

## <과제 1>

#include <stdio.h>

int main()

{

double num1, num2, num3, num4, num5;

double avg, min = 0, max = 0;

printf("\n1 번째 숫자를 입력하시오. "); // 숫자를 입력받는다.

scanf\_s("%lf", &num1);

printf("\n2 번째 숫자를 입력하시오. ");

scanf\_s("%lf", &num2);

printf("\n3 번째 숫자를 입력하시오. ");

scanf\_s("%lf", &num3);

printf("\n4 번째 숫자를 입력하시오. ");

scanf\_s("%lf", &num4);

printf("\n5 번째 숫자를 입력하시오. ");

scanf\_s("%lf", &num5);

avg = (num1 + num2 + num3 + num4 + num5) / 5;

printf("\n평균은 % lf입니다.", avg);

if (num1 < num2) // num1과 num2를 비교해 최솟값 min과 최댓값 max를 설정한다.

{

max = num2;

min = num1;

}

else

{

max = num1;

min = num2;

}

if (max < num3) // 상기 과정으로 설정한 min/max값을 입력받은 값과 비교한다.

{

max = num3;

}

else if (num3 < min)

{

min = num3;

}

if (max < num4)

{

```

        max = num4;
    }
    else if (num4 < min)
    {
        min = num4;
    }
    if (max < num5)
    {
        max = num5;
    }
    else if (num5 < min)
    {
        min = num5;
    }
    printf("\n최댓값은 %lf입니다.", max); // 비교 결과를 출력한다.
    printf("\n최솟값은 %lf입니다.", min);

    return 0;
}

```

```

1 번째 숫자를 입력하시오.  1
2 번째 숫자를 입력하시오.  1.5
3 번째 숫자를 입력하시오.  2
4 번째 숫자를 입력하시오.  2.5
5 번째 숫자를 입력하시오.  3

평균은  2.000000입니다.
최댓값은  3.000000입니다.
최솟값은  1.000000입니다.

```

## <과제 2>

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

```
int main()
```

```
{
```

```
    int fibo1 = 0, fibo2 = 1, fibo3, i, num;
```

```
    printf("숫자를 입력하시오 : ");
```

```
    scanf_s("%d", &num);
```

```
    if (num == 1) // 첫번째 피보나치 수는 0이므로 for문 밖에서 예외처리한다.
```

```
    {
```

```
        printf("1번째 피보나치 수는 0입니다.\n");
```

```
    }
```

```
    else if (num == 2) // 두번째 피보나치 수는 1이므로 for문 밖에서 예외처리한다.
```

```
    {
```

```
        printf("2번째 피보나치 수는 1입니다.\n");
```

```
    }
```

```
    if (num > 3) // 3이상의 수 대상으로 피보나치 수 계산
```

```
    {
```

```
        for (i = 0; i < (num - 2); i++)
```

```
        {
```

```
            fibo3 = fibo1 + fibo2; // 앞의 두 수의 합이 피보나치 수가 된다.
```

```
            fibo1 = fibo2; // 두번째 자리의 있던 수를 첫째 자리로
```

```
            fibo2 = fibo3; // 세번째 자리의 수를 둘째 자리로
```

```
        }
```

```
    }
```

```
    printf("%d번째 피보나치 수는 %d입니다.\n", num, fibo3); // 한 번만 출력하기 위해  
    for문 밖에서 출력한다.
```

```
    return 0;
```

```
}
```

```
숫자를 입력하시오 : 6
6번째 피보나치 수는 5입니다.
```

### <과제 3>

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

```
int main()
```

```
{
```

```
    int year;
```

```
    printf("년도를 입력하세요 : ");
```

```
    scanf_s("%d", &year);
```

```
    if (year % 4 == 0) // 연도가 4의 배수인지 확인
```

```
    {
```

```
        if (year % 100 == 0) // 4의 배수이면서 100의 배수인지 확인
```

```
        {
```

```
            if (year % 400 == 0) // 4와 100 배수이며, 400의 배수인지 확인
```

```
            {
```

```
                printf("윤년");
```

```
            }
```

```
            else // 4와 100의 배수이나 400의 배수가 아닐 경우
```

```
            {
```

```
                printf("윤년이 아님");
```

```
            }
```

```
        }
```

```
        else // 4의 배수이며 100의 배수가 아닐 경우
```

```
        {
```

```
            printf("윤년");
```

```
        }
```

```
    }
```

```
    else // 4의 배수가 아닐 경우
```

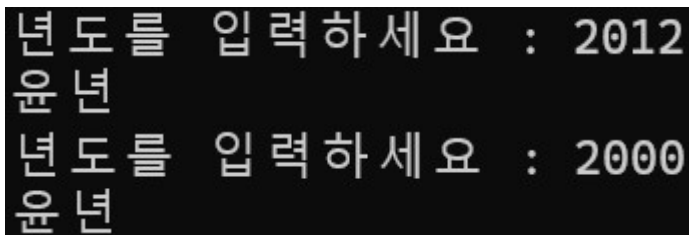
```
    {
```

```
        printf("윤년이 아님");
```

```
    }
```

```
    return 0;
```

```
}
```



The image shows a terminal window with a black background and white text. It displays the program's output for two different inputs. The first input is 2012, and the output is "윤년" (Leap Year). The second input is 2000, and the output is "윤년" (Leap Year). The text is arranged in two lines, with the input and output for each case shown together.

```
년도를 입력하세요 : 2012
윤년
년도를 입력하세요 : 2000
윤년
```

#### <과제 4>

```
#define _CRT_SECURE_NO_WARNINGS
```

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

```
int main(void)
```

```
{
```

```
    double num1, num2, p, sum = 0, min = 0, div = 0, mul = 0;
```

```
    double power = 1;
```

```
    char operator;
```

```
    printf("input Arithmetic Operation\n");
```

```
    printf("ex)  3.4 * 8.5\n      2.9 - 5.4\n      3.9 * 8.0\n      3.9^8\n");
```

```
    printf("input : ");
```

```
    scanf("%lf %c %lf", &num1, &operator, &num2); // 연산자는 c를 통해 입력
```

```
    if (operator == '+') // 입력받은 연산자가 + 일 경우
```

```
    {
```

```
        sum = num1 + num2;
```

```
        printf("%.2lf + %.2lf = %.2lf", num1, num2, sum);
```

```
    }
```

```
    else if (operator == '-') // 입력받은 연산자가 - 일 경우
```

```
    {
```

```
        min = num1 - num2;
```

```
        printf("%.2lf - %.2lf = %.2lf", num1, num2, min);
```

```
    }
```

```
    else if (operator == '/') // 입력받은 연산자가 / 일 경우
```

```
    {
```

```
        if (num2 == 0) // 분모가 0일 경우 재입력
```

```
        {
```

```
            printf("wrong input. try again");
```

```
        }
```

```
        else // 분모가 0이 아닐 경우 정상적으로 계산
```

```
        {
```

```
            div = num1 / num2;
```

```
            printf("%.2lf / %.2lf = %.2lf", num1, num2, div);
```

```
        }
```

```
    }
```

```
    else if (operator == '*') // 입력받은 연산자가 * 일 경우
```

```
    {
```

```
        mul = num1 * num2;
```

```
        printf("%.2lf * %.2lf = %.2lf", num1, num2, mul);
```

```

    }
    else if (operator == '^') // 입력받은 연산자가 ^ 일 경우
    {
        for (p = 1; p <= num2; p++) // 변수 p가 num2 크기와 같아질 때까지 증가
        {
            power = power * num1; // 조건 충족시까지 반복
        }
        printf("%.2lf ^ %.2lf = %.2lf", num1, num2, power);
    }
    else
    {
        printf("wrong input. try again"); // 연산자가 위에 속하지 않을 시 재입력
    }

    return 0;
}

```

```

input Arithmetic Operation
ex)  3.4 * 8.5
      2.9 - 5.4
      3.9 * 8.0
      3.9^8
input : 3 + 5
3.00 + 5.00 = 8.00

```

```

input Arithmetic Operation
ex)  3.4 * 8.5
      2.9 - 5.4
      3.9 * 8.0
      3.9^8
input : 5.57 ^ 4
5.57 ^ 4.00 = 962.54

```

```

input Arithmetic Operation
ex)  3.4 * 8.5
      2.9 - 5.4
      3.9 * 8.0
      3.9^8
input : 8.6 - 3
8.60 - 3.00 = 5.60

```

```
input Arithmetic Operation
ex)  3.4 * 8.5
      2.9 - 5.4
      3.9 * 8.0
      3.9^8
input : 5.0 * 5.78
5.00 * 5.78 = 28.90
```

```
input Arithmetic Operation
ex)  3.4 * 8.5
      2.9 - 5.4
      3.9 * 8.0
      3.9^8
input : 5 / 2.5
5.00 / 2.50 = 2.00
```

### <과제 5>

```
#define _CRT_SECURE_NO_WARNINGS
```

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

```
int main()
```

```
{
```

```
    int num, i, j;
```

```
    printf("값을 입력하세요. ");
```

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

```
    for (i = 0; i < num; i++) // i=0에서부터 입력값보다 작을동안 반복
    {
```

```
        for (j = 1; j <= i; j++) // 위 부분 그리기, 좌상단 리본
```

```
        {
```

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

```
        }
```

```
        for (j = ((num * 2) - 1); j > ((i * 2) - 1); j--) // *을 채우지 않는 공간을
```

공백으로 채움

```
        {
```

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

```
        }
```

```
        for (j = 1; j <= i; j++) // 우상단 리본
```

```
        {
```

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

```
        }
```

```
        printf("\n"); // 줄 변경
```

```
    }
```

부분 출력 포함

```
    for (i = 1; i <= num; i++) // i=1에서부터 입력값과 같아질 때까지 반복, 가운데 허리
```

```
    {
```

```
        for (j = num; j >= i; j--) // 아랫 부분 그리기, 좌하단 리본
```

```
        {
```

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

```
        }
```

```
        for (j = 1; j <(( 2 * i) - 1); j++) // *을 채우지 않는 공간을 공백으로 채움
```

```
        {
```

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

```
        }
```

```
        for (j = num; j >= i; j--) // 우하단 리본
```

```
        {
```

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

```
        }
```



[illegible]

## <과제 6>

```
#define _CRT_SECURE_NO_WARNINGS
```

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

```
int main()
```

```
{
```

```
    int num, i, j, k=0;
```

```
    printf("값을 입력하세요. ");
```

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

```
    for (i = 1; i < num; i++) // 삼각형 높이 결정
```

```
    {
```

```
        for (j = i; j < num; j++)
```

```
        {
```

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

```
        }
```

```
        while (k != ((2 * i) - 1)) // k의 값이 (2*i)-1이 아닌 동안 아래 행위 반복
```

```
        {
```

```
            if (k == 0 || k == ((2 * i) - 2)) // k값이 0이거나 k값이 (2*i)-2일 경우
```

```
                printf("*"); // *을 쌓음
```

```
            else
```

```
                printf(" "); // 그렇지 않으면 공백으로 남김
```

```
            k++; // 일련의 행동 마무리 후 k값 증가
```

```
        ;
```

```
        }
```

```
        k = 0; // k값 초기화
```

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

```
    }
```

```
    for (i = 0; i < ((2 * num) - 1); i++) // 삼각형의 밑변 출력
```

```
    {
```

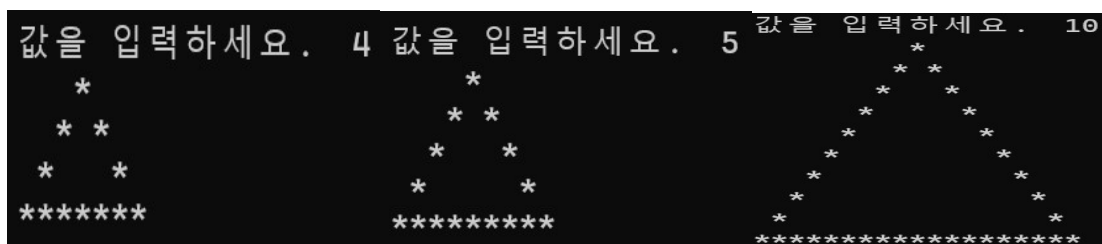
```
        printf("*");
```

```
    }
```

```
    printf("\n"); // 줄 변경
```

```
    return 0;
```

```
}
```



### <과제 7>

```
#define _CRT_SECURE_NO_WARNINGS
```

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

```
int main()
```

```
{
```

```
    int num1 = 0, num2 = 0;
```

```
    int n1, t, n2, pR, cR1, cR2;
```

```
    double num1Fact = 1, num2Fact = 1, totalFact = 1, Fact = 1, Fact2 = 1; //순서  
    대로 num1!, num2!, (num1-num2)!, (num1+num2-1)!, (num1-1)!
```

```
    double permuRepeat = 1, combination = 1, comRepeat = 1;
```

```
    printf("두 숫자를 입력하시오. ");
```

```
    scanf("%d %d", &num1, &num2);
```

```
    for (n1 = num1; n1 >= 1; n1--) // num1의 팩토리얼의 값
```

```
    {
```

```
        num1Fact = num1Fact*n1;
```

```
    }
```

```
    for (n2 = num2; n2 >= 1; n2--) // num2의 팩토리얼의 값
```

```
    {
```

```
        num2Fact = num2Fact * n2;
```

```
    }
```

```
    for (t = (num1 - num2); t >= 1; t--) // (num1-num2)의 팩토리얼의 값
```

```
    {
```

```
        totalFact = totalFact * t;
```

```
    }
```

```
    printf("순열 계산값 : %.2lf \n", num1Fact/totalFact); // 순열(permutation)의 값
```

```
    for (pR = 1; pR <= num2; pR++) // 변수 p가 num2 크기와 같아질 때까지 증가
```

```
    {
```

```
        permuRepeat = permuRepeat * num1; // 조건 충족시까지 반복
```

```
    }
```

```
    printf("중복 순열 계산값 : %.2lf \n", permuRepeat); // 중복순열의 값
```

```
    combination = num1Fact / (totalFact * num2Fact);
```

```
    printf("조합 계산값 : %.2lf \n", combination);
```

```
    for (cR1 = (num1+num2-1); cR1 >= 1; cR1--) // (num1+num2-1)의 팩토리얼의 값
```

```
    {
```

```
        Fact = Fact * cR1;
```

```
    }
```

```

for (cR2 = (num1 - 1); cR2 >= 1; cR2--) // (num1+1)의 팩토리얼의 값
{
    Fact2 = Fact2 * cR2;
}

comRepeat = Fact / (Fact2 * num2Fact);

printf("중복 조합 계산값 : %lf", comRepeat);

return 0;
}

```

```

두 숫자를 입력하시오 . 5 2
순열 계산값 : 20.00
중복 순열 계산값 : 25.00
조합 계산값 : 10.00
중복 조합 계산값 : 15.000000

```