



쉽게 풀어쓴 C언어 Express

제7장 반복문





이번 장에서 학습할 내용



- 반복의 개념 이해
- while 반복문
- do-while 반복문
- for 반복문
- break와 continue문

반복 구조는 일련의 처리를 반복할 수 있게 한다. 반복의 개념을 먼저 이해하고 C에서 제공되는 3가지의 반복 구조에 대하여 학습한다.

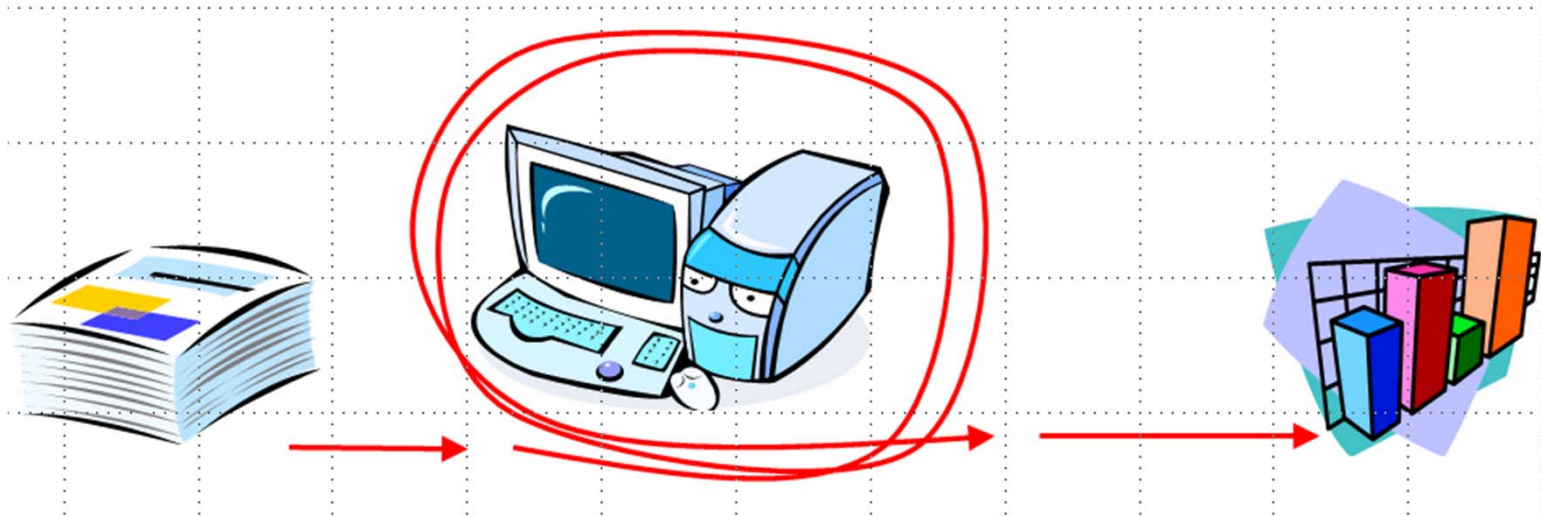




반복문

Q) 반복 구조는 왜 필요한가?

A) 같은 처리 과정을 되풀이하는 것이 필요하기 때문이다. 학생 30명의 평균 성적을 구하려면 같은 과정을 30번 반복하여야 한다.

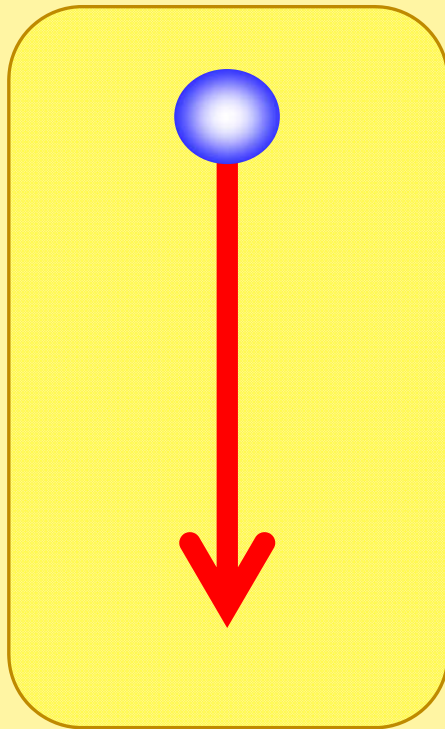




프로그램의 흐름을 제어하는 방법

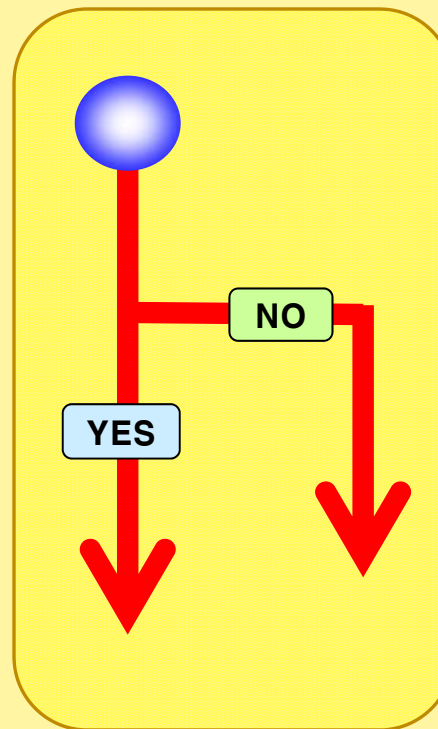
1) 순차구조

- 차례대로 실행



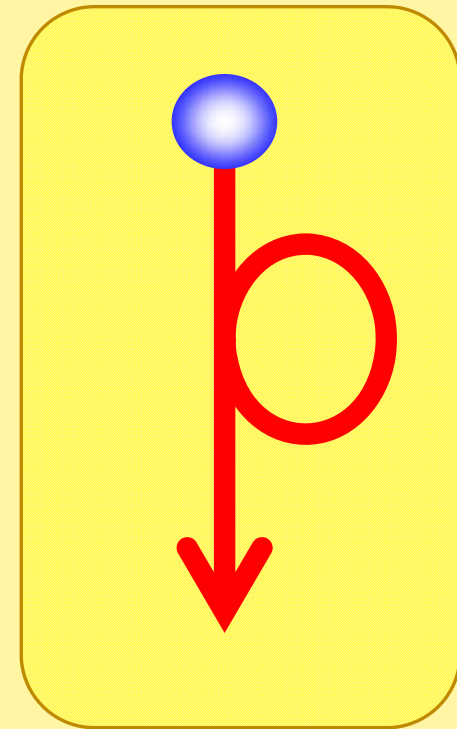
2) 선택구조

- 조건을 검사하여 여러 개의 실행 경로 중에서 하나를 선택



3) 반복구조

- 조건이 만족될 때까지 반복





마일을 미터로 바꾸는 프로그램

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

```
int main(void)
```

```
{
```

```
    int meter;
```

```
    meter = 0 * 1609;  
    printf("0 마일은 %d미터입니다\n", meter);
```

```
    meter = 1 * 1609;  
    printf("1 마일은 %d미터입니다\n", meter);
```

```
    meter = 2 * 1609;  
    printf("2 마일은 %d미터입니다\n", meter);
```

```
    return 0;
```

```
}
```

같은 처리과정 #1

같은 처리과정 #2

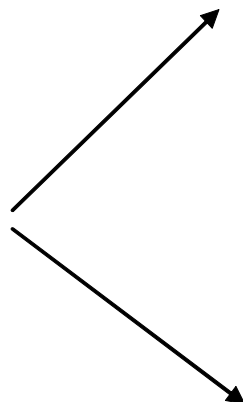
같은 처리과정 #3



반복문의 종류



반복문



while



for





중간 점검

1. 프로그램에 반복 구조가 필요한 이유는 무엇인가?
2. 반복문에는 _____, _____문이 있다.

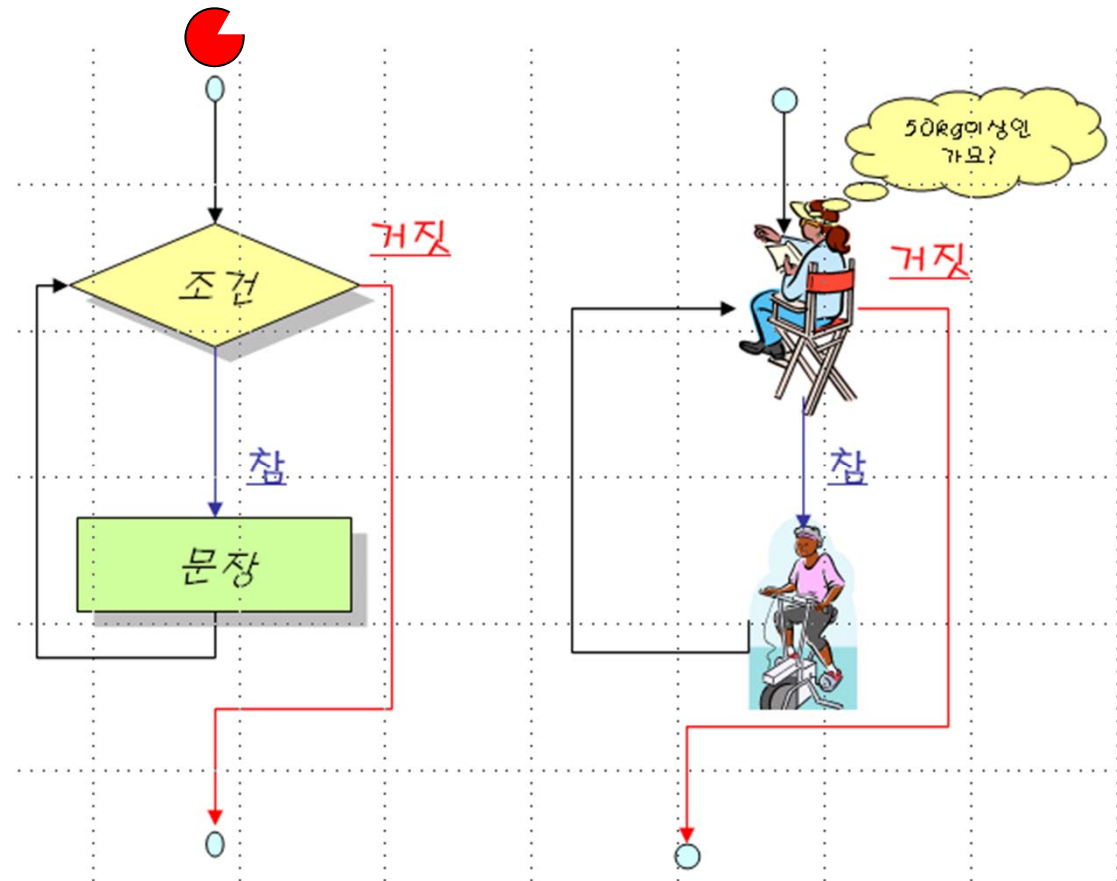




while 문

- 주어진 조건이 만족되는 동안 문장들을 반복 실행한다.

while(조건식)
문장;





예제

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

```
int main(void)
```

```
{
```

```
    int meter=0;  
    int i = 0;
```

```
    while(i < 3)
```

```
    {
```

```
        meter = i * 1609;
```

```
        printf("%d 마일은 %d 미터입니다\n", i, meter);
```

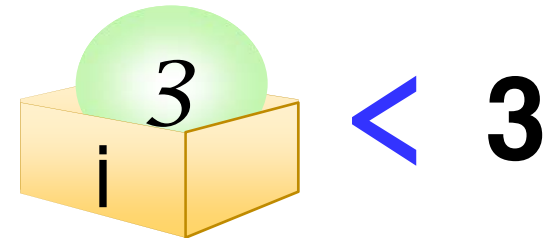
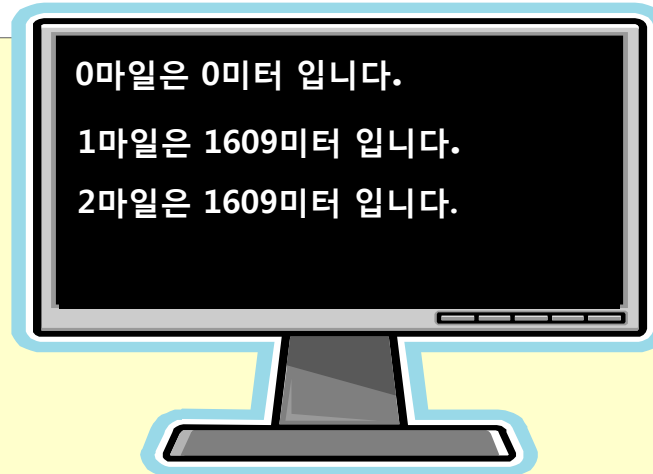
```
        i++; // i = i+1 과 동일
```

```
    }
```

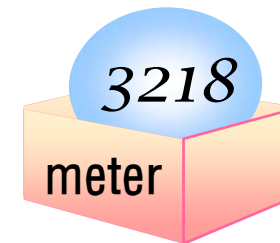
```
    return 0;
```

```
}
```

i 값이 3으로 증가하였지만 조건에 만족하지 않아
출력하지 못하고 프로그램을 종료한다.



1609





```
while( i < 3 )  
{  
    meter = i * 1609;  
    printf("%d 마일은%d 미터입니다\n", i, meter);  
    i++;  
}
```



```
while( i < 3 )  
{  
    meter = i * 1609;  
    printf("%d 마일은%d 미터입니다\n", i, meter);  
    i++;  
}
```



```
while( i < 3 )  
{  
    meter = i * 1609;  
    printf("%d 마일은%d 미터입니다\n", i, meter);  
    i++;  
}
```



조건식이 거짓이 되어 반복종단

```
while( i < 3 )  
{  
    meter = i * 1609;  
    printf("%d 마일은%d 미터입니다\n", i, meter);  
    i++;  
}
```



예제

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

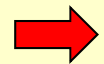
```
int main(void)  
{
```

```
    int n;  
    int i = 1;
```

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

```
    while (i <= 9)
```

```
    {
```



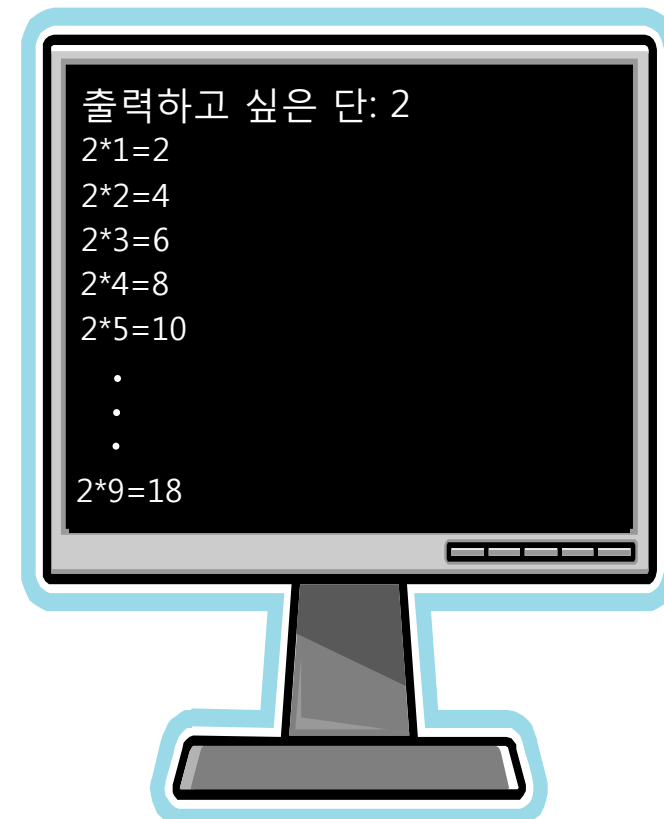
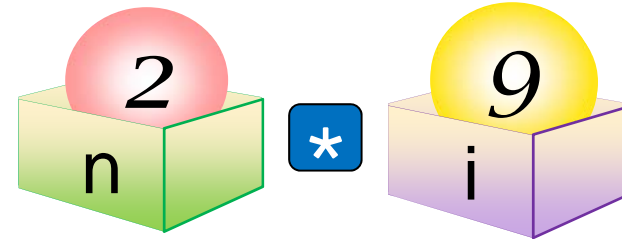
```
        printf("%d*d = %d \n", n, i, n*i);
```

```
        i++; // i = i+1 과 동일
```

```
    }
```

```
    return 0;
```

```
}
```





예제

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

```
int main(void)  
{
```

```
    int n;
```

```
    printf("=====\n");
```

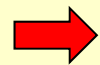
```
    printf("  n      n의 제곱 \n");
```

```
    printf("=====\n");
```

```
    n = 1;
```

```
    while (n <= 10)
```

```
    {
```



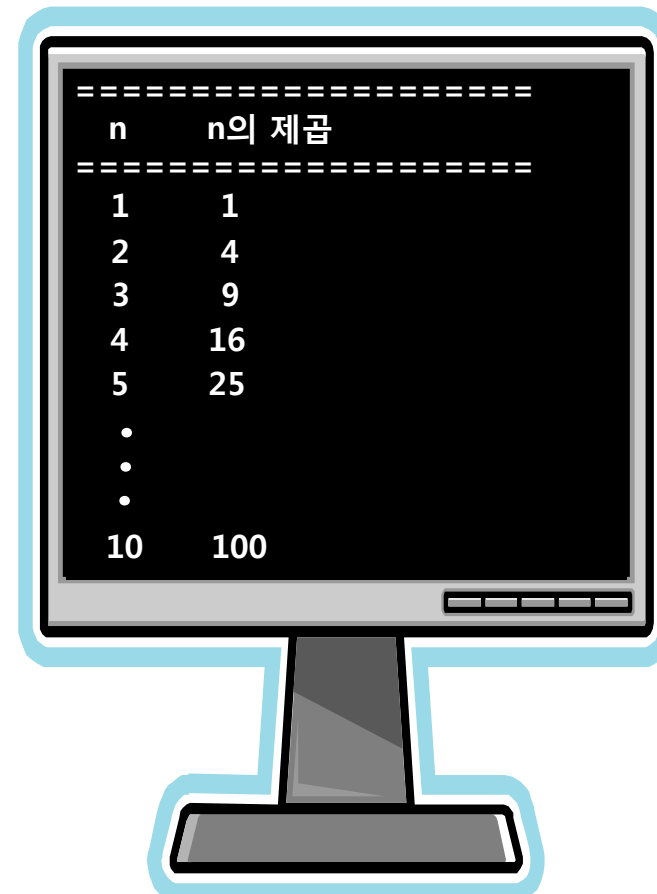
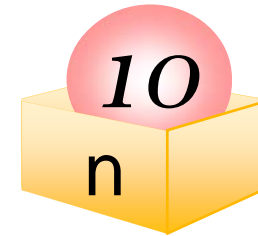
```
        printf("%5d  %5d\n", n, n*n);
```

```
        n++; // n= n+1 과 동일
```

```
    }
```

```
    return 0;
```

```
}
```





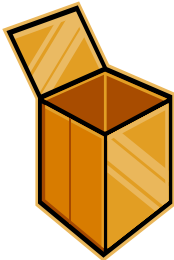
예제

- 1부터 n 까지의 합을 계산하는 프로그램

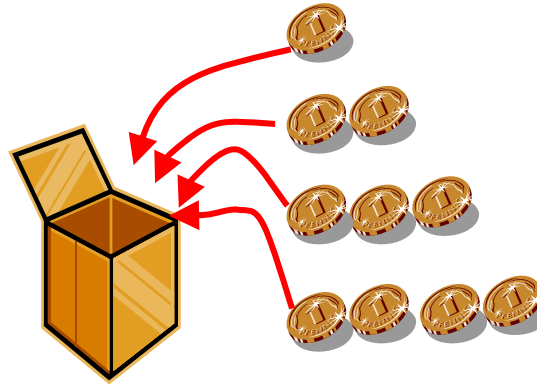
$$1 + 2 + 3 + \dots + n$$

- n 이 무엇이 될지 모르는 경우라면 다음과 같이 계산

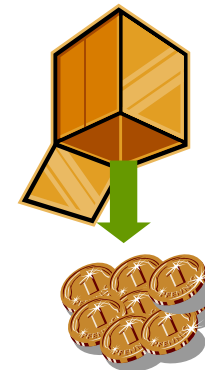
① 빈통을 준비한다.



② 통에 1부터 n 까지를 넣는다.



③ 통에 들어 있는 동전의 개수를 출력한다.





예제

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

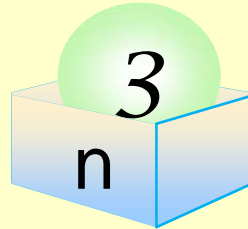
```
int main(void)
```

```
{
```

```
    int i, n, sum; // 변수 선언
```

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

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



```
    while(i <= n)
```

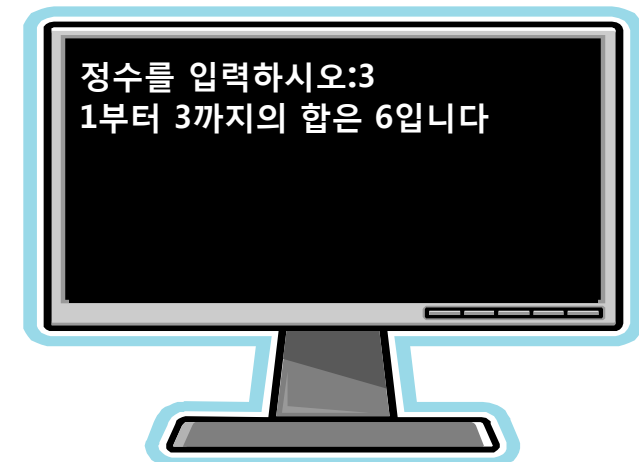
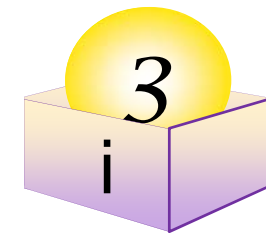
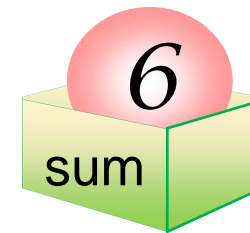
```
    {
```

```
        ➔ sum += i; sum = sum + i 와 동일  
        i++; // i = i + 1 와 동일
```

```
    }
```

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

```
}
```





예제

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

```
int main(void)  
{
```

```
    int i, n, sum;
```

```
    i = 0;
```

```
    // 변수 초기화
```

```
    sum = 0;
```

```
    // 변수 초기화
```

```
    while (i < 3)
```

```
    {
```



```
        printf("값을 입력하시오: ");
```

```
        scanf("%d", &n);
```

```
        sum = sum + n; // sum += n;과 같다.
```

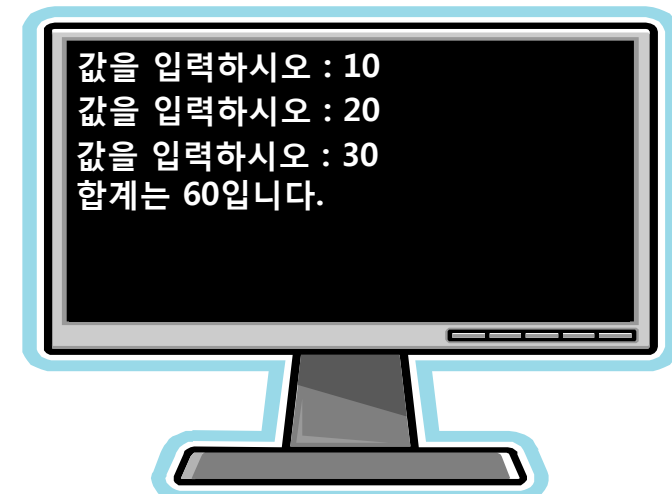
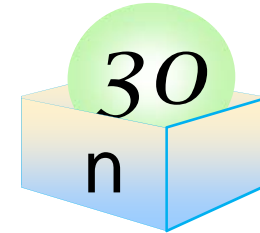
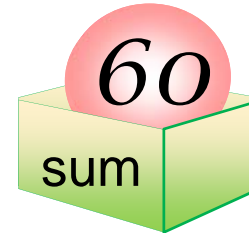
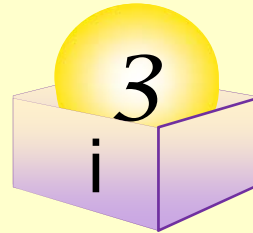
```
        i++;
```

```
    }
```

```
    printf("합계는 %d입니다.\n", sum);
```

```
    return 0;
```

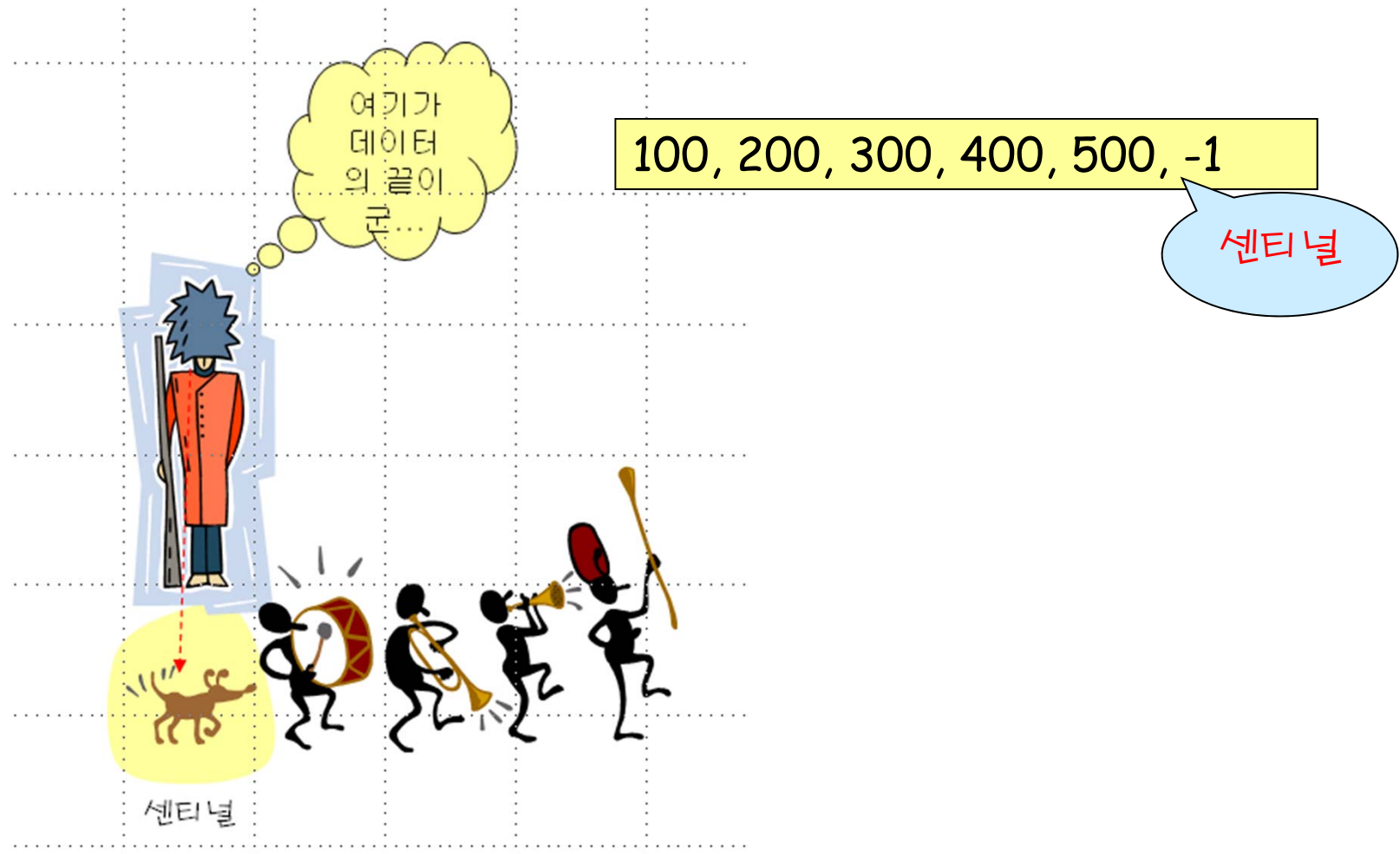
```
}
```





센티널(보초값의 이용)

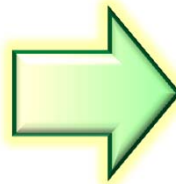
- 센티널: 입력되는 데이터의 끝을 알리는 특수한 값





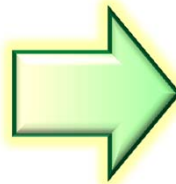
성적들의 평균을 구하는 문제

- 성적의 평균을 구한다.



1. 필요한 변수들을 초기화한다.
2. 성적을 입력 받아서 합계를 구하고 성적의 개수를 센다.
3. 평균을 계산하고 화면에 출력한다.

1. 필요한 변수들을 초기화한다.

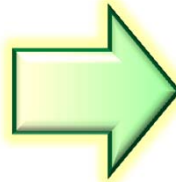


- (1) sum을 0으로 초기화한다.
- (2) n을 0으로 초기화한다.
- (3) grade를 0으로 초기화한다.



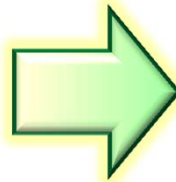
성적들의 평균을 구하는 문제

2. 성적을 입력 받아서
합계를 구하고 성적의
개수를 센다.



while 성적이 0보다 작지 않으면
(1) 사용자로부터 성적을 읽어서
grade에 저장한다.
(2) sum에 이 점수를 누적한다.
(3) n을 하나 증가한다.

3. 평균을 계산하고
화면에 출력한다.



(1) sum을 n으로 나누어서
average에 저장한다.
(2) average를 화면에 출력한다.



센티넬 예제 1/2

```
// while 문을 이용한 성적의 평균 구하기 프로그램
#include <stdio.h>

int main(void)
{
    int grade, n;
    float sum, average;

    // 필요한 변수들을 초기화한다.
    n = 0;
    sum = 0;
    grade = 0;

    printf( " 종료 시 음수 입력\n");
```

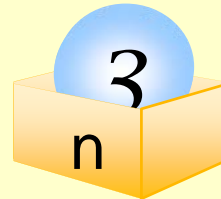


센티넬 예제 2/2

// 성적을 입력 받아서 합계를 구하고 학생 수를 센다.

```
while (grade >= 0)
{
    printf("성적을 입력하시오: ");
    scanf("%d", &grade);

    sum += grade;
    n++;
}
```



sum = sum - grade; // 마지막 데이터를 제거한다.

n--; // 마지막 데이터를 제거한다.

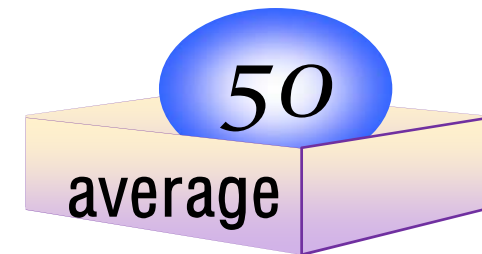
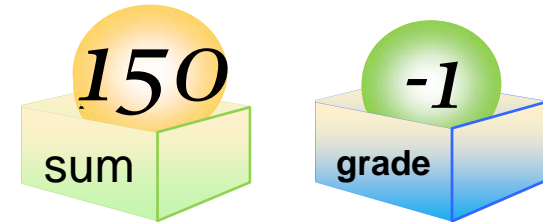
// 평균을 계산하고 화면에 출력한다.

average = sum / n;

printf("성적의 평균은 %f입니다.\n", average);

return 0;

}



```
종료 시 음수 입력
성적을 입력하시오 : 30
성적을 입력하시오 : 50
성적을 입력하시오 : 70
성적을 입력하시오 : -1
성적의 평균은 50.000000입니다.
```



예제: 최대값

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

int main(void)
{
    int number, min_value = INT_MAX;
    printf("정수를 입력하시오\n종료는 Ctrl+z\n");
    while(scanf("%d", &number) != EOF)
    {
        if( number < min_value )
            min_value = number;
    }
    printf("최소값은 %d", min_value);
    return 0;
}
```

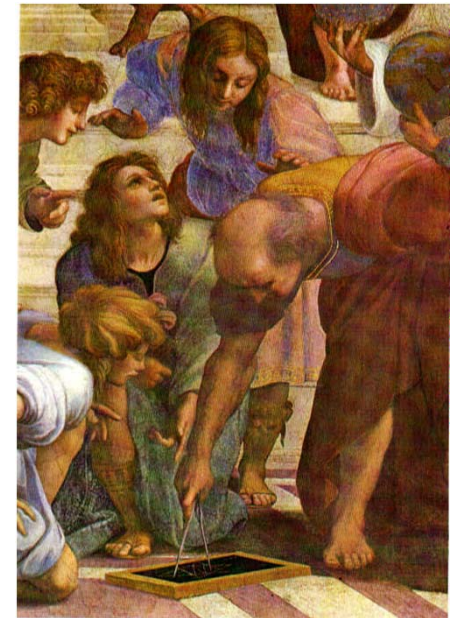




예제

- 두수의 최대 공약수 구하기
- 유클리드 알고리즘

- ① 두 수 가운데 큰 수를 x , 작은 수를 y 라 한다.
- ② y 가 0이면 공약수는 x 와 같다.
- ③ $r \leftarrow x \% y$
- ④ $x \leftarrow y$
- ⑤ $y \leftarrow r$
- ⑥ 단계 ②로 되돌아간다.





예제

// while 문을 이용한 최대 공약수 구하기 프로그램

#include <stdio.h>

int main(void)

{

int x, y, r;

printf("두 정수를 입력 (큰 수, 작은 수): ");

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

while (y != 0)

{

r = x % y;

x = y;

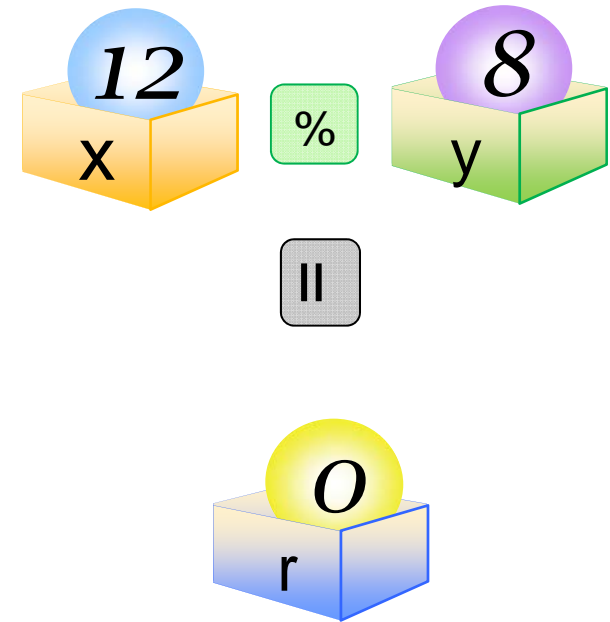
y = r;

}

printf("최대 공약수는 %d입니다.\n", x);

return 0;

}



두 정수를 입력 (큰 수, 작은 수): 12 8
최대공약수는 4입니다.



if 문과 while 문의 비교

```
if( 조건 )  
{  
  ...  
  ...  
}
```

조건이 만족되면
한번만 실행
된다.

```
while( 조건 )  
{  
  ...  
  ...  
}
```

조건이 만족되면
여러 번 반복 실행
된다.



while 문에서 주의할 점

```
int i = 1;
while(i < 10)
{
    printf("반복중입니다\n");
    i--;
}
```

변수가 증가 아니라 감소

```
int i = 0;
while(i < 3)
    printf("반복중입니다\n");
    i++;
```

반복 루프에 포함되어 있지 않다.

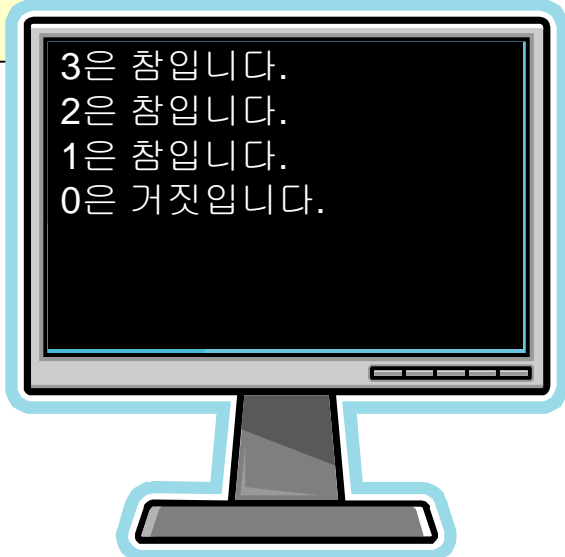
```
int i = 0;
while(i < 3) ;
{
    printf("반복중입니다\n");
    i++;
}
```

조건뒤에 ;이 있음



참과 거짓

```
#include <stdio.h>
int main(void)
{
    int i = 3;
    while (i)
    {
        printf("%d은 참입니다.", i);
        i--;
    }
    printf("%d은 거짓입니다.", i);
}
```



3은 참입니다.
2은 참입니다.
1은 참입니다.
0은 거짓입니다.



중간 점검

1. **if** 문과 **while** 문을 비교하여 보라. 똑같은 조건이라면 어떻게 동작하는가?
2. **while** 루프를 이용하여 무한 루프를 만들어 보라.
3. 다음 코드의 출력을 쓰시오.

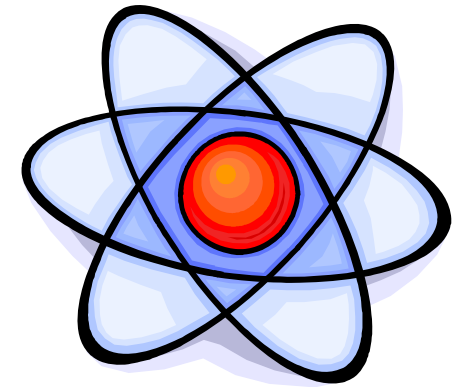
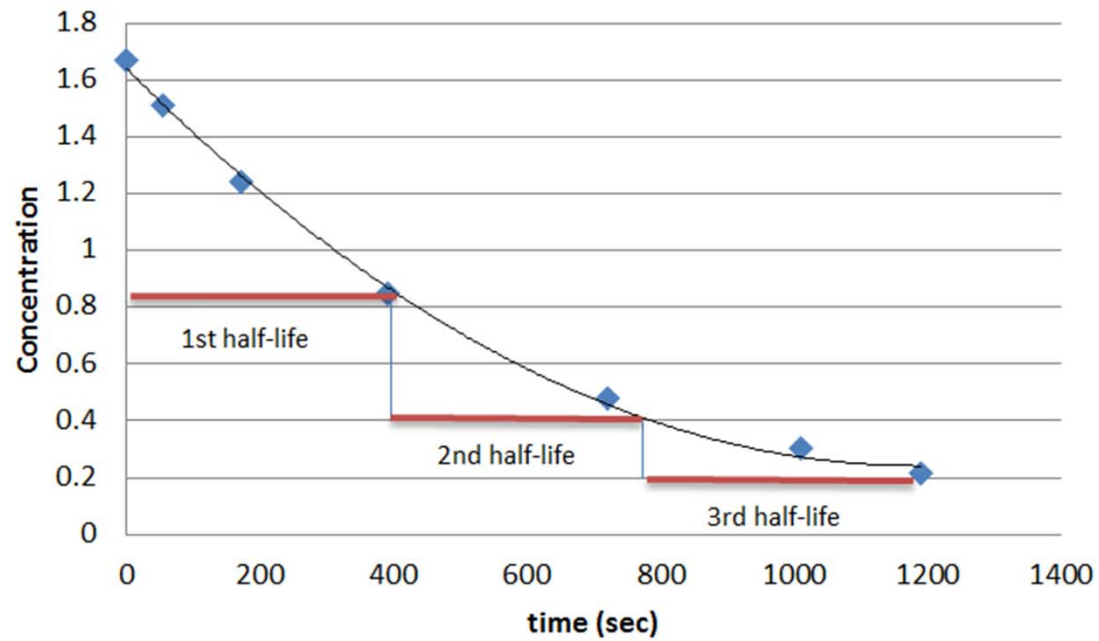
```
int n = 10;
while (n > 0) {
    printf("%d\n", n);
    n = n - 3;
}
```





실습: 반감기

- 반감기: 방사능 물질의 양이 $\frac{1}{2}$ 로 되는 시간





실행 결과

반감기를 입력하시오(년):
10
10년 후에 남은 양
=50.000000
20년 후에 남은 양
=25.000000
30년 후에 남은 양
=12.500000
40년 후에 남은 양
=6.250000
1/10 이하로 되기 까지 걸린
시간=40년

- 단 로그 함수는 사용하지 않는다!
- 반복문을 사용한다.



알고리즘

- 사용자로부터 반감기를 입력받는다.
- *while*(물질의 양 > 초기 물질의 양*0.1)
- 반감기만큼 시간을 더한다.
- 물질의 양은 **1/2**로 줄어든다.
- 현재 물질의 양을 출력한다.
- **10%** 이하로 되기까지 걸린 시간을 출력한다.



소스

```
#include <stdio.h>
int main(void)
{
    int halflife;
    double initial;
    double current;
    int years=0;

    printf("반감기를 입력하시오(년): ");
    scanf("%d", &halflife);

    initial = 100.0;
    current = initial;
    while( current > initial/10.0 )

        years += halflife;
        current = current / 2.0;
        printf("%d년 후에 남은 양=%f", years, current);

    printf("1/10 이하로 되기까지 걸린 시간=%d년", years);
    return 0;
}
```



도전문제

- 위와 비슷한 문제를 하나 더 작성해보자. 세균이 1시간마다 4배씩 증가한다고 가정하자. 이 세균 10마리를 배양하면 7시간 후의 세균의 수는 얼마나 될까? 역시 지수 함수나 로그 함수를 이용하지 말고 반복 구조만을 사용하여 해결하여 보자.
- 종이를 한번 접으면 면적이 $1/2$ 로 줄어든다. 종이를 몇 번 접어야 원래 면적의 $1/100$ 로 줄어드는가? 역시 로그 함수나 지수 함수를 사용하지 말고 반복 구조를 이용하여서 해결하여 보자.

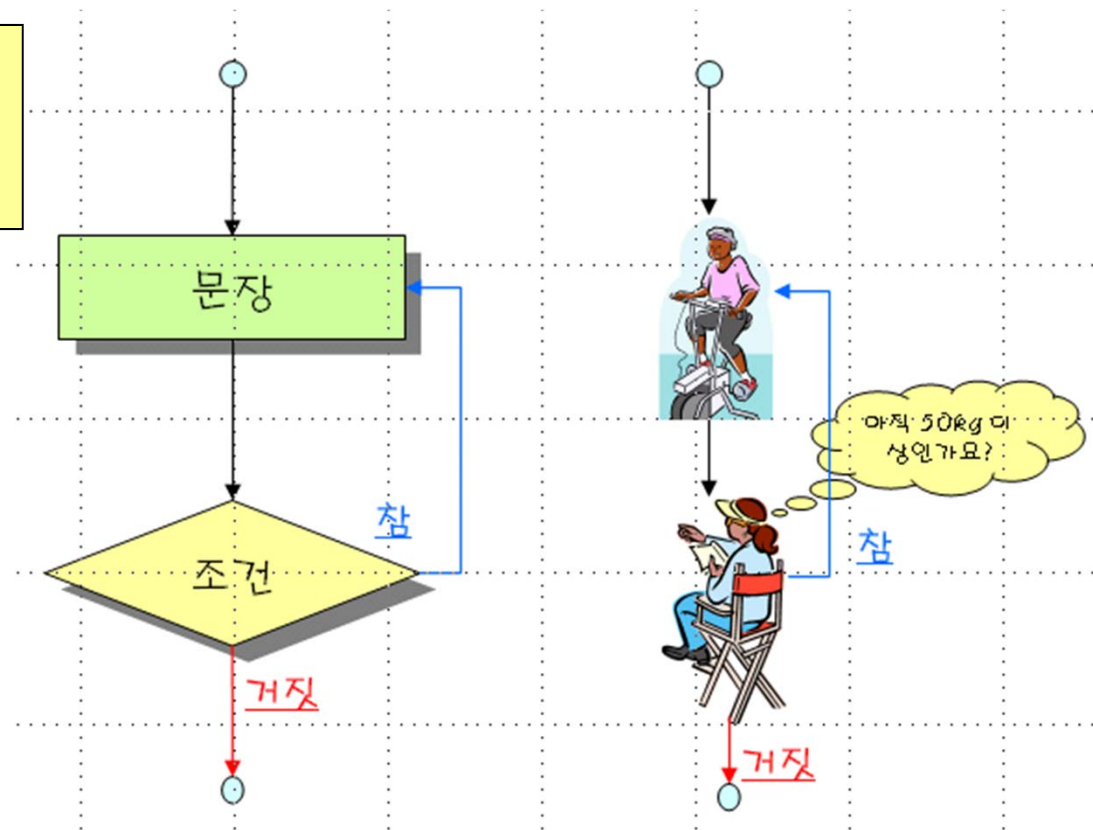




do...while문

- 반복 조건을 루프의 끝에서 검사

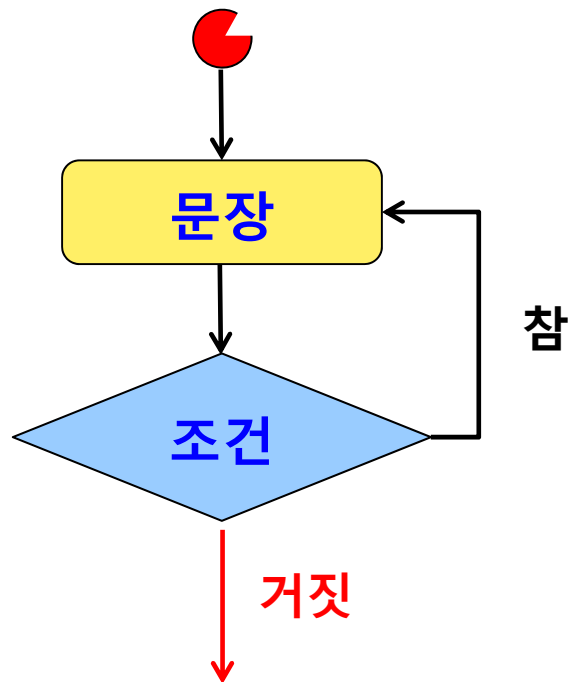
do
문장
while(조건)





do-while 문

- 적어도 한번은 반복문장을 실행한다.





예제

// do..while 문을 이용한 메뉴

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

```
int main(void)
```

```
{
```

```
    int i = 0;
```

```
    do
```

```
    {
```



```
        printf("1---새로 만들기\n");
```

```
        printf("2---파일 열기\n");
```

```
        printf("3---파일 닫기\n");
```

```
        printf("하나를 선택 하시오:\n");
```

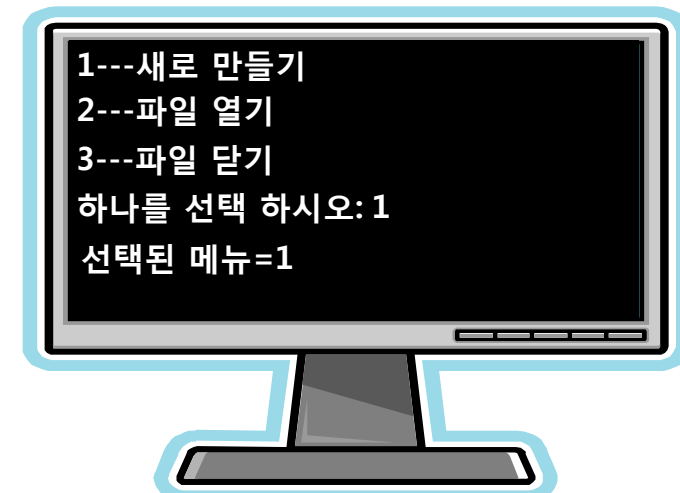
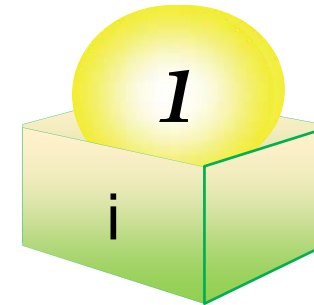
```
        scanf("%d", &i);
```

```
    } while(i < 1 || i > 3);
```

```
    printf("선택된 메뉴=%d\n", i);
```

```
    return 0;
```

```
}
```





중간 점검

1. 다음 코드의 출력을 쓰시오.

```
int n = 0;  
do {  
    printf("%d\n", n);  
    n = n + 3;  
} while(n < 10);
```





실습: 숫자 추측 게임

- 프로그램이 가지고 있는 정수를 사용자가 알아맞히는 게임
- 사용자가 답을 제시하면 프로그램은 자신이 저장한 정수와 비교하여 제시된 정수가 더 높은지 낮은지 만을 알려준다.





실행 결과

정답을 추측하여 보시오: 10
제시한 정수가 낮습니다.
정답을 추측하여 보시오: 30
제시한 정수가 낮습니다.
정답을 추측하여 보시오: 60
제시한 정수가 높습니다.
정답을 추측하여 보시오: 59
축하합니다. 시도횟수=4



알고리즘

- *do*
- 사용자로부터 숫자를 *guess*로 입력받는다.
- 시도횟수를 증가한다.
- *if(guess < answer)*
- 숫자가 낮다고 출력한다.
- *if(guess > answer)*
- 숫자가 높다고 출력한다.
- *while(guess != answer);*
- “축하합니다”와 시도횟수를 출력한다.



소스

```
#include <stdio.h>

int main(void)
{
    int answer =59;    // 정답
    int guess;
    int tries = 0;
    // 반복 구조
    do {
        printf("정답을 추측하여 보시오: ");
        scanf("%d", &guess);
        tries++;

        if (guess >answer) // 사용자가 입력한 정수가 정답보다 높으면
            printf("제시한 정수가 높습니다.");
        if (guess <answer) // 사용자가 입력한 정수가 정답보다 낮으면
            printf("제시한 정수가 낮습니다.");
    } while (guess !=answer);
    printf("축하합니다. 시도횟수=%d", tries);
    return 0;
}
```




도전문제

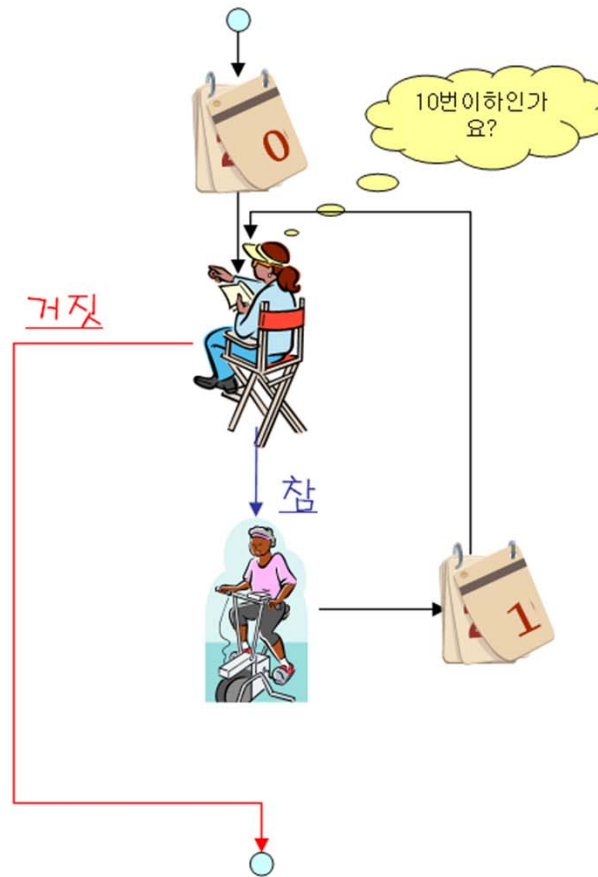
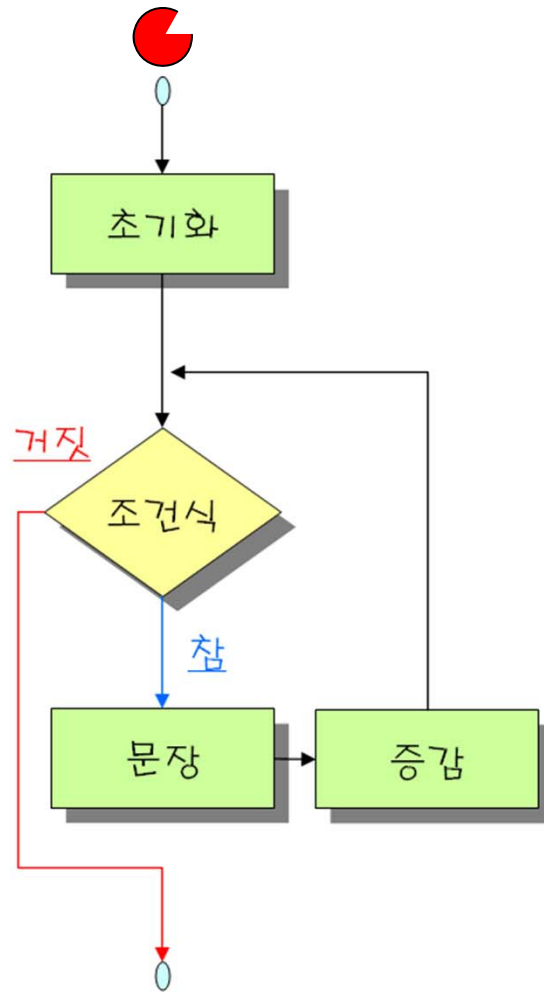
- 위의 프로그램이 게임이 되려면 난수를 발생시키는 것이 좋다. 난수는 $(\text{rand()} \% 100)$ 으로 발생이 가능하다. `stdlib.h` 헤더 파일도 포함시켜야 한다.





for 루프

- 정해진 횟수만큼 반복하는 구조



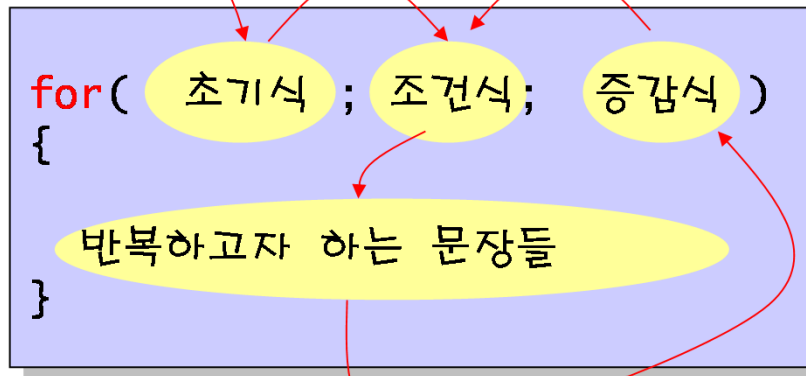


for 문의 구조

for (초기식; 조건식; 증감식)

문장;

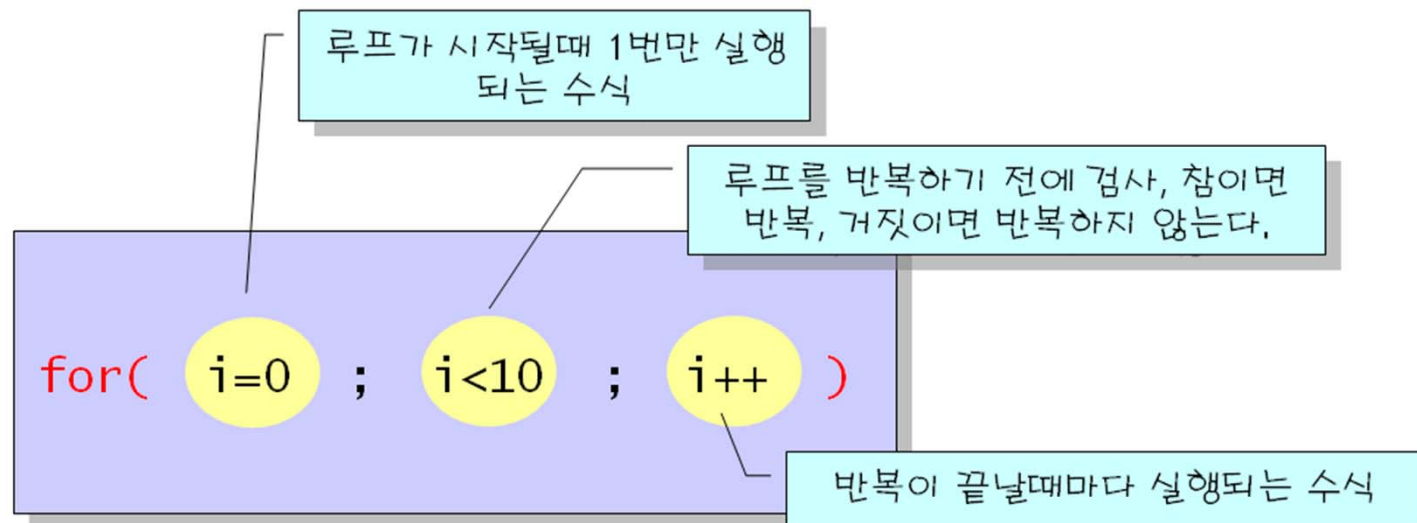
- ① 초기식을 실행한다.
- ② 반복 조건을 나타내는 조건식을 계산한다.
- ③ 수식의 값이 거짓이면 **for** 문의 실행이 종료된다.
- ④ 수식의 값이 참이면 문장이 실행된다.
- ⑤ 증감식을 실행하고 ②로 돌아간다.





for 문의 구조

```
int i;  
for(i = 0; i < 10; i++)  
    printf("Hello World!\n");
```





for문의 실행과정

```
for( i=0 ; i<10 ; i++ )  
    printf("Hello world!\n");
```

1번째 루프
i값은 0

```
for( i=0 ; i<10 ; i++ )  
    printf("Hello world!\n");
```

2-10번째 루프
i값은



...

...

```
for( i=0 ; i<10 ; i++ )  
    printf("Hello world!\n");
```

11번째 루프
i값은 10



for문의 실행과정

```
int i;  
for(i = 0; i < 10; i++)  
    printf("Hello World!\n");
```

반복횟수	i의 값	i < 10	반복여부
1번째 반복	0	참	반복
2번째 반복	1	참	반복
3번째 반복	2	참	반복
4번째 반복	3	참	반복
5번째 반복	4	참	반복
6번째 반복	5	참	반복
7번째 반복	6	참	반복
8번째 반복	7	참	반복
9번째 반복	8	참	반복
10번째 반복	9	참	반복
11번째 반복	10	거짓	중지





예제

// 반복을 이용한 정수합 프로그램

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

```
int main(void)
```

```
{
```

```
    int i, sum;
```

```
    sum = 0;
```

```
    for(i = 1; i <= 10; i++)
```

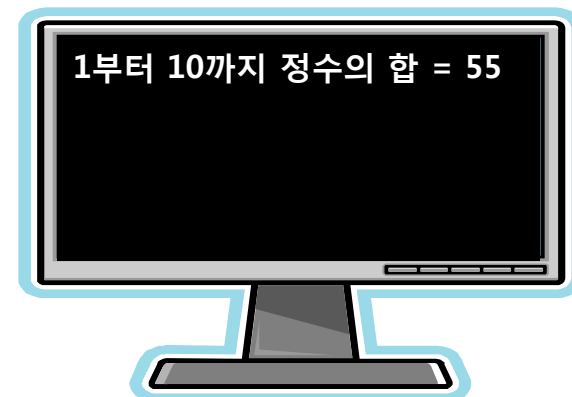
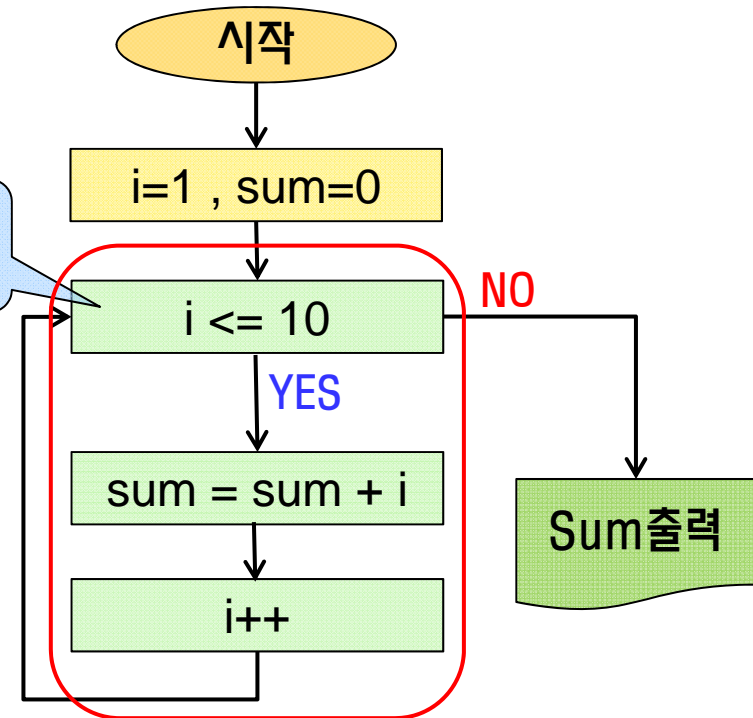
```
        sum += i; // sum = sum + i;와 같음
```

```
    printf("1부터 10까지 정수의 합 = %d\n", sum);
```

```
    return 0;
```

```
}
```

i가 10보다 작거나 같은 때
까지 10번 반복





예제

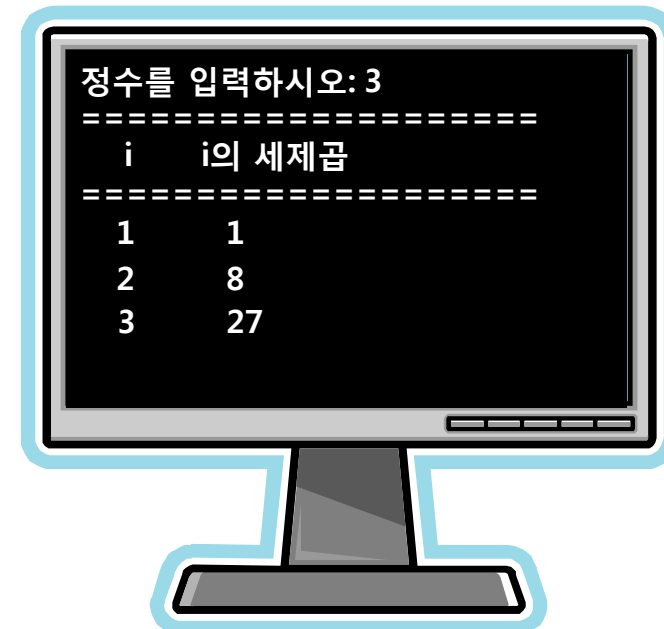
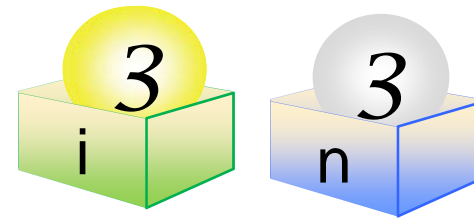
```
// 반복을 이용한 세제곱값구하기
#include <stdio.h>

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    %5d\n", i, i*i*i);

    return 0;
}
```





예제

```
// 반복을 이용한 네모 그리기
#include <stdio.h>

int main(void)
{
    int i;
    printf("*****");

    for(i = 0; i < 5; i++)
        printf("**");

    printf("*****");

    return 0;
}
```





예제

// 반복을 이용한 팩토리얼 구하기

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

```
int main(void)
{
```

```
    long fact=1;
```

```
    int i, n;
```

```
    printf("정수를 입력하시오:");
```

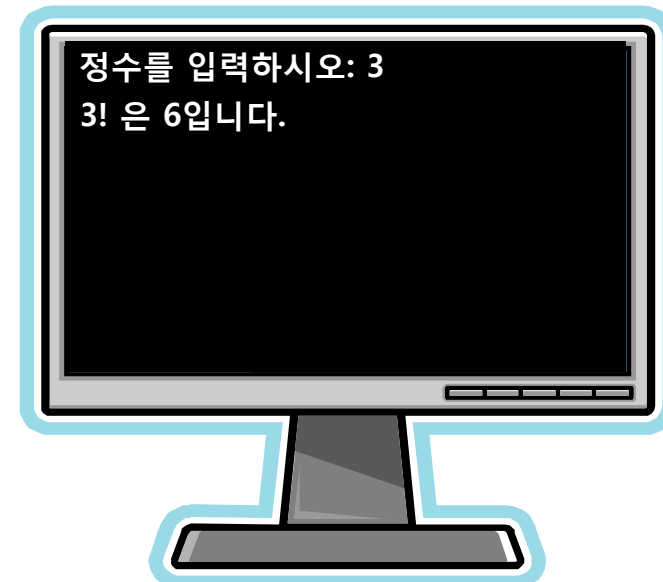
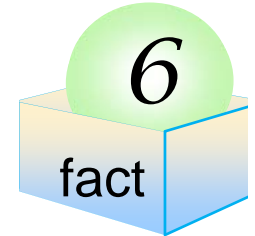
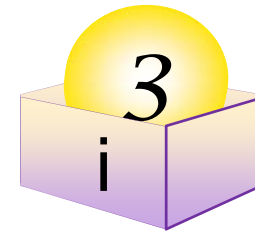
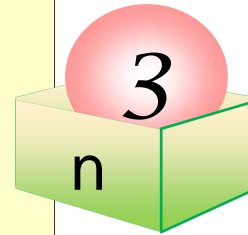
```
    scanf("%d", &n);
```

```
    → for(i = 1; i <= n; i++)
        fact = fact * i;
```

```
    printf("%d! 은 %d입니다.\n", n, fact);
```

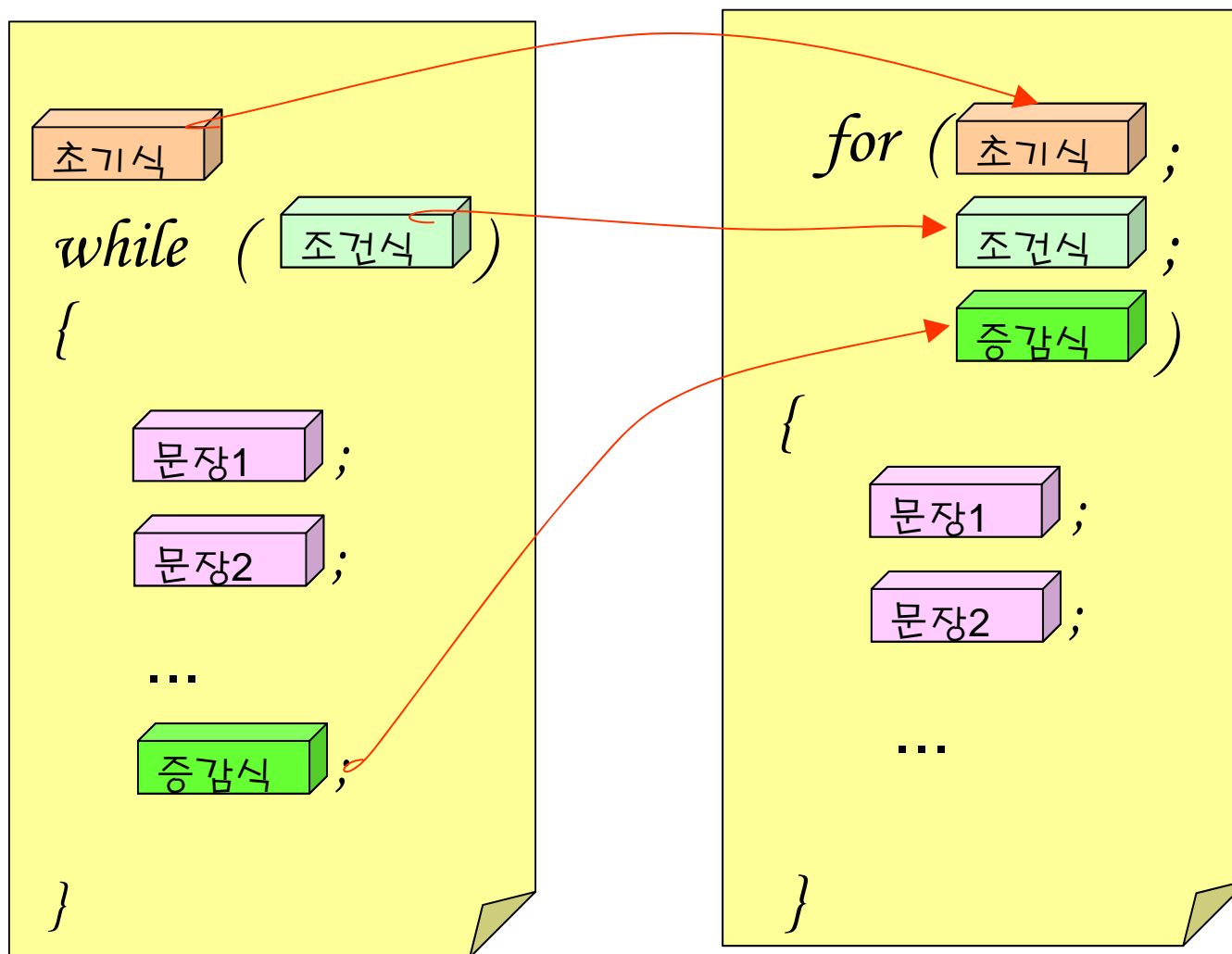
```
    return 0;
```

```
}
```





while 루프와 for 루프와의 관계





팩토리얼 계산 예제(while 버전)

```
// 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;
}
```



다양한 증감수식의 형태

```
for (i = 10; i > 0; i-- )  
    printf("Hello World!\n");
```

뺄셈 사용

```
for (i = 0; i < 10; i += 2 )  
    printf("Hello World!\n");
```

2씩 증가

```
for (i = 1; i < 10; i *= 2 )  
    printf("Hello World!\n");
```

2를 곱한다.

```
for (i = 0; i < 100; i = (i * i) + 2 )  
    printf("Hello World!\n");
```

어떤 수식이라도 가능



다양한 for 루프

```
for ( ; i < 100; i++ )  
    printf("Hello World!\n");
```

한부분이 없을 수도 있다.

```
for ( i = 0, k = 0; i < 100; i++ )  
    printf("Hello World!\n");
```

2개 이상의 변수 초기화

```
for ( printf("반복시작"), i = 0; i < 100; i++ )  
    printf("Hello World!\n");
```

어떤 수식도 가능

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

무한 반복 루프



중간 점검

1. 다음 코드의 출력을 쓰시오.

```
for(i = 1; i < 5; i++)  
    printf("%d ", 2 * i);
```

2. 다음 코드의 출력을 쓰시오.

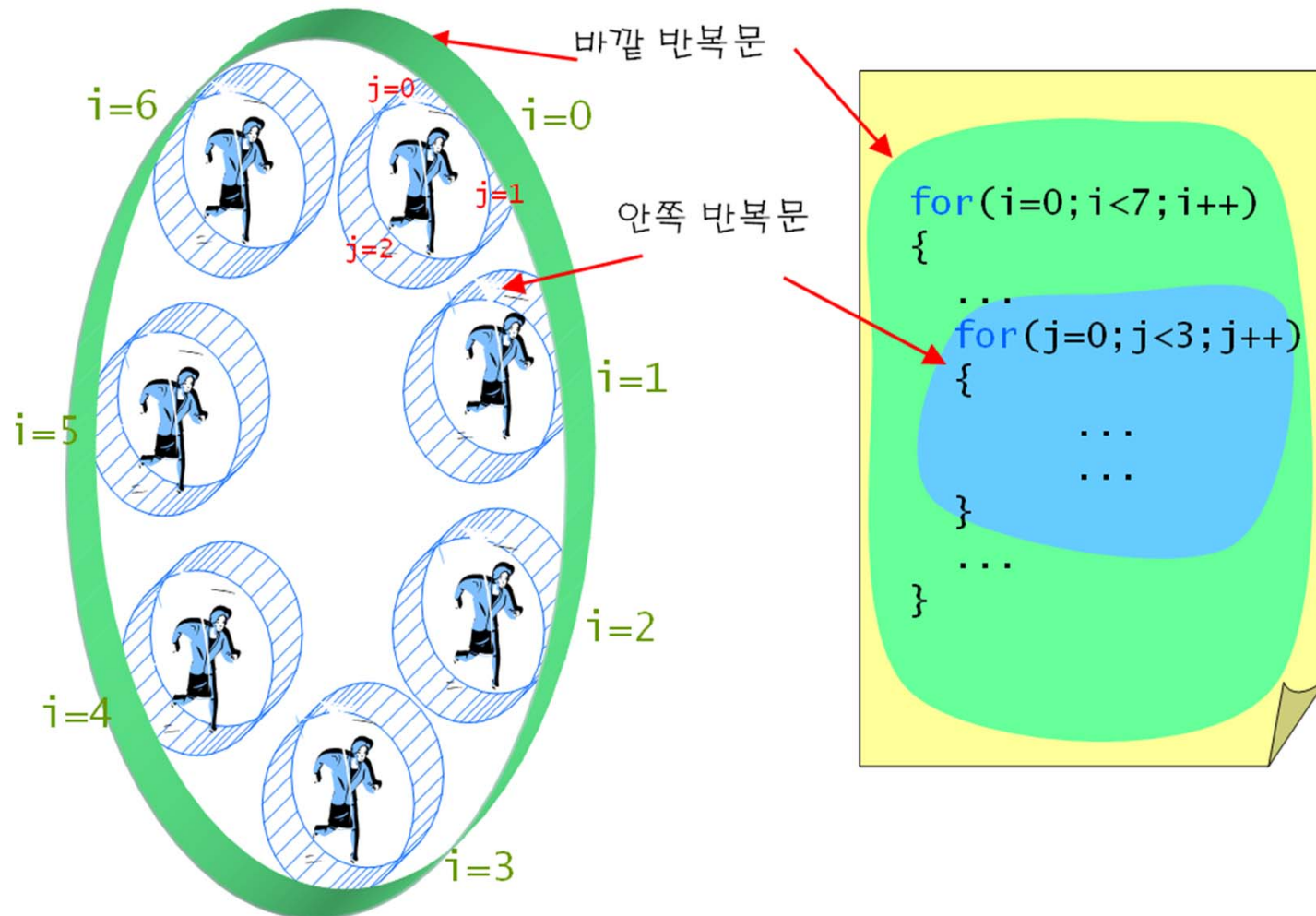
```
for(i = 10; i > 0; i = i - 2)  
    printf("Student%d\n", i);
```





중첩 반복문

- 중첩 반복문(nested loop): 반복문 안에 다른 반복문이 위치





예제

```
// 중첩 for 문을 이용하여 *기호를 사각형  
// 모양으로 출력하는 프로그램
```

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

```
int main(void)
```

```
{
```

```
    int x, y;
```

```
    for(y = 0; y < 5; y++)
```

②

```
    {
```

```
        for(x = 0; x < 10; x++)
```

①

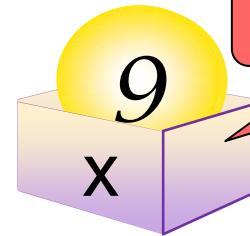
```
            printf("*");
```

```
            printf("\n");
```

```
    }
```

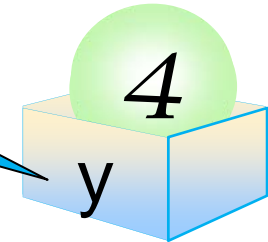
```
    return 0;
```

```
}
```



0부터 9까지 10번 반복

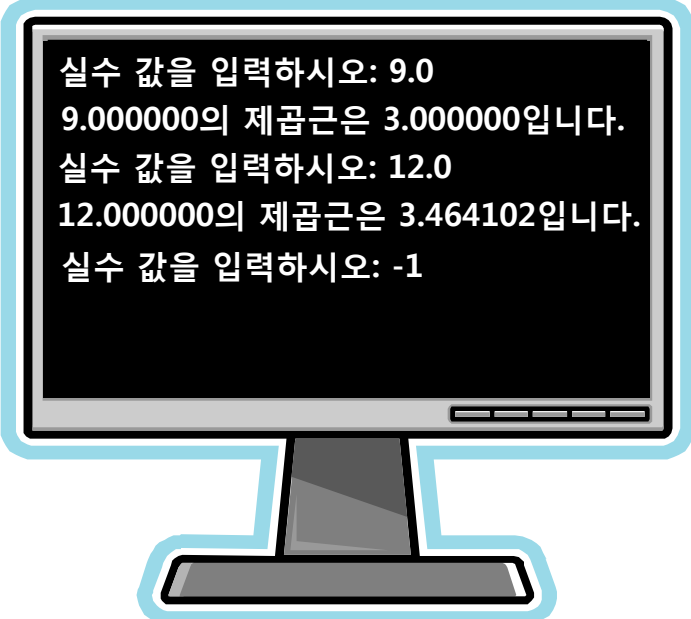
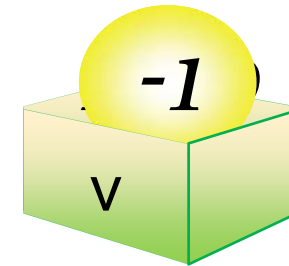
0부터 4까지 5번 반복





예제

```
// break를 이용하여 무한루프를 탈출한다.  
#include <stdio.h>  
#include <math.h>  
  
int main(void)  
{  
    double v;  
  
    while(1)  
    {  
        printf("실수 값을 입력하시오: ");  
        scanf("%lf", &v);  
        if( v < 0.0 )  
            break;  
        printf("%f의 제곱근은 %f\n", v, sqrt(v));  
    }  
  
    return 0;  
}
```



```
실수 값을 입력하시오: 9.0  
9.000000의 제곱근은 3.000000입니다.  
실수 값을 입력하시오: 12.0  
12.000000의 제곱근은 3.464102입니다.  
실수 값을 입력하시오: -1
```



예제

// break를 이용하여 무한루프를 탈출한다.

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

```
int main(void)
```

```
{
```

```
    float grade, sum = 0.0, average;
```

```
    int count = 0;
```

```
    while(1)
```

```
    {
```

```
        printf("학생 성적을 입력하시오: ");  
        scanf("%f", &grade);
```

```
        if( grade < 0.0 )  
            break;
```

```
        count++;
```

```
        sum += grade;
```

```
    }
```

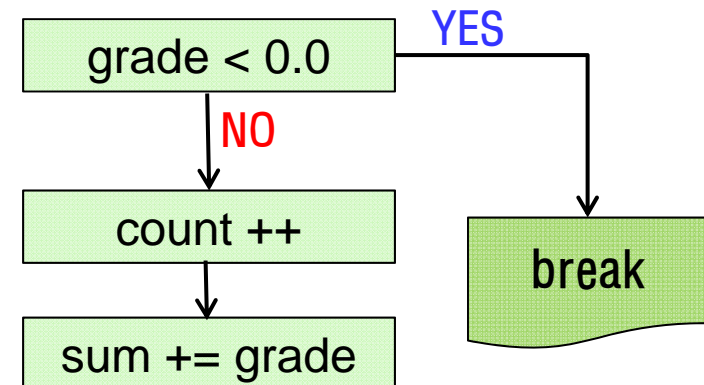
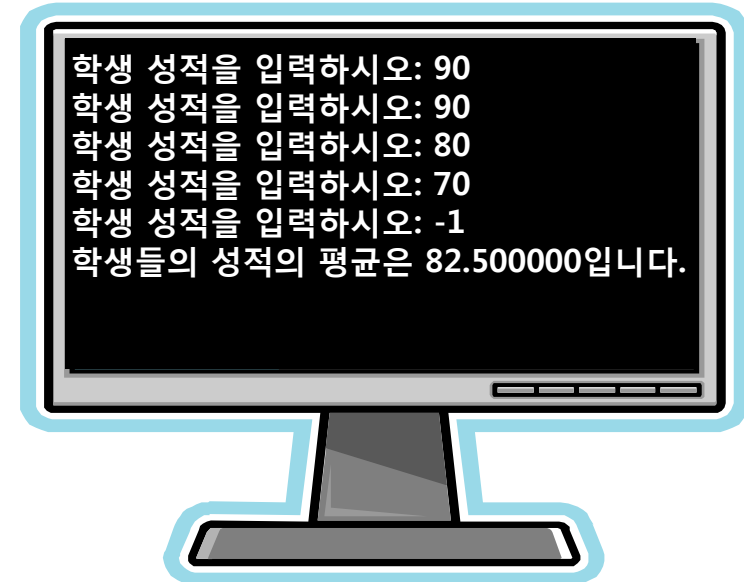
```
    average = sum / count;
```

```
    printf("학생들의 평균은 %f입니다.\n", average);
```

```
    return 0;
```

```
}
```

음수 입력 시
while문을 나옴





예제

```
#include <stdio.h>
int main(void)
{
    int x, y;

    for(y = 1; y <= 5; y++)
    {
        for(x = 0; x < y; x++)
            printf("*");
        printf("\n");
    }
    return 0;
}
```

// 내부 반복문이 종료될 때마다 실행





중간 점검

1. 다음 코드의 출력을 쓰시오.

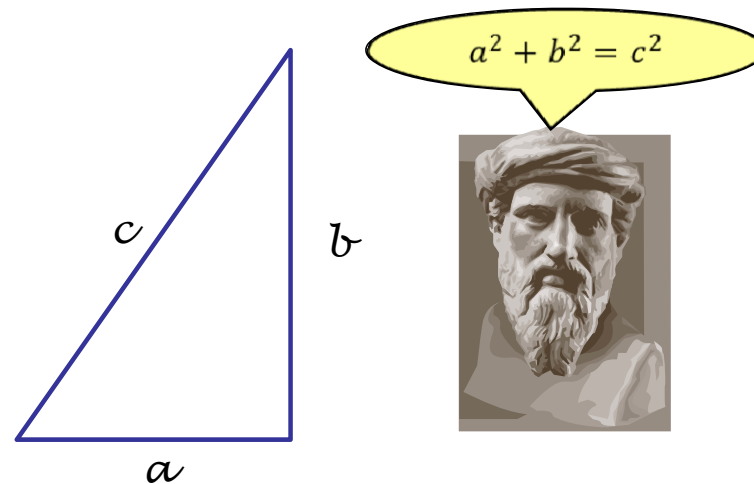
```
for(i = 1; i < 6; i++)  
    for(j = 5; j >= 1; j--)  
        printf("%d 곱하기 %d은 %d\n", i, j, i*j);
```





실습: 직각 삼각형 찾기

- 각 변의 길이가 **100**보다 작은 삼각형 중에서 피타고라스의 정리가 성립하는 직각 삼각형은 몇 개나 있을까?





실행 결과





알고리즘

- *for(a=1;a<=100;a++)*
- *for(b=1;b<=100;b++)*
- *for(c=1;c<=100;c++)*
- *if(a*a + b*b == c*c)*
- *a와 b와 c를 화면에 출력한다.*



소 스

```
#include <stdio.h>
int main(void)
{
    int a, b, c;

    for(a=1; a<=100; a++)
        for(b=1; b<=100; b++)
            for(c=1; c<=100; c++)
                if( (a*a+b*b)==c*c )
                    printf("%d %d %d", a, b, c);

    return 0;
}
```



도전문제

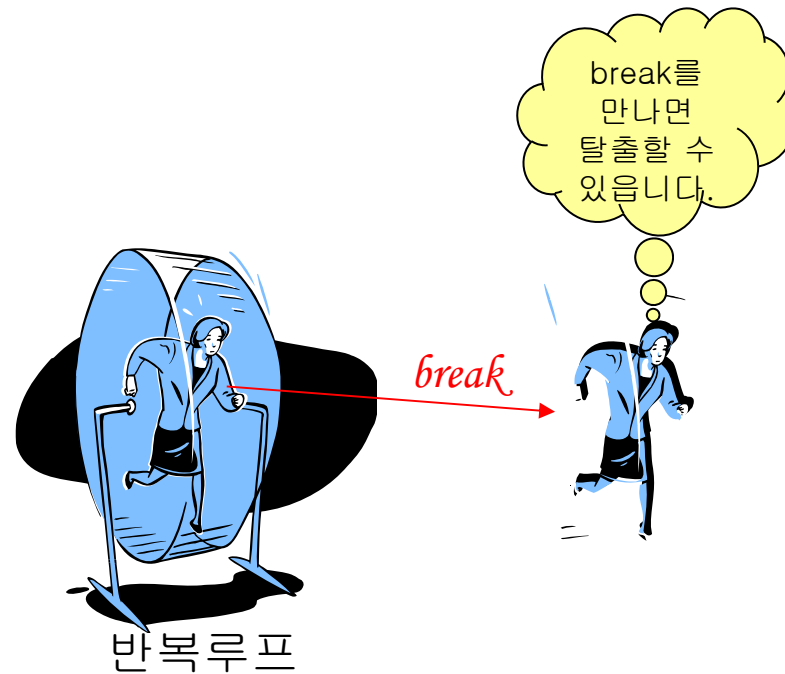
- 위와 비슷한 문제를 하나 더 작성해보자. 라스베가스와 같은 도박장에 가면 주사위 게임이 있다. 주사위 2개를 던졌을 때, 합이 6이 되는 경우를 전부 출력하여 보자. 예를 들어서 (1, 5), (2, 4),...와 같이 출력되면 된다. 또 주사위 3개를 사용하여서 합이 10이 되는 경우를 전부 출력하여 보자.





break 문

- break 문은 반복 루프를 빠져 나오는데 사용된다.





break 문

```
#include <stdio.h>

int main(void)
{
    int i;

    for(i=1; i<100; i++)
    {
        printf("%d\n", i);
        if( i == 5 ) break;
    }

    return 0;
}
```

for 반복 루프



예제

```
#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", year);
    return 0;
}
```

원금의 10배가 되면



예제

// break를 이용하여 무한루프를 탈출한다.

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

```
#include <math.h>
```

```
int main(void)
```

```
{
```

```
    double v;
```

```
    while(1)
```

```
    {
```

```
        printf("실수값을 입력하시오: ");
```

```
        scanf("%lf", &v);
```

```
        if( v < 0.0 )
```

```
            break;
```

```
        printf("%f의 제곱근은 %f입니다.", v, sqrt(v));
```

```
    return 0;
```

```
}
```

실수값을 입력하시오: 9.0

9.000000의 제곱근은

3.000000입니다.

실수값을 입력하시오: 25.0

25.000000의 제곱근은

5.000000입니다.

실수값을 입력하시오: -1

음수이면 빠져나간다



goto문의 사용

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

```
int main(void)
```

```
{
```

```
    int x, y;
```

```
    for(y = 1; y < 10000; y++)
```

```
    {
```

```
        for(x = 1; x < 50; x++)
```

```
        {
```

```
            if( _kbhit() ) goto OUT;
```

```
            printf("*");
```

```
        }
```

```
        printf("\n");
```

```
    }
```

```
OUT:
```

```
    return 0;
```

```
}
```

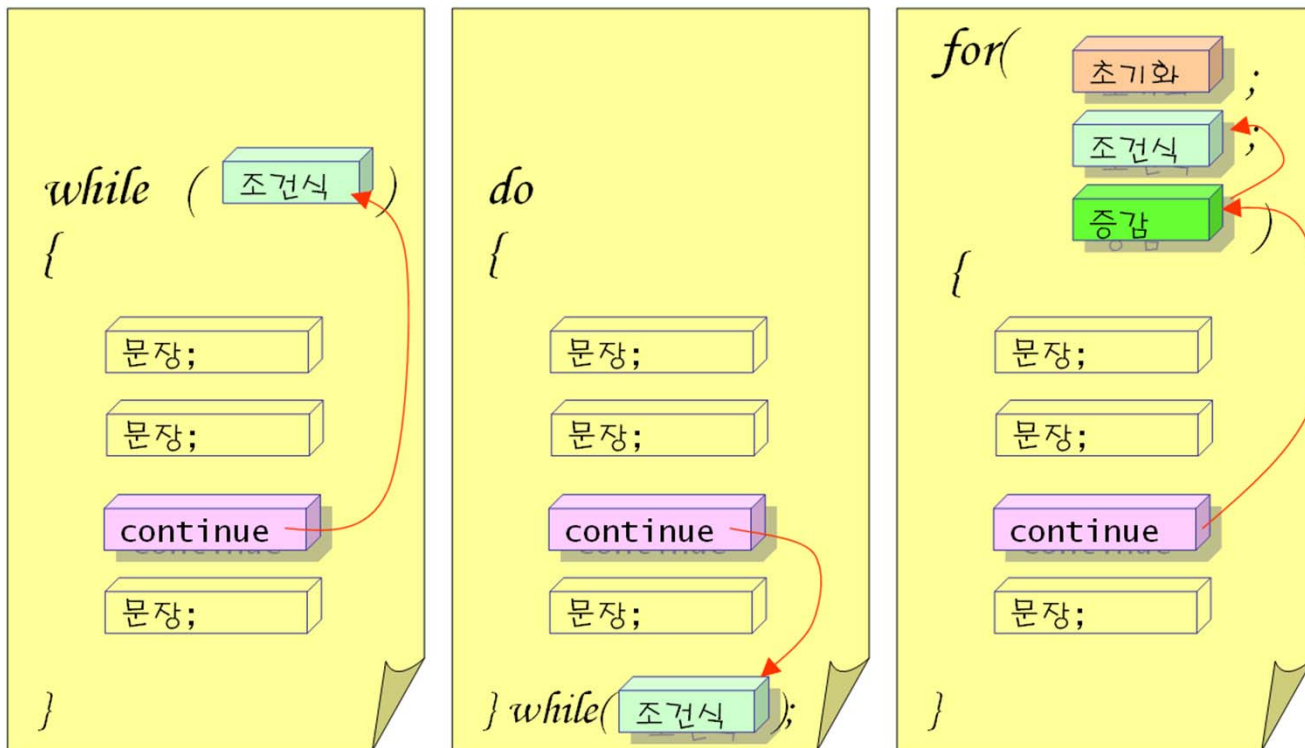
OUT 으로 goto

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



continue 문

- 현재의 반복을 중단하고 다음 반복을 시작하게 한다.





예제

```
#include <stdio.h>
int main(void)
{
    int i;
    for(i=0 ; i<10 ; i++)
        if( i%3 == 0 )
            continue;
        printf("%d ", i);

    return 0;
}
```

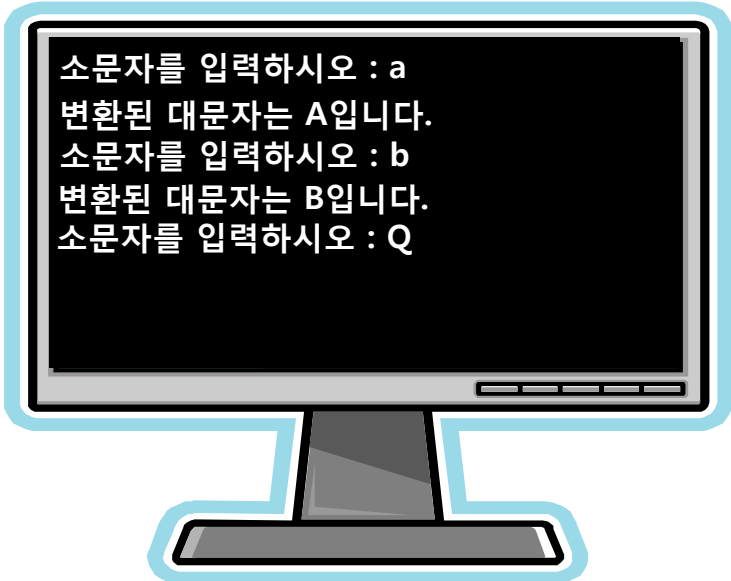
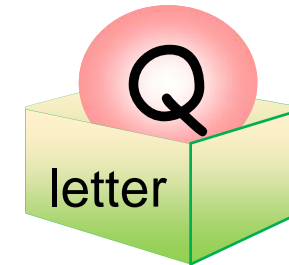
3의 배수 는 건너뛴다.

1 2 4 5 7 8



예제

```
// 소문자를 대문자로 변경한다.  
#include <stdio.h>  
  
int main(void)  
{  
    char letter;  
  
    while(1)  
    {  
        printf("소문자를 입력하시오: ");  
        scanf(" %c", &letter);  
  
        if( letter == 'Q' )  
            break ;  
        if( letter < 'a' || letter > 'z' )  
            continue ;  
  
        letter -= 32;  
        printf("변환된 대문자는 %c입니다.\n", letter);  
    }  
  
    return 0;  
}
```



```
소문자를 입력하시오 : a  
변환된 대문자는 A입니다.  
소문자를 입력하시오 : b  
변환된 대문자는 B입니다.  
소문자를 입력하시오 : Q
```



중간 점검

1. _____ 문이 반복문에서 실행되면 현재의 반복을 중단하고 다음번 반복 처리가 시작된다.
2. _____ 문이 반복문에서 실행되면 반복문을 빠져 나온다.
3. 다음 코드의 출력을 쓰시오.

```
int i;  
for(i = 1; i < 10; i++) {  
    if( i % 3 == 0 ) break;  
    printf("%d\n", i);  
}
```

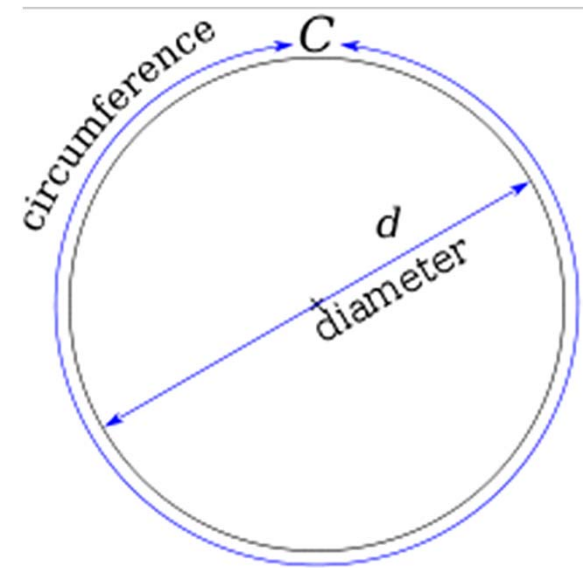




실습: 파이 구하기

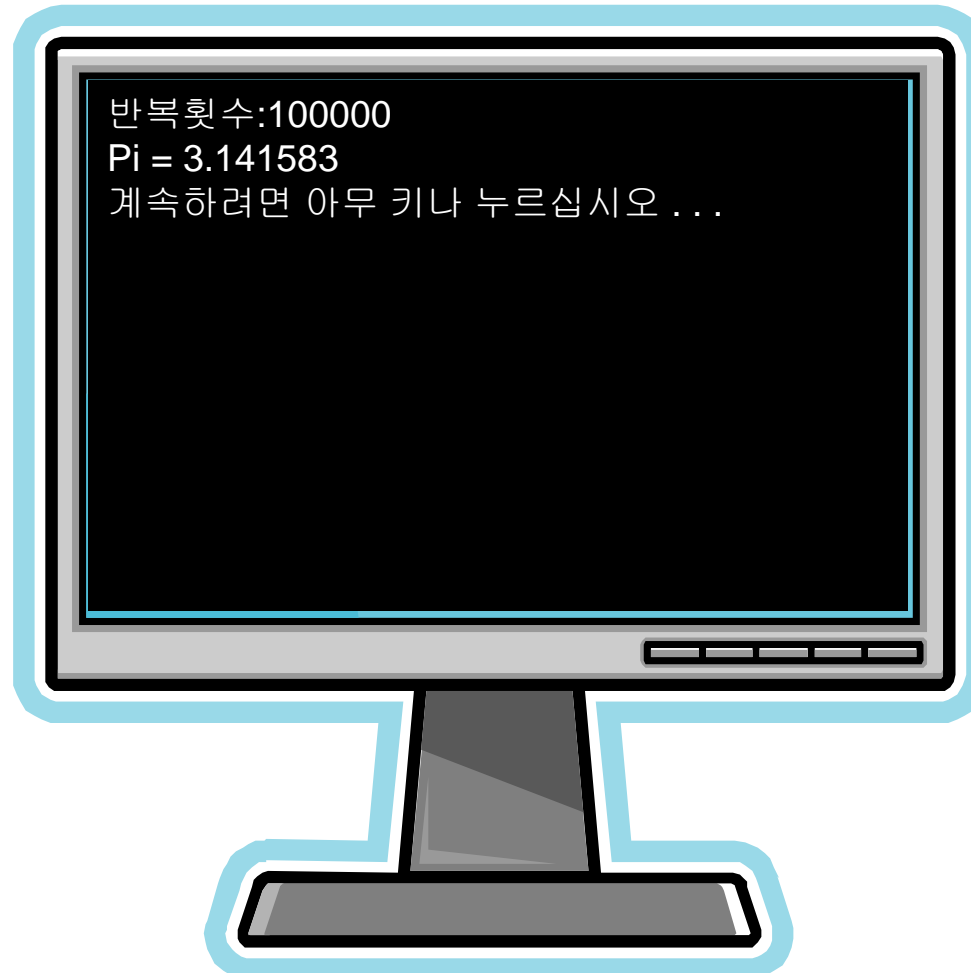
- 파이를 계산하는 가장 고전적인 방법은 *Gregory-Leibniz* 무한 수열을 이용하는 것

$$\pi = \frac{4}{1} - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} + \dots$$





실행 결과





알고리즘

- 사용자로부터 반복횟수 `loop_count`를 입력받는다.
- 분자 = 4.0;
- 분모 = 1.0;
- `sum = 0.0;`
- `while(loop_count > 0)`
 - `sum = sum + 분자 / 분모;`
 - `분자 = -1.0* 분자;`
 - `분모 = 분모 + 2.0;`
 - `--loop_count;`
- `sum`을 출력한다.



실습 코드

```
#include <stdio.h>

int main(void)
{
    double divisor, dividend, sum;
    int loop_count;

    divisor = 1.0;
    dividend = 4.0;
    sum = 0.0;
    printf("반복횟수:");
    scanf("%d", &loop_count);

    while(loop_count > 0) {
        sum = sum + dividend / divisor;
        dividend = -1.0 * dividend;
        divisor = divisor + 2;
        loop_count--;
    }
    printf("Pi = %f", sum);
    return 0;
}
```



예제

```
#include <stdio.h>
int main(void)
{
    int i, years;
    double total, rate, investment;

    printf("원금: ");
    scanf("%lf", &investment);
    printf("이율(%): ");
    scanf("%lf", &rate);
    printf("기간(년): ");
    scanf("%d", &years);

    printf("=====");
    printf("연도 원리금");
    printf("=====");
    total = investment;
    rate /= 100.0;
    for(i = 0; i < years; i++)
    {
        total = total * ( 1 + rate );
        printf("%2d %10.1f", i+1, total);
    }

    return 0;
}
```





예제

```
#include <stdio.h>
#define START_DAY      3    // 첫번째 날이 수요일
#define DAYS_OF_MONTH  31   // 달의 일수

int main(void)
{
    int day, date;
    printf("===== \n");
    printf("일 월 화 수 목 금 토 \n");
    printf("===== \n");
    // 월요일부터 수요일까지
    for(day = 0; day < START_DAY ; day++)
        printf(" ");          // 공백 출력
    for(date = 1; date <= DAYS_OF_MONTH ; date++)
    {
        if( day == 7 )
        {
            day = 0;    // 일요일이면 줄바꿈을 출력
            printf("\n");
        }
        day++;
        printf("%2d ", date);    // 날을 출력한다.
    }
    printf("\n===== \n");
    return 0;
}
```

```
=====
일 월 화 수 목 금 토
=====
                1  2  3  4
    5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31
=====
```



Q & A

