

기초프로그래밍

제7장 반복문

Sangsoo Lim

CSAI

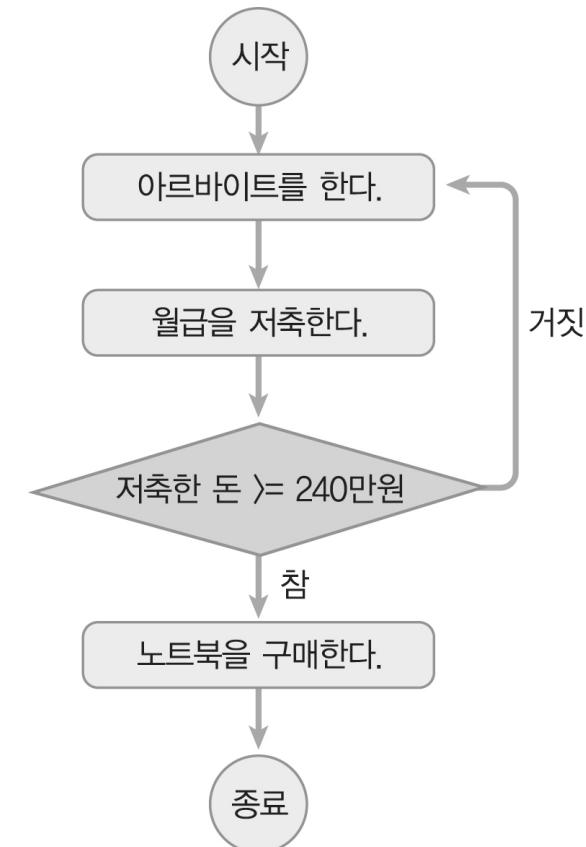
Dongguk University

차례

- 반복문이란
- 반복문을 만드는 방법 1 – while문
- 반복문을 만드는 방법 2 – for문
- 반복문을 만드는 방법 3 – do~while문

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



반복문이란

- 반복문
 - 어떤 행동을 반복적으로 수행하는 문장
- 반복문의 종류
 - while문
 - for문
 - do ~ while문

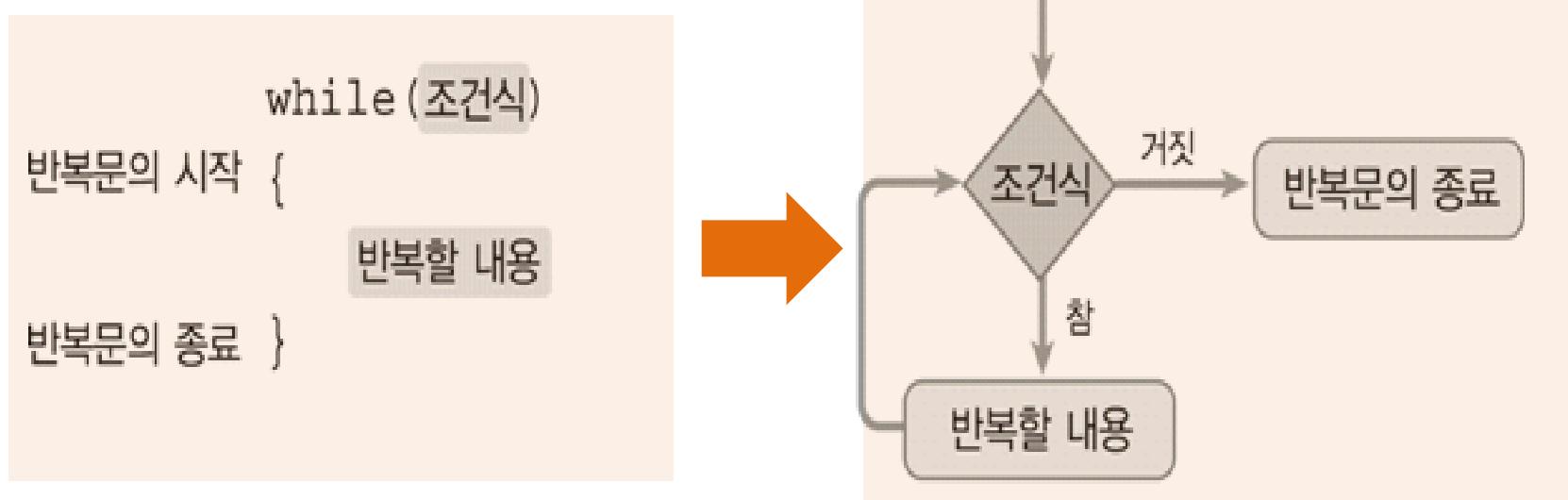
반복문을 만드는 방법1 – while문

- **while문을 위해 배울 내용**
 - ① while문의 기본 문법
 - ② while 무한 루프(무한 반복문)
 - ③ 중첩 while문

반복문을 만드는 방법1 – while문

① while 문의 기본 문법

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



반복문을 만드는 방법1 – while문

```
/* 7-1.c */
#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;
}
```

```
반복 내용 : 0
반복 내용 : 1
반복 내용 : 2
반복 내용 : 3
반복 내용 : 4
반복문을 종료한 후 : 5
```

반복문을 만드는 방법1 – while문



반복문을 만드는 방법1 – while문

```
/* 7-2.c */
#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;
}
```

```
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 -----
```

반복문을 만드는 방법1 – while문

② while 무한 루프(무한 반복문)

- 종료되지 않고 무한히 실행되는 반복문

```
/* 7-3.c */
#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;
}
```

반복문을 만드는 방법1 – while문

③ 중첩 while문 : while문 내부에 또 다른 while문이 있는 것

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

큰 반복
작은 반복
작은 반복
큰 반복
작은 반복
작은 반복

반복문을 만드는 방법1 – while문

```
/* 7-4.c */
#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;
}
```

| | |
|------------|-------|
| 2 * 1 = 2 | ----- |
| 2 * 2 = 4 | ----- |
| 2 * 3 = 6 | ----- |
| 2 * 4 = 8 | ----- |
| 2 * 5 = 10 | ----- |
| 2 * 6 = 12 | ----- |
| 2 * 7 = 14 | ----- |
| 2 * 8 = 16 | ----- |
| 2 * 9 = 18 | ----- |
| ----- | ----- |
| 3 * 1 = 3 | ----- |
| 3 * 2 = 6 | ----- |
| 3 * 3 = 9 | ----- |
| 3 * 4 = 12 | ----- |
| 3 * 5 = 15 | ----- |
| 3 * 6 = 18 | ----- |
| 3 * 7 = 21 | ----- |
| 3 * 8 = 24 | ----- |
| 3 * 9 = 27 | ----- |
| ----- | ----- |

| | |
|------------|-------|
| 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 | ----- |
| ----- | ----- |

반복문을 만드는 방법1 – while문

```
/* 7-5.c */
#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;
}
```

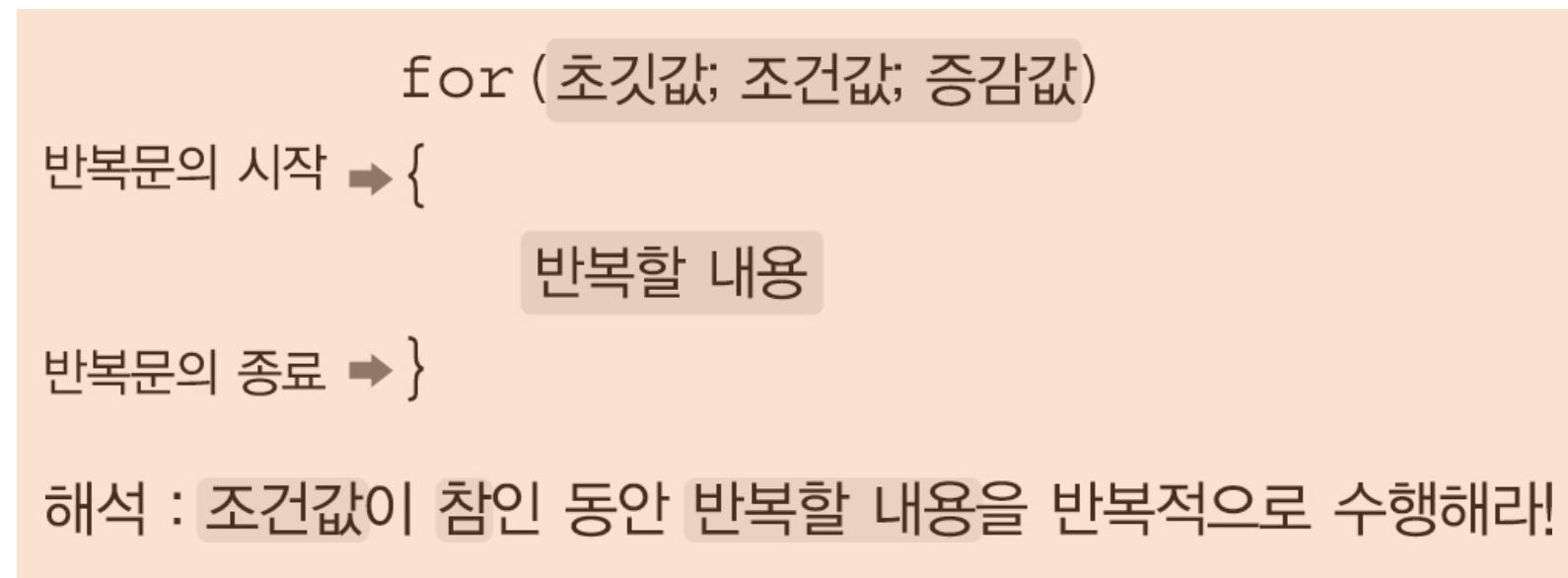
반복문을 만드는 방법2 – for문

- **for문을 위해 배울 내용**
 - ① for문의 기본 문법
 - ② for 무한 루프(무한 반복문)
 - ③ 중첩 for문
 - ④ for문의 여러 가지 변형

반복문을 만드는 방법2 – for문

① for 문의 기본 문법

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



반복문을 만드는 방법2 – for문

```
/* 7-6.c */
#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;
}
```

반복문을 만드는 방법2 – for문

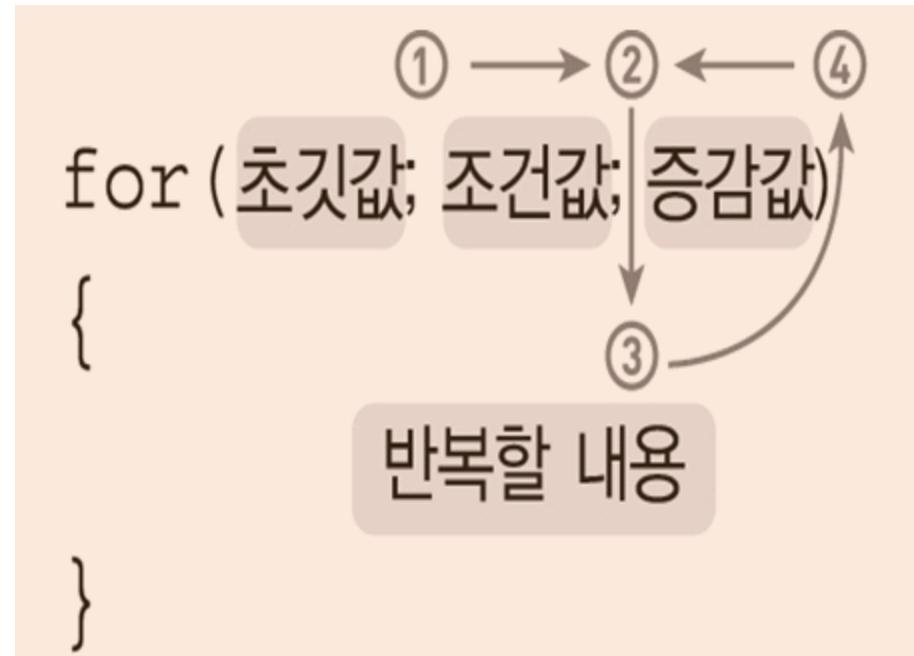
- while 문과 for문의 비교
 - 7-1.c 와 7-6.c를 비교

```
/* 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++)  
{  
    초기값  
    조건값  
    증감값  
}  
  
printf("반복 내용: %d \n", num);  
}
```

반복문을 만드는 방법2 – for 문

- **for 문의 실행 순서**



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

반복문을 만드는 방법2 – for문

```
/* 7-7.c */
#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;
}
```

```
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-----
```

반복문을 만드는 방법2 – for문

```
/* 7-8.c */
#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;
}
```

숫자를 입력하세요 : 5
5 * 1 = 5입니다.
5 * 3 = 15입니다.
5 * 5 = 25입니다.
5 * 7 = 35입니다.
5 * 9 = 45입니다.

반복문을 만드는 방법2 – for문

② for 무한 루프(무한 반복문)

- 종료되지 않고 무한히 실행되는 반복문

```
/* 7-9.c */
#include <stdio.h>
int main(void)
{
    int i;
    for(i=0; 1; i++)
    {
        printf("반복 횟수 : %d\n", i);
    }
    return 0;
}
```

반복문을 만드는 방법2 – for문

③ 중첩 for문

- for문 내부에 또 다른 for문이 있는 것

```
/* 7-10.c */
#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;
}
```

```
2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
2 * 4 = 8
2 * 5 = 10
2 * 6 = 12
2 * 7 = 14
2 * 8 = 16
2 * 9 = 18
-----
3 * 1 = 3
3 * 2 = 6
3 * 3 = 9
3 * 4 = 12
3 * 5 = 15
3 * 6 = 18
3 * 7 = 21
3 * 8 = 24
3 * 9 = 27
-----
```

```
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
-----
```

반복문을 만드는 방법2 – for문

```
/* 7-11.c */
#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;
}
```

반복문을 만드는 방법2 – for문

④ for문의 여러 가지 변형

```
/* 7-13.c */
#include <stdio.h>
int main(void)
{
    int i=1, factorial=1;
    for( ; i<=10; i++)          // 초기값이 없는 경우
    {
        factorial = factorial * i;
    }
    printf("1부터 10까지의 곱 : %d\n", factorial);

    return 0;
}
```

반복문을 만드는 방법2 – for문

```
/* 7-14.c */
#include <stdio.h>
int main(void)
{
    int i, factorial=1;
    for(i=1; ; i++) // 조건값이 없는 경우
    {
        factorial=factorial*i;

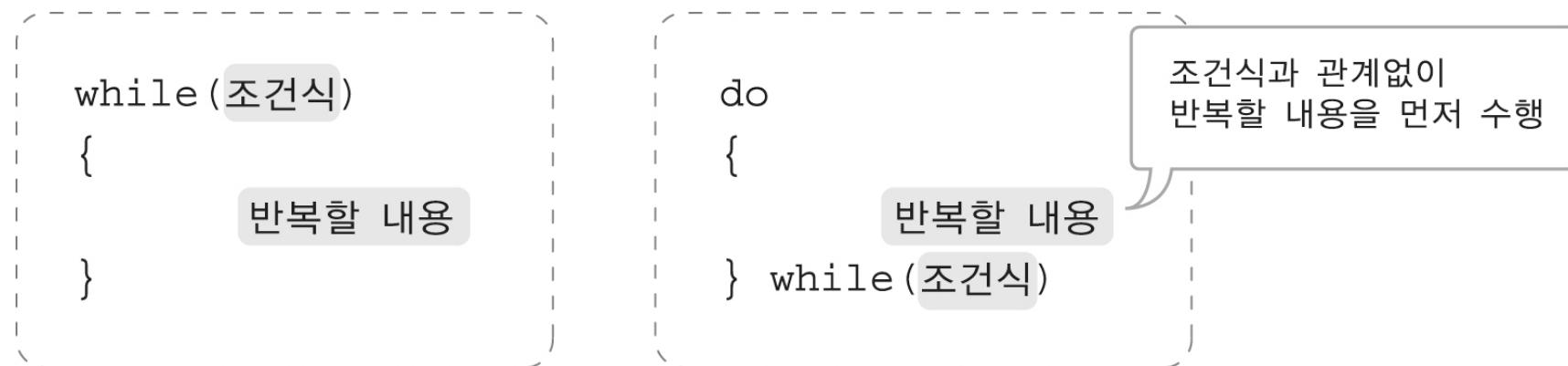
        if (i>10)
            break;      // 무한 반복문을 탈출하기 위한 break 문
    }
    printf("1부터 10까지의 곱 : %d\n", factorial);
    return 0;
}
```

반복문을 만드는 방법2 – for문

```
/* 7-15.c */
#include <stdio.h>
int main(void)
{
    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;
}
```

반복문을 만드는 방법 3 – do~while 문

- **do ~ while 문**
 - ‘while 문과 for 문에 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.’



반복문을 만드는 방법 3 – do~while 문

```
/* 7-16.c */
#include <stdio.h>
int main(void)
{
    int num=10;

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

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

10

while 문을 종료합니다.

Summary

- 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

1. 사용자 입력에 따른 누적 합계

```
#include <stdio.h>
int main(void) {
    int num, sum = 0;
    printf("정수를 입력하세요 (음수를 입력하면 종료): ");
    scanf("%d", &num);

    while (num >= 0) {
        sum += num;
        printf("현재까지 합계: %d\n", sum);
        printf("정수를 입력하세요 (음수를 입력하면 종료): ");
        scanf("%d", &num);
    }

    printf("최종 합계: %d\n", sum);
    return 0;
}
```

2. for 문을 이용한 별 출력

```
#include <stdio.h>
int main(void) {
    int i, j, height;
    printf("삼각형 높이를 입력하세요: ");
    scanf("%d", &height);

    for (i = 1; i <= height; i++) {
        for (j = 1; j <= i; j++) {
            printf("*");
        }
        printf("\n");
    }
    return 0;
}
```

3. do~while을 이용한 비밀번호 확인

```
#include <stdio.h>
#define PASSWORD 1234

int main(void) {
    int input;
    do {
        printf("비밀번호를 입력하세요: ");
        scanf("%d", &input);
    } while (input != PASSWORD);

    printf("비밀번호가 맞습니다. 환영합니다!\n");
    return 0;
}
```

4. 구구단 역순 출력

```
#include <stdio.h>
int main(void) {
    int dan;
    printf("출력할 구구단 단 수를 입력하세요: ");
    scanf("%d", &dan);

    for (int i = 9; i >= 1; i--) {
        printf("%d * %d = %d\n", dan, i, dan * i);
    }

    return 0;
}
```

5. 중첩 for 문 - 별 피라미드 만들기

```
#include <stdio.h>
int main(void) {
    int i, j, n;
    printf("피라미드의 높이를 입력하세요: ");
    scanf("%d", &n);

    for (i = 1; i <= n; i++) {
        for (j = 1; j <= n - i; j++) {
            printf(" ");
        }
        for (j = 1; j <= (2 * i - 1); j++) {
            printf("*");
        }
        printf("\n");
    }

    return 0;
}
```

1. 짝수의 합 구하기 (while 문 연습)

1. 문제 설명

- 정수 N을 입력받아, 1부터 N까지의 수 중에서 짝수의 합을 구하는 프로그램을 while 문을 이용하여 작성하세요

2. 조건

- 입력은 양의 정수 하나입니다.
- while문만 사용하세요.

정수를 입력하세요: 10
1부터 10까지 짝수의 합은 30입니다.

2. 별로 이루어진 사각형 출력 (for 문 중첩 사용)

1. 문제 설명

- 두 개의 정수 행, 열을 입력받아, 해당 크기의 별(*)로 이루어진 사각형을 출력하는 프로그램을 작성하세요.

2. 요구사항

- for문을 중첩해서 사용하세요.
- 행, 열은 1 이상 20 이하의 자연수로 가정합니다.

행과 열을 입력하세요: 3 5

3. 1부터 N까지 숫자 중 3의 배수는 건너뛰고 출력 (continue 활용)

1. 문제 설명

- 사용자로부터 양의 정수 N을 입력받아 1부터 N까지 수 중에서 3의 배수는 출력하지 않고, 나머지 수만 출력하는 프로그램을 작성하세요. for문과 continue문을 함께 사용해 보세요.

2. 요구사항

- continue 문을 반드시 사용하세요.
- 출력은 한 줄에 공백으로 구분하여 출력합니다.

정수를 입력하세요: 10

1 2 4 5 7 8 10

4. 1부터 N까지의 숫자 중 "숫자에 3, 6, 9가 포함된 수"만 출력 (일명 369 게임)

1. 문제 설명

- 사용자로부터 정수 N을 입력받아, 1부터 N까지의 수 중에서 '숫자에 3, 6, 9가 하나라도 포함된 수'만 출력하는 프로그램을 작성하세요. 단, 출력 형식은 각 수를 공백으로 구분해서 한 줄에 출력합니다.
- 예를 들어, 13은 숫자에 3이 포함되므로 출력 대상이며, 36은 3, 6이 모두 포함되어도 출력 대상입니다. 반면, 14, 25, 80 등은 제외됩니다.

2. 요구사항

- 숫자 하나하나의 자릿수를 검사해야 합니다.
- while문이나 for문 모두 사용 가능하며, 자유롭게 선택 가능.
- if, continue 등의 조건문 자유 사용.

정수를 입력하세요: 40

3 6 9 13 23 30 31 32 33 34 35 36 37 38 39