
p. 16 수정

C 프로그래밍 및 실습

6. 반복문

세종대학교

목차

- 1) 반복문
- 2) while 문
- 3) do while 문
- 4) for 문
- 5) 중첩 반복문
- 6) break 문
- 7) continue 문

제어문 (Review)

▪ 제어문

- C 언어는 순차처리언어로, 특별한 지정이 없으면, 소스 코드 첫 줄부터 차례대로 처리
- 그러나 문제 해결 위해 처리 흐름 제어 필요 → 제어문
- C언어에서는 조건문(5장), [반복문\(6장\)](#)과 같은 제어문을 제공한다.

▪ 제어문 종류

분류	종류
조건문	if 문, if~else 문, switch 문
반복문	for 문, while 문, do~while 문
기타	break 문, continue 문, goto 문, return 문

1) 반복문

■ 반복문

- 특정 조건을 만족할 때까지 문장을 반복하여 실행할 때 사용하는 제어문
- 같은 처리 과정을 여러 번 되풀이 하고자 할 때 사용
- 루프(loop)라고도 불림
- while 문, do while 문, for 문
- 반복문이 필요한 예

```
// 동일한 내용을 반복할 경우
```

```
printf("Hello World\n");
```

```
printf("Hello World\n");
```

```
printf("Hello World\n");
```

```
// 일정한 규칙으로 반복하는 작업을 수행하는 경우
```

```
int sum = 1 + 2 + 3 + 4 + 5;
```

2) while 문

- while 문

- 조건식이 참이면 **중괄호**로 둘러 쌓인 블록 안의 문장들이 반복 실행된다.
- 반복문을 종료하고자 할 때, 증감식을 이용하여 조건식을 거짓으로 만들어 준다.
- 구문

```
while( 조건식 ) {  
    조건식이 참일 때 실행되는 문장들  
    증감식;  
}
```

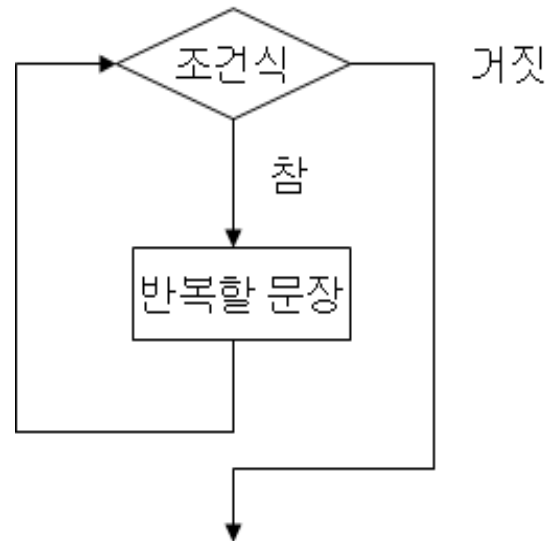
- 예) "Hello World"를 3회 반복 출력

```
int i = 1;           // 초기화  
while( i <= 3 ) {  
    printf("Hello World\n");  
    i++;  
}
```

2) while 문

- 일반적으로 while 문 do while 문은 일반적으로 **반복해야 될 횟수**를 정확하게 알지 못하지만 **반복의 조건**을 알고 있을 때 사용
 - while 문은 반복 조건을 처음에 판단
 - do while 문은 반복 조건을 나중에 판단
- 반면에 for 문은 **반복하는 횟수**가 정해진 경우에 사용
- while 문의 동작 과정

```
while( 조건식 ) {  
    반복할 문장들  
}
```



2) while 문

- while문을 이용하여 1부터 10까지의 합을 구하기

```
#include <stdio.h>

int main(void) {
    int i, sum;
    i = 1;
    sum = 0;

    while ( i <= 10 ) {
        sum += i;
        i++;
    }

    printf("1부터 10까지의 합은 %d입니다 \n", sum);
    return 0;
}
```

2) while 문

- 문자가 입력되기 전까지 입력 받은 정수의 합

```
#include <stdio.h>
```

```
int main(void) {
    int i, total;
    i = 0;
    total = 0;
```

```
    printf("문자가 입력되지 전까지 입력 받은 정수의 합 구하기 \n");
    printf("정수를 입력하세요: ");
    while ( scanf("%d", &i) ) {
        total += i;
        printf("정수를 입력하세요: ");
    }
```

```
    printf("입력한 정수의 합은 %d입니다 \n", total);
    return 0;
```

```
}
```

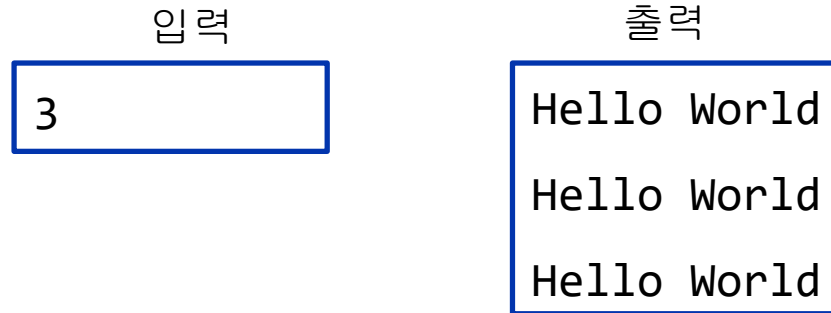
코 드	입 력	출 력	의 미
int i, j; j = scanf("%d", &i) ; printf("%d", j);	123	1	정상 입력
int i, j; j = scanf("%d", &i) ; printf("%d", j);	A	0	비정상 입력

2) while 문

- 다음 각 문장에 해당하는 while 문을 작성하시오.
 1. 변수 i가 10보다 작은 동안, "Hello World" 를 출력하고 i를 1만큼 증가
 2. 변수 i가 10보다 작은 동안, 변수 x에 0을 대입하고 i를 1만큼 감소
 3. 변수 num이 50보다 클 때까지, 숫자를 읽어 들여 num에 저장
 4. 최소값 구하기. 음수가 입력될 때 까지

2) while 문 (실습)

- (실습1) 정수를 입력 받아, 그 값만큼 "Hello World" 출력하는 프로그램을 작성하시오.



- (실습2) 알파벳 (대문자, 소문자) 순서대로 출력하는 프로그램을 작성하시오.

영어 문자	십진수
A	65
B	66
Z	90
a	97
b	98
z	122

실행 결과

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz

2) while 문 (실습)

- (실습3) 정수 변수 a가 1, 2, 3, 4, ... 증가한다. 변수 a의 합이 최초로 100 이상이 되는 때의 정수 변수 a의 값을 출력하는 프로그램을 작성하시오.

실행 결과

14

-
- (실습4) 정수를 입력 받아 그 정수에 해당하는 구구단을 출력하는 프로그램을 작성하시오.

입력

5

출력

5 * 1 = 5
5 * 2 = 10
...
5 * 9 = 45

2) while 문 (실습)

- (실습5) 사용자에게 정수 값들을 반복해서 입력 받아(종료 조건까지), 이 중에서 **가장 작은 값**을 출력하는 프로그램을 작성하시오.
 - 종료조건: **0 입력**
 - 처음 입력되는 정수는 0이 아니라고 가정하고, 종료 조건으로 입력되는 0은 비교 대상에서 제외한다.

입력 예시 1

10 2 15 7 0

출력 예시 1

2

입력 예시 2

10 2 -15 0

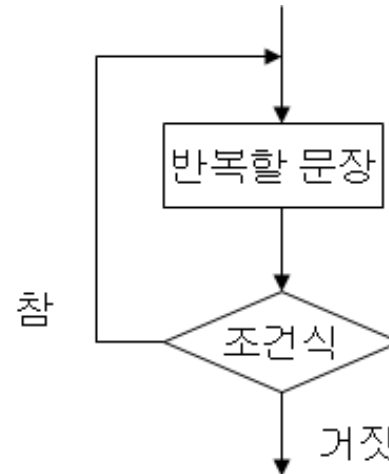
출력 예시 2

-15

3) do while 문

- do 안에 있는 문장을 먼저 한번 실행하고 나중에 while문의 조건식을 비교한다.
- 반복문 내에 있는 문장을 **최소한 한 번** 실행하고자 할 때 유용하다.
- 마지막에 세미콜론(;)을 반드시 추가해야 한다.
- 구문 및 동작과정

```
do {  
    반복할 문장들  
} while( 조건식 );
```



3) do while 문

- do while문을 이용하여 1부터 10까지의 합을 구하는 예제

```
/* do while 문을 이용하여 1부터 10까지의 합 */  
  
#include <stdio.h>  
  
int main(void) {  
    int i, sum;  
    i = 1;  
    sum = 0;  
  
    do {  
        sum += i;  
        i++;  
    } while ( i <= 10) ;      // 1+2+ ... +10  
  
    printf("1부터 10까지의 합은 %d입니다 \n", sum);  
    return 0;  
}
```

3) do while 문

- 음의 정수가 입력되기 전까지 입력 받은 정수의 합

```
#include <stdio.h>

int main(void) {
    int x, sum=0;

    do {
        sum = sum + x;    // scanf( ) 다음으로 가면 안됨
        printf("정수를 입력하세요: ");
        scanf("%d", &x);
    } while (x > 0);

    printf("입력한 정수의 합은 %d입니다 \n", sum);
    return 0;
}
```

3) do while 문

- do while문을 문자가 입력되기 전까지 입력 받은 정수의 합 예제

```
#include <stdio.h>

int main(void) {
    int i, sum;
    i = 0;
    sum = 0;

    do {
        sum += i;
        printf("정수를 입력하세요: ");
    } while ( scanf("%d", &i) );

    printf("입력한 정수의 합은 %d입니다 \n", sum);
    return 0;
}
```


3) do while 문

- do while 문 반복 횟수 구하는 예제

```
/* do while 문 반복 횟수 예제 */
#include <stdio.h>

int main(void)
{
    int count = 1;
    do {
        printf("do while 문 반복: %d\n", count);
        count++;
    } while(count < 5);
    printf("do while 문 반복 횟수:%d\n", count - 1);
}
```

3) do while 문 (실습)

- (실습6) 정수를 입력 받아, 그 값만큼 “Hello World”를 출력하시오.

입력

3

출력

Hello World
Hello World
Hello World

-
- (실습7) 정수를 입력 받아 그 정수에 해당하는 구구단을 출력하시오.

입력

5

출력

5 * 1 = 5
5 * 2 = 10
...
5 * 9 = 45

3) do while 문 (실습)

- (실습8) 1 부터 더한 값이 최초로 100 이상이 되는 a 값을 구하는 프로그램을 작성하시오.

실행 결과

14

-
- (실습9) 1 부터 더한 값이 1000 보다 작은 최대합과 자연수 a를 구하는 프로그램을 작성하시오.

실행 결과

990

44

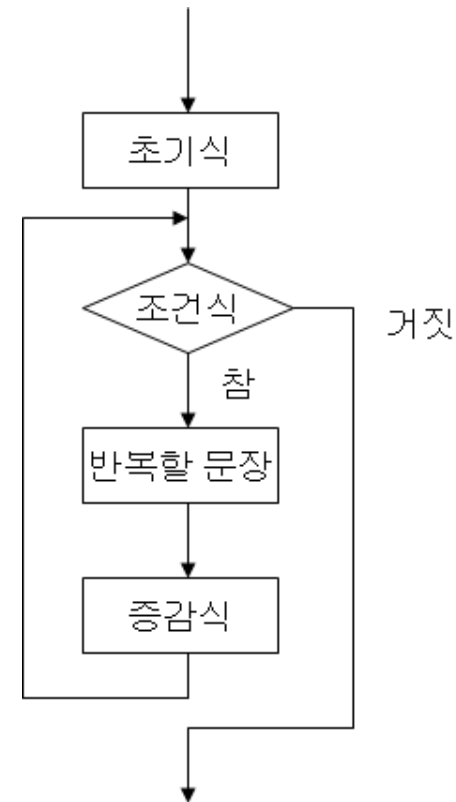
4) for 문

- 반복해야 할 문장을 수행할지 말지를 미리 검사하는 반복문이다.
- 일반적으로 반복하는 횟수가 정해진 경우에 사용한다.
- 초기문, 조건식, 증감식으로 구성된다.
- 구문 및 동작방식

```
for(초기문; 조건식; 증감식) {
```

```
}  
↑ 반복할 문장들
```

```
int i = 1;           // 초기화  
while( i <= 3 ) {  
    printf("Hello World\n");  
    i++;  
}
```



4) for 문

- for 문의 구조

- 초기문

- ✓ 초기문은 수행할 때 한 번만 실행된다.
 - ✓ 콤마(,)로 구분하여 여러 문장을 나열할 수 있다.

```
for (i=0, sum=0; i<=10 ; i++)
```

- ✓ 초기문을 빈 상태로 두어도 된다.

- 조건식

- ✓ 조건식의 결과가 참(true)인 경우에 반복이 계속된다.
 - ✓ 반복할 문장이 수행되기 전에 먼저 수행된다.
 - ✓ 논리형 변수나 논리연산만 사용 가능하다.

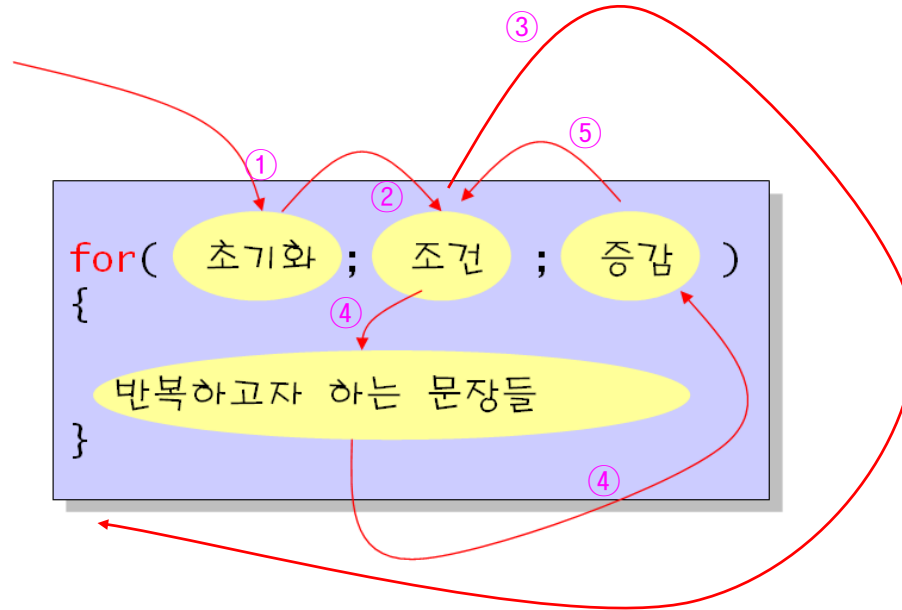
```
for (i=0; i<=10 && sum<100 ; i++)
```

- 증감식

- ✓ for 문 내의 반복할 문장이 실행된 후 증감식이 실행된다.
 - ✓ 변수의 증가 또는 감소 연산이 사용된다.

4) for 문

- for 문의 상세 동작 방식



- ① 초기화를 실행한다.
- ② 반복 조건을 나타내는 조건식을 계산한다.
- ③ 수식의 값이 거짓이면 for 문의 실행이 종료된다.
- ④ 수식의 값이 참이면 문장이 실행된다.
- ⑤ 증감을 실행하고 ②로 돌아간다.

4) for 문

- 다음 각 문장에 해당하는 for 문을 작성하시오.

1. 변수 i 의 초기값은 0이고, i 의 값이 1씩 증가하면서, i 가 10보다 작거나 같으면 반복
2. 변수 i 의 초기값은 10이고, i 의 값이 3씩 감소하면서, i 가 0보다 크거나 같으면 반복
3. 변수 i 의 초기값은 0이고, i 의 값이 $i*i+2$ 로 증가하면서, i 가 50보다 작거나 같으면 반복

4) for 문

- for 문을 이용하여 1부터 10까지의 합을 구하는 예제

```
#include <stdio.h>

int main(void) {
    int i, sum;
    sum = 0;

    for(i=1; i<=10 ; i++)
        sum += i;

    printf("1부터 10까지의 합은 %d입니다 \n", sum);
    return 0;
}
```


4) for 문

- for 문을 이용하여 팩토리얼(factorial)을 계산하는 예제

```
/* for 문을 이용하여 5! 계산 */  
  
#include <stdio.h>  
  
int main(void) {  
    int i, fac;  
    fac = 1;  
  
    for(i=1; i<=5; i++)  
        fac = fac * i;  
  
    printf("5!은 %d입니다 \n", fac);  
    return 0;  
}
```

4) for 문

- while문과 for문의 형식 비교

// while 문

초기화; ①

while(조건식) { ②

③ 반복할 문장;

④ 증감식;

}

// for 문

for(초기문; ① 조건식; ② 증감식) { ④

③ 반복할 문장;

}

4) for 문

- 무한반복

- 종료되지 않는 반복문을 무한반복(infinite loop)이라 한다.

```
while( 1 ) {  
    반복할 문장;  
}
```

```
for( ; ; ) {  
    반복할 문장;  
}
```

- **break** 문을 사용하여 무한반복에서 빠져나올 수 있다.

4) for 문

- 다음 for 문을 while 문으로 변환하시오.

1.

```
for(i=1; i<5; i++)  
    printf("%d", i);
```

2.

```
for(i=10; i>=0; i-=2)  
    printf("%d", i);
```

- 다음 while 문을 for 문으로 변환하시오.

1.

```
int i=5;  
while(i>=0){  
    printf("%d", i);  
    i--;  
}
```

2.

```
int i=0;  
while(i<=50){  
    if(i%5==0)  
        printf("%d", i);  
    i+=2;  
}
```

4) for 문 (실습)

- (실습10) 정수 1개를 입력 받아 입력 받은 정수의 팩토리얼(factorial)을 계산하는 프로그램을 작성하시오.



- (실습11) 문자 1개와 정수 1개를 입력 받아 입력 받은 정수 개수 만큼 문자를 출력하는 프로그램을 작성하시오.



여기서 □는 공백을 의미한다

4) for 문 (실습)

- (실습12) for 문을 이용하여 1부터 10까지의 홀수의 합, 짝수의 합을 구하는 프로그램을 작성하시오.

실행 결과

25
30

- (실습13) 정수를 입력 받아 그 정수에 해당하는 요일(1이면 일요일, 2이면 월요일, ..., 7이면 토요일)에 1일이 시작하는 달력을 출력하는 프로그램을 작성하시오.

- 한 달은 31일 까지 있다고 가정한다.
- 각 날짜는 공백 3개를 확보하여 사용한다.

입력

5

출력

				1	2	3
<input type="checkbox"/> 4	5	6	7	8	9	10
<input type="checkbox"/> 11	12	13	14	15	16	17
<input type="checkbox"/> 18	19	20	21	22	23	24
<input type="checkbox"/> 25	26	27	28	29	30	31

5) 중첩 반복문

- 반복문 안에 또다른 반복문을 사용하는 경우 중첩 반복문이라 한다.
- while 문 안에 while문을 겹쳐 사용하거나, for 문 안에 for 문을 사용할 수 있다.
- while 문과 for 문을 중첩하여 사용할 수도 있다.
- 중첩 for 문 구문

```
for(초기문; 조건식; 증감식) {  
    반복할 문장들1  
    for(초기문; 조건식; 증감식) {  
        반복할 문장들2  
    }  
    반복할 문장들3  
}
```

5) 중첩 반복문

```
/* 두 개의 for 문을 중첩한 구구단 프로그램 */

int i, j;
for( i = 2; i < 10; i++)
    for(j = 1; j < 10; j++)
        printf ("%d * %d = %d\n", i, j, i * j);
```

```
/* while 문, for 문을 이용한 구구단 프로그램 */

int i, j;
i = 2;
while( i < 10 ) {
    for (j = 1; j < 10; j++)
        printf ("%d * %d = %d\n", i, j, i * j);
    i++;
}
```


5) 중첩 반복문 (실습)

- (실습14) 중첩 for 문을 이용하여 다음과 같은 실행결과 1, 2, 3을 나오게 하는 프로그램을 각각 작성하시오.

✓ 결과3에서 공백 3개를 확보하여 각 숫자를 출력한다.

실행 결과 3

결과 1

```
*
**
***
****
*****
```

결과 2

```
*~~~~
**~~~
***~~
****~
*****
```

```
□□1  2  3  4  5
□□6  7  8  9
□10 11 12
□13 14
□15
```

- (실습15) 중첩 for 문을 이용하여 2 ~ 50 사이의 소수를 구하는 프로그램을 작성하시오.

✓ 각 줄에 소수 5개씩 출력하시오.

✓ 각 줄 처음에 공백이 있고, 숫자 사이에는 공백 하나가 있고, 각 줄의 끝에는 공백이 없어야 한다. 예) □2□3□5□7□11

실행 결과

```
□2 3 5 7 11
□13 17 19 23 29
□31 37 41 43 47
```

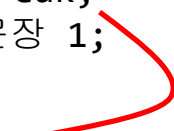
6) break 문

- 반복문의 실행 상태를 직접 제어하고자 break 문과 continue 문을 사용한다.
- break 문은 현재 사용 중인 반복문을 조건식에 관계 없이 모두 중단할 때 사용한다. 즉, 반복 루프를 빠져 나오는데 사용된다.

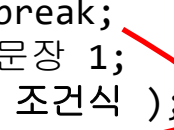
6) break 문

- break 문 동작 과정

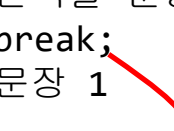
```
while( 조건식 ) {  
    반복할 문장;  
    break;  
    문장 1;  
}  
문장 2;
```



```
do {  
    반복할 문장;  
    break;  
    문장 1;  
while( 조건식 );  
문장 2;
```



```
for(초기문; 조건식; 증감식) {  
    반복할 문장;  
    break;  
    문장 1;  
}  
문장 2;
```



6) break 문

- break 문 예제

```
/* break 문의 사용 예 */  
  
#include <stdio.h>  
  
int main()  
{  
    int count;  
    for (count = 0; count < 5; count++) {  
        if (count == 3)  
            break;  
    }  
    printf("count 값은 %d 입니다\n", count);  
}
```

6) break 문 (실습)

- (실습16) 1부터 n까지의 합이 최초로 100이 넘을 때의 n값과 합을 구하는 프로그램을 작성하시오. 단, n까지의 합은 **무한반복** 안에 구현하고 합이 100을 넘으면 break 문을 이용하여 무한반복을 벗어나도록 하시오.

실행 결과

14

105

- (실습17) 입력 받은 정수의 합을 구하는 프로그램을 작성하시오. 단, **무한반복**을 사용하고 입력 받은 정수가 0이면 무한반복을 벗어나도록 하시오.

입력

1

5

3

0

출력

9


7) continue 문

- continue 문은 현재 반복을 중단하고, 다음 조건 값에 대한 처리를 수행한다.
- continue 문을 만나면 제어는 반복문의 마지막 부분으로 전달되고 다음 반복 동작이 시작된다.
 - continue 문과 반복문의 마지막 부분 사이에 있는 문장은 실행되지 않음


7) continue 문

- continue 문 동작 과정

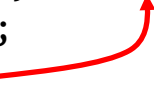
```
while( 조건식 ) {  
    반복할 문장;  
    continue;  
    문장 1;  
}  
문장 2;
```



```
do {  
    반복할 문장;  
    continue;  
    문장 1;  
} while( 조건식 );  
문장 2;
```



```
for( 초기문; 조건식; 증감식 ) {  
    반복할 문장;  
    continue;  
    문장 1;  
}  
문장 2;
```



7) continue 문

- continue 문을 이용한 1부터 10 사이의 짝수 출력 예제

```
/* continue 문의 사용 예 */  
  
#include <stdio.h>  
  
int main()  
{  
    int i;  
    printf("1부터 10사이의 짝수만 출력 합니다 \n");  
  
    for (i = 1; i <= 10; i++) {  
        if (i % 2 != 0)  
            continue;  
        printf("\n%d", i);  
    }  
}
```


7) continue 문 (실습)

- (실습18) continue 문을 이용하여 1과 10사이의 홀수의 합을 구하는 프로그램을 작성하시오.

실행 결과

25

-
- (실습19) 1과 20 사이에서 2의 배수이면서 3의 배수인 수를 제외한 모든 정수를 출력하는 프로그램을 작성하시오.

- Hint:

✓ 2의 배수인 조건 $(i\%2)==0$

✓ 3의 배수인 조건 $(i\%3)==0$

실행 결과

□ 1 2 3 4 5 7 8 9 10 11 13 14 15 16 17 19 20