실습 3주차

반복문

같은 명령어를 반복하고 싶을 때, 혹은 특정 명령어를 반복적으로 호출할 때
반복문을 사용하면 함수 호출문의 호출 빈도수를 낮출 수 있고, 프로그램의 코드 양도 줄일 수 있다.

- while 문
- do ~ while 문
- for 문

반복문

- 증가, 감소 연산자

→ ++num : num 값을 1 증가시킨 후, 속한 문장의 나머지를 진행 (선 증가, 후 연산)
→ num++ : 속한 문장을 먼저 수행한 후, num 값을 1 증가 (선 연산, 후 증가)
→ --num : num 값을 1 감소 후, 속한 문장의 나머지를 수행 (선 감소, 후 연산)
→ num-- : 속한 문장을 먼저 수행한 후, num 값을 1 감소 (선 연산, 후 감소)

- 복합 대입 연산자

→ num1 += num2
: num1 = num1 + num2
→ num1 -= num2
: num1 = num1 - num2
→ num1 *= num2
: num1 = num1 * num2
→ num1 /= num2
: num1 = num1 / num2
→num1 %= num2
: num1 = num1 % num2

while

```c
#include <stdio.h>

int main(void) {

    //while문의 기본 구조
    while (조건식) {
        //조건식이 참일 경우 while문 안의 명령어 수행한다.
        //조건식이 거짓일 경우 while문 안의 명령어는 수행되지 않는다.
    }

    return 0;
}
```

* 조건식이 “1”이라고 되어 있을 경우

-> while(1) { … }

1은 ‘참’을 의미 함으로 while문 안의 명령어를 항상 수행하고, 무한 반복하게 된다.

무한 루프(loop)에서 빠져 나오려면 Ctrl+C를 입력하면 종료된다.

무한 루프에 빠지지 않도록 조건식을 잘 세우는 것이 중요하다.

(단, 필요에 의해 의도적으로 무한 루프를 생성하는 경우도 있다.)

while

예제

```c
#include <stdio.h>

int main(void) {
    int num = 0;

    while (num < 5)
    {
        printf("num : %d₩n", num);
        num++;
    }
    return 0;
}
```

결과

```
num : 0
num : 1
num : 2
num : 3
num : 4
```

do ~ while

```c
#include <stdio.h>

int main(void) {

    //do~while문의 기본 구조
    do
    {
        //while 조건식을 확인하기 앞서 우선 첫번째 루프에서 일단 실행한다.
    } while (조건식);
    // do에 나오는 명령어가 실행 된 후, while의 조건식을 따진 다음 반복을 진행할지 여부를 결정한다.

    return 0;
}
```

do ~ while

예제

```c
#include <stdio.h>

int main(void) {
    int num = 0;

    do
    {
        printf("num : %d₩n", num);
        num++;
    } while (num < 5);

    return 0;
}
```

결과

```
num : 0
num : 1
num : 2
num : 3
num : 4
```

for

```c
#include <stdio.h>

int main(void) {

    //for문의 기본 구조
    for (초기식; 조건식; 증감식) {
        //반복시킬 명령어
    }

    return 0;
}
```

* 초기식 : 반복을 위한 변수의 선언 및 초기화에 사용
* 조건식 : 반복의 조건을 검사하는 목적으로 선언
* 증감식 : 반복의 조건을 '거짓'으로 만드는 증가 및 감소 연산

for

예제

```c
#include <stdio.h>

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

    printf("num : ");
    scanf("%d", &num);
    for (i = 0; i < num + 1; i++) {
        sum += i;
    }
    printf("0부터 num=%d 까지의 합 : %d₩n", num, sum);
    return 0;
}
```

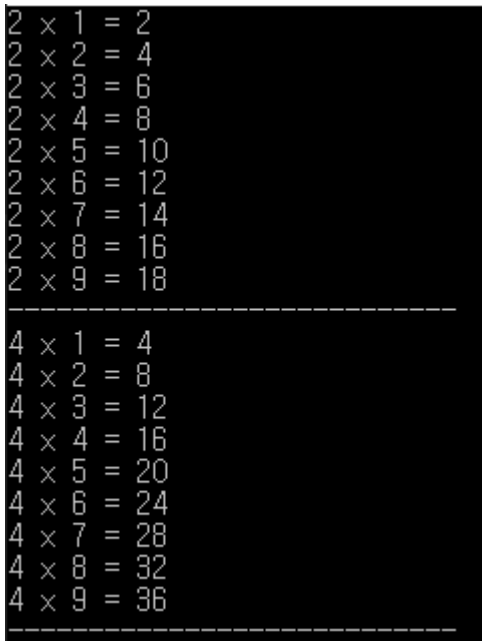
결과

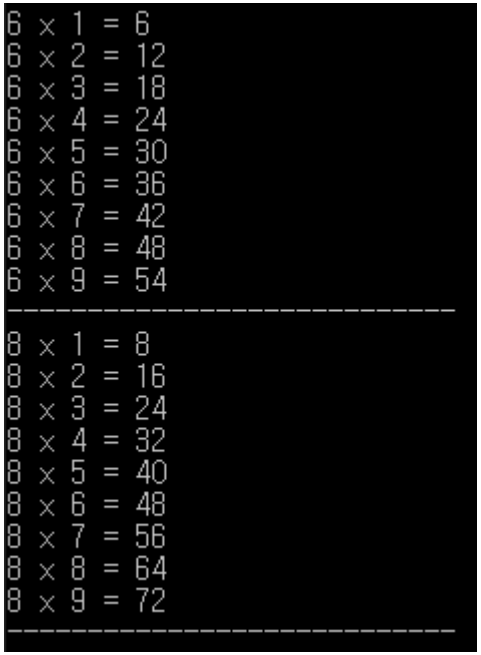
```
num : 10
0부터 num=10 까지의 합 : 55
```

실습1

While문과 for문을 사용하여 구구단(2~9단) 중 짝수 단(2,4,6,8단)을 출력하세요.

결과화면





실습1

While문과 for문을 사용하여 구구단 중 짝수 단(2,4,6,8단)을 출력하세요.

% 연산자

왼쪽의 피연산자 값을 오른쪽의 피연산자 값으로 나눴을 때 얻게 되는 나머지를 반환

Ex) num = 9 % 4;

=> 9/4의 나머지 값인 1 반환

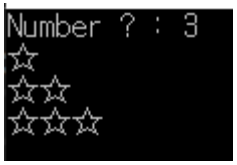
num = 5 % 3;

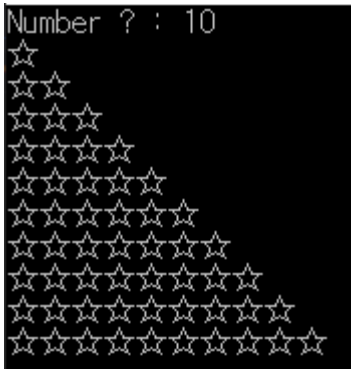
=> 5/3의 나머지 값인 2 반환

실습2-1

"For문을 사용"하여 아래의 결과화면과 같이 출력될 수 있도록 하세요.
(단순하게 printf로 다 찍는 것 안됨)

결과화면





실습2-2

"For문을 사용"하여 아래의 결과화면과 같이 출력될 수 있도록 하세요.
(단순하게 printf로 다 찍는 것 안됨)

결과화면