

-Part1-

제7장 반복문이란 무엇인가

학습목차

7.1 반복문이란

7.2 반복문을 만드는 방법 1 –while문

7.3 반복문을 만드는 방법 2 –for문

7.4 반복문을 만드는 방법 3 –do~while문

7.1 반복문이란

반복문을 배우기 전에...

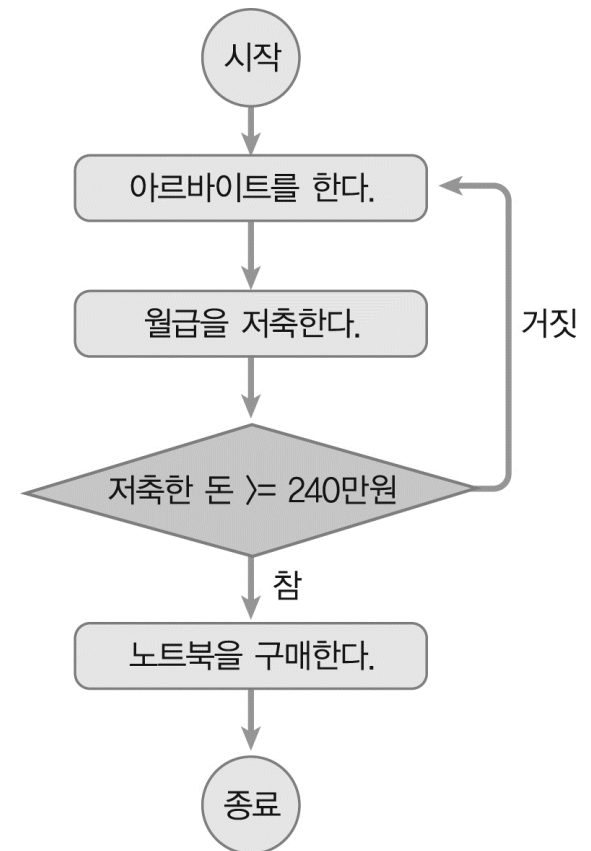
▶ “대학생 정민은 최신 노트북을 사려고 아르바이트를 한다.”

▶ 필요한 돈은 240만원

▶ 아르바이트로 매달 50만원을 받아 저축

▶ 240만원이 모일 때 까지 반복해서 저축

▶ 노트북 구매 후 아르바이트 종료



7.1 반복문이란 (1/1)

▶ 반복문

- ✓ 어떤 행동을 반복적으로 수행하는 문장

▶ 반복문의 종류

- ✓ while문
- ✓ for문
- ✓ do ~ while문

7.2 반복문을 만드는 방법1 – while 문

7.2 반복문을 만드는 방법1 - while문(1/9)

▶ while문을 위해 배울 내용

- ① while문의 기본 문법
- ② while 무한 루프(무한 반복문)
- ③ 중첩 while문

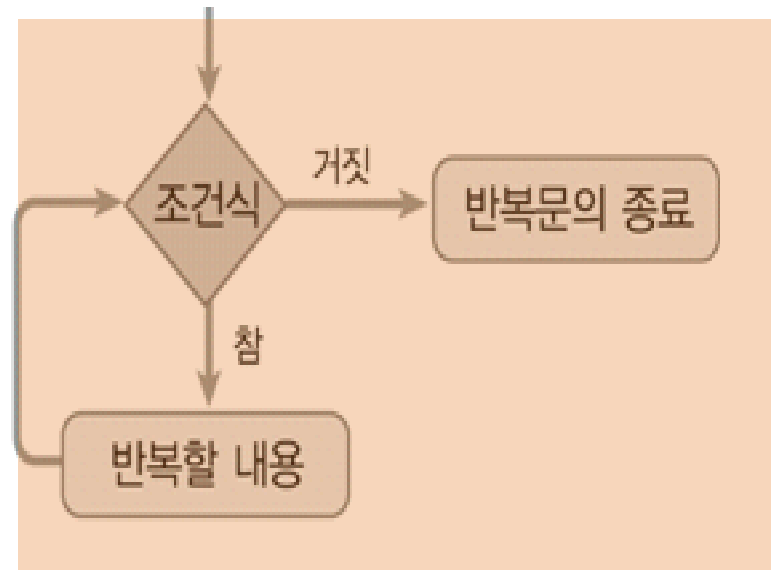
7.2 반복문을 만드는 방법1 - while문(2/9)

① while 문의 기본 문법

✓ while 문의 의미

- '~하는 동안'
- '조건식'이 '참'인 동안 '반복할 내용'을 반복해라

```
while (조건식)
반복문의 시작 {
    반복할 내용
반복문의 종료 }
```

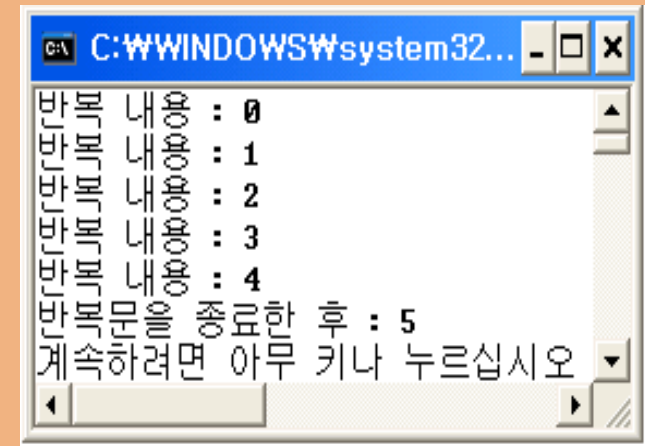


7.2 반복문을 만드는 방법1 – while문(3/9)---[7-1.c 실습]

```
#include <stdio.h>
int main(void)
{
    int num=0;

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

    return 0;
}
```



```
C:\WINDOWS\system32...
반복 내용 : 0
반복 내용 : 1
반복 내용 : 2
반복 내용 : 3
반복 내용 : 4
반복문을 종료한 후 : 5
계속하려면 아무 키나 누르십시오
```

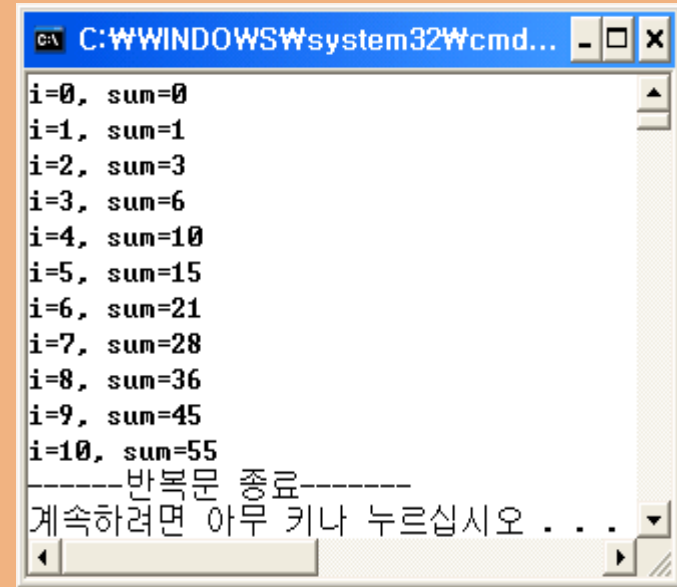
7.2 반복문을 만드는 방법1 - while문(4/9)---[7-1.c 분석]



7.2 반복문을 만드는 방법1 – while문(5/9)---[7-2.c 실습]

```
#include <stdio.h>
int main(void)
{
    int i=0, sum=0;
    while(i<=10)
    {
        sum=sum+i;
        printf("i=%d, sum=%d\n", i, sum);
        i++;
    }
    printf("-----반복문 종료-----\n");

    return 0;
}
```



```
C:\WINDOWS\system32\cmd...
i=0, sum=0
i=1, sum=1
i=2, sum=3
i=3, sum=6
i=4, sum=10
i=5, sum=15
i=6, sum=21
i=7, sum=28
i=8, sum=36
i=9, sum=45
i=10, sum=55
-----반복문 종료-----
계속하려면 아무 키나 누르십시오 . . .
```

7.2 반복문을 만드는 방법1 - while문 (6/9)---[7-3.c 실습]

② while 무한 루프(무한 반복문)

✓ 종료되지 않고 무한히 실행되는 반복문

```
#include <stdio.h>
int main(void)
{
    int i=0;
    while(1) // 무조건 참
    {
        printf("반복 횟수 : %d \\\n", i);
        i++;

        // if(i>10)
        // break;
    }
    return 0;
}
```

7.2 반복문을 만드는 방법1 - while문(7/9)

③ 중첩 while문 : 'while문 내부에 또 다른 while문이 있는 것을 말한다.'

```
#include <stdio.h>
int main(void)
{
    int i=0, j=0;
    while(i<2)
    {
        printf("큰 반복 %d\n", i);
        while(j<2)
        {
            printf("작은 반복 %d\n", j);
            j++;
        }
        i++;
        j=0;
    }
    return 0;
}
```



```
C:\WINDOWS\system32\cmd.exe
큰 반복
작은 반복
작은 반복
큰 반복
작은 반복
작은 반복
계속하려면 아무 키나 누르십시오...
```

7.2 반복문을 만드는 방법1 - while문(8/9)---[7-4.c 실습]

```
#include <stdio.h>
int main(void)
{
    int i=2;           // 2단부터
    int j=1;           // 2*1에서 1의 의미로 초기화
    int result=0;      // 구구단의 결과 저장 변수

    while(i<10)        // 9단까지
    {
        while(j<10)
        {
            result=i*j;
            printf("%d * %d = %d \n", i, j, result);
            j++;
        }
        i++;           // 단을 증가
        j=1;           // 단의 시작
        printf("-----\n");
    }
    return 0;
}
```



```
C:\WINDOWS\system32...
7 * 3 = 21
7 * 4 = 28
7 * 5 = 35
7 * 6 = 42
7 * 7 = 49
7 * 8 = 56
7 * 9 = 63
-----
8 * 1 = 8
8 * 2 = 16
8 * 3 = 24
8 * 4 = 32
8 * 5 = 40
8 * 6 = 48
8 * 7 = 56
8 * 8 = 64
8 * 9 = 72
-----
9 * 1 = 9
9 * 2 = 18
9 * 3 = 27
9 * 4 = 36
9 * 5 = 45
9 * 6 = 54
9 * 7 = 63
9 * 8 = 72
9 * 9 = 81
-----
계속하려면 아무 키나 누르십시오
```

7.2 반복문을 만드는 방법1 - while문(9/9)---[7-5.c 실습]

```
#include <stdio.h>
int main(void)
{
    int num=0, j=9, result=0;
    printf("숫자를 입력하세요 : ");
    scanf("%d", &num);

    while(num>0)
    {
        while(j>0)
        {
            result=num*j;
            printf("%d * %d = %d\n", num, j, result);
            j--;
        }
        num--;
        j=9;
        printf("-----\n");
    }
    return 0;
}
```



```
C:\WINDOWS\system32\cmd.exe
숫자를 입력하세요 : 3
3 * 9 = 27
3 * 8 = 24
3 * 7 = 21
3 * 6 = 18
3 * 5 = 15
3 * 4 = 12
3 * 3 = 9
3 * 2 = 6
3 * 1 = 3
-----
2 * 9 = 18
2 * 8 = 16
2 * 7 = 14
2 * 6 = 12
2 * 5 = 10
2 * 4 = 8
2 * 3 = 6
2 * 2 = 4
2 * 1 = 2
-----
1 * 9 = 9
1 * 8 = 8
1 * 7 = 7
1 * 6 = 6
1 * 5 = 5
1 * 4 = 4
1 * 3 = 3
1 * 2 = 2
1 * 1 = 1
-----
계속하려면 아무 키나 누르십시오
```

7.3 반복문을 만드는 방법2 – for 문

7.2 반복문을 만드는 방법2 – for문(1/14)

▶ for문을 위해 배울 내용

- ① for문의 기본 문법
- ② for 무한 루프(무한 반복문)
- ③ 중첩 for문
- ④ for문의 여러 가지 변형

7.3 반복문을 만드는 방법2 – for문(2/14)

① for 문의 기본 문법

✓ for 문의 의미

- ‘~하는 동안’
- ‘조건식’이 ‘참’인 동안 ‘반복할 내용’을 반복해라

```
for (초기값; 조건값; 증감값)  
반복문의 시작 → {  
    반복할 내용  
반복문의 종료 → }
```

해석 : 조건값이 참인 동안 반복할 내용을 반복적으로 수행해라!

7.3 반복문을 만드는 방법2 – for문(3/14)---[7-6.c 실습]

```
#include <stdio.h>
int main(void)
{
    int num;
    for(num=0; num<5; num++)
    {
        printf("반복 내용 : %d \n", num);
    }
    printf("반복문을 종료한 후 : %d \n", num);
    return 0;
}
```

7.3 반복문을 만드는 방법2 – for문(4/14)

▶ while 문과 for문의 비교

✓ 7-1.c 와 7-6.c를 비교

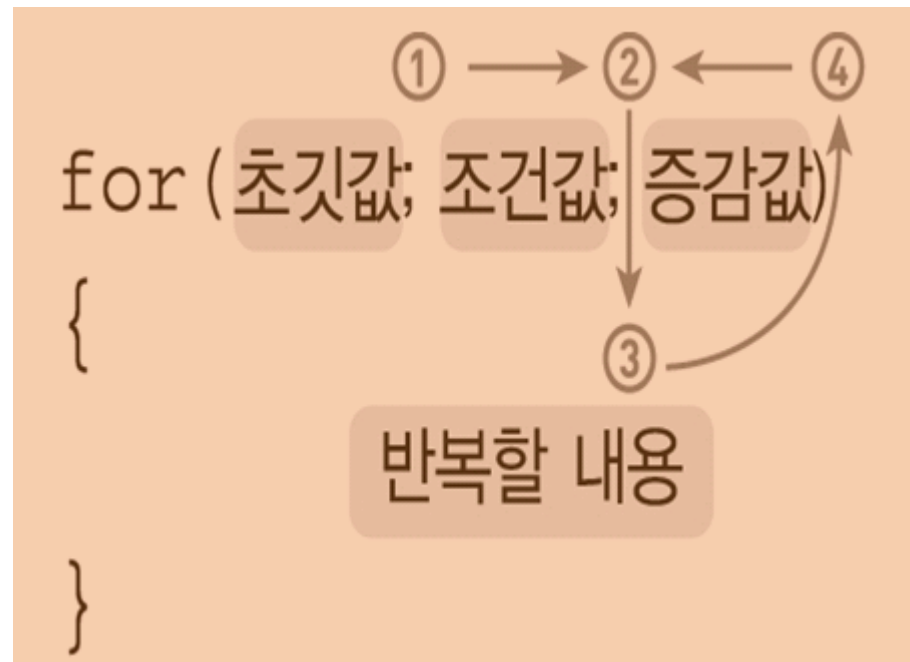
```
/* 7-1.c */  
int num = 0; -----> 초기값  
  
while(num < 5) --> 조건값  
{  
    printf("반복 내용: %d \n", num);  
    num++; -----> 증감값  
}
```

==

```
/* 7-6.c */  
int num;  
  
for( num = 0; num < 5; num++ )  
{  
    printf("반복 내용: %d \n", num);  
}
```

7.3 반복문을 만드는 방법2 – for문(5/14)

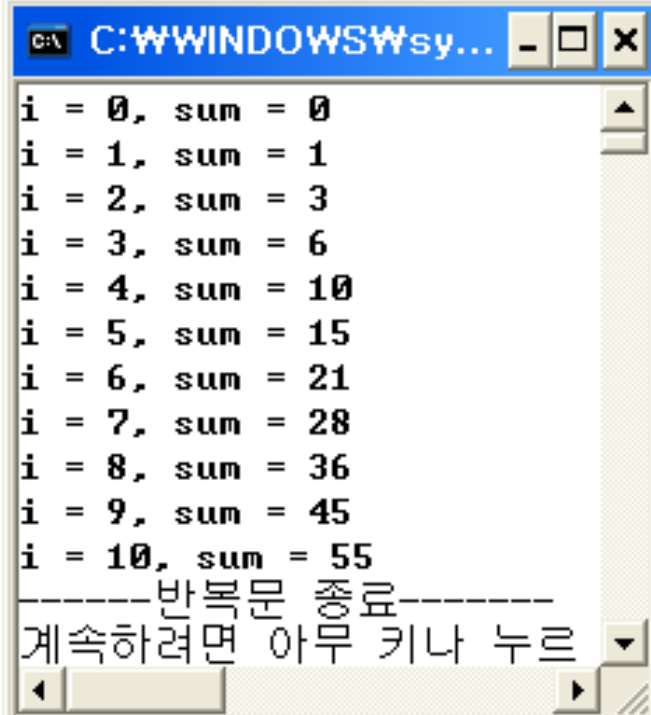
▶ for 문의 실행 순서



해석 : ①초깃값은 무조건 한 번만 수행하고 ②조건값, ③ 반복할 내용, ④ 증감값은 반복적으로 수행해라!

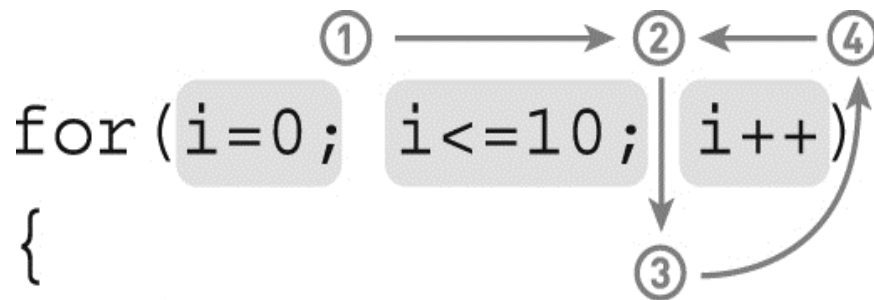
7.3 반복문을 만드는 방법2 – for문(6/14)---[7-7.c 실습]

```
#include <stdio.h>
int main(void)
{
    int i, sum=0;
    for(i=0; i<=10; i++)
    {
        sum=sum+i;
        printf("i = %d, sum = %d\n", i, sum);
    }
    printf("-----반복문 종료-----\n");
    return 0;
}
```



```
C:\WINDOWS\system32\cmd.exe
i = 0, sum = 0
i = 1, sum = 1
i = 2, sum = 3
i = 3, sum = 6
i = 4, sum = 10
i = 5, sum = 15
i = 6, sum = 21
i = 7, sum = 28
i = 8, sum = 36
i = 9, sum = 45
i = 10, sum = 55
-----반복문 종료-----
계속하려면 아무 키나 누르
```

7.3 반복문을 만드는 방법2 – for문(7/14)---[7-7.c 분석]



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

```
    sum=sum+i;
```

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

```
}
```

7.3 반복문을 만드는 방법2 – for문(8/14)---[7-8.c 실습]

```
#include <stdio.h>
int main(void)
{
    int num=0, i, result=0;
    printf("숫자를 입력하세요 : ");
    scanf("%d", &num);

    for(i=1; i<10; i=i+2)
    {
        result=num*i;
        printf("%d * %d = %d입니다. \n", num, i, result);
    }
    return 0;
}
```


7.3 반복문을 만드는 방법2 – for문(9/14)---[7-9.c 실습]

② for 무한 루프(무한 반복문)

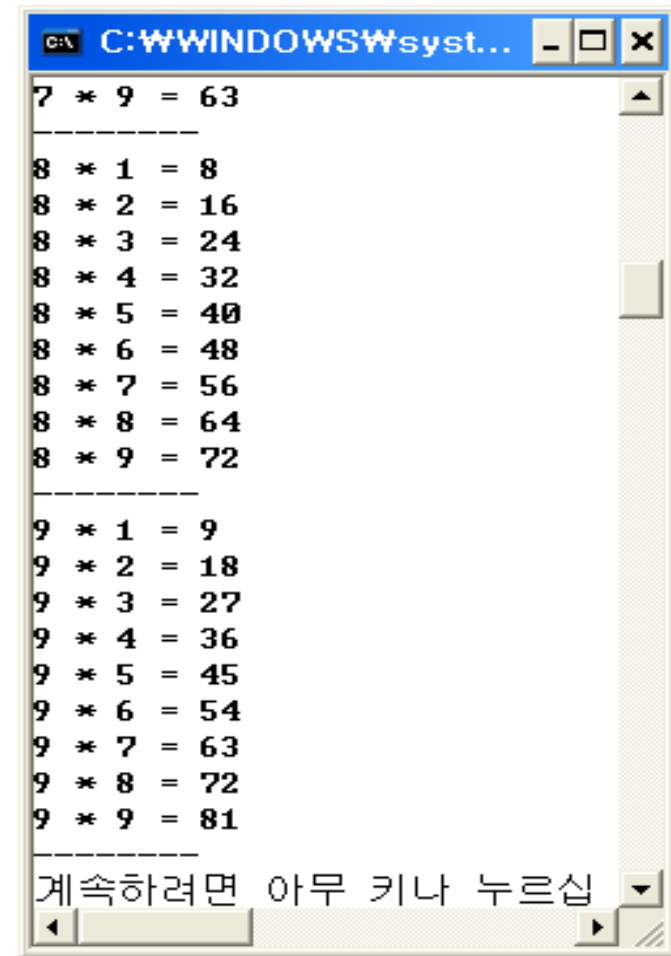
✓ 종료되지 않고 무한히 실행되는 반복문

```
#include <stdio.h>
int main(void)
{
    int i;
    for(i=0; 1; i++)
    {
        printf("반복 횟수 : %d\n", i);
    }
    return 0;
}
```

7.3 반복문을 만드는 방법2 – for문(10/14)---[7-10.c 실습]

③ 중첩 for문 : 'for문 내부에 또 다른 for문이 있는 것을 말한다.'

```
#include <stdio.h>
int main(void)
{
    int i, j;
    int result=0;    // 구구단의 결과 저장 변수
    for(i=2; i<10; i++)
    {
        for(j=1; j<10; j++)
        {
            result=i*j;
            printf("%d * %d = %d\n", i, j, result);
        }
        printf("-----\n");
    }
    return 0;
}
```

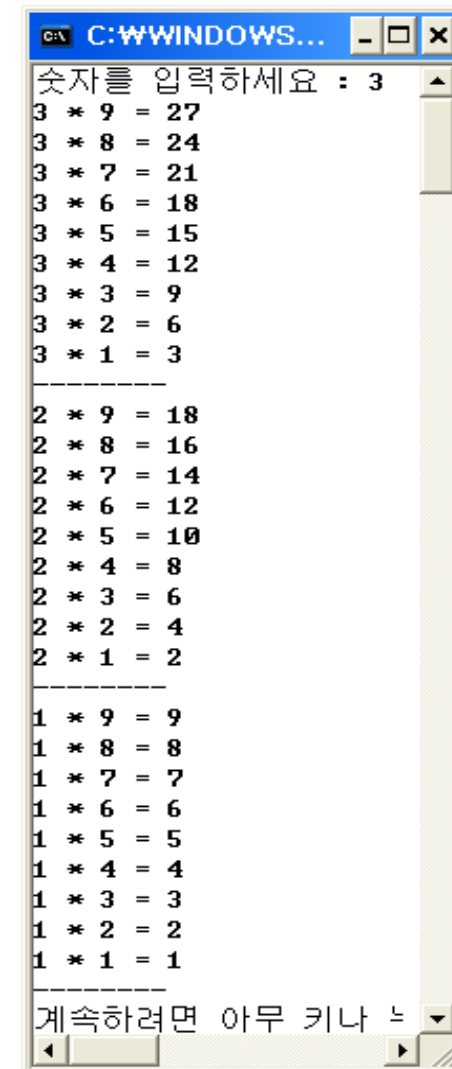


```
C:\WINDOWS\system32\cmd.exe
7 * 9 = 63
-----
8 * 1 = 8
8 * 2 = 16
8 * 3 = 24
8 * 4 = 32
8 * 5 = 40
8 * 6 = 48
8 * 7 = 56
8 * 8 = 64
8 * 9 = 72
-----
9 * 1 = 9
9 * 2 = 18
9 * 3 = 27
9 * 4 = 36
9 * 5 = 45
9 * 6 = 54
9 * 7 = 63
9 * 8 = 72
9 * 9 = 81
-----
계속하려면 아무 키나 누르십시오
```

7.3 반복문을 만드는 방법2 – for문(11/14)---[7-11.c 실습]

```
#include <stdio.h>
int main(void)
{
    int num, j=9, result=0;
    printf("숫자를 입력하세요 : ");
    scanf("%d", &num);

    for( ; num>0; num--)
    {
        for(j=9; j>0; j--)
        {
            result=num*j;
            printf("%d * %d = %d\\n", num, j, result);
        }
        printf("-----\\n");
    }
    return 0;
}
```



```
C:\WINDOWS... - □ ×
숫자를 입력하세요 : 3
3 * 9 = 27
3 * 8 = 24
3 * 7 = 21
3 * 6 = 18
3 * 5 = 15
3 * 4 = 12
3 * 3 = 9
3 * 2 = 6
3 * 1 = 3
-----
2 * 9 = 18
2 * 8 = 16
2 * 7 = 14
2 * 6 = 12
2 * 5 = 10
2 * 4 = 8
2 * 3 = 6
2 * 2 = 4
2 * 1 = 2
-----
1 * 9 = 9
1 * 8 = 8
1 * 7 = 7
1 * 6 = 6
1 * 5 = 5
1 * 4 = 4
1 * 3 = 3
1 * 2 = 2
1 * 1 = 1
-----
계속하려면 아무 키나 누르십시오
```

7.3 반복문을 만드는 방법2 – for문(12/14)---[7-13.c 실습]

④ for문의 여러 가지 변형

```
#include <stdio.h>
int main( )
{
    int i=1, factorial=1;
    for( ; i<=10; i++)           // 초깃값이 없는 경우
    {
        factorial = factorial * i;
    }
    printf("1부터 10까지의 곱 : %d\n", factorial);

    return 0;
}
```

7.3 반복문을 만드는 방법2 – for문(13/14)---[7-14.c 실습]

```
#include <stdio.h>
int main( )
{
    int i, factorial=1;
    for(i=1; ; i++) // 조건값이 없는 경우
    {
        factorial=factorial*i;

        if (i>10)
            break; // 무한 반복문을 탈출하기 위한 break 문
    }
    printf("1부터 10까지의 곱 : %d\n", factorial);
    return 0;
}
```

7.3 반복문을 만드는 방법2 – for문(14/14)---[7-15.c 실습]

```
#include <stdio.h>
int main( )
{
    int i, factorial=1;
    for(i=1; i<=10; )    // 증감값이 없는 경우
    {
        factorial=factorial*i;
        i++;            // 증감값을 여기에...
    }
    printf("1부터 10까지의 곱 : %d\n", factorial);
    return 0;
}
```

7.4 반복문을 만드는 방법3 – do ~ while 문

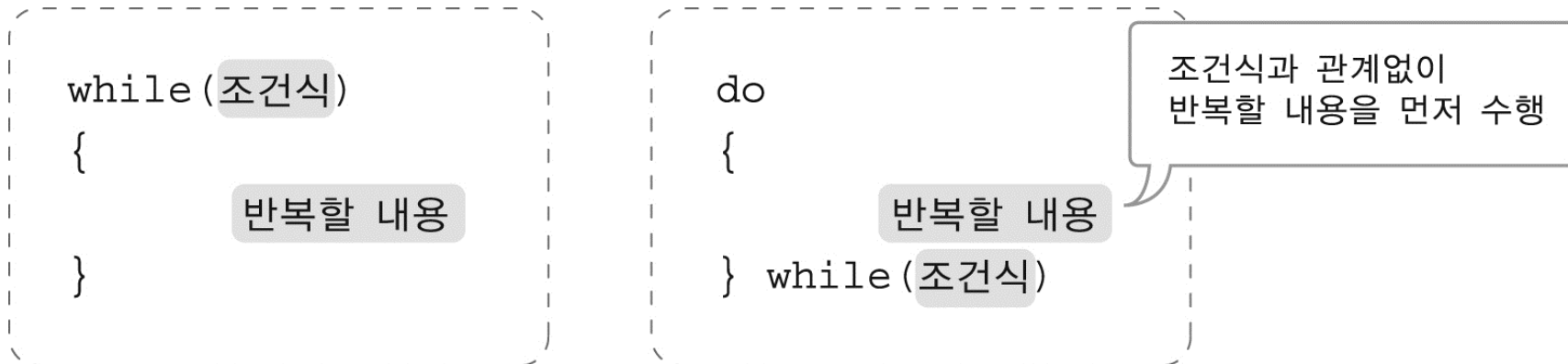
7.4 반복문을 만드는 방법 3 – do~while 문 (1/2)

▶ do ~ while 문

- ✓ 'while 문과 for문에 비해 사용 빈도가 적다.'

▶ while 문과 do~while 문 비교

- ✓ while 문: '조건이 만족하지 않으면 반복할 내용을 실행하지 않는다.'
- ✓ do~while 문: '최소한 한번은 반복할 내용을 실행한다.'



7.4 반복문을 만드는 방법3 – do~while문(2/2)---[7-16.c 실습]

```
#include <stdio.h>
int main( )
{
    int num=10;

    do
    {
        printf("%d", num);
        num++;
    }while(num<10);

    printf("\n **while 문을 종료합니다.** \n");
    return 0;
}
```

공부한 내용 떠올리기

- ▶ 반복문의 의미
- ▶ while 문, for 문, do~while 문을 이용해 반복문을 만드는 방법
- ▶ do~while 문과 while 문의 차이
- ▶ 무한 루프(무한 반복문)
- ▶ 중첩 반복문을 사용하는 방법

꽃 도둑을 배려한 가위 (출처: 사랑과 지혜의 탈무드)

