

```

// for 반복문 예제1
#include<stdio.h>

int main()
{
    int i;
    for (i = 0; i < 10; i++)
    {
        printf("%d ", i);
    }
    printf("\n");
    printf("i의 값은 : ");
    printf("%d\n", i);

    return 0;
}
// 결과
// for 반복문 예제1 : 0 1 2 3 4 5 6 7 8 9
// i의 값은 : 10

```

```

// for 반복문 예제2
#include<stdio.h>

int main()
{
    for (int i = 0; i < 10; i++)
    {
        printf("%d ", i);
    }
    printf("\n");
    printf("i의 값은 : ");
    printf("%d\n", i); // 오류남

    return 0;
}
// 결과없음. 오류남

```

```
// 1~10까지 누적 합 구하기
#include<stdio.h>

int main()
{
    int count, hap = 0;

    for (count = 1; count <= 10; count++)
    {
        hap += count;
        printf("%2d 까지의 누적합\t hap = %2d\n", count, hap);
    }

    printf("-----\n");
    printf("1~10까지의 누적합 : %d\n", hap);

    return 0;
}

// 결과
//  1 까지의 누적합      hap = 1
//  2 까지의 누적합      hap = 3
//  3 까지의 누적합      hap = 6
//  4 까지의 누적합      hap = 10
//  5 까지의 누적합      hap = 15
//  6 까지의 누적합      hap = 21
//  7 까지의 누적합      hap = 28
//  8 까지의 누적합      hap = 36
//  9 까지의 누적합      hap = 45
// 10 까지의 누적합      hap = 55
// -----
// 1~10까지의 누적합 : 55
```

```
// 1~10까지 누적되는 홀수의 합 구하기
#include<stdio.h>

int main()
{
    int count, hap = 0;

    for (count = 1; count <= 10; count = count + 2)
    {
        hap += count;
        printf("%2d 까지의 누적합\t hap = %2d\n", count, hap);
    }
}
```

```

printf("-----\n");
printf("1~10까지의 누적된 홀수의 합 : %d\n", hap);

return 0;
}

// 결과
// 1까지의 누적합      hap = 1
// 3까지의 누적합      hap = 4
// 5까지의 누적합      hap = 9
// 7까지의 누적합      hap = 16
// 9까지의 누적합      hap = 25
// -----
// 1~10까지의 누적된 홀수의 합 : 25
// 1~10까지 누적되는 짝수의 합 구하기
#include<stdio.h>

int main()
{
    int count, hap = 0;

    for (count = 2; count <= 10; count = count + 2)
    {
        hap += count;
        printf("%2d까지의 누적합\t hap = %2d\n", count, hap);
    }

    printf("-----\n");
    printf("1~10까지의 누적된 짝수의 합 : %d\n", hap);

    return 0;
}

// 결과
// 2까지의 누적합      hap = 2
// 4까지의 누적합      hap = 6
// 6까지의 누적합      hap = 12
// 8까지의 누적합      hap = 20
// 10까지의 누적합     hap = 30
// -----
// 1~10까지의 누적된 짝수의 합 : 30

```

```
// 구구단 출력 1
#include<stdio.h>

int main()
{
    int dan, count, gob;

    dan = 4; // 4단 출력

    printf("%d단 출력\n-----\n", dan);

    for (count = 1; count <= 9; count++)
    {
        gob = dan * count;
        printf("%d x %d = %2d\n", dan, count, gob);
    }

    return 0;
}

// 결과
// 4단 출력
// -----
// 4 x 1 = 4
// 4 x 2 = 8
// 4 x 3 = 12
// 4 x 4 = 16
// 4 x 5 = 20
// 4 x 6 = 24
// 4 x 7 = 28
// 4 x 8 = 32
// 4 x 9 = 36
```

```
// 구구단 출력 2(이중 반복문 사용하여 세로로 출력)
#include<stdio.h>

int main()
{
    for (int i = 2; i <= 9; i++)
    {
        for (int j = 1; j <= 9; j++)
        {
            printf("%d x %d = %2d\n", i, j, (i * j));
        }
        printf("\n");
    }
}
```

```
}
return 0;
}

// 결과
// 2 x 1 = 2
// 2 x 2 = 4
// 2 x 3 = 6
// 2 x 4 = 8
//
//
//
// (중간 색락)
//
//
//
// 9 x 1 = 9
// 9 x 2 = 18
// 9 x 3 = 27
// 9 x 4 = 36
// 9 x 5 = 45
// 9 x 6 = 54
// 9 x 7 = 63
// 9 x 8 = 72
// 9 x 9 = 81

// 구구단 출력 3(이중 반복문 사용하여 가로로 출력)
#include<stdio.h>

int main()
{
    for (int i = 2; i <= 9; i++)
    {
        for (int j = 1; j <= 9; j++)
        {
            printf("%d x %d = %2d\t", i, j, (i * j));
        }
        printf("\n");
    }
    return 0;
}

// 결과
// 2 x 1 = 2    2 x 2 = 4    2 x 3 = 6    2 x 4 = 8    2 x 5 = 10    2 x 6 = 12    2 x 7 = 14    2 x 8 = 16    2 x 9 = 18
// 3 x 1 = 3    3 x 2 = 6    3 x 3 = 9    3 x 4 = 12    3 x 5 = 15    3 x 6 = 18    3 x 7 = 21    3 x 8 = 24    3 x 9 = 27
// 4 x 1 = 4    4 x 2 = 8    4 x 3 = 12    4 x 4 = 16    4 x 5 = 20    4 x 6 = 24    4 x 7 = 28    4 x 8 = 32    4 x 9 = 36
// 5 x 1 = 5    5 x 2 = 10    5 x 3 = 15    5 x 4 = 20    5 x 5 = 25    5 x 6 = 30    5 x 7 = 35    5 x 8 = 40    5 x 9 = 45
// 6 x 1 = 6    6 x 2 = 12    6 x 3 = 18    6 x 4 = 24    6 x 5 = 30    6 x 6 = 36    6 x 7 = 42    6 x 8 = 48    6 x 9 = 54
// 7 x 1 = 7    7 x 2 = 14    7 x 3 = 21    7 x 4 = 28    7 x 5 = 35    7 x 6 = 42    7 x 7 = 49    7 x 8 = 56    7 x 9 = 63
```

```
// 8 x 1 = 8      8 x 2 = 16      8 x 3 = 24      8 x 4 = 32      8 x 5 = 40      8 x 6 = 48      8 x 7 = 56      8 x 8 = 64      8 x 9 = 72
// 9 x 1 = 9      9 x 2 = 18      9 x 3 = 27      9 x 4 = 36      9 x 5 = 45      9 x 6 = 54      9 x 7 = 63      9 x 8 = 72      9 x 9 = 81
```

```
// 구구단 출력 4(for문을 while문으로 변환하기)
```

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

```
int main()
```

```
{
```

```
    int i, j;
```

```
    i = 2;
```

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

```
        j = 1;
```

```
        while (j <= 9) {
```

```
            printf("%d x %d = %2d\n", i, j, (i * j));
```

```
            j++;
```

```
        }
```

```
        i++;
```

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

```
    }
```

```
    return 0;
```

```
}
```

```
// 결과
```

```
// 2 x 1 = 2
```

```
// 2 x 2 = 4
```

```
// 2 x 3 = 6
```

```
// 2 x 4 = 8
```

```
//      .
```

```
//      .
```

```
//      .
```

```
// (중간 색락)
```

```
//      .
```

```
//      .
```

```
//      .
```

```
// 9 x 1 = 9
```

```
// 9 x 2 = 18
```

```
// 9 x 3 = 27
```

```
// 9 x 4 = 36
```

```
// 9 x 5 = 45
```

```
// 9 x 6 = 54
```

```
// 9 x 7 = 63
```

```
// 9 x 8 = 72
```

```
// 9 x 9 = 81
```

```

// 직사각형 만들기
#include<stdio.h>

int main()
{
    for (int i = 0; i < 5; i++) {
        for (int j = 0; j < 5; j++) {
            printf("*");
        }
        printf("\n");
    }
    return 0;
}

// 결과
// *****
// *****
// *****
// *****
// *****

```

```

// 직삼각형 만들기
#include<stdio.h>

int main()
{
    for (int i = 0; i < 5; i++) {
        for (int j = 0; j <= i; j++) {
            printf("*");
        }
        printf("\n");
    }
    return 0;
}

// 결과
// *
// **
// ***
// ****
// *****

```

```
// 역직삼각형 만들기
```

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

```
int main()
```

```
{
```

```
    for (int i = 0; i < 5; i++) {
```

```
        for (int j = 0; j < 5 - i; j++) {
```

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

```
        }
```

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

```
    }
```

```
    return 0;
```

```
}
```

```
// 결과
```

```
// *****
```

```
// ****
```

```
// ***
```

```
// **
```

```
// *
```

```
// 피라미드 만들기
```

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

```
int main()
```

```
{
```

```
    for (int i = 0; i < 5; i++)
```

```
    {
```

```
        for (int j = i; j < 4; j++)
```

```
        {
```

```
            printf(" "); //피라미드의 공백을 출력
```

```
        }
```

```
        for (int k = 0; k < i * 2 + 1; k++)
```

```
        {
```

```
            printf("*"); //피라미드의 별을 출력
```

```
        }
```

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

```
    }
```

```
    return 0;
```

```
}
```

```
// 결과
```

```
//      *
```

```
//     ***
```

```
//    *****
```

```
//   *****
```

```
//  *****
```



```

// do-while문
#include<stdio.h>

int main()
{
    int i = 1;

    do                                // 처음 한 번은 아래 코드가 실행됨
    {
        printf("Hello, world! %d\n", i); // Hello, world!와 i의 값을 함께 출력
        i++;                             // i를 1씩 증가시킴
    } while (i < 5);                     // i가 5보다 작을 때 반복. 1부터 4까지 증가하
    면서 5번 반복

    return 0;
}
// 결과
// Hello, world! 1
// Hello, world! 2
// Hello, world! 3
// Hello, world! 4

```

```

// while문과 do-while문 비교하기
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>

int main()
{
    int x = 10;
    int y = 1;
    int sum = 0;

    while (x <= y)
    {
        sum = sum + x;
        x++;
    }
    printf("while문을 사용한 sum : %d\n", sum);

    // -----
    x = 10;
    y = 1;
    sum = 0;

```

```
do
{
    sum = sum + x;
    x++;
} while (x <= y);
printf("do-while문을 사용한 sum : %d\n", sum);

return 0;
}
// 결과
// while문을 사용한 sum : 0
// do - while문을 사용한 sum : 10
```

```
// break문을 사용해서 하나의 반복문 빠져나오기
#include<stdio.h>

int main()
{
    int m, n;

    for (m = 5; m < 7; m++)                // m은 5 ~ 6 까지 2회 반복
    {
        for (n = 0; n < 3; n++)            // n은 0 ~ 2 까지 3회 반복
        {
            if (m == 5 && n == 1) break;    // m이 5이고 n이 1일 때 break를 수행한다
            printf("m(%d) - n(%d)\n", m, n);
        }
    }
}
// 결과
// m(5) - n(0)
// m(6) - n(0)
// m(6) - n(1)
// m(6) - n(2)
```

```
// break문을 사용해서 반복문 모두 빠져나오기
#include<stdio.h>

int main()
{
    int m, n;

    for (m = 5; m < 7; m++)                // m은 5 ~ 6 까지 2회 반복
    {
        for (n = 0; n < 3; n++) {           // n은 0 ~ 2 까지 3회 반복
            if (m == 5 && n == 1) break;    // m이 5이고 n이 1일 때 break를 수행한다
            printf("m(%d) - n(%d)\n", m, n);
        }
        if (m == 5 && n == 1) break;
    }
}
// 결과
// m(5) - n(0)
```

```
// continue문을 사용해서 반복문의 실행 흐름 변경하기
#include<stdio.h>

int main()
{
    int m;

    for (m = 1; m < 8; m++) {        /* m은 1 ~ 7 까지 8회 반복 */
        if (m == 6) continue;      /* m이 6일 때 continue를 수행한다 */
        printf("m(%d)\n", m);
    }
}

// 결과
// m(1)
// m(2)
// m(3)
// m(4)
// m(5)
// m(7)
```