



학습자가 강의 저작물을 다운로드·캡처 받아 **외부로 유출하는 행위**는 저작권자의  
이용허락 없이 저작물을 복제·공중송신 또는 배포 하는 것으로 **저작권 침해 행위**에 해당함.

# C 프로그래밍

(001/002)

## 제 5 강

신 한 대 학 교  
소프트웨어융합학과  
교수 송 진 희

# C 프로그래밍

## 제 5 강

- 제어문
  - 1) 반복문
    - for()문
    - while()문
    - do-while()문
    - break / continue

# 학습 목표

- for()문을 이용한 반복 처리를 표현할 수 있다.
- while()문을 이용한 반복 처리를 표현할 수 있다.
- do-while()문을 이용한 반복 처리를 표현할 수 있다.
- for()문과 while()문을 이용한 반복 처리 표현을 상호 (相互) 변경해서 표현할 수 있다.

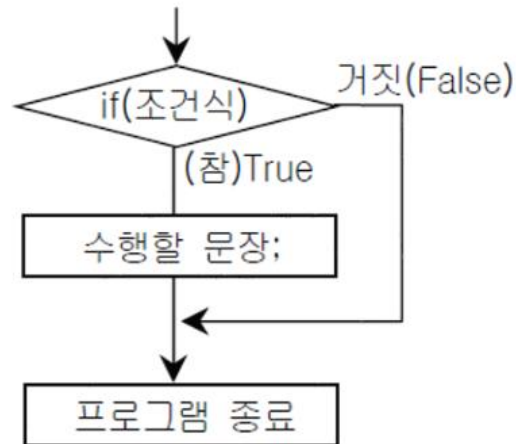
# 4강 - 정리 요약

○다음의 조건문 기술 형식으로 논리식을 표현할 수 있다.

- if(조건1)  
명령문1;

- if( 조건식1 )  
명령문1;  
else  
명령문2;

- if(조건식1)  
명령문1;  
else if ( 조건식2)  
명령문2;



switch (수식 또는 변수)  
{

case 값1 : 명령문1;  
break;

case 값2 : 명령문2;  
break;

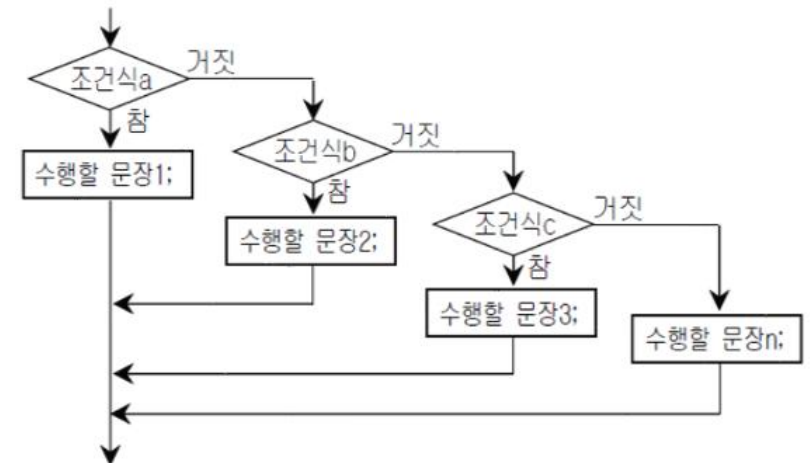
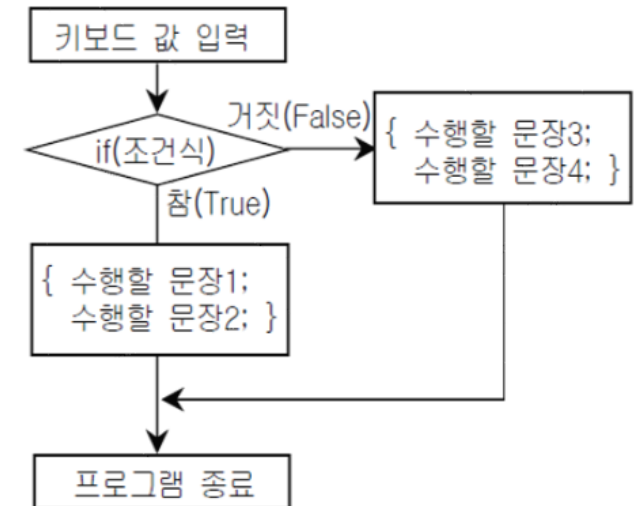
case 값3 : 명령문3;  
break;

case 값4 : 명령문4;  
break;

default : 명령문4;  
break;

}

정수 / 문자



# 4강 - 정리 요약 (if)

- 키보드로 입력 받은 값이 2030, -96일 때 아래 실행 결과가 나오도록 블록({}) if문을 이용해서 작성하시오.

C:\w [실행결과]

■ 입력받은 값이 조건식의 참, 거짓에 따라 출력되는 프로그램

=====

> 정수입력 : 2030

-----

-----

> 키보드로 입력한 값 2030 는(은) 양의 정수입니다.

> 조건식은 참입니다.

> 프로그램을 종료합니다...

양수 판별 처리

C:\w [실행결과]

■ 입력받은 값이 조건식의 참, 거짓에 따라 출력되는 프로그램

=====

> 정수입력 : -96

-----

-----

> 프로그램을 종료합니다...

음수이면 '종료' 출력

# 4강 - 정리 요약

- 키보드로 입력 받은 값이 2030, -96일 때 블록({ }) 실행 문 처리 결과를 각각 화면 캡처해서 제출하시오.

```
1  /*-- 기본 if문 출력에서 코드블록의 중요성 인식 프로그램 --*/
2  #include <stdio.h>
3  void main()
4  {
5      int num;
6
7      printf(" ■ 입력받은 값이 조건식의 참, 거짓에 따라 출력되는 프로그램  \n");
8      printf("===== \n");
9
10     printf(" > 정수입력 : ");
11     scanf("%d", &num);          // scanf_s("%d", &num);
12     printf("----- \n");
13
14     if(num > 0)
15     {
16         printf(" > 키보드로 입력한 값 %d는(은) 양의 정수입니다. \n", num);
17         printf(" > 조건식은 참입니다. \n");
18     }
19     printf(" > 프로그램을 종료합니다... \n");
20 }
```

# 4강 - 정리 요약 (switch)

## ○다중 if(조건식)문을 switch문으로 변경 가능

- 'case(상수)'로 선언 : 문자 상수 또는 정수 상수만 허용
- 각 'case'문의 끝에는 'break'문을 선언
- 'default:'문은 'case'에 해당되는 조건이 없을 경우 수행

## ○switch문 사용 주의 사항

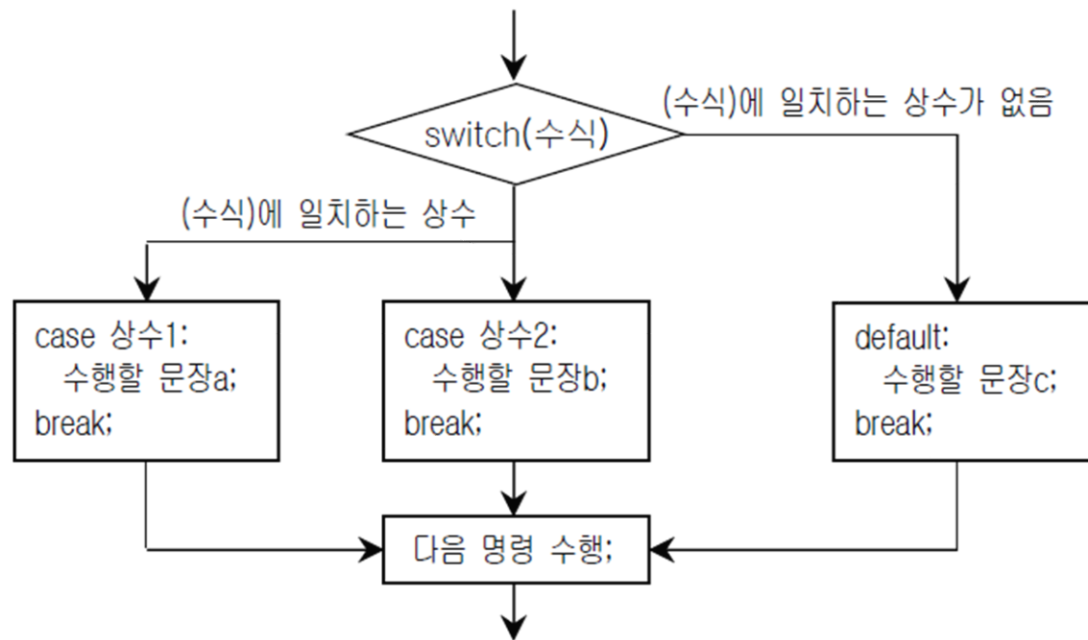
- switch(수식) ... 수식은 반드시 정수형(문자형 포함)
- case 상수: ..... 상수는 반드시 1개의 정수 값만 인정 (논리형 값은 올바르지 못함)
- 콜론(:) ..... 'case 상수:'에서 상수 다음에는 세미콜론(;)이 아닌 콜론(:)을 사용
- 코드블록 ..... 'case 상수:'에 해당하는 문장은 2개 이상이라도 중괄호를 사용하지 않음
- break; ..... 생략하면 다음 case문까지 계속해서 처리하게 됨
- default: ..... 해당 'case문'이 없을 경우에 수행할 문장



# 4강 - 정리 요약 (switch)

## ○switch문 선언과 순서도

```
switch(수식)
{
  case 상수1:
    실행할 문장a;
    break;
  case 상수2:
    실행할 문장b;
    break;
  default:
    실행할 문장c;
    break;
}
```



# 반 복 문

# 반복

○반복(iteration)은 동일한 처리 과정을 여러 번 되풀이하는 것

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



```
for(int i = 0; i < 5; i++)  
    printf("Hello World! \n");
```

# 반복문(Iteration)

- 특정 조건에 따라 반복 실행하는 명령문
  - `for()` / `while()` / `do-while()`
- `for()`문의 제어 구조

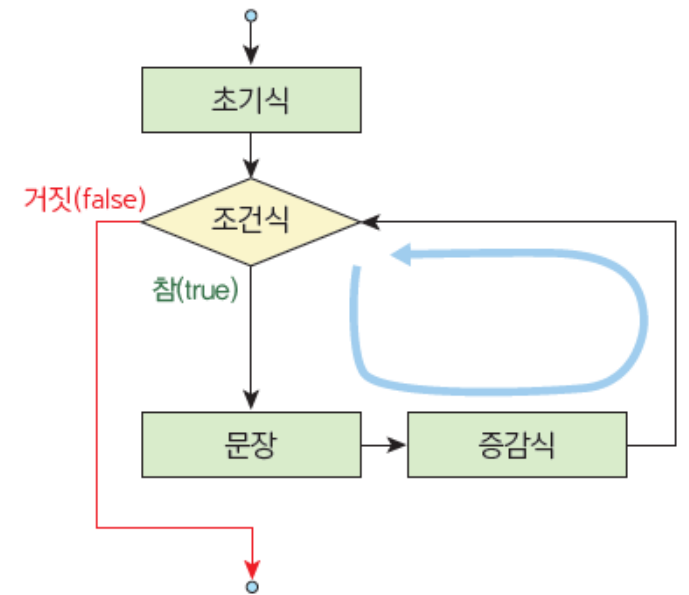
## Syntax for 문

예

```
for( 초기식; 조건식; 증감식 ) {  
    printf("Hello World!");  
}
```

초기식      조건식      증감식

반복되는 문장



# for()문

## ○초기식

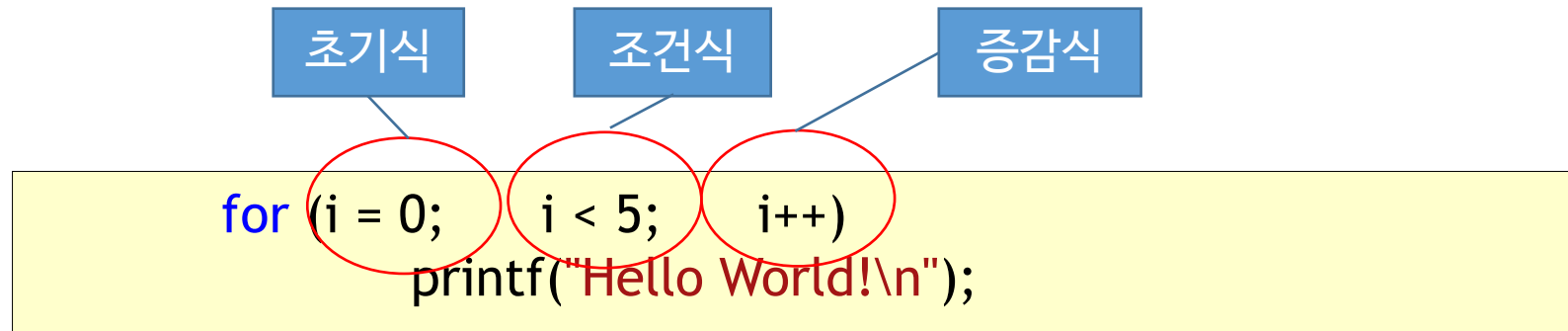
- 초기식은 반복 루프를 시작하기 전에 한번만 실행된다.  
주로 변수 값을 초기화하는 용도로 사용

## ○조건식

- 반복의 조건을 검사하는 수식이다. 이 수식의 값이 거짓이면 반복을 중단

## ○증감식

- 한 번의 루프 실행이 끝나면 증감식이 실행



# for()문

```
// "Hello World!" 5번 출력하기
#include <stdio.h>

int main(void)
{
    int i;

    for (i = 0; i < 5; i++)        // i는 0부터 4까지 1씩 증가하면서 반복 실행
        printf("Hello World!\n");

    return 0;
}
```

```
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
```

# 예제 #2

○ 1부터 10까지의 정수를 더하여 합계를 구하는 프로그램  
을 작성하시오.

○ 실행 결과 :



1부터 10까지의 정수의 합 = 55

```
// “Hello World!” 5번 출력하기
#include <stdio.h>

int main(void)
{
    int i, sum;

    sum = 0;
    for (i = 1; i <= 10; i++)           // i는 1부터 10까지 1씩 증가하면서 반복 실행
        sum += i;

    printf(“1부터 10까지의 정수의 합 = %d \n”, sum );
    return 0;
}
```

# 예제 #3

○이번 예제에서는 for 루프를 이용하여 일정 범위의 정수에 대하여 세제곱 값을 구하여 보자. 즉 1의 세제곱부터 시작해서 **사용자가 입력하는 수**까지의 세제곱까지를 나열하는 프로그램을 작성하시오.

정수를 입력하시요:5

=====

i	i의 세제곱
---	--------

=====

1	1
2	8
3	27
4	64
5	125



# 예제 #3

```
// 반복을 이용한 세제곱값 구하기
#include <stdio.h>

int main(void)
{
    int i, n;

    printf("정수를 입력하시오: ");
    scanf("%d", &n);

    printf("=====\\n");
    printf(" i   i의 세제곱\\n");
    printf("=====\\n");
    for(i = 1; i <= n; i++)
        printf("%5d   %8d \\n", i, i * i * i);

    return 0;
}
```

정수를 입력하시오:5

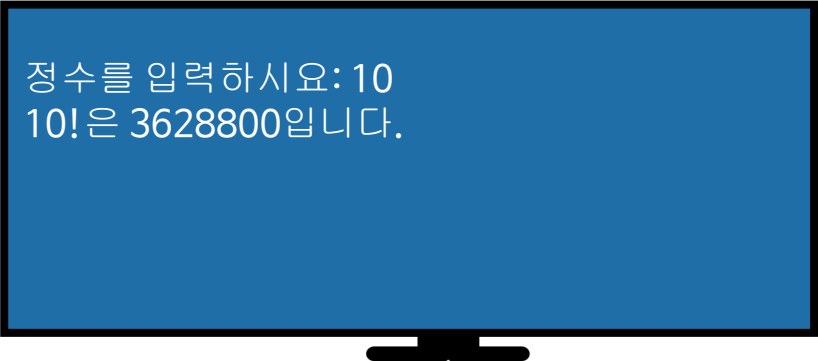
```
=====
 i   i의 세제곱
=====
 1      1
 2      8
 3     27
 4     64
 5    125
```

# [문제 #1]

○이번 예제에서는 팩토리얼 값을 계산하여 보자.

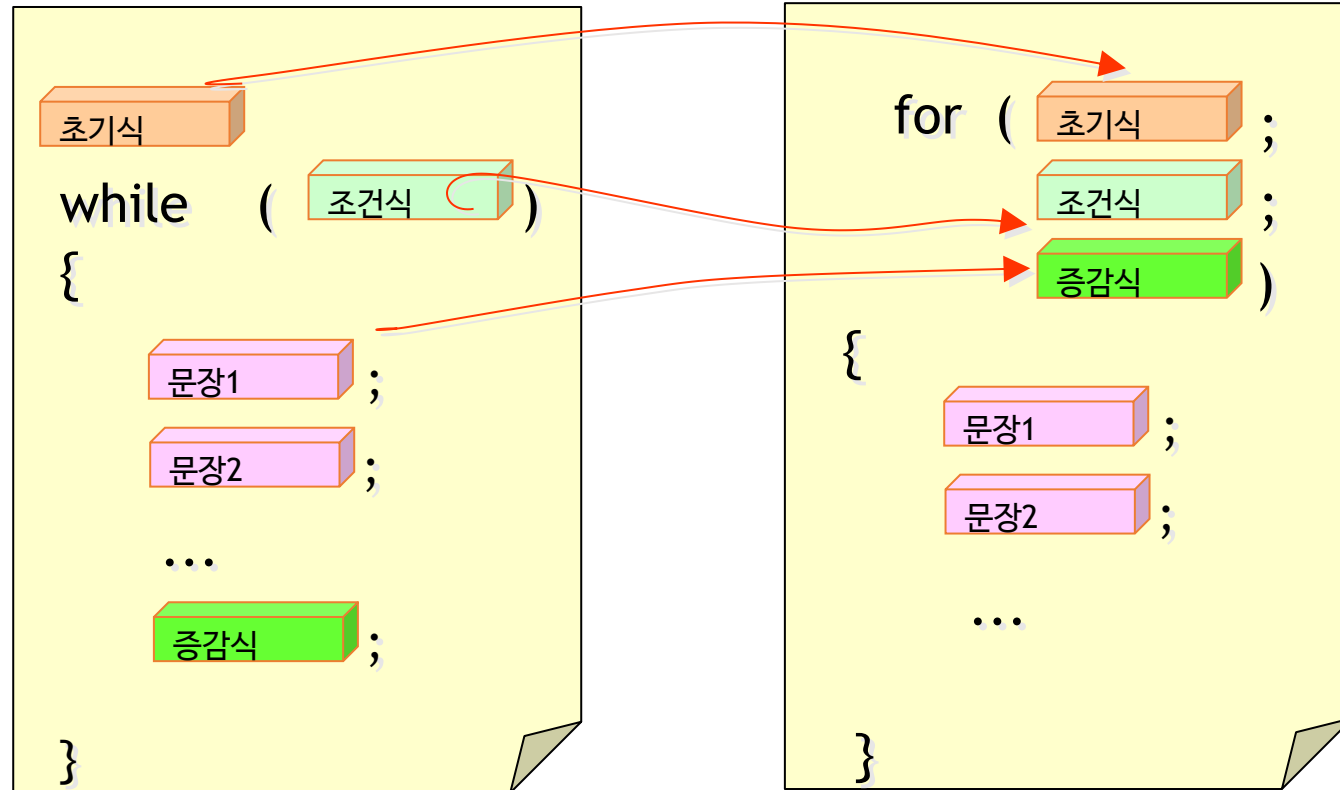
▶팩토리얼은 다음과 같이 정의된다.

$$n! = 1 \times 2 \times 3 \times \dots \times n$$



정수를 입력하시요: 10  
10!은 3628800입니다.

# while()문과 for()문의 변환



# [예제 #5]의 while() 처리

정수를 입력하시요: 10  
10!은 3628800입니다.

// 반복을 이용한 팩토리얼 구하기

#include <stdio.h>

int main(void)

{

    long fact = 1;

    int i = 1, n;

    printf("정수를 입력하시요: ");

    scanf("%d", &n);

    while (i <= n)

    {

        fact = fact \* i;

        i++;

    }

    printf("%d!은 %d입니다.", n, fact);

    return 0;

}

for(int i = 1; i <= 10; i++) {

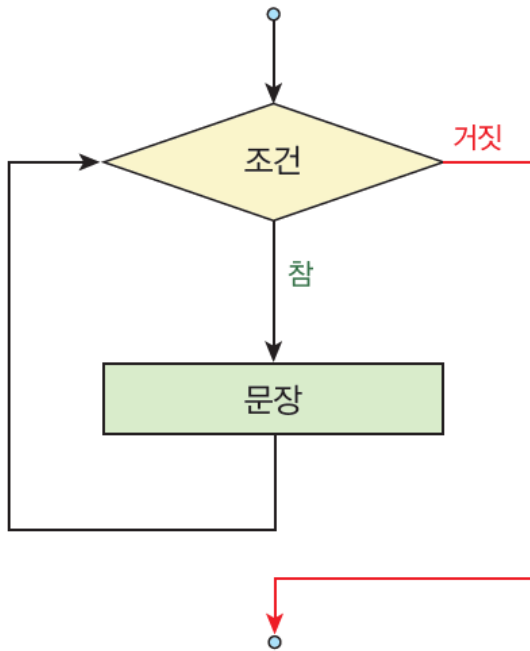
    ...

}

# while()문

○조건이 '참'인 경우에만 반복 실행

○while()문의 제어 구조



Syntax

while 문

예

```
while( i < 10 ) {  
    printf("Hello World!\n");  
    i++;  
}
```

조건식

조건식이 참이면 문장을 반복 실행한다.

# while()문

○조건이 '참'인 경우에만 반복 실행

```
while(i != 0)
{
    ...
}
```



```
while( i )
{
    ...
}
```

```
while(i = 2)
{
    ...
}
```

수식의 값이 2이므로 항상 참이 되어서 무한 루프

# 예제

```
#include <stdio.h>
int main(void)
{
    int i = 0;
    while( i < 5 )
    {
        printf("Hello World! \n");
        i++;
    }

    return 0;
}
```

반복 조건

반복 내용

Hello World!  
Hello World!  
Hello World!  
Hello World!  
Hello World!

# 예제 #1

## ○ 1~n까지의 합을 계산하는 프로그램 작성

```
#include <stdio.h>

int main(void)
{
    int i, n, sum;                // 변수 선언

    printf("정수를 입력하시오:"); // 입력 안내 메시지 출력
    scanf("%d", &n);              // 정수값 입력

    i = 1;                        // 변수 초기화
    sum = 0;

    while(i <= n)
    {
        sum += i;                // sum = sum + i;와 같다.
        i++;                     // i = i + 1과 같다.
    }

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

정수를 입력하시오: 3  
1부터 3까지의 합은 6입니다



# 예제 #2

```
// while 문을 이용한 구구단 출력 프로그램
#include <stdio.h>

int main(void)
{
    int n;
    int i = 1;

    printf("출력하고 싶은 단: ");
    scanf("%d", &n);

    while (i <= 9)
    {
        printf("%d*%d = %d \\\n", n, i, n*i);
        i++;
    }

    return 0;
}
```

출력하고 싶은 단을 입력하시오: 9  
9\*1 = 9  
9\*2 = 18  
9\*3 = 27  
....  
9\*9 = 81

# 예제 #3

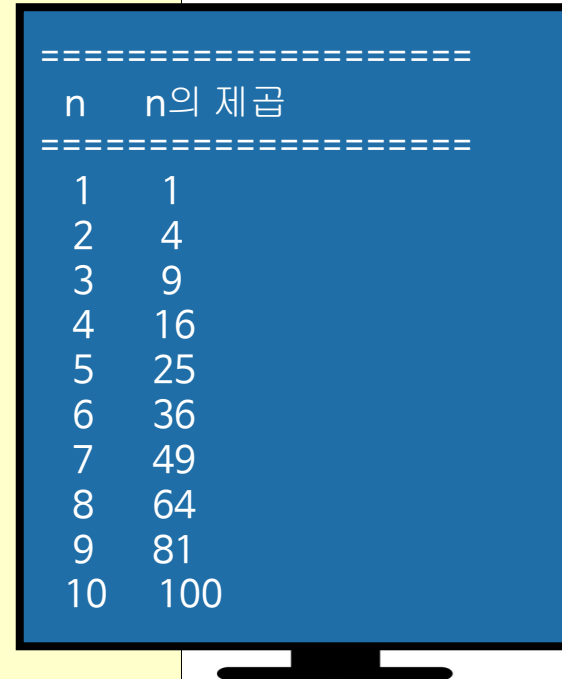
```
// while 문을 이용한 제곱값 출력 프로그램
#include <stdio.h>

int main(void)
{
    int n;

    printf("=====\n");
    printf("  n      n의 제곱 \n");
    printf("=====\n");

    n = 1;
    while (n <= 10)
    {
        printf("%5d   %8d\n", n, n*n);
        n++;
    }

    return 0;
}
```



=====	
n	n의 제곱
=====	
1	1
2	4
3	9
4	16
5	25
6	36
7	49
8	64
9	81
10	100

# [문제 #2]

○  $n$ 이하의 모든 짝수의 합을 계산하는 프로그램을 작성  
하십시오.

▶ 짝수의 합을 출력하려면 짝수들만 sum에 누적.



정수를 입력하십시오: 10  
1부터 10까지의 짝수합은 30입니다.

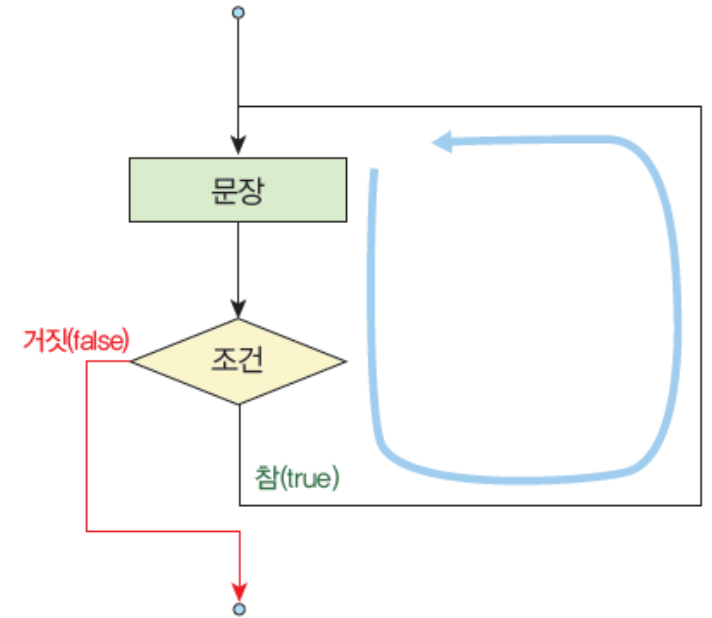
# do~while 문

○do ~ while() 사이의 문장들을 한번은 실행 후, 조건이 참이면 반복 실행하고 거짓이면 while()문을 벗어난다.

Syntax

do...while 문

예



# 예제 #1

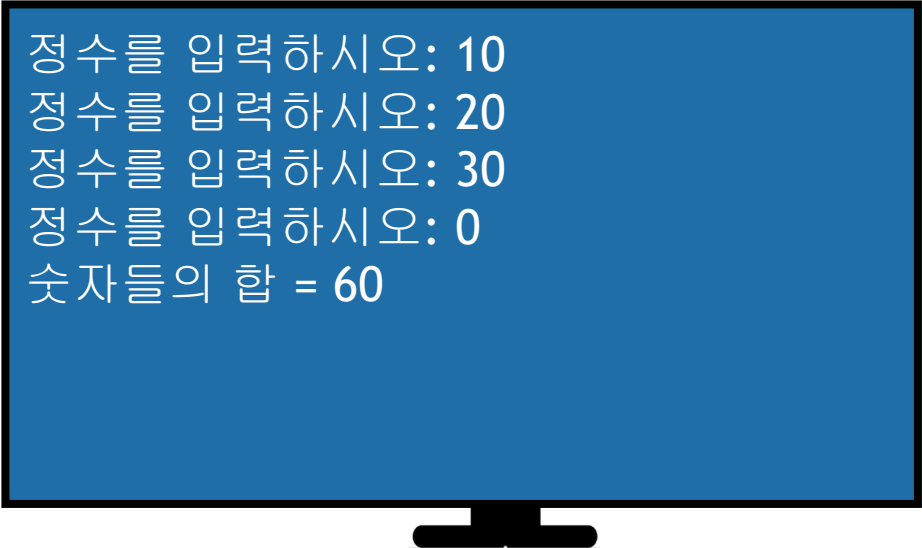
○do~while 문을 이용하여 사용자가 0을 입력할 때까지 입력된 숫자들을 더하는 프로그램을 작성해보자.

```
#define _CRT_SECURE_NO_WARNINGS
// 사용자가 0을 입력할 때까지 숫자를 더한다.
#include <stdio.h>
int main(void)
{
    int number, sum = 0;

    // 루프 몸체가 적어도 한번은 실행된다.
    do
    {
        printf("정수를 입력하시오: ");
        scanf("%d", &number);
        sum += number;
    } while (number != 0);

    printf("숫자들의 합 = %d \n", sum);

    return 0;
}
```



정수를 입력하시오: 10  
정수를 입력하시오: 20  
정수를 입력하시오: 30  
정수를 입력하시오: 0  
숫자들의 합 = 60

# 예제 #2

## ○ do~while 문은 입력을 처리하는 부분에서 많이 사용

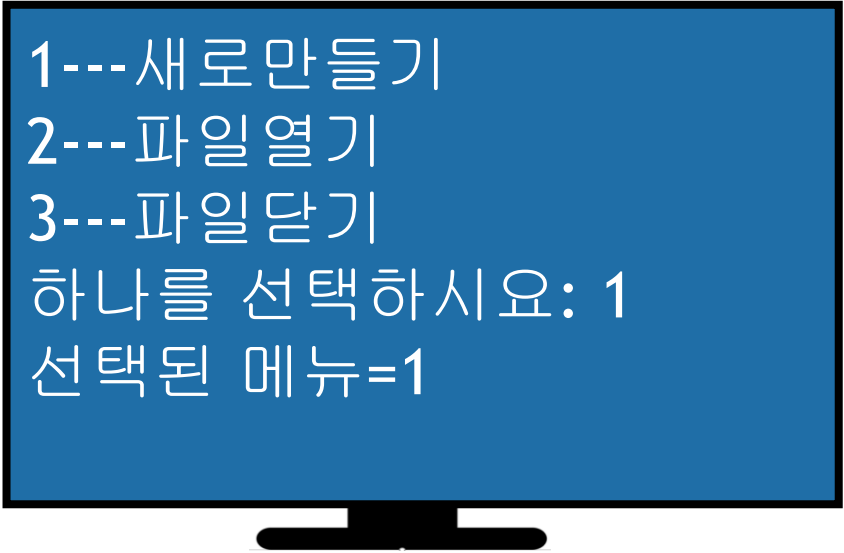
```
#define _CRT_SECURE_NO_WARNINGS
// 사용자의 파일 선택 메뉴 출력하기
#include <stdio.h>
int main(void)
{
    int i = 0;

    // 루프 몸체가 적어도 한번은 실행된다.
    do
    {
        printf( "1--- 새로 만들기 \n");
        printf( "2--- 파일 열기 \n " );
        printf( "3--- 파일 닫기 \n");
        printf( "하나를 선택하시요 : \n " );

        scanf( " %d " , &i);
    } while (i < 1 || i > 3);

    printf( " 선택된 메뉴 = %d \n " , i);

    return 0;
}
```



1--- 새로 만들기  
2--- 파일 열기  
3--- 파일 닫기  
하나를 선택하시요: 1  
선택된 메뉴=1

# 난수 발생 함수 : rand() / srand()

- 임의의 수(난수)를 발생시키는 함수
- 헤더 파일 정의 : **<stdlib.h>**
- **rand(seed)** : 난수의 생성 패턴이 한 개(0 ~ 32767)
  - 매 실행할 때마다 동일한 난수 발생

The image shows two screenshots of a C program in Visual Studio. The program is named 'rand1.c' and is part of 'Project5'. The code is as follows:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main() {
5     int i;
6     for (i = 1; i <= 10; i++)
7         printf("%d ", (rand() % 100) + 1);
8     printf("\n");
9 }
```

The first screenshot shows the program running, and the output window displays the sequence: 42 68 35 1 70 25 79 59 63 65. The second screenshot shows the program running again, and the output window displays the same sequence: 42 68 35 1 70 25 79 59 63 65. This demonstrates that the random number sequence is identical for the same program execution.

→ : 첫 번째 실행 결과 발생 난수

=

동일함

→ : 두 번째 실행 결과 발생 난수

# 난수 발생 함수 : srand()

○srand() : seed에 따라 다른 난수 발생

➤C++ / <iostream>, <cstdlib>



# 난수 발생 함수 : rand() / srand()

○ rand(**seed**) : 난수의 생성 패턴이 한 개(0 ~ 32767)

➤ seed : 현재 시간(time)을 사용 ➔ 실행할 때마다 다른 난수 발생

```
rand2.c  rand1.c
Project5 (전역 범위) main()
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <time.h>
4
5 int main() {
6     int i;
7     srand(time(NULL));
8
9     for (i = 1; i <= 10; i++)
10         printf("%d ", (rand() % 100) + 1);
11     printf("\n");
12 }
```

Microsoft Visual Studio 디버그 콘솔

```
22 35 20 53 72 54 60 88 94 51
C:\Users\User\source\repos\Project5\64\Debug\Project5.exe(프로세스 11060개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

: 첫 번째 실행 결과 발생 난수

```
rand2.c  rand1.c
Project5 (전역 범위) main()
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <time.h>
4
5 int main() {
6     int i;
7     srand(time(NULL));
8
9     for (i = 1; i <= 10; i++)
10         printf("%d ", (rand() % 100) + 1);
11     printf("\n");
12 }
```

Microsoft Visual Studio 디버그 콘솔

```
42 35 37 62 93 26 68 27 30 33
C:\Users\User\source\repos\Project5\64\Debug\Project5.exe(프로세스 14280개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

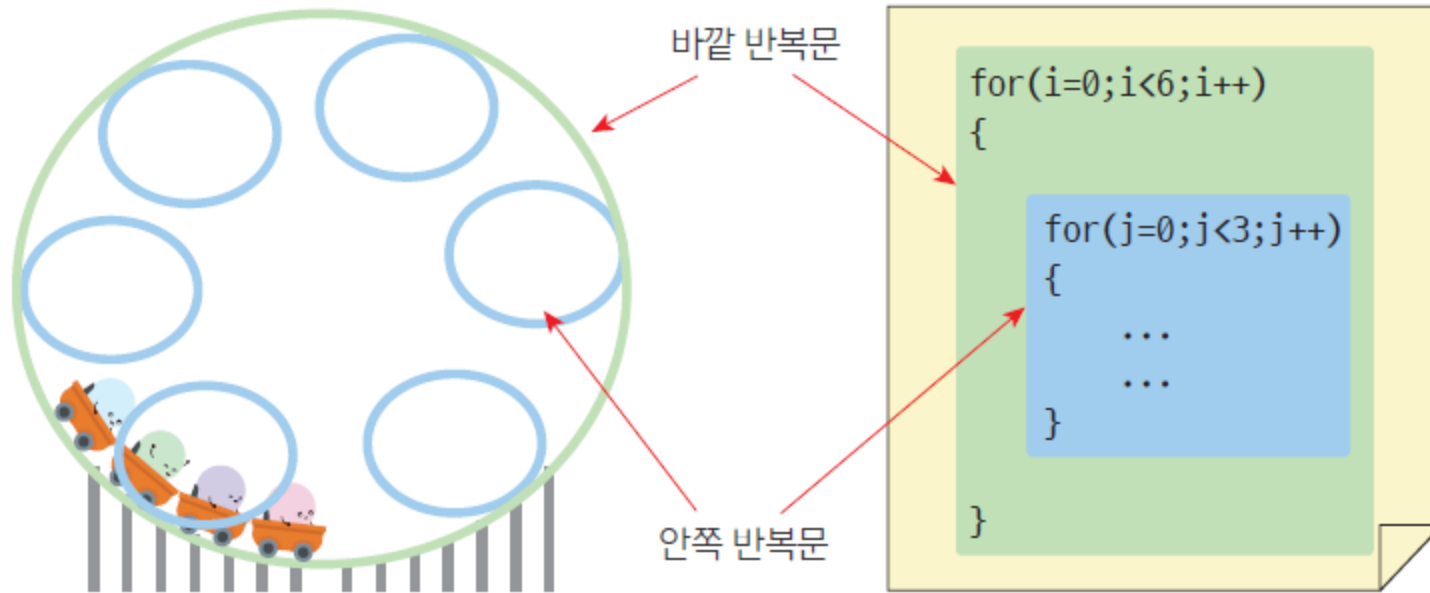
: 두 번째 실행 결과 발생 난수



서로 다름

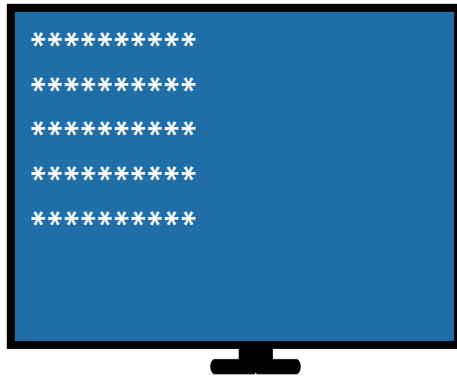
# 중첩 for()문

○ 중첩 for()문(nested loop): for()문 안에 다른 for()문 사용 가능



# 예제 #1

○ 다음 예제는 \*기호를 사각형 모양으로 출력하는 프로그램 작성



```
// 중첩 for 문을 이용하여 *기호를 사각형 모양으로 출력하는 프로그램
#include <stdio.h>

int main(void)
{
    int x, y;

    for(y = 0; y < 5; y++)
    {
        for(x = 0; x < 10; x++)
        {
            printf("*");
            printf("\n");
        }

        return 0;
    }
}
```

# 무한 루프와 break문

## ○무한 루프(infinite loop)

- 조건 제어 루프에서 프로그램이 무한히 반복하는 것
- **break** : break문이 포함된 가장 인접한 반복 루프를 빠져나감
- **continue** : continue문이 포함된 가장 인접한 반복 루프의 시작 조건으로 제어를 이동하는 명령문

Syntax

무한 루프 문

Syntax

```
while (1) {  
    if (조건)  
        break;    # 반복을 중단한다.  
    if (조건)  
        continue; # 다음 반복을 시작한다.  
}
```

# 무한루프의 사용 예

○사용자가 입력한 수가 3의 배수이거나 음수인 경우,  
while 루프를 빠져나가야 한다고 하자.

```
while((x % 3 != 0) && (x >= 0)) {  
    ...  
    ...  
    ...  
}
```



```
while (1) {  
    if (x%3 == 0) break;  
    if (x<0) break;  
    ...  
}
```

\* Break문이 포함되어 있는 가장 가까운  
반복문을 빠져나감

# 실습 #1 (break문)

○ 100만원으로 재테크를 시작한 사람이 1년에 30%의 수익을 얻는다면 몇 년 만에 원금의 10배가 되는지를 계산하는 프로그램을 작성하시오.

- 무한 반복 구조를 사용하고 조건이 만족되었을 때 break문이 실행하여 반복 구조를 탈출



# 실습 #1

```
#include <stdio.h>
#define SEED_MONEY 1000000

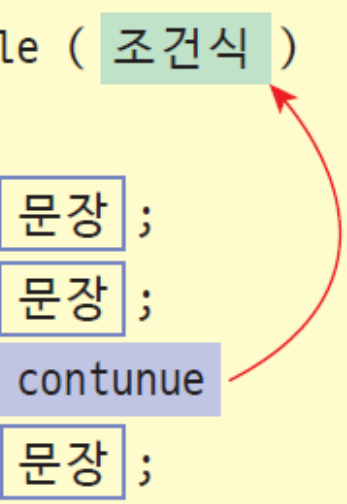
int main(void)
{
    int year=0, money=SEED_MONEY;
    while(1)
    {
        year++;
        money += money * 0.30;
        if( money > 10 * SEED_MONEY )
            break;
    }
    printf("%d 년", year);
    return 0;
}
```

원금의 10배를 초과하면

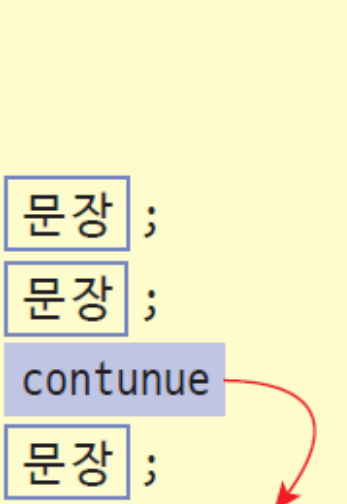
# continue 문

- ❖ **continue** : continue문이 포함된 가장 인접한 반복 루프의 시작 조건으로 제어를 이동하는 명령문

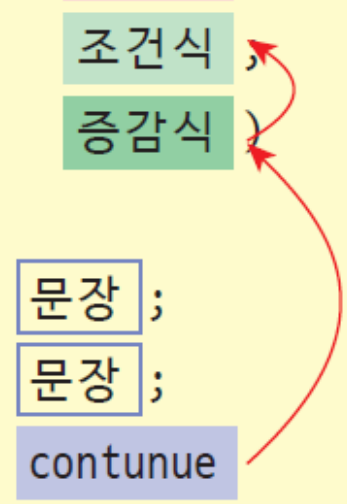
```
while ( 조건식 )  
{  
    문장 ;  
    문장 ;  
    continue  
    문장 ;  
}
```



```
do  
{  
    문장 ;  
    문장 ;  
    continue  
    문장 ;  
} while ( 조건식 );
```



```
for ( 초기식 ;  
      조건식 ;  
      증감식 )  
{  
    문장 ;  
    문장 ;  
    continue  
    문장 ;  
}
```





# 실습 #2 (continue문)

○ 0부터 10까지의 정수 중에서 3의 배수가 아닌 숫자만 출력하는 프로그램을 작성하시오.

```
#include <stdio.h>

int main(void)
{
    int i;

    for(i = 0; i < 10; i++)
    {
        if(i % 3 == 0)
            continue; // i가 3의 배수이면 i 다음 숫자를 체크

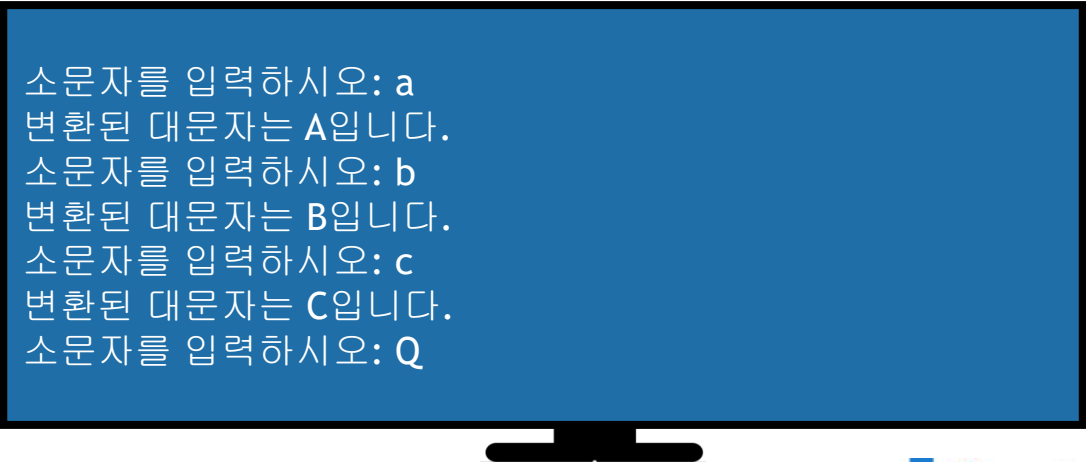
        printf("%d ", i);
    }
    return 0;
}
```

1 2 4 5 7 8

# [문제 #3]

○사용자로부터 알파벳 소문자를 받아서 대문자로 바꾸도록 while()문을 이용한 프로그램을 작성한다.

- 사용자로부터 받은 문자가 소문자가 아니면 사용자로부터 다시 문자를 입력 받도록 while()문을 반복 실행
- 사용자가 입력한 문자가 'Q'이면 while()문을 빠져나가 프로그램을 종료한다.



소문자를 입력하시오: a  
변환된 대문자는 A입니다.  
소문자를 입력하시오: b  
변환된 대문자는 B입니다.  
소문자를 입력하시오: c  
변환된 대문자는 C입니다.  
소문자를 입력하시오: Q

# 5강 – 정리 요약

## ○반복 실행문

- **for**(초기값; 최종값; 증감식)
- **while**(조건식) : 조건식이 참일 경우에만 반복 실행
- **do~while()** : 1회는 반복 수행한 후, 조건을 체크해서 참이면 다시 반복 실행

## ○반복 구조에서 제어 빠져나가기

- **break**문
- **continue**문

## ○난수 발생 함수의 사용

- **rand()** : 프로그램 실행할 때마다 동일한 난수 발생한다.
- **srand()** : seed 값에 따라 발생하는 난수가 다르다.

