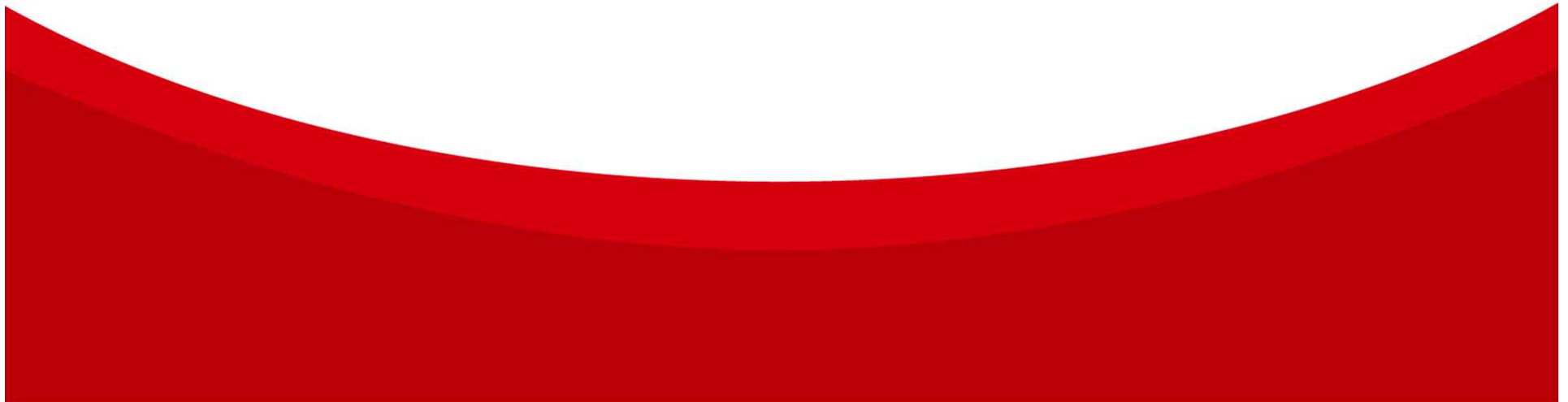


# Chapter 7

## 반복제어문 2



01

Question

10 이하의 정수를 입력받아 입력받은 정수만큼 “C언어 프로그래밍” 이라고 출력하는 프로그램을 작성하시오.

입력 예

5

출력 예

C언어 프로그래밍

C언어 프로그래밍

C언어 프로그래밍

C언어 프로그래밍

C언어 프로그래밍

## 소스

```
#include <stdio.h>

int main()
{
    int n, i;

    scanf("%d", &n);

    for (i = 1; i <= n; i++) {
        printf("C언어 프로그래밍 \n");
    }

    return 0;
}
```

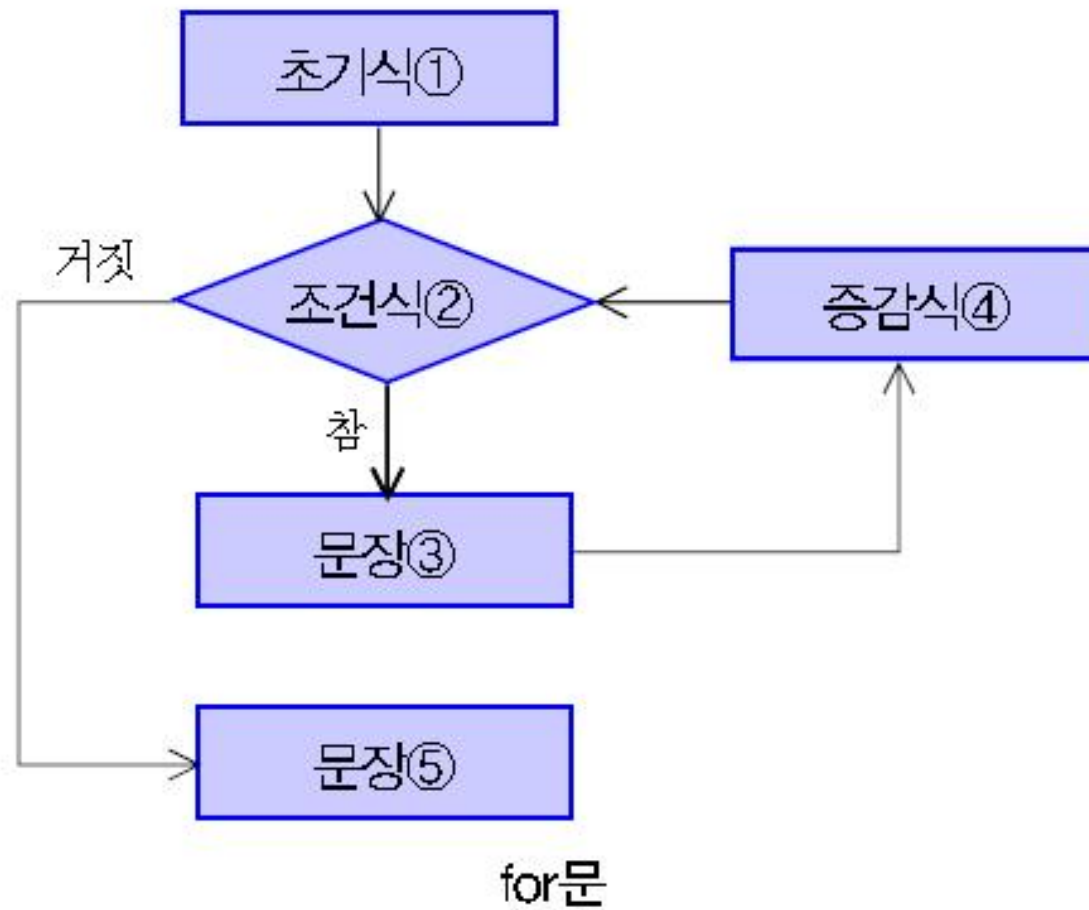
### 설명

- for문은 while문과 같이 블록 내의 문장을 반복하여 실행하는 제어문이다.
- for문의 구조는 다음과 같다.

```
for ( 초기식① ; 조건식② ; 증감식④ )  
{  
    실행문장 ③  
}  
이후 실행문장 ⑤
```

- ① 초기식 : for문을 시작할 때 처음 단 한 번만 실행되는 식으로 어떤 변수의 초기 값을 정해주는 식이다.
- ② 조건식 : 조건이 참이면 for문 블록 내의 문장을 실행하고, 거짓이면 for문을 끝내고 블록을 벗어난다. 처음 실행할 때와 for문 내의 실행문장을 모두 처리하고 증감식이 실행될 때마다 이 식을 검사한다.
- ④ 증감식 : for문 내의 실행문장이 처리될 때마다 실행하며 초기식에 지정된 변수의 값을 변화시키는 데 주로 사용된다.

- 위 for문의 실행순서는 다음과 같다.



- `for (i = 1; i <= n; i++) { }`  
변수 `i`의 초기값을 1로 지정하고 그 값이 `n`보다 커질 때까지 블록 내의 문장을 실행한다. 블록 내의 문장이 한 번 실행 완료될 때마다 `i`값을 1씩 증가시키고 다시 조건을 검사한다. 결국 이 문장은 `i`가 1부터 `n`까지 증가되면서 블록 내의 문장을 `n`번 반복하여 실행하게 된다.
- 위의 `for`문은 다음과 같이 `while`문으로 수정하여도 결과는 같다. 초기식은 `while`문 시작하기 전에, 증감식은 블록의 마지막 부분에 넣으면 같은 식이 된다.

```
i = 1;
while ( i<=n ) {
    printf("C언어 프로그래밍 %d\n", i);
    i++;
}
```

## 자가진단 1

문자를 입력받아서 입력받은 문자를 20번 반복하여 출력하는 프로그램을 작성하시오

입력 예 A

출력 예 AAAAAAAAAAAAAAAAAAAAAA



02

Question

대문자를 'A'부터 'Z'까지 차례로 출력하는 프로그램을 작성하시오.

출력 예      ABCDEFGHIJKLMNOPQRSTUVWXYZ

## 소스

```
#include <stdio.h>

int main()
{
    char i;

    for (i = 'A'; i <= 'Z'; i++) {
        printf("%c", i);
    }

    printf("\n");

    return 0;
}
```

## 설명

- `char i;`  
for문에서 영문자를 사용하기 위해 `i`를 문자형으로 선언한다.
- `for (i = 'A'; i <= 'Z'; i++)`

`i = 'A'` : `i`변수에 초기값으로 'A'를 저장한다.

`i <= 'Z'` : `i`변수의 값이 'Z'보다 작은 동안 블록 내의 문장을 반복하여 실행한다.

`i++` : 블록 내의 문장이 실행완료될 때마다 `i`의 값을 1씩 증가시킨다. ('A'에서 1을 증가하면 'B'가 된다.)

## 자가진단 2

10부터 20까지의 숫자를 차례대로 출력하는 프로그램을 작성하시오.

출력 예     10 11 12 13 14 15 16 17 18 19 20

03

Question

1부터 20까지의 홀수를 차례대로 출력하는 프로그램을 작성하시오.

출력 예     1 3 5 7 9 11 13 15 17 19

## 소스1

```
#include <stdio.h>

int main()
{
    int i;

    for (i = 1; i <= 20; i++) {
        if (i % 2 == 1) {
            printf("%d ", i);
        }
    }

    printf("\n");

    return 0;
}
```

## 설명

- `for (i = 1; i <= 20; i++)`  
i의 값을 1부터 20까지 1씩 증가시키면서 반복하여 실행한다.
- `if (i % 2 == 1) {`  
    `printf ("%d ", i);`  
    `}`

i의 값이 홀수이면 출력한다. 결과적으로 1부터 20까지 차례대로 실행하면서 홀수인 경우에만 출력을 한다.

## 소스2

```
#include <stdio.h>

int main()
{
    int i;

    for (i = 1; i <= 20; i += 2) {
        printf("%d ", i);
    }

    printf("\n");

    return 0;
}
```



## 설명

- for (i = 1; i <= 20; i += 2)

i의 값을 1부터 20까지 2씩 증가시키면서 반복하여 실행한다.  
2씩 증가하게 되므로 홀수만 실행이 된다. (1, 3, 5, 7, 9, ...)  
따라서 블록 내에서 홀수인지 아닌지 판단하는 if문은 필요가 없다.  
i += 2는 i = i + 2와 같다.

### 자가진단 3

한 개의 수를 입력받아 1부터 입력받은 수까지의 짝수를 차례대로 출력하는 프로그램을 작성하시오. (입력 받는 수는 2이상 50 이하의 정수이다.)

입력 예 10

출력 예 2 4 6 8 10

04

Question

자연수를 입력받아서 1부터 입력받은 수까지의 합을 출력하는 프로그램을 작성하시오.

입력 예      10

출력 예      55

소스

```
#include <stdio.h>

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

    scanf("%d", &n);

    for (i = 1; i <= n; i++) {
        sum += i;
    }

    printf("%d \n", sum);

    return 0;
}
```

## 설명

- `int i, n;`  
for문에서 사용할 변수 `i`와 입력받을 변수 `n`을 선언한다. `i`는 for문 안에서 초기값을 지정하고 `n`은 입력을 받으면서 값이 지정되므로 초기값은 지정하지 않아도 된다.
- `int sum = 0;`  
합계를 저장할 변수 `sum`을 선언하고 초기값으로 0을 저장한다. 처음에는 아무것도 더하지 않은 상태이므로 반드시 0으로 초기화해야 한다.
- `for (i = 1; i <= n; i++) {`  
    `sum += i;`  
}

`i`값을 1부터 `n`까지 1씩 증가시켜 가면서 그 값을 `sum`에 누적하여 더한다.  
`sum += i;` 는 `sum = sum + i;`와 같다.

## 자가진단 4

정수를 입력받아서 입력받은 정수부터 100까지의 합을 출력하는 프로그램을 작성하시오. (입력받는 정수는 100 이하의 자연수이다.)

입력 예      95

출력 예      585

05

Question

10개의 정수를 입력받아 그 수들 중 짝수의 개수가 몇 개인지 출력하는 프로그램을 작성하시오

입력 예      15 22 3 129 66 81 35 1 46 888

출력 예      입력받은 짝수는 4개입니다.

소스

```
#include <stdio.h>

int main()
{
    int i, num;
    int count = 0;
    for (i = 1; i <= 10; i++) {
        scanf("%d", &num);
        if (num % 2 == 0) count++;
    }
}
```



## Chapter 7

### 반복제어문 2

---

```
    printf("입력받은 짝수는 %d개입니다. \n", count);  
    return 0;  
}
```

## 설명

- `int count = 0;`  
    짝수의 개수를 저장할 변수를 선언하고 0으로 초기화한다.
- `for (i = 1; i <= 10; i++) {`  
    `scanf("%d", &num);`  
    `if (num % 2 == 0) count++;`  
}

i가 1부터 10을 벗어날 때까지 1씩 증가하면서 블록 내의 문장 즉, 입력을 받고 입력 받은 값이 짝수이면 count를 1 증가시키는 작업을 10번 반복하여 실행하게 된다.

- `printf("입력받은 짝수는 %d개 입니다. \n", count);`  
for문이 끝나면 count에는 입력받은 짝수의 개수가 저장되어 있다.

## 자가진단 5

10개의 정수를 입력받아 3의 배수의 개수와 5의 배수의 개수를 각각 출력하는 프로그램을 작성하시오

입력 예      10 15 36 99 100 19 46 88 87 13

출력 예      Multiples of 3 : 4  
              Multiples of 5 : 3

- *hint*

3의 배수의 개수와 5의 배수의 개수를 저장할 변수를 각각 선언하여 처리한다.

if(3의 배수) ~ else if(5의 배수)와 같이 사용하게 되면 3의 배수이면서 5의 배수인 경우(예:15) 3의 배수만 증가하고 5의 배수는 누락이 된다. 따라서 각각 if문으로 확인해야 한다.

06

Question

5명의 성적을 입력받아서 총점과 평균을 출력하는 프로그램을 작성하시오. (평균은 반올림하여 소수 첫째자리까지 출력한다.)

입력 예      90 85 100 66 88

출력 예      총점 : 429  
              평균 : 85.8

소스

```
#include <stdio.h>

int main()
{
    int i, score;
    int sum = 0;
    double avg;

    for (i = 1; i <= 5; i++) {
        scanf("%d", &score);
        sum += score;
    }
    avg = (double)sum / 5;

    printf("총점 : %d \n", sum);
    printf("평균 : %.1f \n", avg);

    return 0;
}
```

## 설명

- ```
for (i = 1; i <= 5; i++) {  
    scanf("%d", &score);  
    sum += score;  
}
```

블록 내의 문장을 5번 반복하여 실행한다. 성적(score)을 입력받고 입력받은 값을 sum에 누적하여 더한다.

- ```
avg = (double)sum / 5;
```

평균을 구한다. sum이 정수(int)이므로 실수(double)로 형변환을 해야 한다.  

```
avg = sum / 5.0;
```

과 같이 써도 된다.

## 자가진단 6

10 이하의 과목수  $n$ 을 입력받은 후  $n$ 개 과목의 점수를 입력받아서 평균을 구하여 출력하고 평균이 80점 이상이면 "pass", 아니면 "fail"이라고 출력하는 프로그램을 작성하시오. 평균은 반올림하여 소수 첫째자리까지 출력한다.

입력 예

4

75 80 85 90

출력 예

avg : 82.5

pass

07

Question

아래 출력예와 같이 출력하는 프로그램을 작성하시오.

출력 예

```
1 1 1 1 1
2 2 2 2 2
3 3 3 3 3
4 4 4 4 4
5 5 5 5 5

1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
```



#### 소스

```
#include <stdio.h>

int main()
{
    int i, j;

    for (i = 1; i <= 5; i++) {
        for (j = 1; j <= 5; j++) {
            printf("%d ", i);
        }
        printf("\n");
    }
    printf("\n");

    for (i = 1; i <= 5; i++) {
        for (j = 1; j <= 5; j++) {
            printf("%d ", j);
        }
        printf("\n");
    }

    return 0;
}
```

## 설명

- 반복문 내에 또 다른 반복문이 들어있는 것을 다중반복문이라 한다.

- ```
for (i = 1; i <= 5; i++) {  
    for (j = 1; j <= 5; j++) {  
        printf("%d ", i);  
    }  
    printf("\n");  
}
```

밖에 있는 for문에서 i가 1일 때의 동작을 살펴보자.

- ① for문 내의 또 다른 for문에서 j가 1부터 5까지 5번 반복 실행하여 i값 1을 5번 출력하게 된다.
- ② for문을 빠져나오면 줄을 바꾸어 줌으로써 첫 번째 줄의 출력이 끝난다.
- ③ 다시 바깥 for문의 처음으로 돌아가서 i값을 증가시킨다.  
i값이 2가 되면 다시 위와 같은 작업을 반복하여 2를 5번 출력하고 줄을 바꾼 후 i를 3으로 바꾼다. 이러한 작업이 반복되다가 i값이 5가 되면 5를 5번 출력하고 줄을 바꾼 후 i를 6으로 증가시키는데 이 때  $i \leq 5$ 의 조건이 거짓이 되므로 바깥의 for문을 벗어나게 된다.

- 위 문장에서 i는 5줄을 반복하게 하는 반복횟수를 나타내는 역할과 각 줄마다에 출력할 값을 정해주는 두 가지 역할을 하고 있으나 j는 단순히 5번을 반복하게 하는 역할만을 하고 있다.
- ```
for (i = 1; i <= 5; i++) {  
    for (j = 1; j <= 5; j++) {  
        printf("%d ", j);  
    }  
    printf("\n");  
}
```

이 구조는 위의 경우와 똑같지만 출력할 때  $j$ 값을 출력하는 것이 다르다.  $i$ 가 1일 때  $j$ 가 1부터 5까지 5번 실행되면서  $j$ 값을 출력하게 되므로 1부터 5까지가 차례대로 출력되게 된다.

위의 경우는  $i$ 는 단순히 5줄을 반복하는 횟수를 지정하는 역할만 하지만  $j$ 는 5번 반복하는 일과 출력값을 나타내는 두 가지 역할을 하고 있다. 이렇게 for문에 사용되는 변수는 횟수를 나타내기 위한 역할을 하면서 동시에 어떠한 작업을 위해 값으로 사용되는 두 가지 역할이 있음을 잘 이해하여야 한다.

## 자가진단 7

아래와 같이 출력되는 프로그램을 작성하시오.

출력 예

```
2 3 4 5 6
3 4 5 6 7
4 5 6 7 8
5 6 7 8 9
6 7 8 9 10
```

• *Hint*

i와 j값을 둘 다 활용하여 식을 만들어 본다.

08

Question

아래와 같이 2단부터 4단까지의 구구단을 출력하는 프로그램을 작성하시오.

출력 예

$2 * 1 = 2$	$3 * 1 = 3$	$4 * 1 = 4$
$2 * 2 = 4$	$3 * 2 = 6$	$4 * 2 = 8$
$2 * 3 = 6$	$3 * 3 = 9$	$4 * 3 = 12$
$2 * 4 = 8$	$3 * 4 = 12$	$4 * 4 = 16$
$2 * 5 = 10$	$3 * 5 = 15$	$4 * 5 = 20$
$2 * 6 = 12$	$3 * 6 = 18$	$4 * 6 = 24$
$2 * 7 = 14$	$3 * 7 = 21$	$4 * 7 = 28$
$2 * 8 = 16$	$3 * 8 = 24$	$4 * 8 = 32$
$2 * 9 = 18$	$3 * 9 = 27$	$4 * 9 = 36$

소스

```
#include <stdio.h>

int main()
{
    int i, j;

    for (i = 1; i <= 9; i++) {
        for (j = 2; j < 5; j++) {
            printf("%d * %d = %2d  ", j, i, j * i);
        }
        printf("\n");
    }
    return 0;
}
```



## 설명

- 각 줄마다 바뀌는 것이 무엇인지 확인해 보자.  
첫 번째 줄은 곱하는 값이 1이고 두 번째 줄은 곱하는 값이 2, 세 번째 줄은 3, ...  
이렇게 곱하는 값이 1부터 9까지 달라지게 된다. 따라서 곱하는 값  $i$ 를 1부터 9까지 먼저 정해주었다.
- $i$ 가 1일 때의 동작을 살펴보자.  
왼쪽부터 차례대로 2단부터 4단까지 출력을 해야 한다.  
 $j$ 를 2부터 4까지 증가시키면서  $j$ 단 \* 1을 출력한다.
- for문이 끝나면 줄을 바꾼 다음에  $i$ 를 2로 바꾸어 다시 2단부터 4단까지 \* 2를 출력한다. 이렇게 반복하여 \* 9가 끝나면  $i$ 가 10이 되어 프로그램을 끝내게 된다.

## 자가진단 8

구구단의 일부를 다음과 같이 출력하는 프로그램을 작성하시오  
(곱셈의 결과는 오른쪽으로 맞추어 출력을 하고 결과 값 사이의 공백은 3칸으로 한다.)  
출력형식 예)  $2 * 010 = 002000$  (0는 공백을 나타내는 것임)

출력 예	$2 * 1 = 2$	$2 * 2 = 4$	$2 * 3 = 6$	$2 * 4 = 8$	$2 * 5 = 10$
	$3 * 1 = 3$	$3 * 2 = 6$	$3 * 3 = 9$	$3 * 4 = 12$	$3 * 5 = 15$
	$4 * 1 = 4$	$4 * 2 = 8$	$4 * 3 = 12$	$4 * 4 = 16$	$4 * 5 = 20$

# Thank You!!!

