

개정3판

Visual  
Studio  
2017

쉽게 풀어쓴

# C언어 EXPRESS




천인국 지음

## 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.





# 반복

- 인간은 반복을 싫어하지만 프로그램에서는 반복적인 작업들이 반드시 필요하다.
- 반복(iteration)은 같은 처리 과정을 여러 번 되풀이하는 것이다





# 반복문

Q) 반복 구조는 왜 필요한가?

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





# 왜 반복이 중요한가?

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

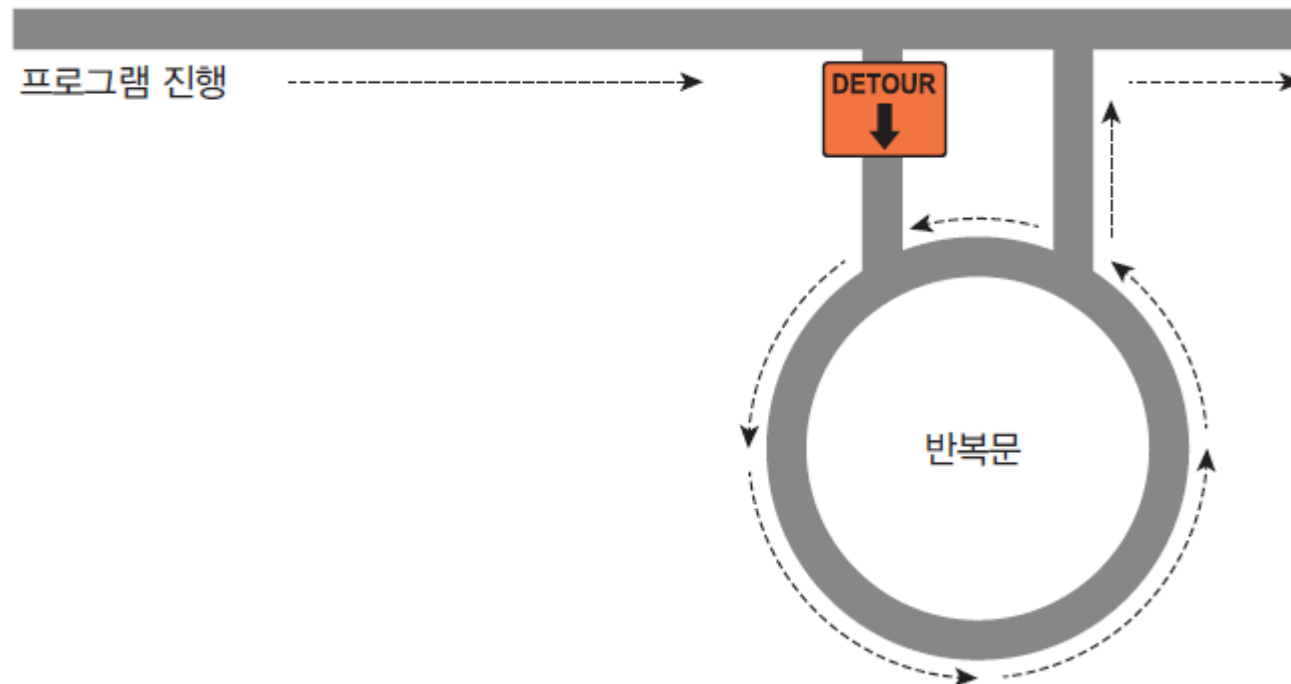


```
for(int i=0; i<5; i++ )  
    printf("Hello World! \n")
```



# 반복 구조

- 어떤 조건이 만족될 때까지 루프를 도는 구조





# 반복문의 종류





## 중간 점검

1. 프로그램에 반복 구조가 필요한 이유는 무엇인가?
2. 반복문에는 \_\_\_\_\_, \_\_\_\_\_문이 있다.

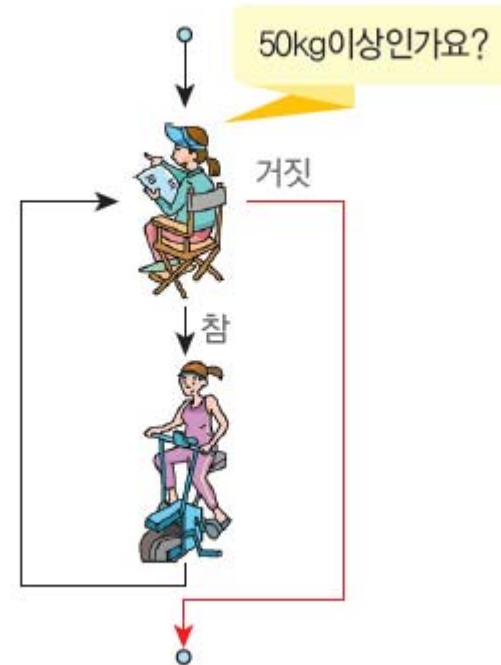
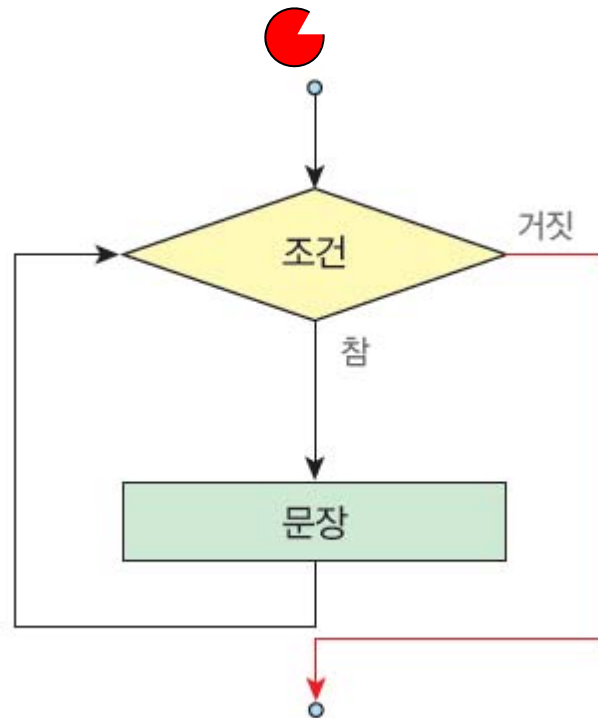






# while 문

- 주어진 조건이 만족되는 동안 문장들을 반복 실행한다.





# while 문

Syntax: while 문

예

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

조건식

조건식이 참이면 문장을 반복 실행한다.



# 예제

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

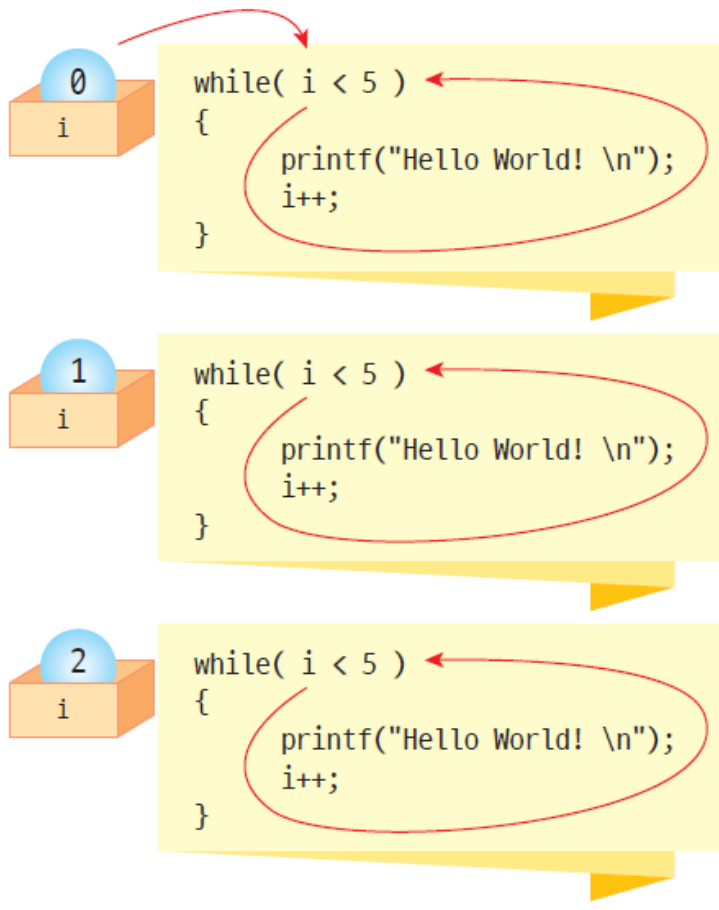
    return 0;
}
```

반복 조건

반복 내용



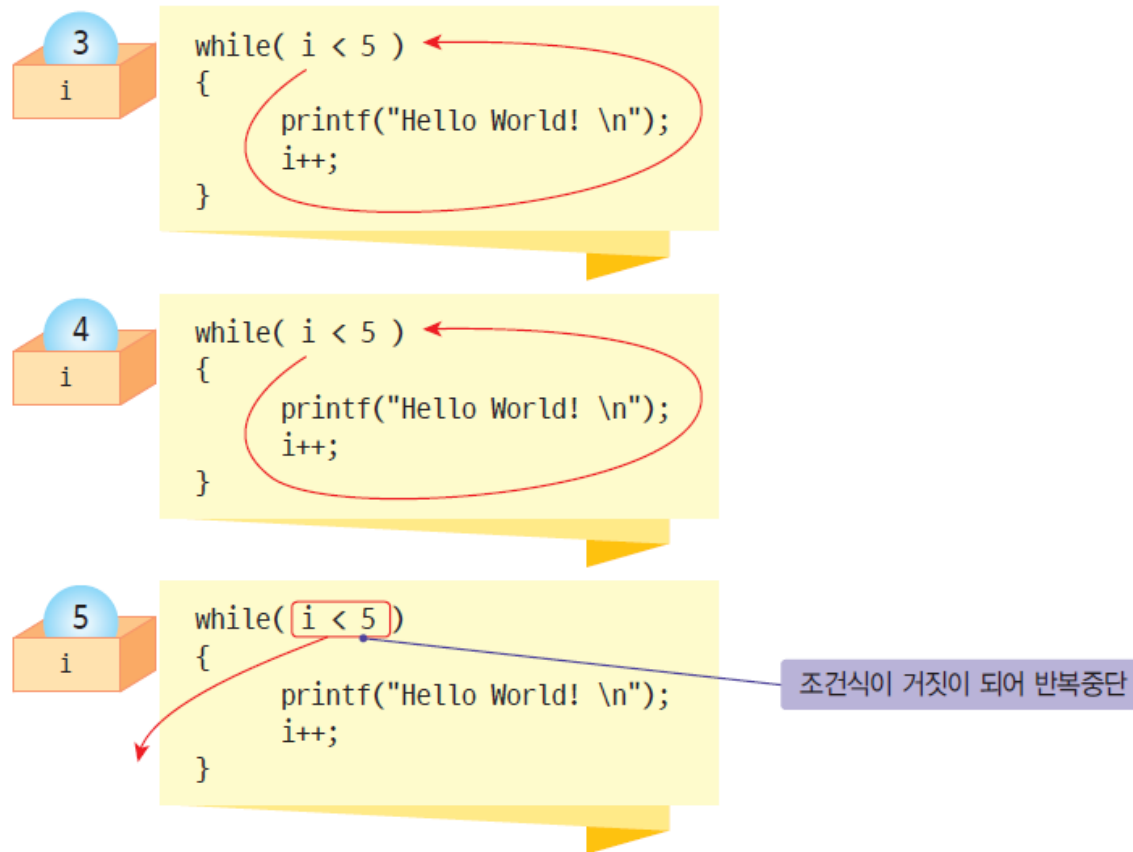
# while 문의 실행 과정



반복횟수	i의 값	(i<5)	반복여부
#1	0	참	반복
#2	1	참	반복
#3	2	참	반복
#4	3	참	반복
#5	4	참	반복
#6	5	거짓	중지



# while 문의 실행 과정





# 예제 #1

// 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 #2

// 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++;
```

```
    }
```

```
    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



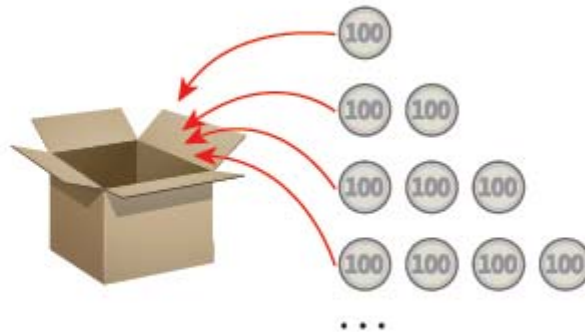
## 예제 #3

### □ 1부터 n까지의 합 계산하는 프로그램

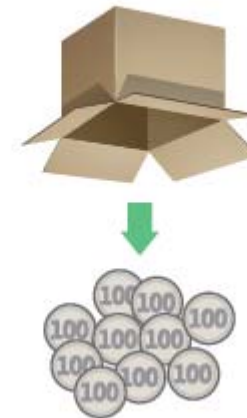
① 빈통을 준비한다.



② 통에 1부터 n까지를 넣는다.



③ 통에 들어있는 동전의 개수를 출력한다.







## 예제 #3

```
#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 #4

```
#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 = 0;                        // 변수 초기화
    sum = 0;

    while(i <= n)
    {
        sum += i;                // sum = sum + i;와 같다.
        i = i + 2;
    }

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

정수를 입력하시오: 10  
1부터 10까지의 짝수합은 30입니다.



## 예제 #5

// 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 < 5)
```

```
    {
```

```
        printf("값을 입력하시오: ");
```

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

```
        sum = sum + n;
```

```
        // sum += n;과 같다.
```

```
        i++;
```

```
    }
```

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

```
    return 0;
```

```
}
```

값을 입력하시오: 10  
값을 입력하시오: 20  
값을 입력하시오: 30  
값을 입력하시오: 40  
값을 입력하시오: 50  
합계는 150입니다.



# if 문과 while 문의 비교

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

조건이 만족되  
면 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.

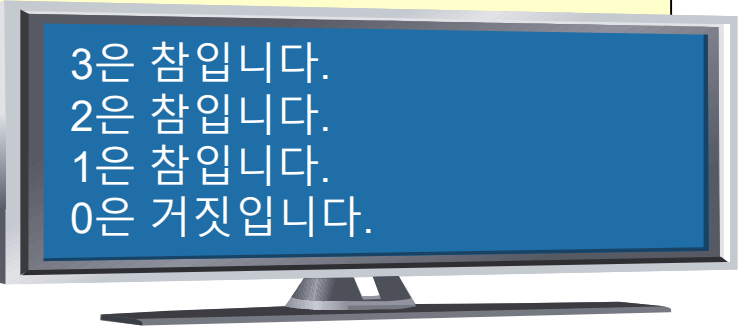
```
int i = 0;
while(i < 3) ;
{
    printf("반복중입니다\n");
    i++;
}
```

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.



# 센티널(보초값의 이용)

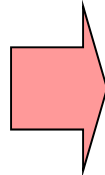
- 센티널: 입력되는 데이터의 끝을 알리는 특수한 값





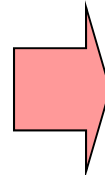
# 성적들의 평균을 구하는 문제

- 성적의 평균을 구한다.



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

1. 필요한 변수들을 초기화한다.



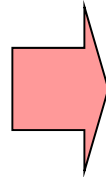
- (1) sum을 0으로 초기화한다.
- (2) n을 0으로 초기화한다.
- (3) grade를 0으로 초기화한다.





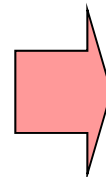
# 성적들의 평균을 구하는 문제

2. 성적을 입력받아서 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.



# 센티널 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");
```



## 센티널 예제 2/2

```
// 성적을 입력받아서 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;
```

```
}
```

성적 입력을 종료하려면 음수를

입력하시오

성적을 입력하시오: 10

성적을 입력하시오: 20

성적을 입력하시오: 30

성적을 입력하시오: 40

성적을 입력하시오: 50

성적을 입력하시오: -1

성적의 평균은 30.000000입니다.



## 예제: 최대값

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

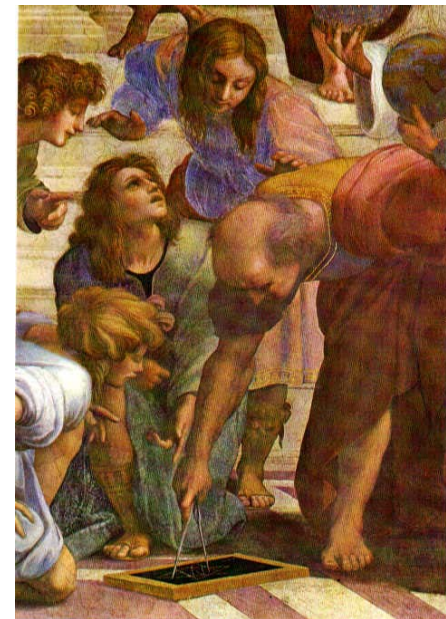
int main(void)
{
    int number, min_value = INT_MAX;
    printf("정수를 입력하시오\n종료는 Ctrl+z\n");
    while(scanf("%d", &number) != EOF)
    {
        if( number < min_value )
            min_value = number;
    }
    printf("최소값은 %d", min_value);
    return 0;
}
```

```
정수를 입력하시오
종료는 Ctrl+z
10
20
30
5
^Z
최소값은 5
```



# lab: 최대 공약수 찾기

두개의 정수를 입력하시오(큰수, 작은수): 12 8  
최대 공약수는 4입니다.





# lab: 최대 공약수 찾기

## □ 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.



// while 문을 이용한 최대 공약수 구하기 프로그램

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

```
int main(void)
```

```
{
```

```
    int x, y, r;
```

```
    printf("두개의 정수를 입력하시오(큰수, 작은수): ");
```

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

```
    while (y != 0)
```

```
    {
```

```
        r = x % y;
```

```
        x = y;
```

```
        y = r;
```

```
    }
```

```
    printf("최대 공약수는 %d입니다.\n", x);
```

```
    return 0;
```

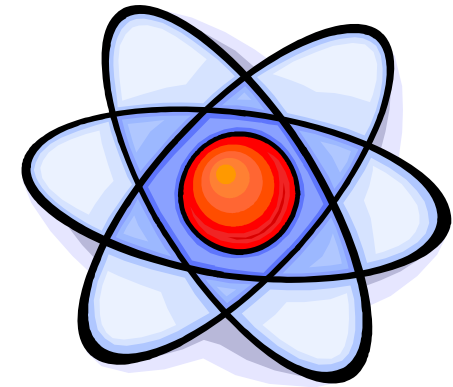
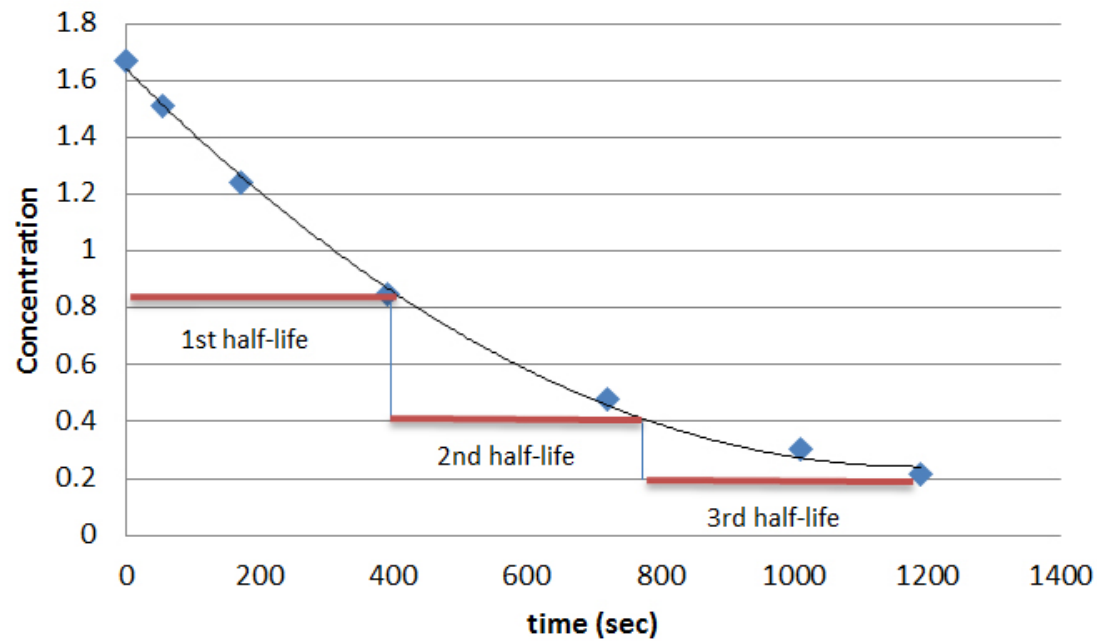
```
}
```

두개의 정수를 입력하시오(큰 수, 작은 수): 12 8  
최대 공약수는 4입니다.



# lab: 반감기

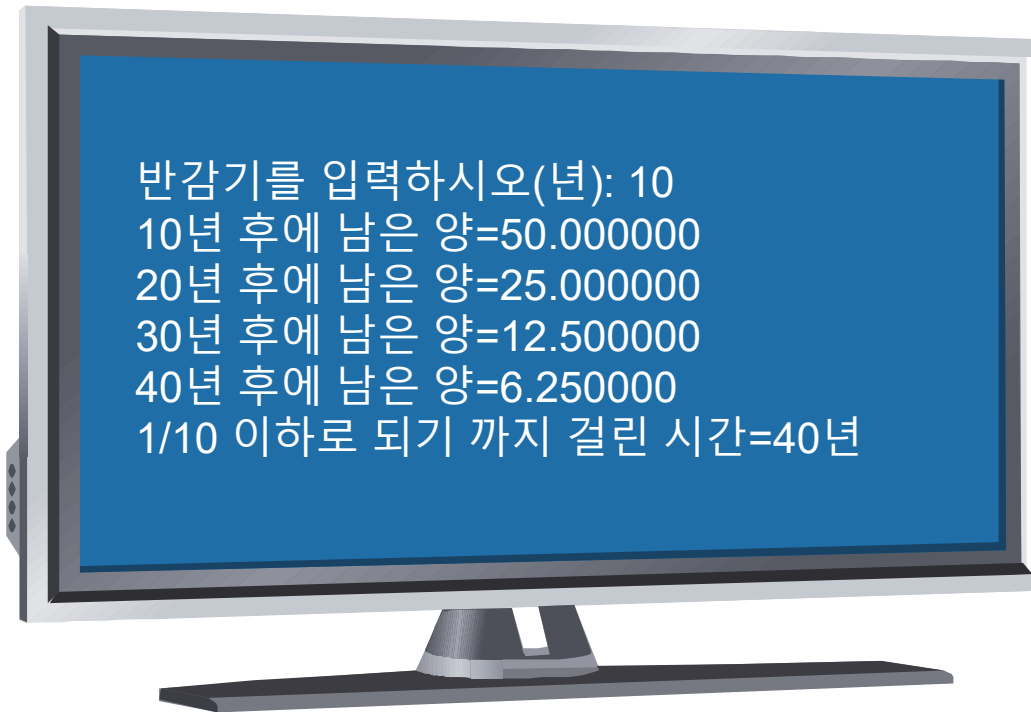
- 반감기: 방사능 물질의 양이  $\frac{1}{2}$ 로 되는 시간







## 실행 결과



- 단 로그 함수는 사용하지 않는다!
- 반복문을 사용한다.



# 알고리즘

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



```
#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 이용하지 말고 반복 구조를 이용하여 해결하여 보자.

