

CHAPTER

07

반복문

- 프로그램 안의 문장들을 반복적으로 실행하기 위한 while 구조와 for 구조를 학습한다.
- do-while 문의 구조를 이해한다.
- 반복문에서의 break와 continue 사용법을 익힌다.
- 무한 반복과 중첩된 반복 구조를 이해하고 익힌다.

Contents

3

7.1

반복문의 개념

7.2

while 문

7.3

반복 루프에서 보조값 사용하기

7.4

do-while 문

7.5

for문

7.6

break와 continue

반복

4


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



반복문

5

 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

6

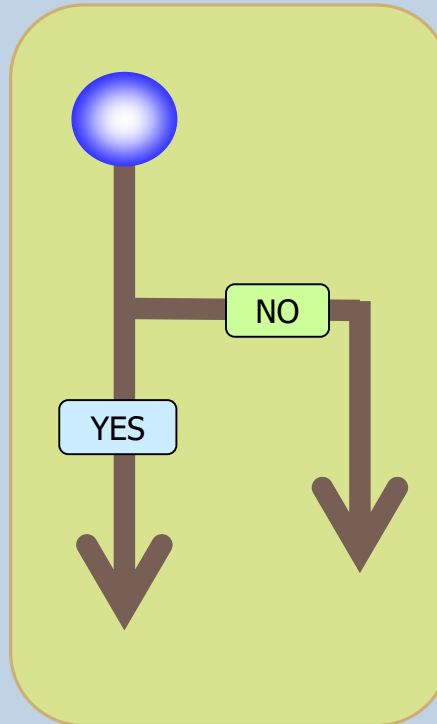
1) 순차구조

- 차례대로 실행



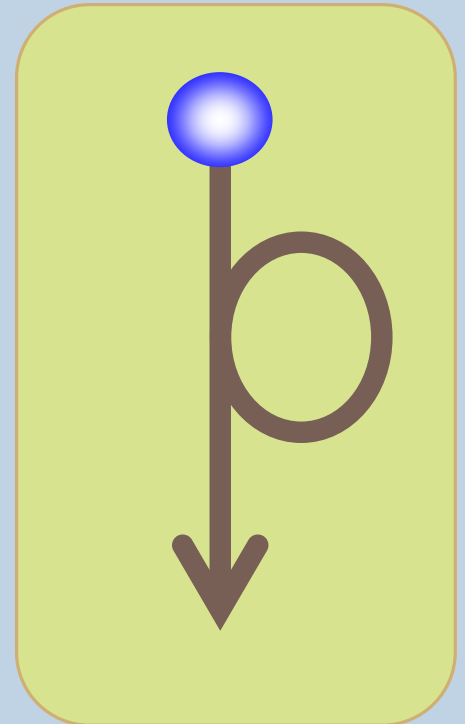
2) 선택구조

- 조건을 검사하여 여러 개의 실행 경로 중에서 하나를 선택



3) 반복구조

- 조건이 만족될 때까지 반복



왜 반복이 중요한가?

7

같은 처리과정

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

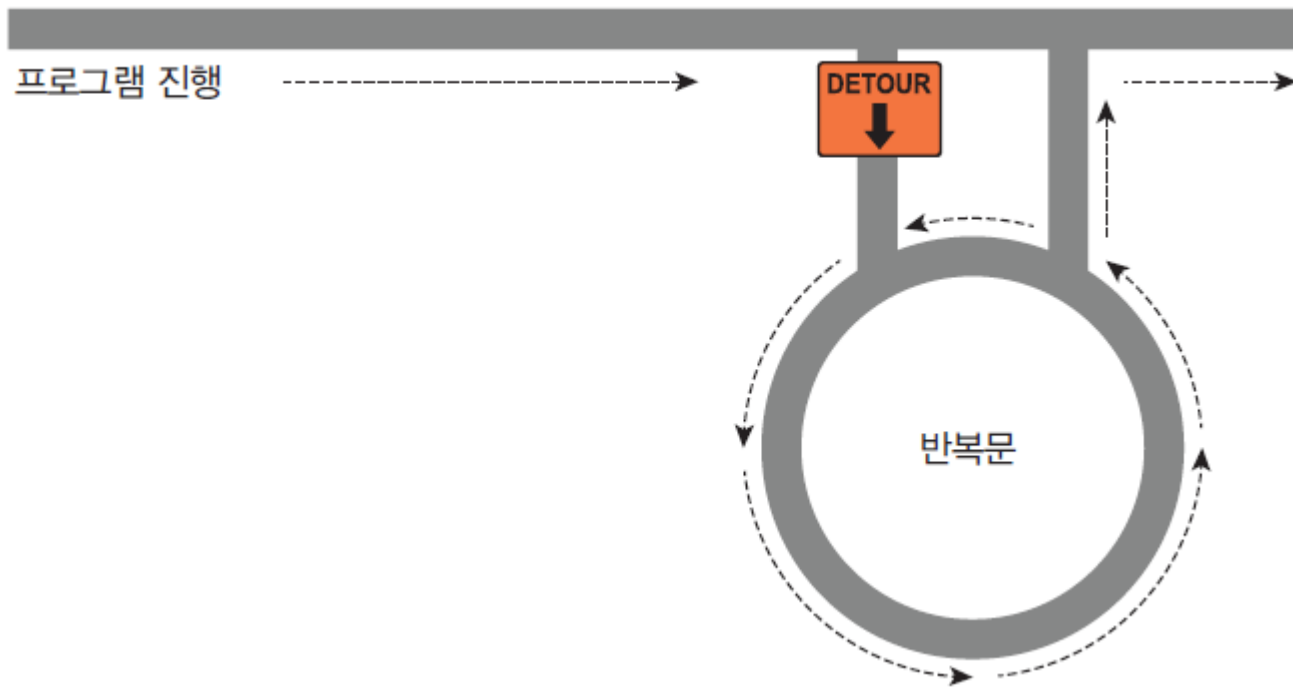


```
for(int i=0; i<5; i++ )  
    printf("Hello World! \n")
```

반복 구조

8

- 어떤 조건이 만족될 때까지 루프를 도는 구조



반복문의 종류

9



while 루프



for 루프

중간 점검

10

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



Contents

11

7.1

반복문의 개념

7.2

while 문

7.3

반복 루프에서 보조값 사용하기

7.4

do-while 문

7.5

for문

7.6

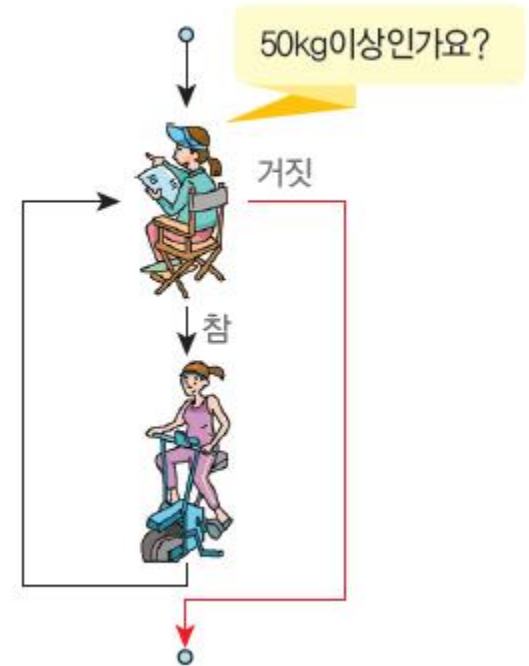
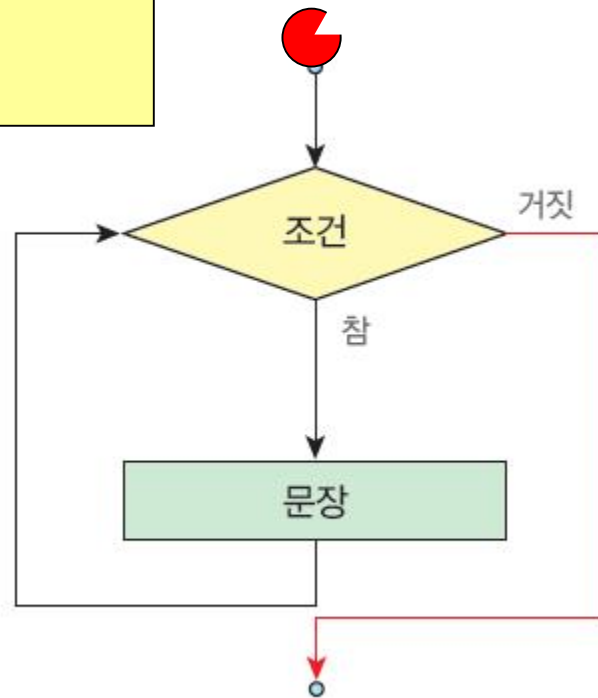
break와 continue

while 문

12

- 주어진 조건이 만족되는 동안 문장들을 반복 실행한다.

while(조건식)
문장;



while 문

13

Syntax: while 문

예

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

조건식

조건식이 참이면 문장을 반복 실행한다.

예제

14

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

```
int main(void)
```

```
{
```

```
    int i = 0;
```

```
    while( i < 5 )
```

```
    {
```

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

```
    }
```

```
    return 0;
```

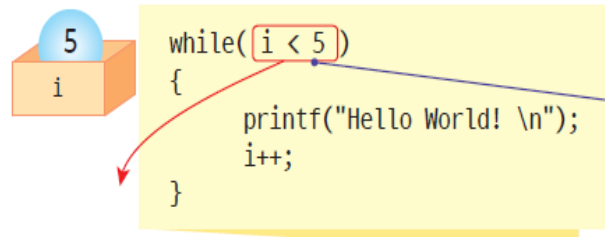
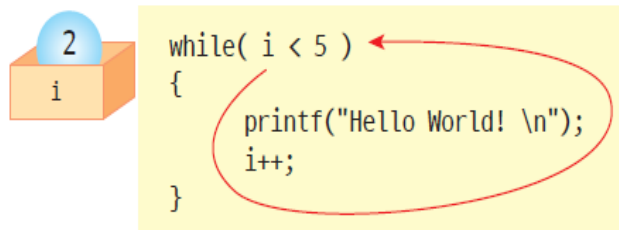
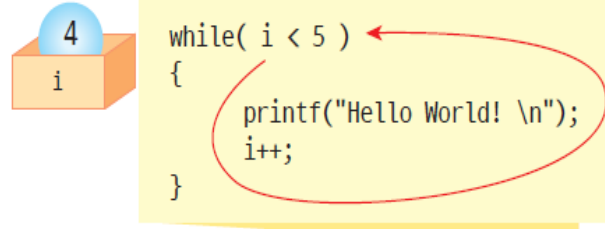
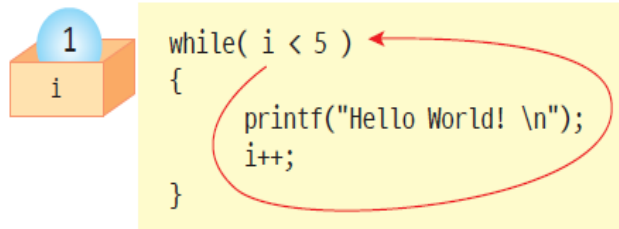
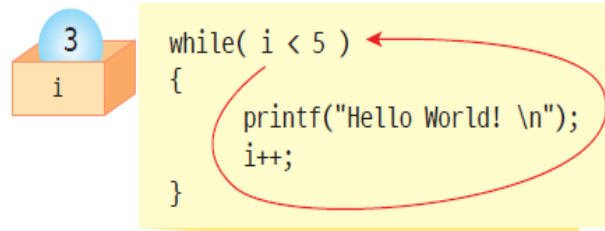
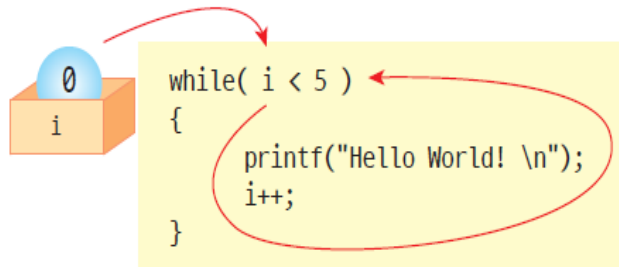
```
}
```

반복 조건

반복 내용

while 문의 실행과정

15



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

조건식이 거짓이 되어 반복종단

예제

16

// 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)

{



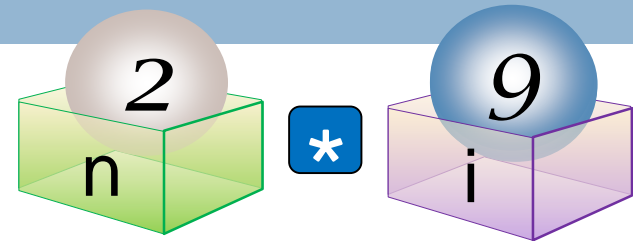
printf("%d*d = %d \n", n, i, n*i);

i++; // i = i+1 과 동일

}

return 0;

}



예제

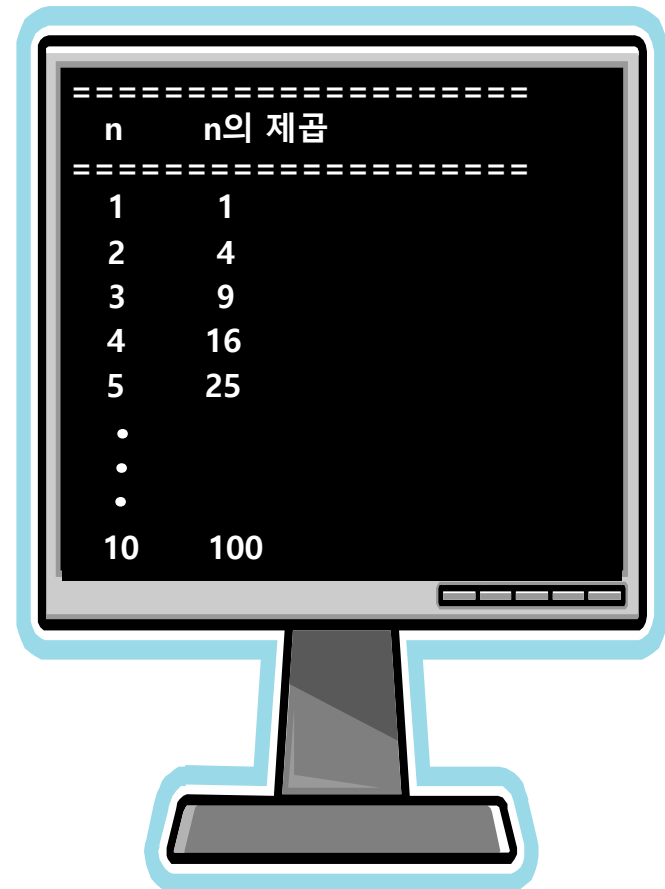
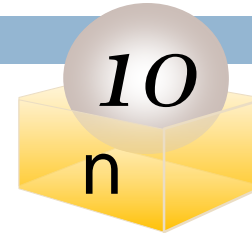
17

```
// 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

18

□ 1부터 n까지의 합을 계산하는 프로그램

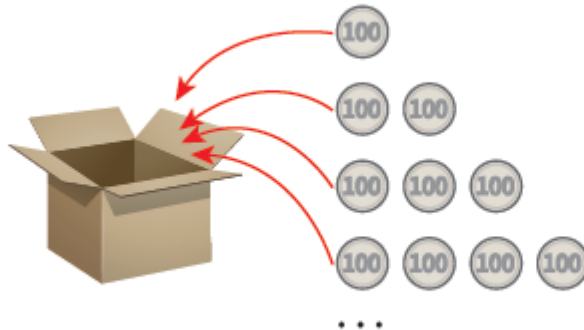
$$1 + 2 + 3 + \dots + n$$

▣ n이 무엇이 될지 모르는 경우라면 다음과 같이 계산

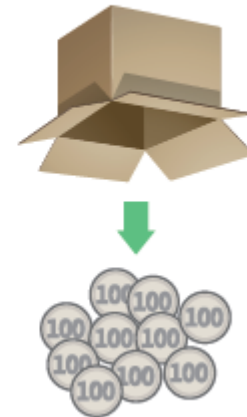
① 빈통을 준비한다.



② 통에 1부터 n까지를 넣는다.



③ 통에 들어있는 동전의 개수를 출력한다.



예제

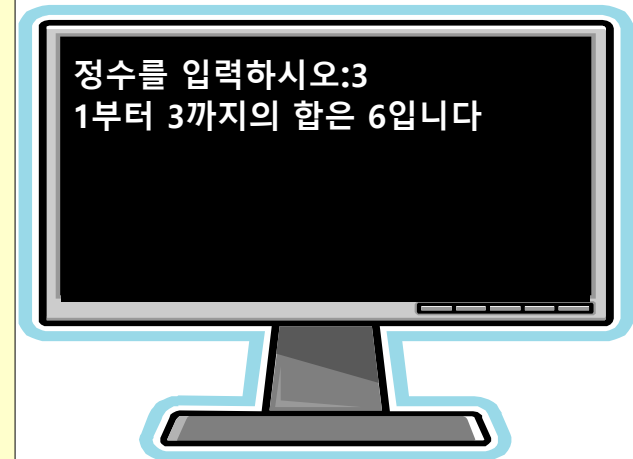
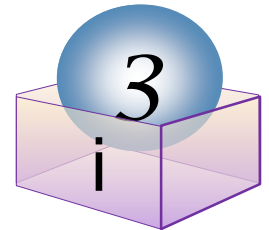
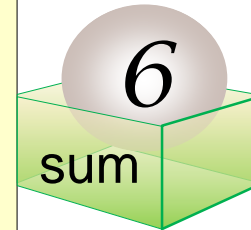
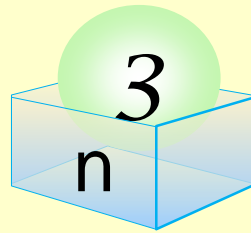
19

```
#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

20

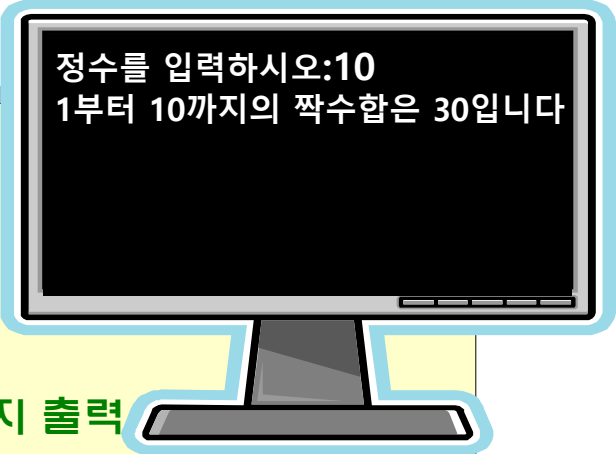
```
#include <stdio.h>
int main(void)
{
    int i, n, sum;                // 변수 선언

    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 + 2;
    }

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



정수를 입력하시오:10
1부터 10까지의 짝수합은 30입니다

예제

21

// while 문을 이용한 합계 프로그램

#include <stdio.h>

int main(void)

{

int i, n, sum;

i = 0;

sum = 0;

while (i < 3)

{



printf("값을 입력하시오: ");

scanf("%d", &n);

sum = sum + n; // sum += n;과 같다.

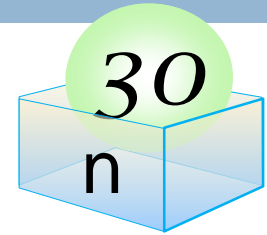
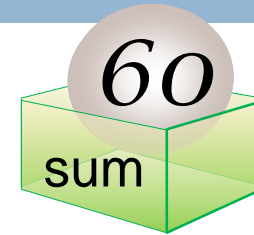
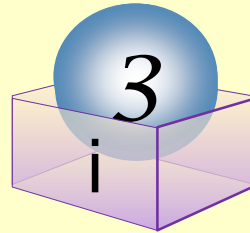
i++;

}

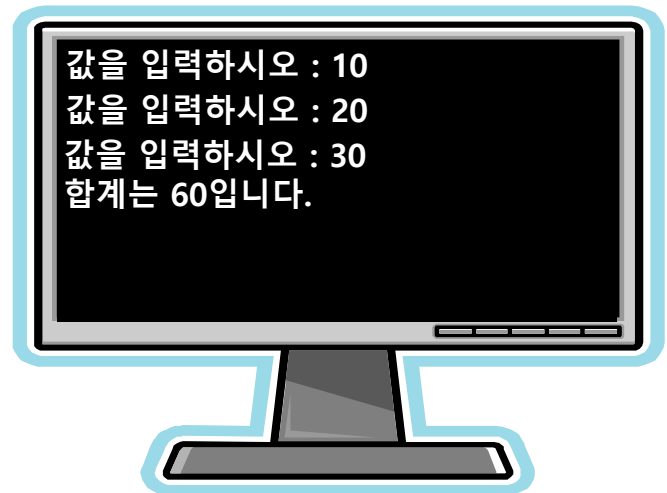
printf("합계는 %d입니다.\n", sum);

return 0;

}



값을 입력하시오 : 10
값을 입력하시오 : 20
값을 입력하시오 : 30
합계는 60입니다.



if 문과 while 문의 비교

22

```
if( 조건 )
```

```
{
```

```
...
```

```
...
```

```
}
```

조건이 만족되면
한번만 실행
된다.

```
while( 조건 )
```

```
{
```

```
...
```

```
...
```

```
}
```

조건이 만족되면
여러 번 반복 실행
된다.

while 문에서 주의할 점

23

```
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

24

```
#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

25

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;
}
```



Contents

26

7.1

반복문의 개념

7.2

while 문

7.3

반복 루프에서 보조값 사용하기

7.4

do-while 문

7.5

for문

7.6

break와 continue

센티널(보초값의 이용)

27

- 센티널: 입력되는 데이터의 끝을 알리는 특수한 값

100, 200, 300, 400, 500, -1

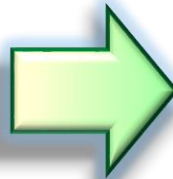
센티널



성적들의 평균을 구하는 문제

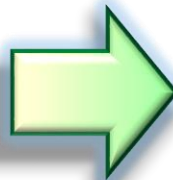
28

- 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

29

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

30

```
// 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

31

// 성적을 입력 받아서 합계를 구하고 학생 수를 센다.

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

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

3
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;

}

150
sum

-1
grade

50
average

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

Lab : 최대 공약수 찾기

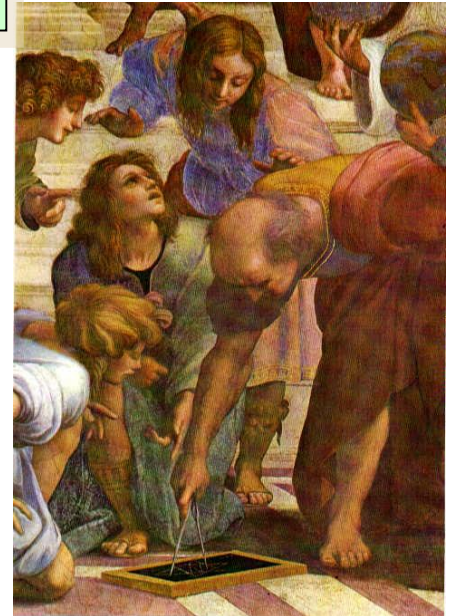
32

□ 두수의 최대 공약수 구하기

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

□ 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

33

// 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)
```

나머지

```
    {
```

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

```
        x = y;
```

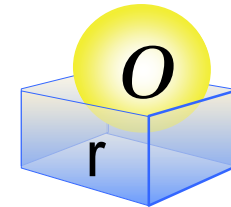
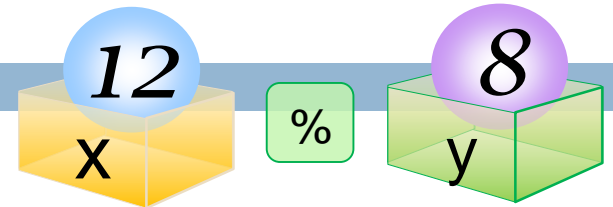
```
        y = r;
```

```
    }
```

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

```
    return 0;
```

```
}
```

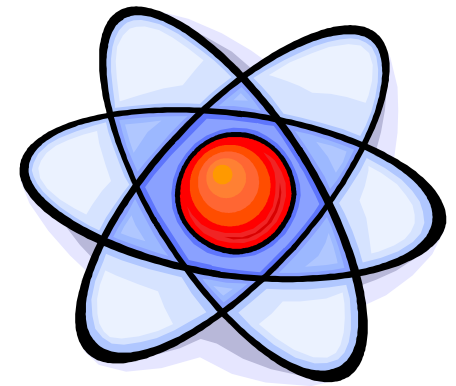
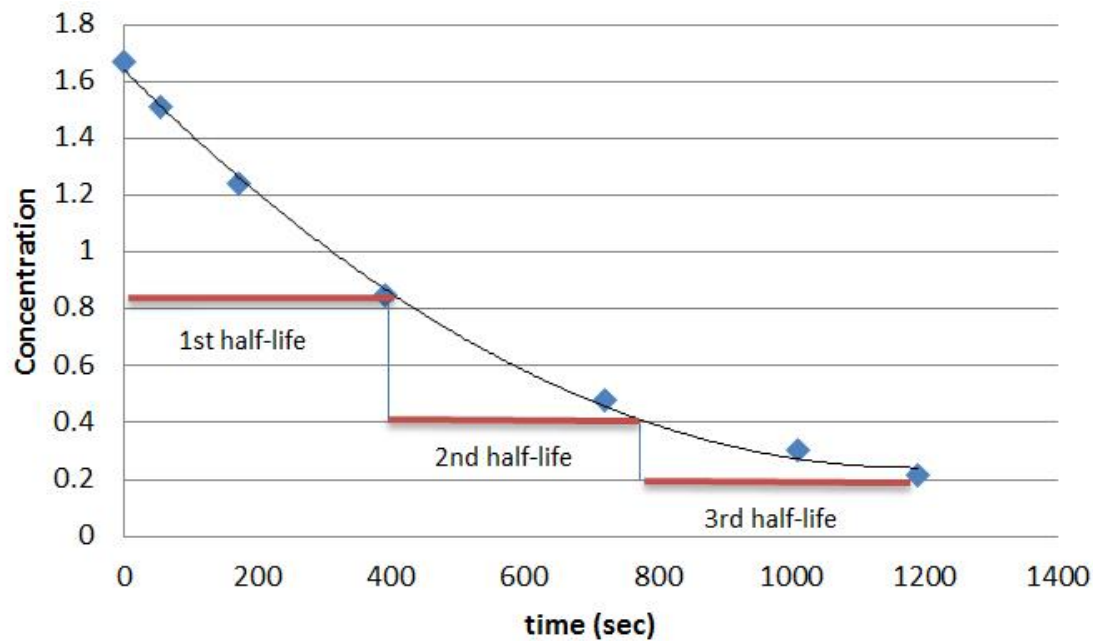


두 정수를 입력 (큰 수, 작은 수): 12 8
최대공약수는 4입니다.

Lab: 반감기

34

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



실행 결과

35

- 단 로그 함수는 사용하지 않는다!
- 반복문을 사용한다.

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



알고리즘

36

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

38

- 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 이용하여 해결하여 보자.

Contents

39

7.1

반복문의 개념

7.2

while 문

7.3

반복 루프에서 보조값 사용하기

7.4

do-while 문

7.5

for문

7.6

break와 continue

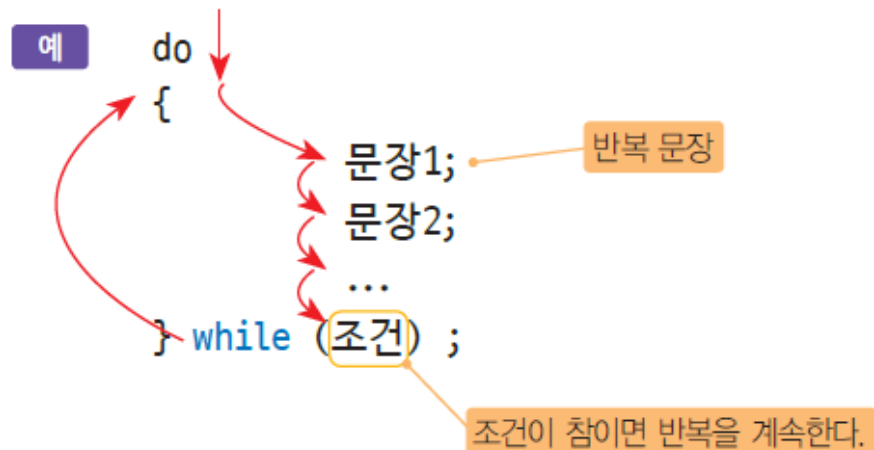
do...while문

40

□ 반복 조건을 루프의 끝에서 검사

```
do  
    문장  
while(조건)
```

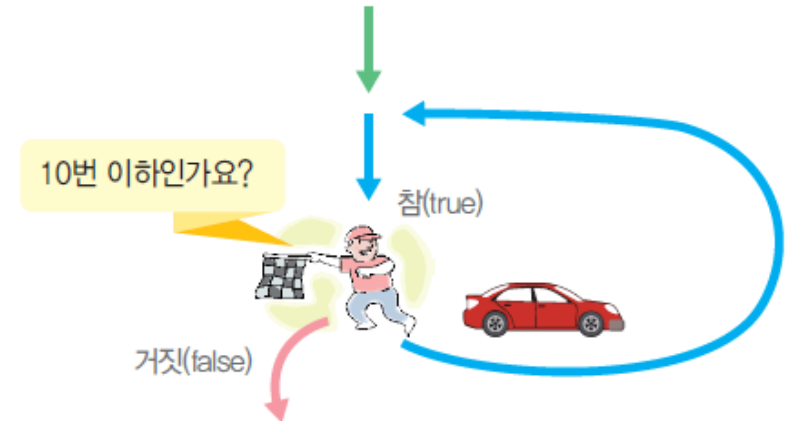
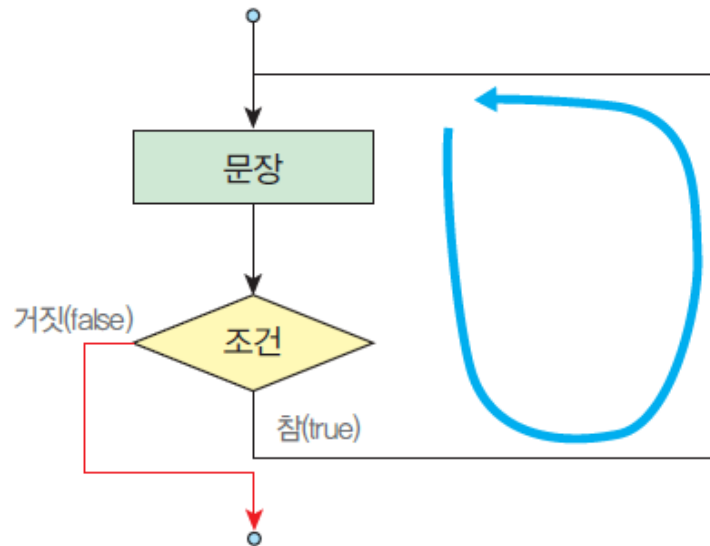
Syntax: do...while문



do-while 문

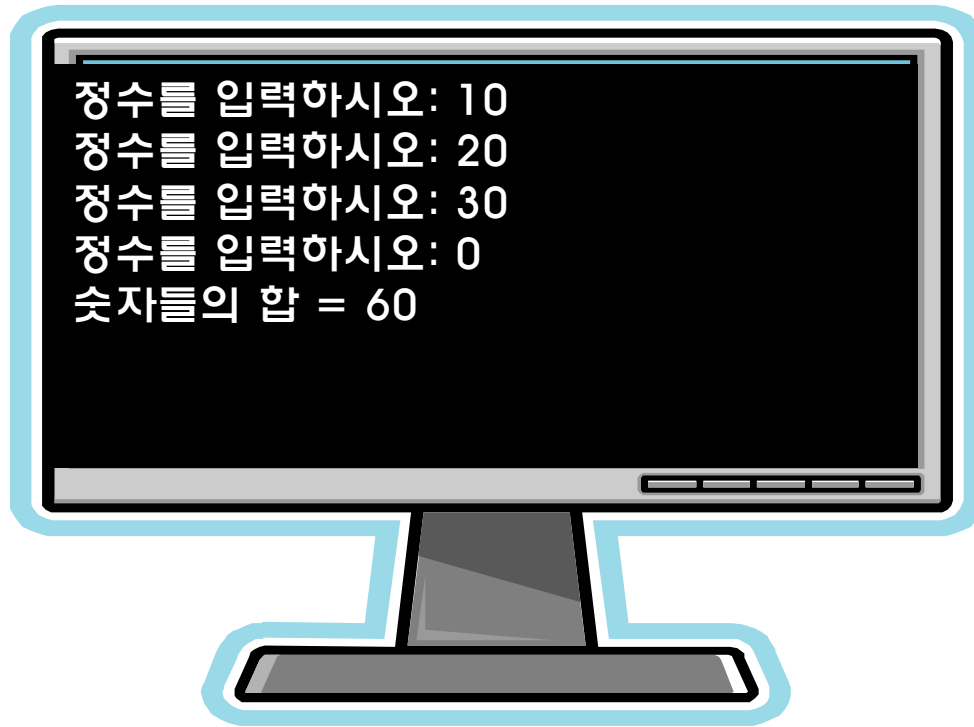
41

- 적어도 한번은 반복문장을 실행한다.



예제

42



예제

43

// 사용자가 0을 입력할 때까지 숫자를 더한다.

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

```
int main(void)
```

```
{
```

```
    int number, sum = 0;
```

```
    do
```

```
    {
```

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

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

```
        sum += number;
```

```
    } while (number != 0);
```

```
    printf("숫자들의 합 = %d \n", sum);
```

```
    return 0;
```

```
}
```

예제

44

// 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");
```

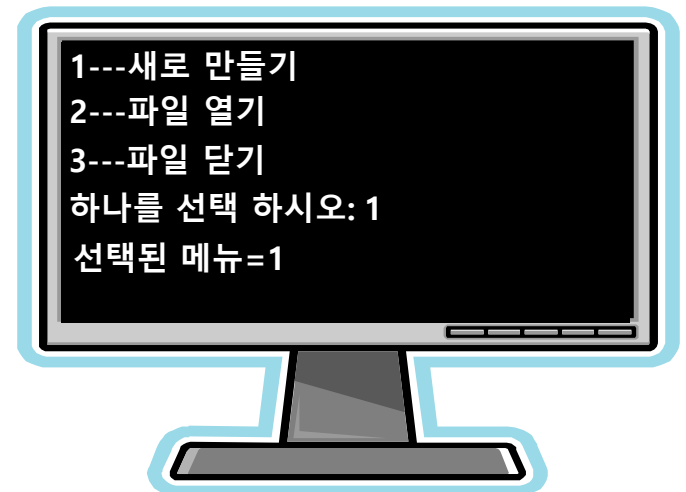
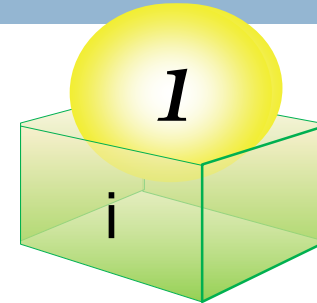
```
        scanf("%d", &i);
```

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

```
    printf("선택된 메뉴=%d\n",i);
```

```
    return 0;
```

```
}
```



중간 점검

45

1. 다음 코드의 출력을 쓰시오.

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




Lab: 숫자 추측 게임

46

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

실행 결과

47

A stylized illustration of a computer monitor with a light blue frame and a grey base. The screen is black and displays white Korean text.

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

48

do

사용자로부터 숫자를 guess로 입력받는다.

시도횟수를 증가한다.

if(guess < answer)

숫자가 낮다고 출력한다.

if(guess > answer)

숫자가 높다고 출력한다.

while(guess != answer);

“축하합니다” 와 시도횟수를 출력한다.


```
#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

50

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

Contents

51

7.1

반복문의 개념

7.2

while 문

7.3

반복 루프에서 보조값 사용하기

7.4

do-while 문

7.5

for문

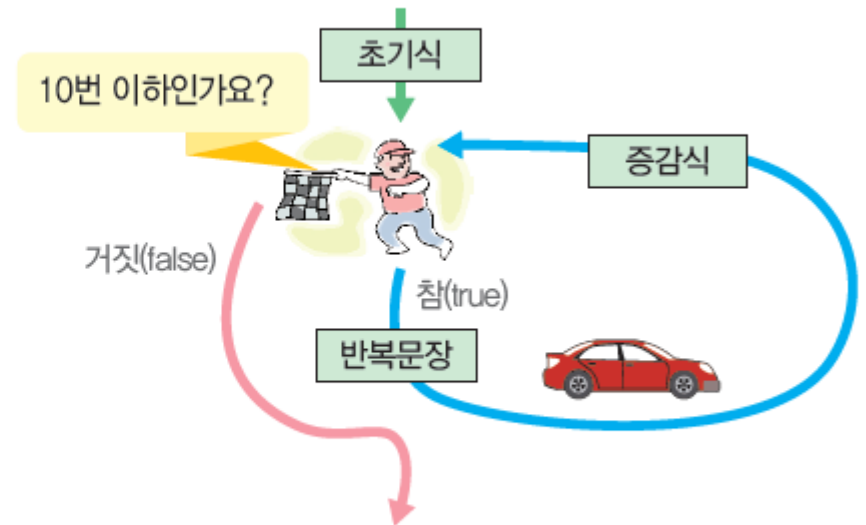
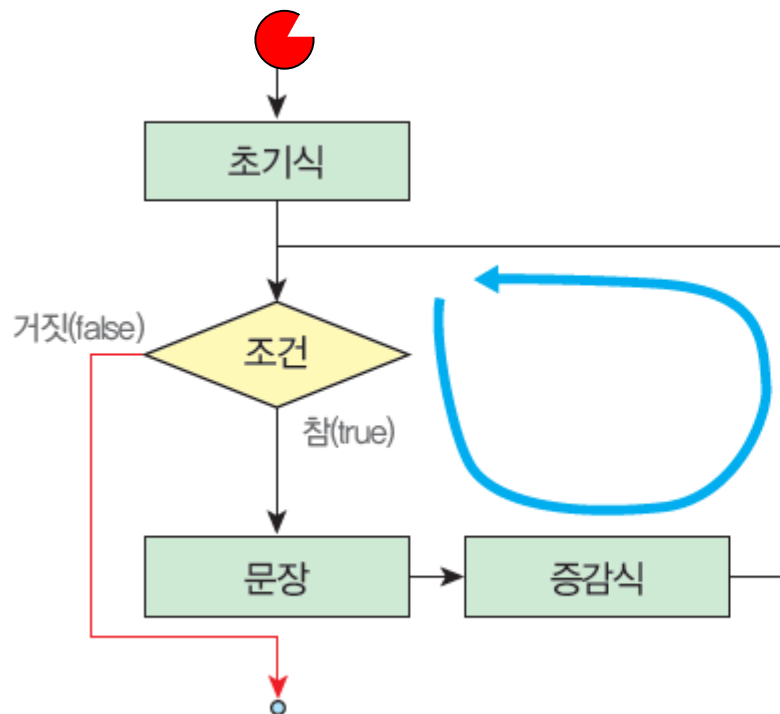
7.6

break와 continue

for 루프

52

□ 정해진 횟수만큼 반복하는 구조

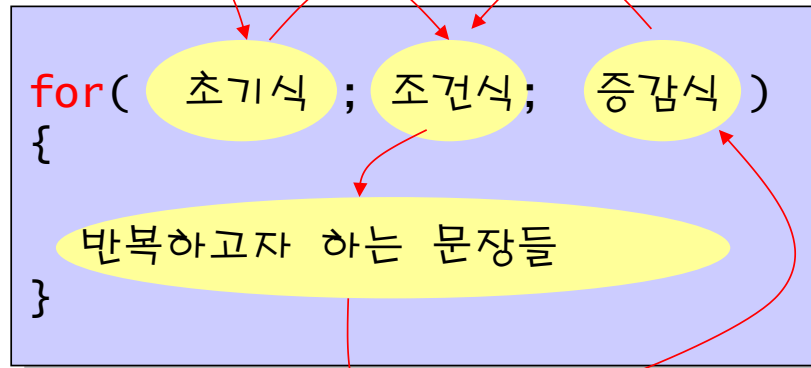


for 문의 구조

53

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.



예제

54

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

int main(void)
{
    int i;

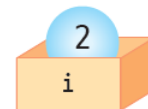
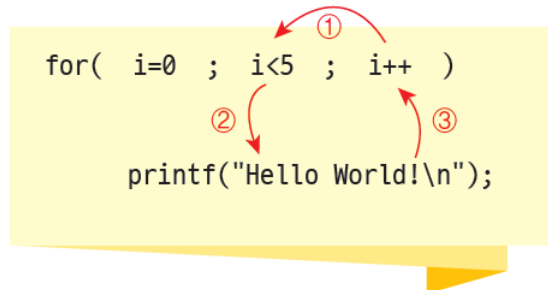
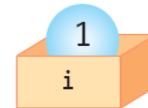
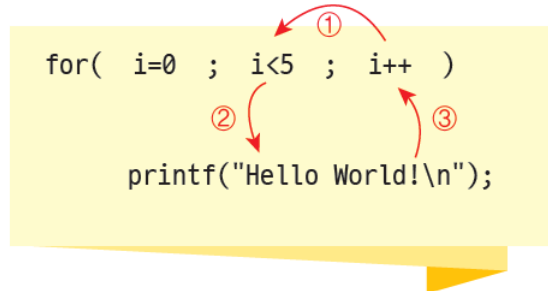
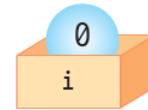
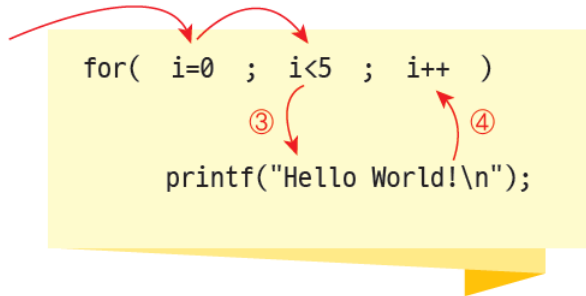
    for (i = 0; i < 5; i++) // i는 0부터 4까지 증가
        printf("Hello World!\n");

    return 0;
}
```

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

for문의 실행과정

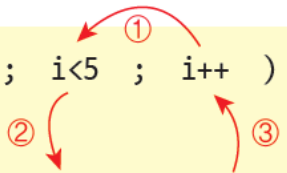
55



for문의 실행과정

56

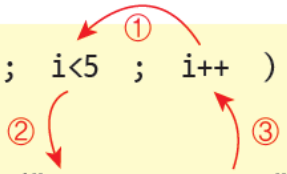
```
for( i=0 ; i<5 ; i++ )  
    printf("Hello World!\n");
```



3
i

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

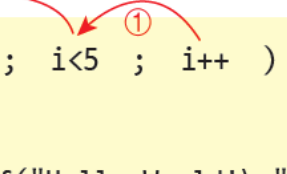
```
for( i=0 ; i<5 ; i++ )  
    printf("Hello World!\n");
```



4
i

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

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



5
i

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


for문의 실행과정

57

```
int i;  
for(i = 0; i < 5; 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	거짓	중지



예제

58

// 반복을 이용한 정수합 프로그램

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

```
int main(void)
```

```
{
```

```
    int i, sum;
```

```
    sum = 0;
```

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

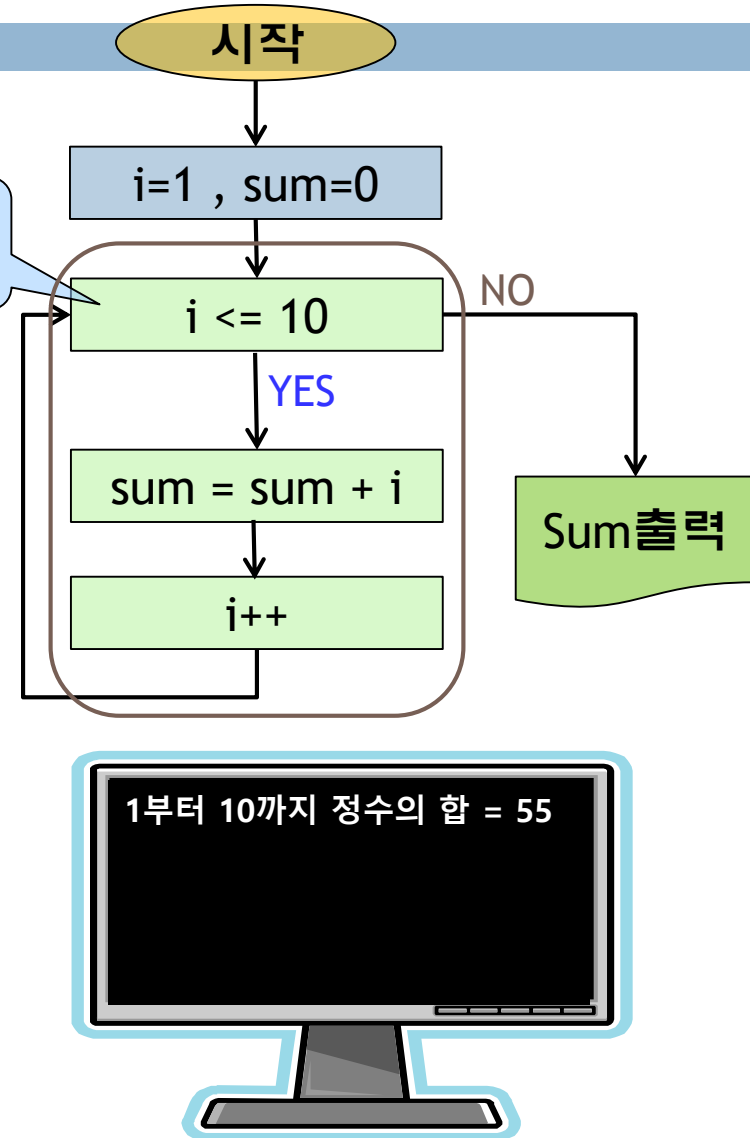
```
        sum += i; // sum = sum + i;와 같음
```

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

```
    return 0;
```

```
}
```

i가 10보다 작거나 같은 때
까지 10번 반복



예제

59

// 반복을 이용한 세제곱값구하기

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

```
int main(void)
```

```
{
```

```
    int i, n;
```

```
    printf("정수를 입력하시오:");
```

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

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

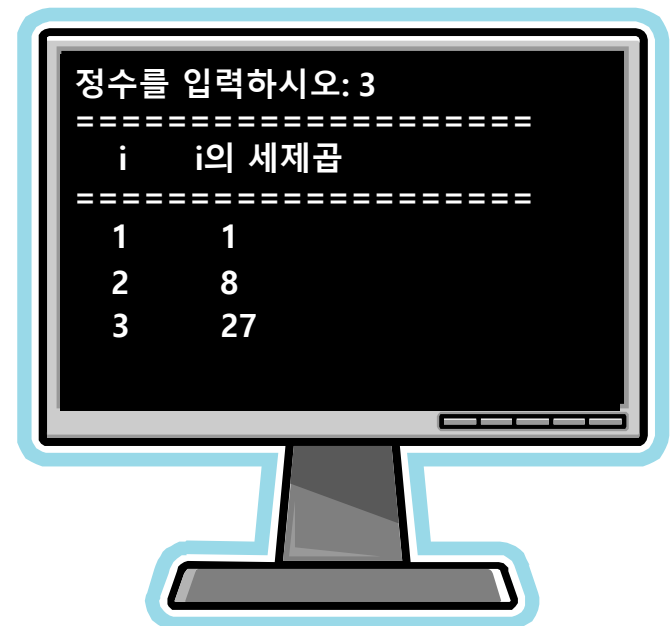
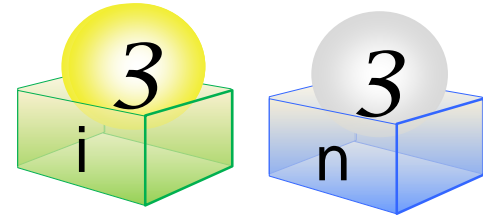
```
    printf("  i      i의 세제곱\n");
```

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

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

```
    return 0;
```

```
}
```



예제

60

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

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

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

    printf("*****");

    return 0;
}
```



예제

61

// 반복을 이용한 팩토리얼 구하기

#include <stdio.h>

int main(void)

{

long fact=1;

int i, n;

printf("정수를 입력하시오:");

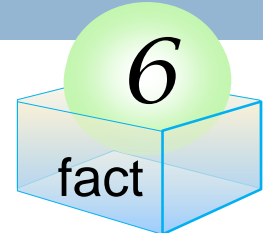
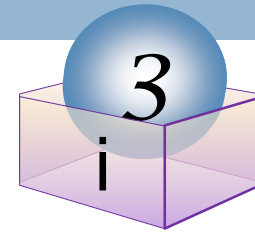
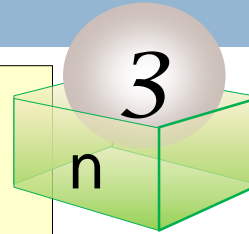
scanf("%d", &n);

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

printf("%d! 은 %d입니다.\n", n, fact);

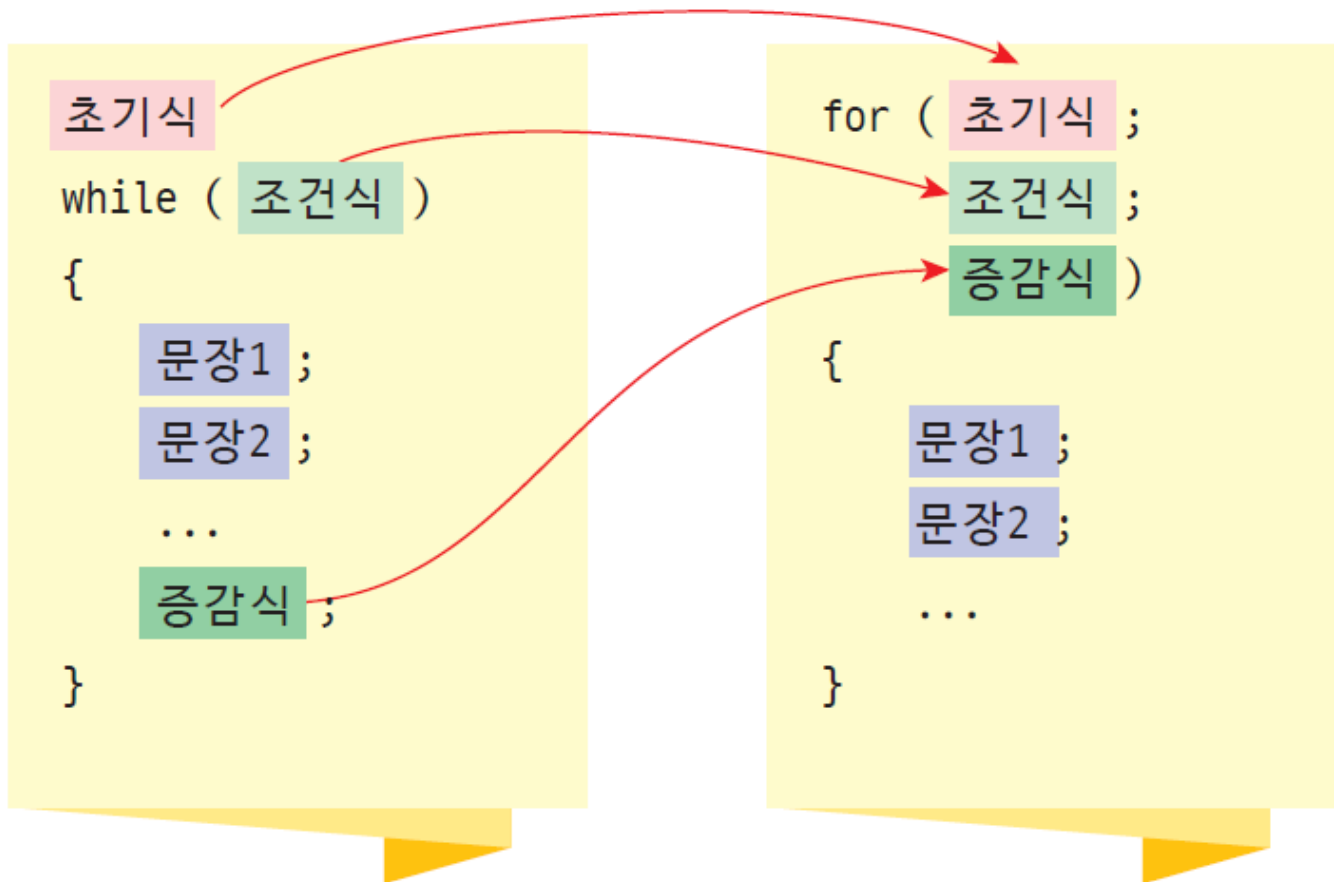
return 0;

}



while 루프와 for 루프와의 관계

62



팩토리얼 계산 예제(while 버전)

63

```
// 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;
}
```

C11부터는 for 루프 안에서 변수 선언 가능

64

```
for(int i =0; i< 10; i++) {  
    ...  
}
```


다양한 중감수식의 형태

65

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

뱀셈 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

66

```
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

67

1. 다음 코드의 출력을 쓰시오.

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

2. 다음 코드의 출력을 쓰시오.

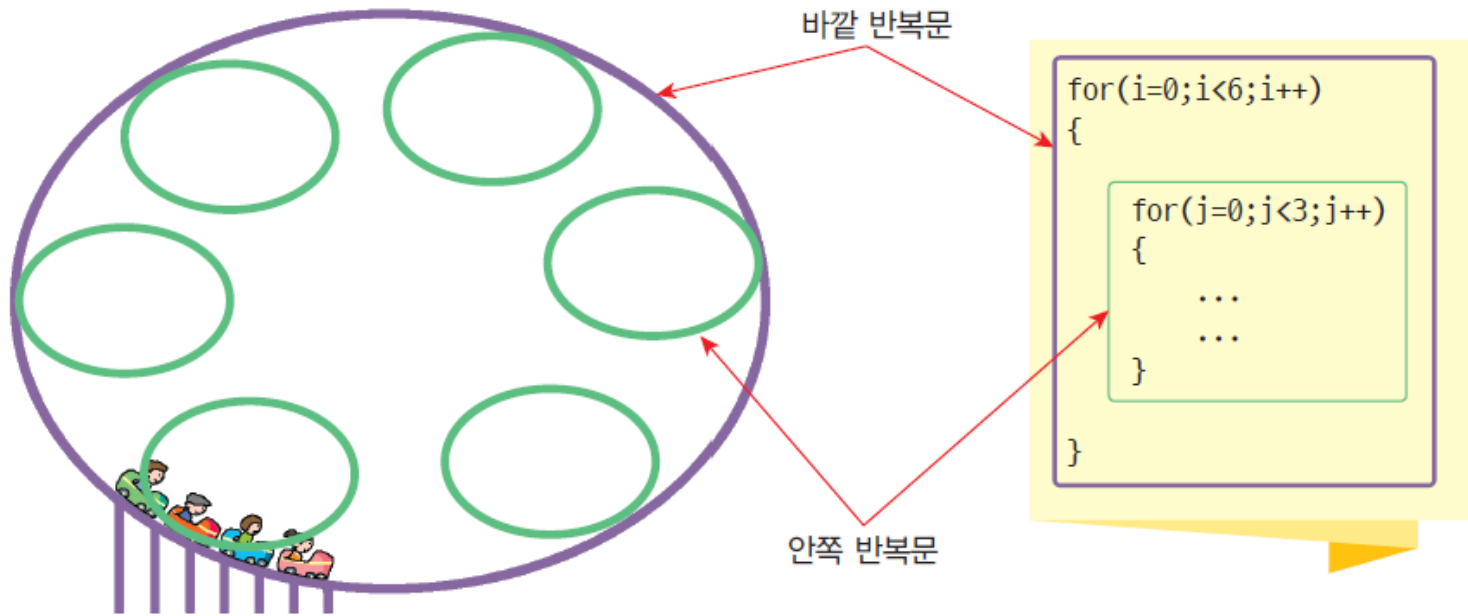
```
for(i = 10; i > 0; i = i - 2)  
    printf("Student%d□\n", i);
```



중첩 반복문

68

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



예제

69

```
// 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

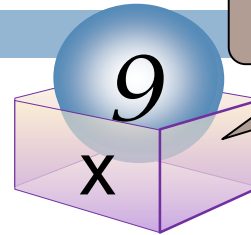
```
        printf("*");
```

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

```
}
```

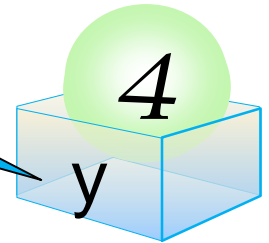
```
    return 0;
```

```
}
```



0부터 9까지 10번 반복

0부터 4까지 5번 반복



```
***** Wn  
***** Wn  
***** Wn  
***** Wn  
***** Wn
```

예제

70

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

    for(y = 1; y <= 5; y++)
    {
        for(x = 0; x < y; x++)
            printf("*");
        printf("\n");           // 내부 반복문이 종료될 때마다 실행
    }
    return 0;
}
```



중간 점검

71

1. 다음 코드의 출력을 쓰시오.

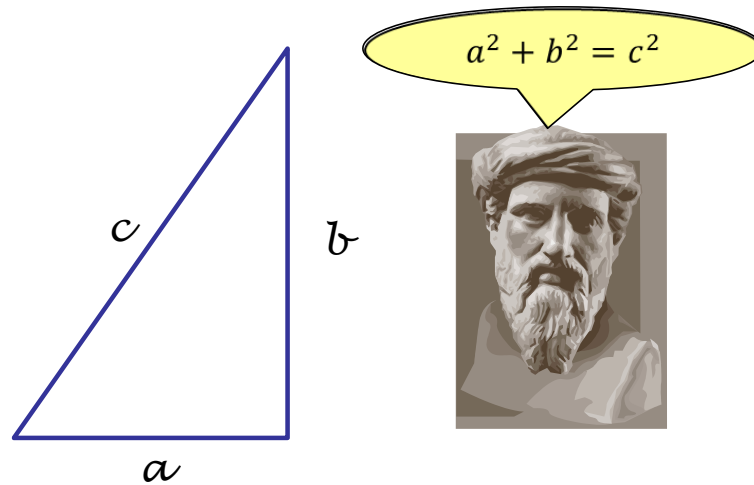
```
for(i = 1; i < 6; i++)  
    for(j = 5; j >= 1; j--)  
        printf("%d 곱하기 %d은 %d\n", i, j, i*j);
```



Lab: 직각 삼각형 찾기

72

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



실행 결과

73



알고리즘

74

```
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.
```



```
#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

76

- 위와 비슷한 문제를 하나 더 작성해보자. 라스베가스과 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.

Contents

77

7.1

반복문의 개념

7.2

while 문

7.3

반복 루프에서 보조값 사용하기

7.4

do-while 문

7.5

for문

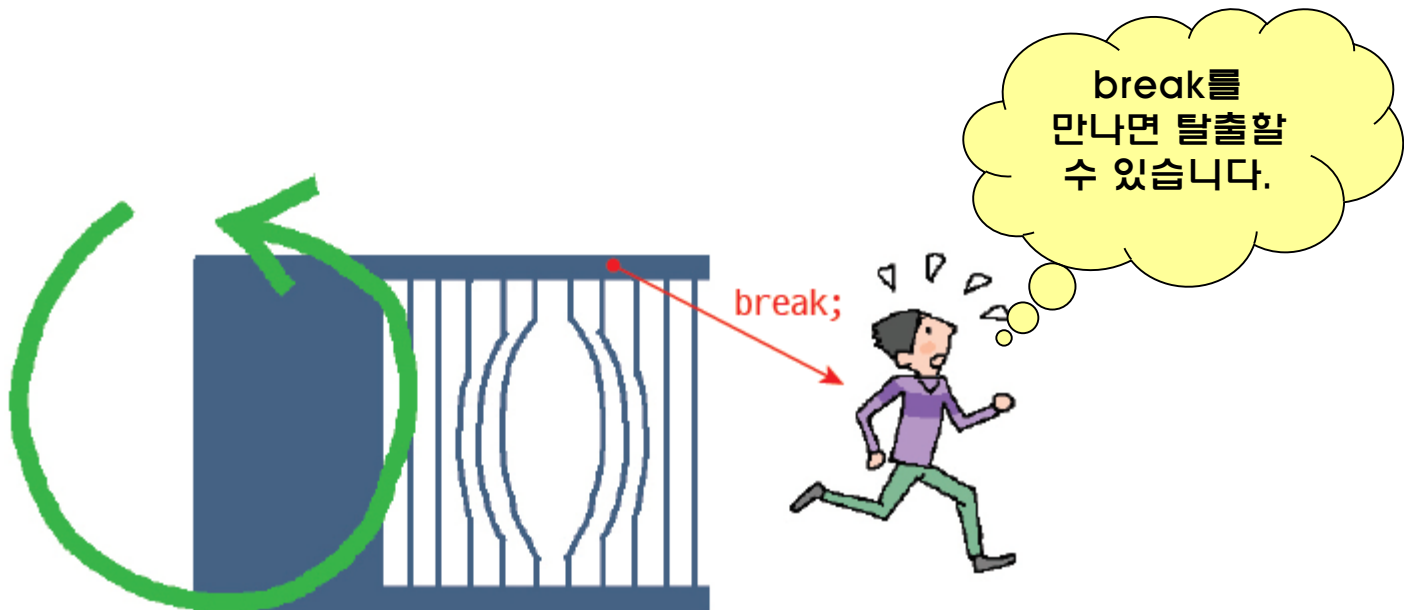
7.6

break와 continue

break 문

78

- break 문은 반복 루프를 빠져 나오는데 사용된다.



반복루프

break 문

79

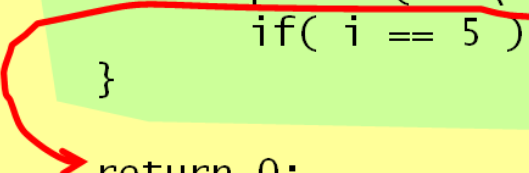
for 반복 루프

```
#include <stdio.h>

int main(void)
{
    int i;

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

    return 0;
}
```



예제

80

```
#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

81

// break를 이용하여 무한루프를 탈출한다.

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

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

```
int main(void)
```

```
{
```

```
    double v;
```

```
    while(1)
```

```
    {
```

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

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

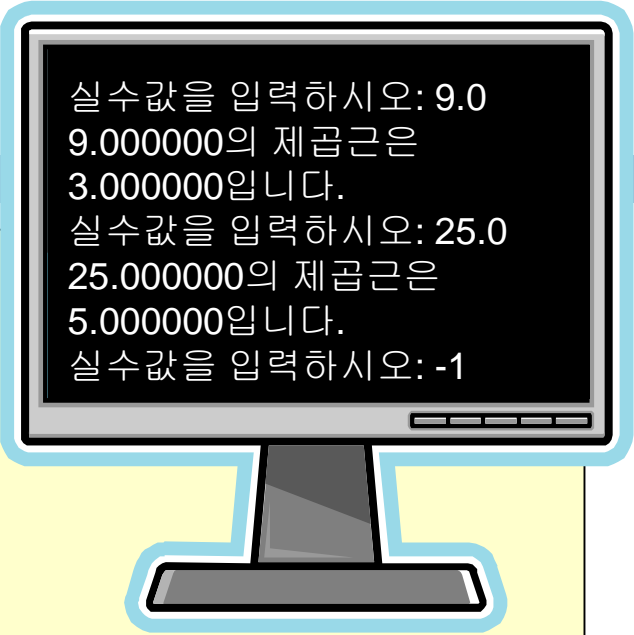
```
        if( v < 0.0 )
```

```
            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

82

```
#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

83

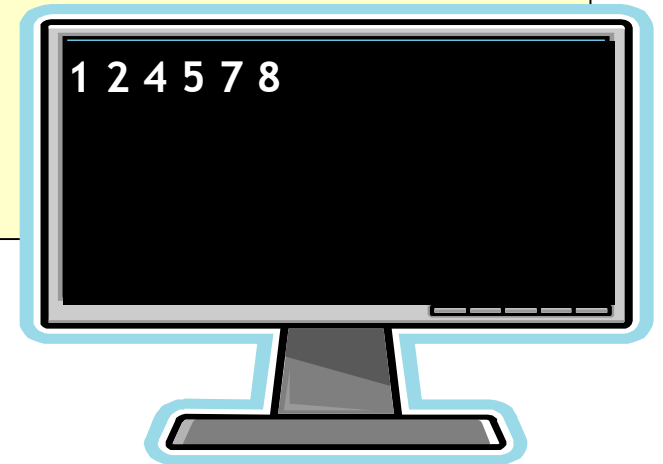
- 현재의 반복을 중단하고 다음 반복을 시작하게 한다.

```
#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 배수는 건너뛰는다.

1 2 4 5 7 8



continue 문

84

```
while ( 조건식 )
```

```
{
```

```
    문장 ;
```

```
    문장 ;
```

```
    continue
```

```
    문장 ;
```

```
}
```



```
do
```

```
{
```

```
    문장 ;
```

```
    문장 ;
```

```
    continue
```

```
    문장 ;
```

```
} while ( 조건식 );
```



```
for ( 초기식 ;
```

```
    조건식 ;
```

```
    증감식 )
```

```
{
```

```
    문장 ;
```

```
    문장 ;
```

```
    continue
```

```
    문장 ;
```

```
}
```



예제

85

```
// 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 ;
```

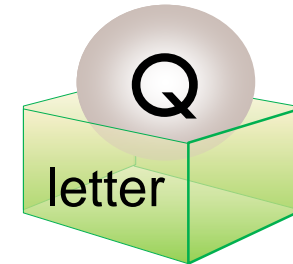
```
        letter -= 32;
```

```
        printf("변환된 대문자는 %c입니다.\n", letter);
```

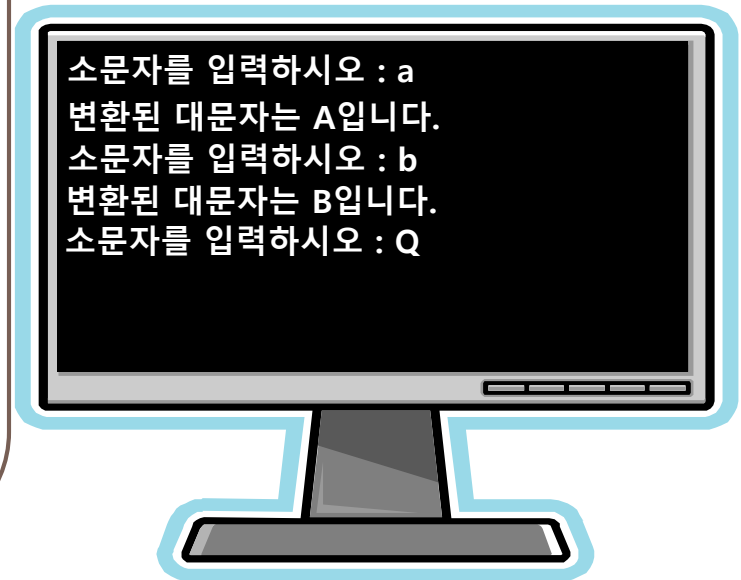
```
    }
```

```
    return 0;
```

```
}
```



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



중간 점검

86

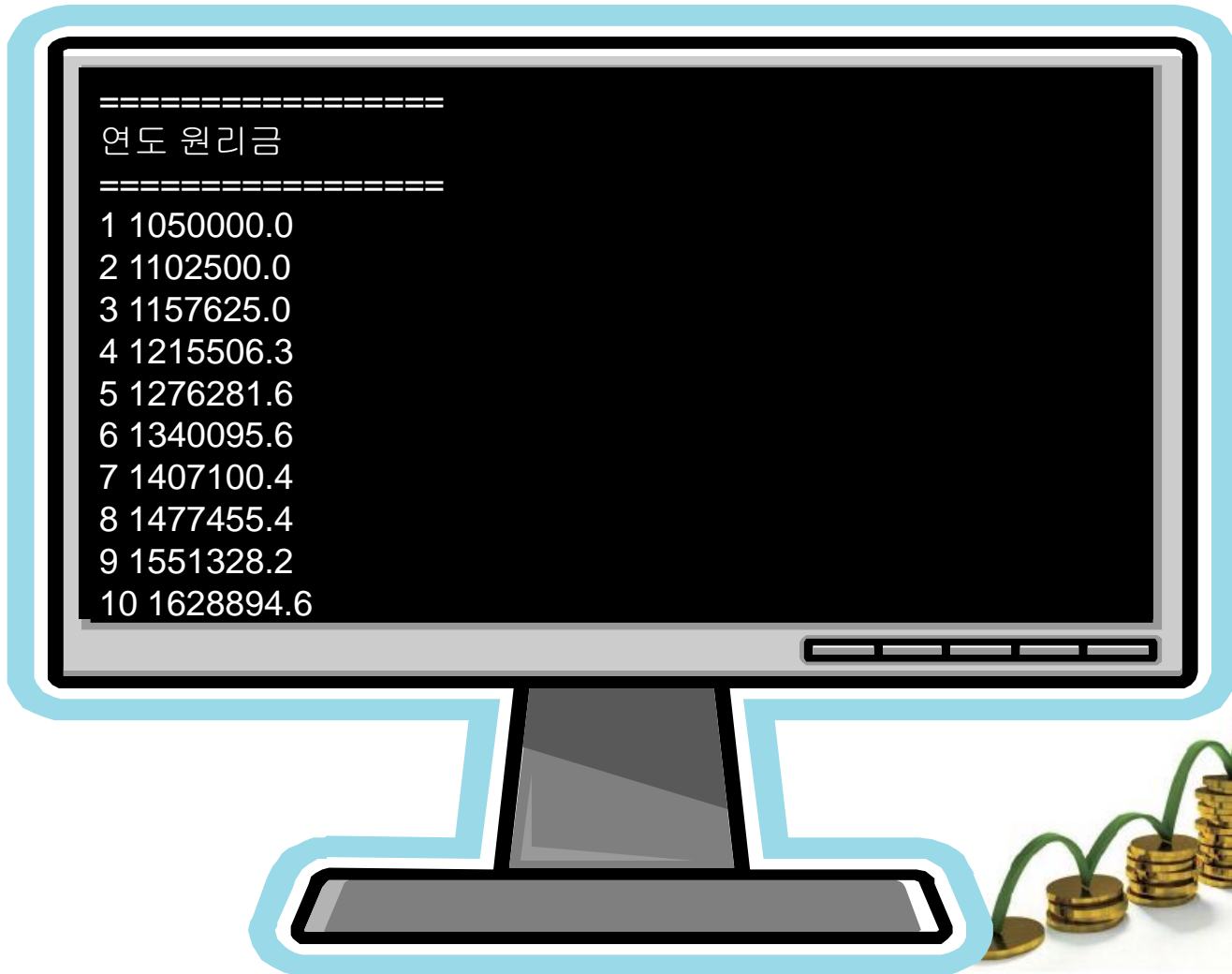
1. _____ 문이 반복문에서 실행되면 현재의 반복을 중단하고 다음번 반복 처리가 시작된다.
2. _____ 문이 반복문에서 실행되면 반복문을 빠져 나온다.
3. 다음 코드의 출력을 쓰시오.

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



lab: 복리 이자 계산

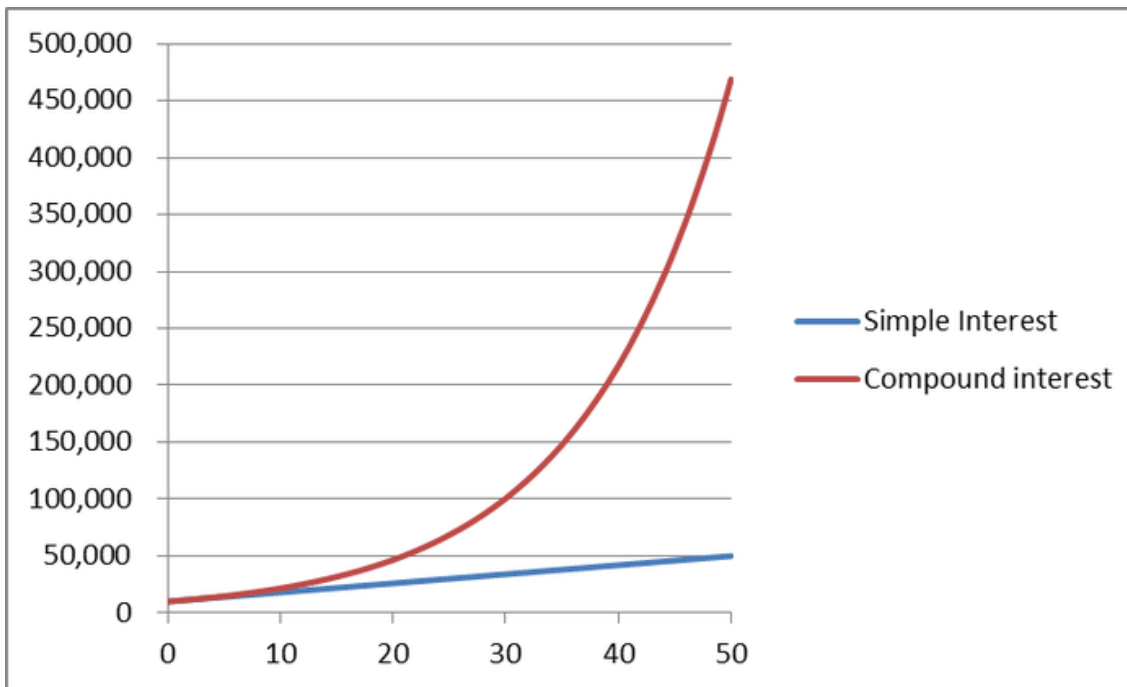
87



복리에서 원리금 합계

88

$$\text{원리합계} = \text{원금} \times (1 + \text{이율})^{\text{기간}}$$



복리에서 원리금 합계

89

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

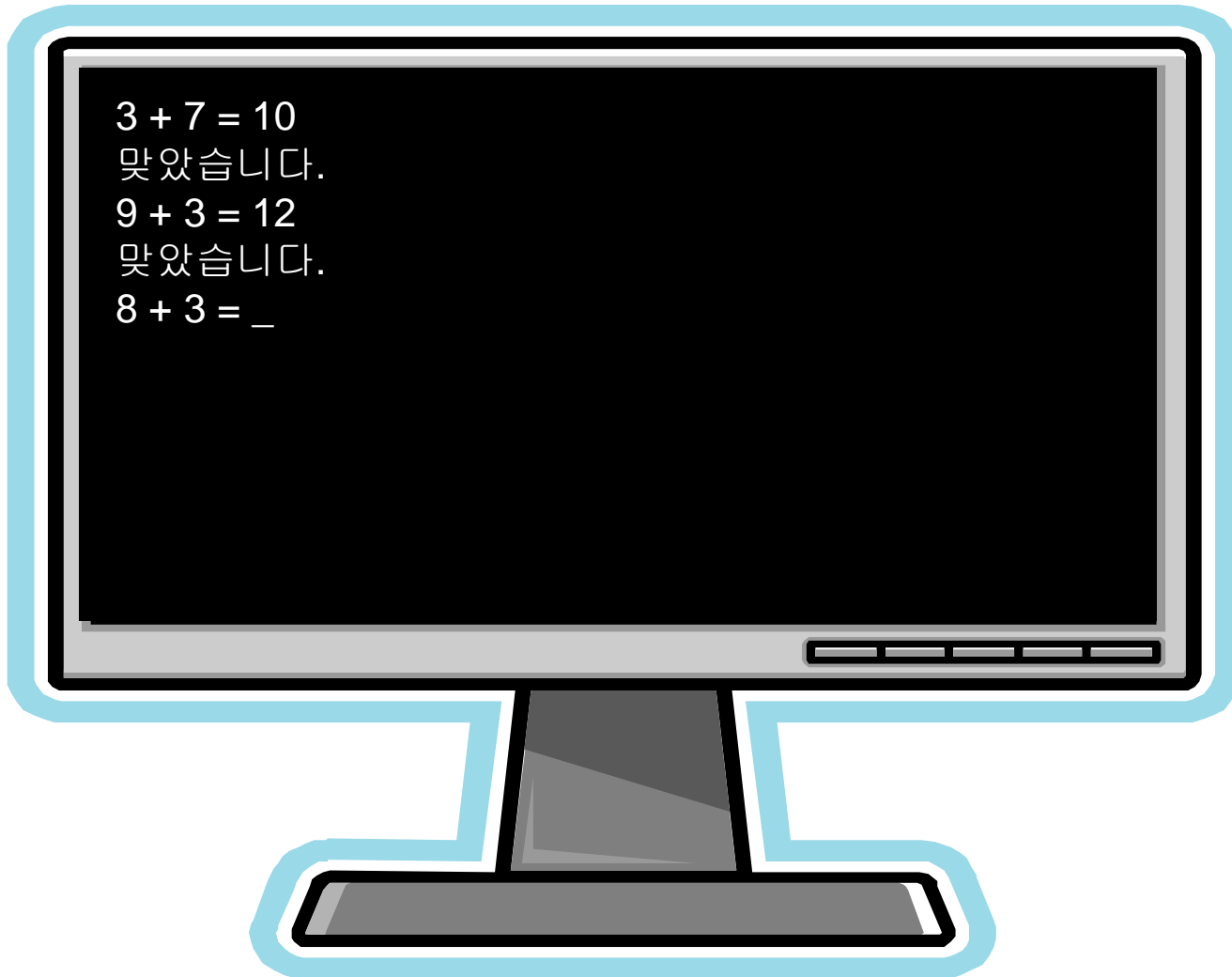
    printf("원금: ");
    scanf("%lf", &investment);
    printf("이율(%): ");
    scanf("%lf", &rate);
    printf("기간(년): ");
    scanf("%d", &years);

    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;
}
```

// 새로운 원리금 계산

Lab: 자동으로 수학문제 생성하기

90



난수 발생

91

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

int main(void)
{
    srand(time(NULL));
    for(int i=0;i<10;i++)
        printf("%d \n", rand());
}
```

자동으로 수학문제 생성하기

92

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

int main(void)
{
    int x, y, answer, i;
    srand(time(NULL));

    for (i = 0; i < 10; i++) {
        x = rand() % 10;
        y = rand() % 10;
        printf("%d + %d = ", x, y);
        scanf("%d", &answer);
        if (x + y == answer)
            printf("맞았습니다.\n");
        else
            printf("틀렸습니다.\n");
    }
    return 0;
}
```

Lab:도박사의 확률

93

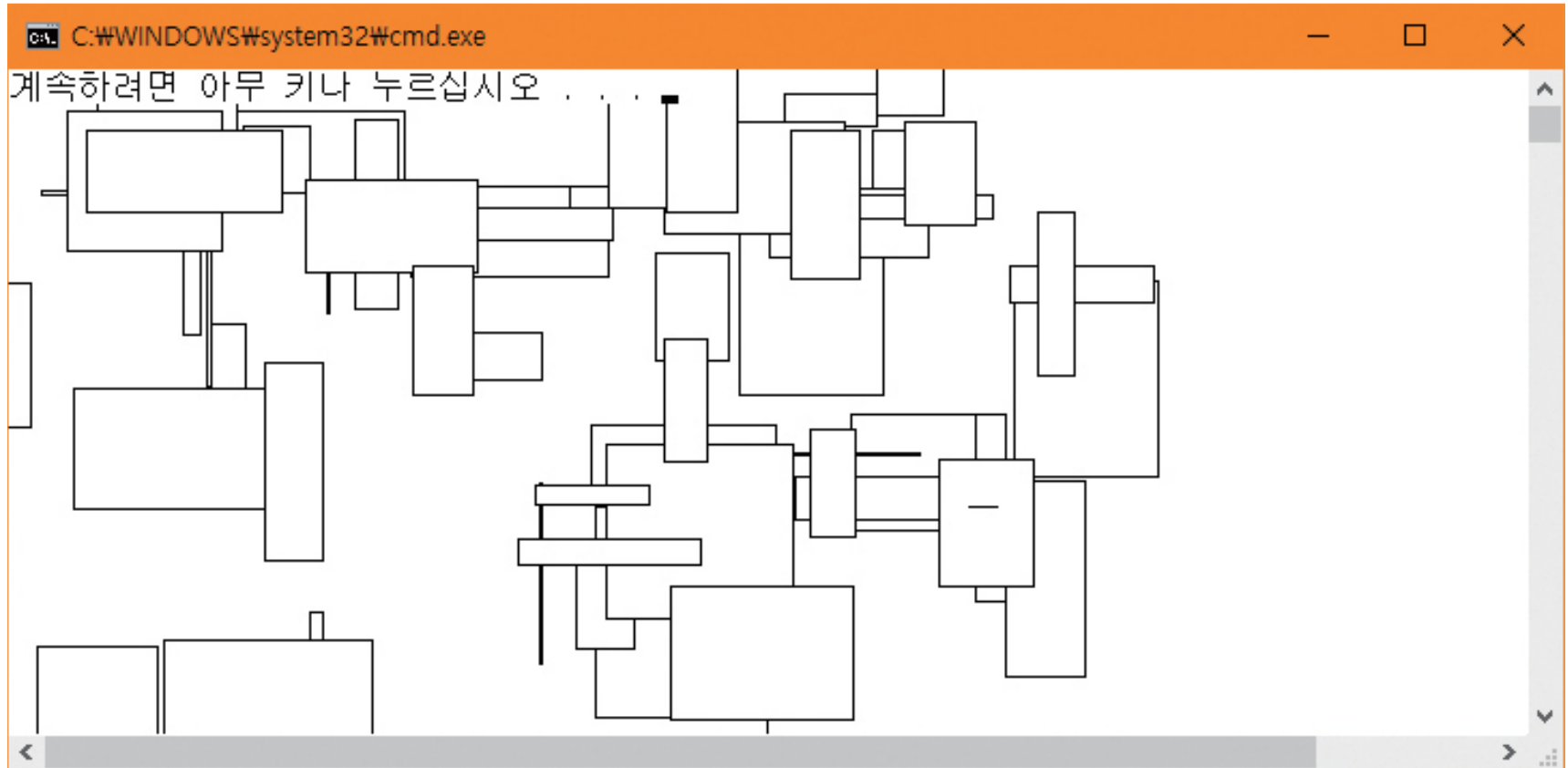


```
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
    int initial_money = 50;
    int goal = 250;
    int i;
    int wins = 0;

    for (i = 0; i < 100; i++) {
        int cash = initial_money;
        while (cash > 0 && cash < goal) {
            if (((double)rand() / RAND_MAX) < 0.5) cash++;
            else cash--;
        }
        if (cash == goal) wins++;
    }
    printf("초기 금액 %d \n", initial_money);
    printf("목표 금액 %d \n", goal);
    printf("100번 중에서 %d번 성공\n", wins);
    return 0;
}
```

Lab: 화면에 사각형 그리기

95



화면에 사각형 그리기

96

```
#include <windows.h>
#include <stdio.h>
int main(void)
{
    HDC hdc = GetWindowDC(GetForegroundWindow()); // 그리기 속성
    // 여기에 그림을 그리는 반복문을 추가한다.
    Rectangle(hdc, x, y, x + w, y + h); // 화면에 사각형을 그린다.
    ...
    return 0;
}
```


화면에 사각형 그리기

97

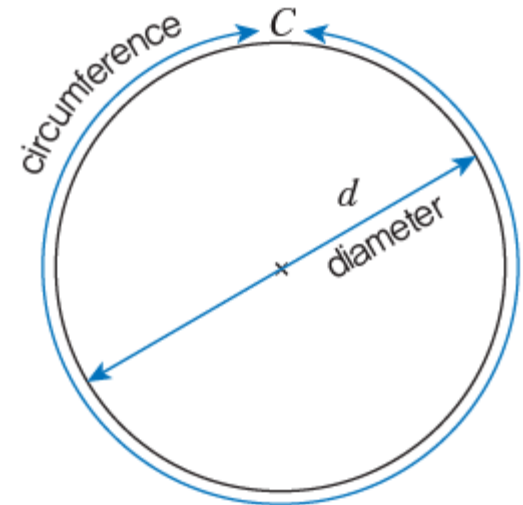
```
#include <windows.h>
#include <stdio.h>
int main(void)
{
    HDC hdc = GetWindowDC(GetForegroundWindow()); // 그리기 속성
    int i;
    for (i = 0; i < 100; i++) {
        int x = rand() % 500;
        int y = rand() % 300;
        int w = rand() % 100;
        int h = rand() % 100;
        Rectangle(hdc, x, y, x + w, y + h); // 화면에 사각형을 그린다.
        Sleep(100);
    }
    return 0;
}
```

Lab: 파이 구하기

98

- 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

99

반복횟수:100000

$\text{Pi} = 3.141583$

계속하려면 아무 키나 누르십시오 ...



알고리즘

100

사용자로부터 반복횟수 loop_count를 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;
```

```
sum을 출력한다.
```

실습 코드

101

```
#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;
}
```