt = cnt + 1;
  printf("%2d", cnt);
}
```

① 4 ② 5 ③ 6
④ 7 ⑤ 8