

# 상명튜터링

3 주 차 반 복 문

## 2주차 숙제

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

```
int main() {
```

```
int x = 0;
```

```
int y = 0;
```

```
printf("00의 숙제\n");
```

```
while(1){
```

```
    printf("2개의 정수를 공백을 기준으로 입력해주세요 >>");
```

```
    scanf("%d %d", &x, &y);
```

```
    if(x > 0 && y > 0){
```

```
        printf("x=%d, y=%d이므로 제 1사분면에 속한다.\n", x, y);
```

```
    }
```

```
    else if( x < 0 && y > 0){
```

```
        printf("x=%d, y=%d이므로 제 2사분면에 속한다.\n", x, y);
```

```
    }
```

```
    else if (x < 0 && y < 0){
```

```
        printf("x=%d, y=%d이므로 제 3사분면에 속한다.\n", x, y);
```

```
    }
```

```
    else if (x > 0 && y < 0){
```

```
        printf("x=%d, y=%d이므로 제 4사분면에 속한다.\n", x, y);
```

```
    }
```

```
    else if (x > 0 && y < 0){
```

```
        printf("x=%d, y=%d이므로 제 4사분면에 속한다.\n", x, y);
```

```
    }
```

```
    else if (x == 0 && y == 0){
```

```
        printf("x=%d, y=%d이므로 프로그램을 종료합니다.\n", x, y);
```

```
        break;
```

```
    }
```

```
    else {
```

```
        printf("\n정수만 입력해주시기 바랍니다.");
```

```
    }
```

```
}
```

```
return 0;
```

```
}
```

## 2주차 숙제

```
> gcc -o main main.c
> ./main
00의 숙제
2개의 정수를 공백을 기준으로 입력해주세요 >>3 3
x=3, y=3이므로 제 1사분면에 속한다.

2개의 정수를 공백을 기준으로 입력해주세요 >>-3 3
x=-3, y=3이므로 제 2사분면에 속한다.

2개의 정수를 공백을 기준으로 입력해주세요 >>-3 -3
x=-3, y=-3이므로 제 3사분면에 속한다.

2개의 정수를 공백을 기준으로 입력해주세요 >>3 -3
x=3, y=-3이므로 제 4사분면에 속한다.

2개의 정수를 공백을 기준으로 입력해주세요 >>0 0
x=0, y=0이므로 프로그램을 종료합니다.
> █
```

< 결과 >

x와 y가 0이 입력되면 프로그램을 종료한다.  
그 외 조건은 2주차 과제와 같다.

# 반복문 while문의 구조

```
while(조건문){  
    //작동 코드  
}
```

## 제약 조건

조건문이 참인 동안에만 실행된다.

# 반복문 while문을 중심으로

```
#include <stdio.h>

int main(){
    int i = 0, sum = 0;
    printf("\nwhile문을 이용하여 0~9까지 출력\n");

    while(i < 10){
        printf("%d\n", i);
        sum += i;
        printf("%d", sum);
    }
}
```

```
gcc -o main main.c
./main
```

while문을 이용하여 1~6까지 합과 곱 출력

```
1. sum=1
1. mutiply=1

2. sum=3
2. mutiply=2

3. sum=6
3. mutiply=6

4. sum=10
4. mutiply=24

5. sum=15
5. mutiply=120

6. sum=21
6. mutiply=720
```

## < 결과 >

총합을 구할 때는 초기값을 0으로 해준다.

총곱을 구할 때는 초기값을 1로 해준다.

# 반복문 do~while문의 구조

```
do{  
    //작동 코드  
}while(조건문)
```

## 제약 조건

조건문이 참인 동안에만 실행된다.

조건문이 거짓이어도 최초 1회는 실행한다.

# 반복문 while문을 중심으로

```
#include <stdio.h>

int main() {
    int i = 0;
    printf("\ndo~while문을 이용하여 0~9까지 출력\n");
    do{
        printf("%d\n", i);
        i++;
    }while(i < 10);

    printf("\ndo~while문의 특징\n");

    do{
        printf("%d\n", i);
        i++;
    }while(i < 0);
    return 0;
}
```

```
gcc -o main main.c
./main

do~while문을 이용하여 0~9까지 출력
0
1
2
3
4
5
6
7
8
9

do~while문의 특징
10
>
```

## < 결과 >

첫번째 do~while문을 통해서 i의 값은 9이다. 이때 두번째 do~while문의 조건식은  $i < 0$ 이므로 거짓이다. 따라서 두번째 do~while은 최초 1회만 반복하고 반복문을 나오게 된다.

# 반복문 do~while문의 구조

```
for(시작조건; 종료조건; 조건 변화식){  
    //작동 코드문  
}
```

## 제약 조건

종료 조건이 참인 경우에만 반복한다.

대부분의 조건 변화식은 ++를 사용한다.



# 반복문 while문을 중심으로

```
#include <stdio.h>

int main (){
printf("\nfor문을 이용하여 홀수단 출력\n");
for(int i=1; i<10; i++){
    for(int j=1; j<10; j+=2){
        printf("%d X %d = %2d |", i, j, i*j);
    }
    printf("\n");
}
return 0;
}
```

```
> gcc -o main main.c
> ./main
```

for문을 이용하여 홀수단 출력

```
1 X 1 = 1 | 1 X 3 = 3 | 1 X 5 = 5 | 1 X 7 = 7 | 1 X 9 = 9 |
2 X 1 = 2 | 2 X 3 = 6 | 2 X 5 = 10 | 2 X 7 = 14 | 2 X 9 = 18 |
3 X 1 = 3 | 3 X 3 = 9 | 3 X 5 = 15 | 3 X 7 = 21 | 3 X 9 = 27 |
4 X 1 = 4 | 4 X 3 = 12 | 4 X 5 = 20 | 4 X 7 = 28 | 4 X 9 = 36 |
5 X 1 = 5 | 5 X 3 = 15 | 5 X 5 = 25 | 5 X 7 = 35 | 5 X 9 = 45 |
6 X 1 = 6 | 6 X 3 = 18 | 6 X 5 = 30 | 6 X 7 = 42 | 6 X 9 = 54 |
7 X 1 = 7 | 7 X 3 = 21 | 7 X 5 = 35 | 7 X 7 = 49 | 7 X 9 = 63 |
8 X 1 = 8 | 8 X 3 = 24 | 8 X 5 = 40 | 8 X 7 = 56 | 8 X 9 = 72 |
9 X 1 = 9 | 9 X 3 = 27 | 9 X 5 = 45 | 9 X 7 = 63 | 9 X 9 = 81 |
```

< 결과 >

2중 반복문을 사용하여 홀수단의 구구단을 출력하는 프로그램이다. 이와 같이 반복문은 중첩이 가능하다.

# 숙제

## 문제해설

두개의 양의 정수를 입력 받고, 해당 숫자들을 포함한 사이값들의 총합과 총곱을 출력하는 프로그램을 작성하시오.

## 결과해설

첫번째 입력값과 두번째 입력값은 순서만 다르고 값은 같아야한다.  
2가지 입력에 대한 총합과 총곱을 출력한다. 마지막에는 0을 입력하여 프로그램을 종료시키면 된다.

```
> gcc -o main main.c
> ./main

두 개의 0을 제외한 양의 정수를 공백을 기준으로 입력해주세요>>3 6

3부터 6까지의 합은 18이다.
3부터 6까지의 곱은 360이다.
0을 입력하면 프로그램은 종료됩니다.

두 개의 0을 제외한 양의 정수를 공백을 기준으로 입력해주세요>>6 3

3부터 6까지의 합은 18이다.
3부터 6까지의 곱은 360이다.
0을 입력하면 프로그램은 종료됩니다.

두 개의 0을 제외한 양의 정수를 공백을 기준으로 입력해주세요>>0 0
n=0, m=0이 입력되었으므로 프로그램을 종료합니다.>
```

<결과 예시>