

K G 아 이 티 뱅 크

C언어

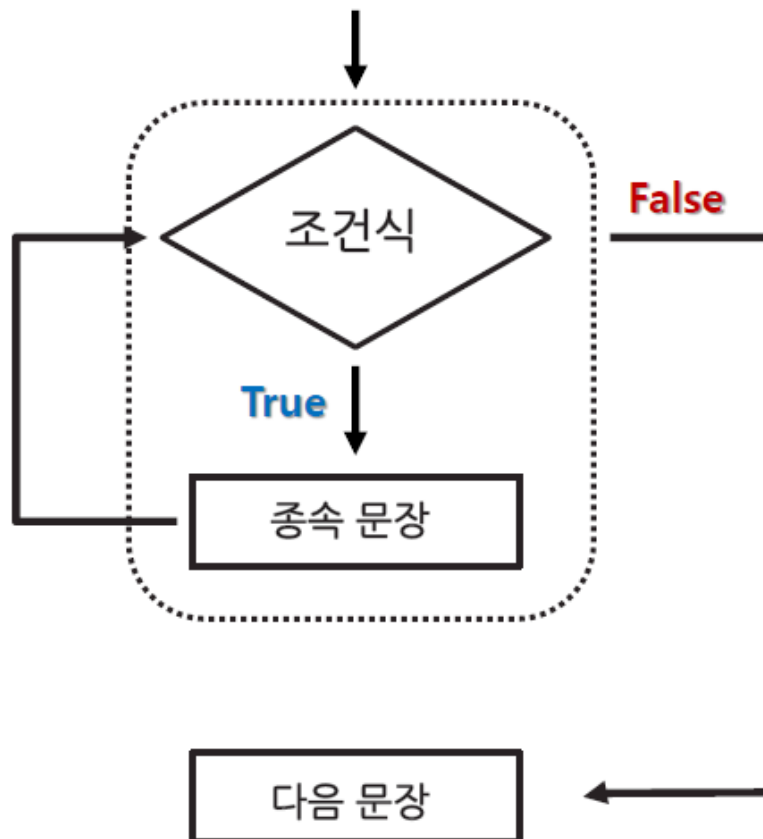
V I S U A L   S T U D I O

반복문

## 반복문 while

### › 정의

- ❖ 주어진 조건식을 평가하여 그 결과가 참인 동안은 while의 종속문장을 반복 실행한다.



```
while ( 조건식 )  
{  
    종속문장;  
}  
다음문장;
```

## 반복문 while

<파일이름 : 01.while.c>

```
#include <stdio.h>

void main()
{
    int num = 0;

    while(num<5)
    {
        printf("반복 내용 : %d\\n",num);
        num++;
    }
    printf("반복문을 종료한 후 : %d\\n",num);
}
```

num가 0일 때 while안에 들어가서 반복내용을 출력 num++로 하나가 더해지고 num이 5보다 작을 때 까지 0, 1, 2, 3, 4 즉 네 번반복을 하고 5가 되면 더 이상 반복을 하지 않고 반복문을 빠져나옵니다.

## 반복문 while

<파일이름 : 02.while.c>

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

```
void main()
```

```
{
```

```
    int num, i = 1;
```

```
    printf("구구단 단수 : "); scanf("%d",&num);
```

```
    while(i<10)
```

```
    {
```

```
        printf("%d x %d = %d\\n",num,i,num*i);
```

```
        i++;
```

```
    }
```

```
}
```

num에 몇 단을 출력할 지 입력받 고 i를 1로 초기화 시켜둔 후  
i++로 숫자를 하나씩 늘려주는데 9까지만 반복하도록 조건 지정

## 반복문 while

<파일이름 : 03.while.c>

```
#include <stdio.h>

void main() {
    int num, sum = 0, count = 0, flag = 1;
    double avg;

    while (flag != 0) {
        count++;
        printf("정수 입력 : "); scanf("%d", &num);
        sum += num;
        flag = num;
    }
    count--; //0을 입력한 횟수를 제외
    avg = sum / (float)count;
    printf("평균 : %f \\\n", avg);
}
```

항상 변수는 밖에서 선언해야 함 -> 안에서 선언한 변수는 함수가 끝나면 사라져버림(지역변수)  
sum에 입력 받은 숫자를 더한 것을 count에 반복한 횟수만큼 ++ 된 값을 나눠주면 평균값이 구해집니다.

# 반복문 while

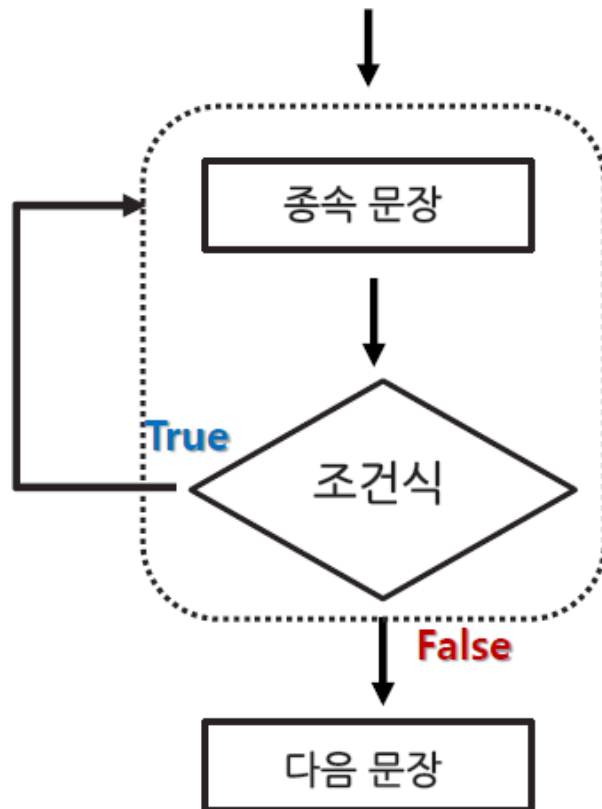
<파일이름 : 04.while.c>

```
#include <stdio.h>
#include <stdlib.h> // system 함수를 사용하게 해주는 헤더파일

void main () {
    int n, i = 0;
    while(1) { // 무한 루프 구조
        i = 1;
        printf("몇 단 ? (0일 경우, 구구단 프로그램 종료) : "); scanf("%d", &n);
        if(n == 0) {
            printf("구구단 프로그램을 종료합니다. \n");
            break; // if문 내에 break문은 if문을 제어하고 있는 while문까지 탈출
        }
        while(i < 10) {
            printf("%d × %d = %d \n", n, i, n*i);
            i++;
        }
        system("pause"); //밑에 코드를 실행 안시키도록 정지
        system("cls"); //화면을 지워줌(clear)
    }
}
```

## 반복문 do ~ while

- 무조건 종속문장을 한번 실행한 후, 조건식을 평가



```
do
{
    종속문장;
} while ( 조건식 );

다음문장;
```



## 반복문 do ~ while

<파일이름 : 05.do-while.c>

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

```
void main ()
```

```
{
```

```
    int num = 10;
```

```
    do
```

```
    {
```

```
        printf("%d\\n",num);
```

```
        num++;
```

```
    }while(num<=9);
```

```
}
```

조건이 거짓이라도 무조건 종속문장을 한번 실행한 후 조건식을 평가

## 반복문 do ~ while

<파일이름 : 07.do-while.c>

```
#include <stdio.h>

void main () {
    int num;
    do {
        printf("양수를 입력 : ");
        scanf("%d", &num);
    } while (num < 0) ;

    printf("프로그램 종료 \n");
}
```

양수를 입력 받을때까지 계속 반복하는 조건식

## 문제

<파일이름 : 07.문제.c>, <파일이름 : 08.문제.c>, <파일이름 : 09.문제.c>

[문제1] 정수를 입력 받아 그 수 만큼의 배수를 출력하시오.

<예> 4 입력 -> 4 8 12 16 (4의 배수를 4개 까지 출력)

[문제2] 정수를 입력 받고, 입력 받은 정수부터 1까지 역순으로 출력하시오.

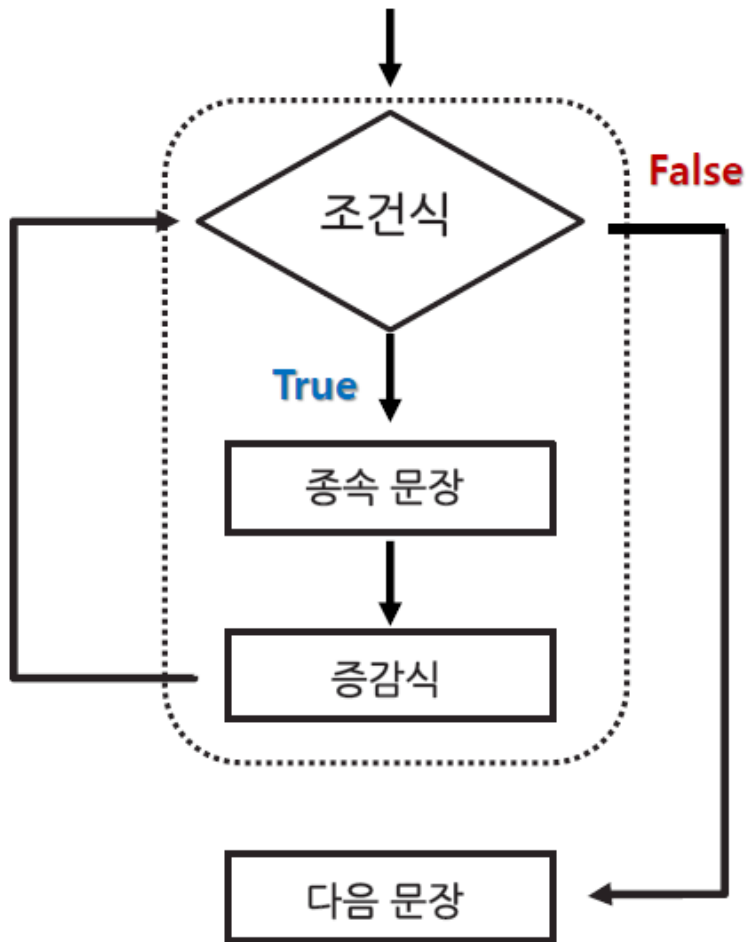
<예> 5 입력 -> 5 4 3 2 1

[문제3] 입력된 세자리 정수를 거꾸로 출력하시오.

<예> 123 입력 -> 321

## 반복문 for

- ▶ 어떠한 조건이 만족되는 동안 지정된 횟수만큼 종속문장을 반복 실행



```
for ( 초기값 ; 조건; 증감식 )  
{  
    종속문장  
}
```

## 반복문 for

<파일이름 :10.for.c>

```
#include <stdio.h>

void main ()
{
    int i;

    for( i = 0; i < 5; i++)
        printf("%d\\n",i);

    printf("프로그램 종료 \\n");
}
```

초기값을 0부터 시작하면  $i < 5$   
초기값을 1부터 시작하면  $i \leq 5$

## 반복문 for

<파일이름 :12.for.c>

```
#include <stdio.h>

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

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

for문의 초기값과 조건, 증감식은 생략 가능  
전부 생략하면 무한 반복문 -> break로 빠져나와야 함

## 반복문 for

<파일이름 :13.for.c>

```
#include <stdio.h>

void main ()
{
    int num1,num2, oddsum=0, evensum=0;

    for(num1=1,num2=2;(num1<=100 && num2<=100); num1+=2, num2+=2)
    {
        oddsum += num1;
        evensum += num2;
    }
    printf("홀수의 합 =%d\\n", oddsum);
    printf("짝수의 합 =%d\\n", evensum);
}
```

홀수에는 1 짝수에는 2를 초기화 해 놓고 2씩 더한 값을  
각각의 sum에 더해 넣으면 홀수와 짝수의 합

## break와 continue

### ➤ break 문

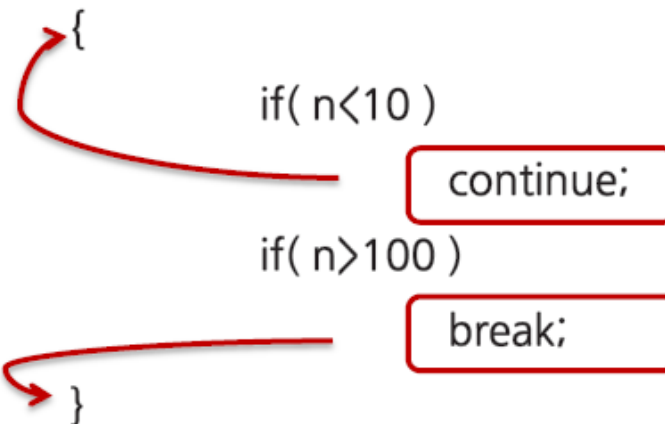
for, while, do~while, switch~case문에서 밖으로 벗어나려고 할 때 사용한다

### ➤ continue문

for, while, do~while, switch~case문에서 continue를 만나면 나머지 실행문장을 실행하지 않고, 다음 번의 반복 실행이 진행된다

EX > int n=0;

while( n++ )





## break와 continue

<파일이름 :15.break.c>

```
#include <stdio.h>

void main ()
{
    int num1;

    for(num1 = 1; num1 <= 5; num1 ++ )
    {
        printf("num1 = %d\n", num1);

        if(num1 == 3)
            break;
    }
    printf("for문을 빠져나온 후 num1 = %d\n",num1);
}
```

break를 만나면 그 시점에서 for문이 종료되고 바깥으로 나감

## break와 continue

<파일이름 : 16.continue.c>

```
#include <stdio.h>

void main()
{
    int num1;

    for (num1 = 1; num1 < 5; num1++)
    {
        if (num1 == 3)
        {
            continue;
            printf("num1 = 3");
            break;
        }
        else
            printf("num1 = %d\n", num1);
    }
}
```

continue문을 만나면 밑에 문장을 실행하지 않고  
반복문을 다시 처음부터 실행

## 문제

<파일이름 : 17.문제.c>, <파일이름 : 18.문제.c>

[문제1] 1 ~ 1000까지의 합을 구하라. 단 3의 배수는 제외하고  
3의 배수이면서 5의 배수는 제외하지 않는 조건으로 구하라.

<출력> 합계 : 366832

[문제2] 첫날에 1원을 예금하고, 다음날에는 전날의 2배를 예금하는 방식으로  
한달(30일) 동안 저축한 금액은?

<출력> 한달 동안 저축한 금액 : 1073741823원

## 문제

<파일이름 : 20.문제.c>

### ❖ for 문제

입력 받은 값의 약수와 개수를 출력하시오.

(약수 : 1 ~ 자기 자신의 수까지 나눠 0으로 떨어지게 하는 수)

예시 >

입력 : 6

출력 결과 : 1, 2, 3, 6 (4개)

## 반복문 for

〈파일이름 : 21.문제.c〉

### ❖ for 문제

숫자를 입력 받고 입력 받은 수가 소수인지 출력하시오

소수(Prime Number) : 약수가 2개인 수

예시 >

입력 : 7

출력 결과 : 7은 소수 입니다.

## 반복문 for

〈파일이름 : 22.문제.c〉

### ❖ for 문제

1부터 입력 받은 숫자까지를 모두 곱해주는 factorial을 구현하시오.

예 > 입력 : 5 =>  $5 \times 4 \times 3 \times 2 \times 1$  => 120

### ➤ 정의

❖ 두 개의 for문을 이용하여 프로그램을 설계

```
for ( 초기값 ; 조건; 증감식 )  
{  
    for ( 초기값 ; 조건; 증감식 )  
    {  
        종속문장  
    }  
}
```

## 반복문 이중 for

<파일이름 :14.이중for.c>

```
#include <stdio.h>

void main ()
{
    int num1,num2;

    for(num1 = 1; num1 <= 5; num1 ++ )
    {
        for(num2 = 1; num2 <= 5; num2++)
            printf("num1 =%d,\tnum2 = %d\tn", num1, num2);
        printf("\n");
    }
}
```

바깥쪽 반복문이 num1 = 1 일 때 안쪽 반복문의 num2는  
1~5까지 다섯 번 반복  
안쪽 반복문이 전부 다 끝나면 num1 =2가 되어 다시 반복



## 다중루프

<파일이름 : 23.직각이등변삼각형.c>

출력

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

num

5

i	j
0	0
1	0 1
2	0 1 2
3	0 1 2 3
4	0 1 2 3 4

i는 별의 행을 결정. j는 한 행에 별을 몇 개 출력할지 결정합니다.

## 다중루프

<파일이름 : 24.직각이등변삼각형역방향.c>

출력

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

num

5

i	j
0	0
1	0 1
2	0 1 2
3	0 1 2 3
4	0 1 2 3 4

j의 앞에 공백을 출력 후 \*을 출력  
공백과 \*의 반복횟수가 num과 같습니다.