3일차 내용 정리

[6132] CO: [SCSA 4-27]

팩토리얼(!)을 구하는 함수

Language: C

```c
#include <stdio.h>

unsigned long long int Factorial(int num)
{
    int f = 1;
    for(int i = 1; i <= num; i++){
        f *= i;
    }
    return f;
}

void main(void)
{
    int value;
    scanf("%d", &value);
    printf("%llu\n", Factorial(value));
}
```

자료형에 관해 묻고 있다
long long int형의 범위
printf에서 llu(long long unsigned) 사용법 등
해당 범위내에서 몇 팩토리얼까지 출력할 수 있는지 확인해보자

3일차 내용 정리

[6133] C1: [SCSA 4-28]

두 정수의 중간 값을 구하는 함수

Language: C

```c
#include <stdio.h>

int Find_Median_Value(a, b)
{
    return (a + b) / 2;
}

void main()
{
    int a, b;
    scanf("%d %d", &a, &b);

    printf("%d", Find_Median_Value(a, b));
    return 0;
}
```

1번 풀이)
중간 값을 구하기 위해
두 수의 합을 2로 나누면 된다는
수학적 지식을 기억해두자
물론, 출제자가 의도한 Two Pointer방식에 관해선 다음장에서 다뤄보자

3일차 내용 정리

[6133] C1: [SCSA 4-28]
두 정수의 중간 값을 구하는 함수

Language: C

```c
#include <stdio.h>

int Find_Median_Value(min, max)
{
    while(++min < --max);
    return min;
}

void main()
{
    int min, max, temp;
    scanf("%d %d", &min, &max);

    if(min > max){
        temp = min;
        min = max;
        max = temp;
    }

    printf("%d", Find_Median_Value(min, max));
    return 0;
}
```

2번 풀이)
중간 값을 구하기 위해 Two Pointer 알고리즘을 사용해보자
증가시킬 min과 감소시킬 max를 구분하기 위해
입력 당시 작은 쪽의 정수를 min에 저장하자
예를 들어, 1과 5 사이 중간값이라면
서로를 향해 한 칸씩 다가오다가 두 값이 같아지는 지점이 중간값이다
while문 안에 조건식을 (++min != --max)로 해도 되지만
(문제에서, 중간 값이 2개가 되지 않는다고 했으므로)
보다 안전한 코드를 위해 min이 max보다 크면 반복문이 종료되게 하자

3일차 내용 정리

[2665] C2: [SCSA 4-29-1]

반복문을 활용한 별 자판기 - 사각별

Language: C

```c
#include <stdio.h>

void Draw_Stars()
{
    for(int j = 1; j <= 3; j++){
        for(int i = 1; i <= 5; i++){
            printf("*");
        }
        printf("\n");
    }
}

void main()
{
    Draw_Stars();
    return 0;
}
```

이중 반복문을 구성할 수 있는지에 관한 문제이다

안쪽, 변수가 i인 반복문을 구성 후

바깥, 변수가 j인 반복문을 구성하자

여러 실행문장을 간결하게 표현한 문법이 반복문임을 기억하자

3일차 내용 정리

[2666] C3: [SCSA 4-29-2]

반복문을 활용한 별 자판기 - 역삼각별

Language: C

```c
#include <stdio.h>

void Draw_Stars()
{
    // for(int j = 1; j <= 5; j++){
    //     for(int i = 1; i <= 6 - j; i++)
    //         printf("*");
    //     }
    //     printf("\n");
    // }
    for(int j = 5; j > 0; j--){
        for(int i = 1; i <= j; i++){
            printf("*");
        }
        printf("\n");
    }
}

void main()
{
    Draw_Stars();
    return 0;
}
```

이전 문제와 마찬가지로
안쪽 변수 i가 있는 반복문을 구성하고
바깥쪽 변수 j가 있는 반복문을 구성하자
j가 5부터 시작해 1까지 감소하는 Logic을
수학 함수를 활용하여 6 - j 등으로 쓸 수 있다
굉장히 많이 활용되는 부분이니 이해하도록 노력해보자

3일차 내용 정리

[1361] C4: [SCSA 기본C 워크샵 문제]
마지막 정수 찾기

Language: C

```c
#include <stdio.h>

void main()
{
    int i = 0, n, sum = 0;
    scanf("%d", &n);

    while(sum < n){
        i++;
        sum += i;
    }
    printf("%d", i);
    return 0;
}
```

반복 횟수를 특정할 수 없으므로
while문을 활용하는 게 좋다
합계를 나타내는 sum에 지속적으로 누적하여 더하다가
sum이 n보다 작지 않게 되면(크거나 같게 되면)
while문을 빠져나와 i를 출력하게 하자

3일차 내용 정리

[1366] C5: [SCSA 기본C 위크샵 문제] 주사위 던지기2

Language: C

```c
#include <stdio.h>

void main()
{
    for(int k = 1; k <= 6; k++){
        for(int j = 1; j <= 6; j++){
            for(int i = 1; i <= 6; i++){
                printf("%d %d %d\n", k, j, i);
            }
        }
    }
    return 0;
}
```

삼중 반복문을 구성할 수 있는지에 관한 문제이다
머릿속에서 바로 구현이 된다면 바깥에서부터
어렵다면 안쪽에서부터 차근차근 구현해보자

3일차 내용 정리

[1369] C6: [SCSA 기본C 워크샵 문제]
3의 배수의 합

Language: C

```c
#include <stdio.h>

void main()
{
    int N = 0, sum = 0;

    while(sum < 1000){
        N += 3;
        sum += N;
    }
    printf("%d", N - 3);
    return 0;
}
```

이전 C4 문제와 전개가 비슷하다
하지만, while문이 종료되었다는건
해당 N을 더했을 때, 1000보다 크다는 뜻이므로
3을 빼줘야 원하는 N을 얻을 수 있다

3일차 내용 정리

[1372] C7: [SCSA 기본C 워크샵 문제] 약수 출력

Language: C

```c
#include <stdio.h>

void Print_Divisor(int n)
{
    for(int i = 1; i <= n; i++){
        if(n % i == 0){
            printf("%d ", i);
        }
    }
    printf("\n");
}

void main()
{
    int n;
    scanf("%d", &n);

    for(int i = 2; i <= n; i++){
        Print_Divisor(i);
    }
    return 0;
}
```

주어진 n에 대하여 n의 모든 약수를 출력하는 Print_Divisor 함수를 설계한 후 2부터 n까지 약수를 출력하였다
수학적 지식이 있다면, Print_Divisor함수 안에 n까지가 아닌 n/2까지만 확인하도록 하여 실행속도를 빠르게 할 수 있다

3일차 내용 정리

[1362] C8: [SCSA 기본C 워크샵 문제]
정수 역순 인쇄

Language: C

```c
#include <stdio.h>

void main()
{
    int n, result = 0;
    scanf("%d", &n);

    while(n > 0){
        result *= 10;
        result += n % 10;
        n /= 10;
    }
    printf("%d", result);
    return 0;
}
```

단순 출력하기만을 원한다면
입력받은 n값의 1의 자리를 출력 후
10으로 나누는 작업을 반복한다

역순으로 숫자를 저장하고자 할 때,
다음과 같이 저장되는 곳(result)의 자릿수를 한 자리씩 늘려가며
기존 값(n)의 1의 자리를 가져오는 작업을 반복한다

3일차 내용 정리

[1678] C9: [TST]
두 수의 거리

Language: C

```c
#include <stdio.h>

int main(void)
{
    int a, b;
    scanf("%d %d", &a, &b);

    // 여기서부터 작성

    int result = a - b;
    if(result < 0){
        result *= -1;
    }
    printf("%d", result);
    return 0;
}
```

두 수의 차가
음수라면 양수로 바꿔주는 Logic을 추가하자

C 카테고리 문제도 수고하셨습니다