1일차 내용 정리

[6113] A0: [SCSA 1-33]

/, % 연산자의 활용 => 10진수 자리수 분리

Language: C

```c
#include <stdio.h>

void main(void)
{

    int money = 78000;
    // -> 1000원짜리는 몇 장인지?
    money / 50000;
    money % 50000 / 10000;
    momey % 10000 / 5000;
    money % 5000 / 1000;
    // 해당 단위의 자릿수를 구하기 위해
    // 상위 단위로 나눈 나머지를 활용할 수 있다.
    // 예) 100원짜리 동전개수를 구하기 위해
    //     500으로 나눈 나머지를 활용한다.


    int a = 12345;
    int a4, a3, a2, a1;

    a4 = a % 10000/ 1000;
    a3 = a % 1000 / 100;
    a2 = a % 100 / 10;
    a1 = a % 10 / 1;

    printf("1000자리=%d, 100자리=%d, 10자리=%d, 1자리=%d\n", a4, a3, a2, a1);
}
```

해당 단위의 자릿수를 구하기 위해
앞선 단위를 제거하는 방법이 필요하다
780원에서 10원짜리 개수를 구하기 위해
780을 50으로 나눈 나머지인 30을 활용해보자

1일차 내용 정리

[2634] A1: [SCSA 2-9-1]

반지름을 입력하면 원의 넓이를 구하는 함수 작성

Language: C

```c
#include <stdio.h>

float compute_circle_area(float radious);

void main(void)
{
    float r;
    scanf("%f", &r);
    printf("%f\n", compute_circle_area(r));
}

float compute_circle_area(float radious)
{
    float pi = 3.14f;

    // 코드 작성
    return pi * radious * radious;
}
```

문제를 해결하기 위해 필요한 수학적 지식이 있을 수 있다
혹시 처음 본다면, 외워두도록 하자
이해가 필요한 영역이라면, 유튜브에 친절한 설명이 있다
이러한 선험적 지식이 쌓일수록 해결하기 위한 다양한 방법을 찾을 수 있다

1일차 내용 정리

[2636] A2: [SCSA 2-9-3]
float 값을 넣으면 가장 가까운 정수 값을 찾아서 넘겨주는 함수 작성

Language: C

```c
#include <stdio.h>

int find_int(float value);

void main(void)
{
    float num;
    scanf("%f", &num);
    printf("%d\n", find_int(num));
}

int find_int(float value)
{
    int i = (int) value;
    float f = value - i;

    if(f >= 0.5){
        return i + 1;
    }else{
        return i;
    }
}
```

우리에게 필요한 실수부(float)를 구하기 위해
C언어에 내장되어 있는 형변환(Casting)을 사용하였다
이처럼 Type을 명시해 강제로 형변환 하는 방식을
명시적 형변환(Explicit Casting)이라 한다
구한 실수부(float)가 0.5이상인지 미만인지를 판단하여
정수부(integer)를 1증가시키거나 그대로 return 할 수 있다

1일차 내용 정리

[2638] A3: [SCSA 2-10]

대문자를 소문자로 바꾸는 함수 작성

Language: C

```c
char Change_Case(char upper)
{
    // 'A' 65 , -> 'a' 97
    return upper + ('a' - 'A');
}

void main(void)
{
    char a;

    scanf("%c" , &a );
    printf("%c => %c\n", a, Change_Case(a));
}
```

ASCII(아스키) 코드를 외울 필요는 없지만
대표적인 값들은 알아두면 도움이 된다
'A'는 65
'a'는 97
'0'은 48이다
예를 들어, 'A'를 'a'로 바꾸기 위해서 32의 값을 증가시킬 필요가 있다
32라는 값은 97과 65의 차이에서 왔다(97 - 65)
1) upper + 32
2) upper + (97 - 65)
3) upper + ('A' - 'a')
위 3가지 방법 중 3번 방법이 직관적이다.
물론, 1번과 2번을 사용해도 무방하다

1일차 내용 정리

[2639] A4: [SCSA 2-11]

ASCII 숫자 문자를 정수 숫자로 반환하는 함수

Language: C

```c
#include <stdio.h>

int Change_Char_to_Int(char num)
{
    return num - '0';
}

void main(void)
{
    char a;

    scanf("%c", &a);
    printf("%d\n", Change_Char_to_Int(a));
}
```

'1'을 정수로 출력해보면 49가 나온다
우리가 원하는 정수 1을 구하기 위해 48을 빼면 된다
1) num - 48
2) num - '0'
둘 중 원하는 쪽으로 사용하자

1일차 내용 정리

[2640] A5: [SCSA 3-6-1]

홀짝을 맞춰라

Language: C

```c
#include <stdio.h>

int Check_Odd_Even(int num)
{
    if(num % 2 == 0){
        return 2;
    }else{
        return 1;
    }
    // 참 -> 1 -> 2
    // 거짓 -> 0 -> 1
    // return num % 2 == 0 + 1;

}

void main(void)
{
    int num;
    scanf("%d", &num);
    printf("%d\n", Check_Odd_Even(num));
}
```

홀수와 짝수의 정의를 살펴보자
짝수는 2의 배수, 홀수는 2의 배수가 아닌 수이다
배수란, 정수 배가 되는 수이다. x1, x2, x3, ...,
(물론, x0과 x(-1), x(-2)등도 있지만 문제에서는 양의 정수 조건이 달려 있다)
해당 조건에 따라 if-else로 구성하거나
2로 나눈 나머지가 참이라면 1, 거짓이라면 0이 나온다
이를 활용하여 1을 2로, 0을 1로 만드는 로직(Logic)을 구현하자

1일차 내용 정리

[6119] A6: [SCSA 3-6-2]

홀짝을 맞춰라 - 변경

Language: C

```c
#include <stdio.h>

int Check_Odd_Even(int num)
{
    if(num % 2 == 0){
        return 0;
    }else{
        return 1;
    }
    // 0 -> 1
    // 1 -> 0
    // return 1 - (num % 2 == 0);
}

void main(void)
{
    int num;
    scanf("%d", &num);
    printf("%d\n", Check_Odd_Even(num));
}
```

해당 조건에 따라 if-else로 구성하거나
2로 나눈 나머지가 참이라면 1, 거짓이라면 0이 나온다
이를 활용하여 1을 0으로, 0을 1로 만드는 로직(Logic)을 구현하자
1 -> 0
0 -> 1
위 두 case의 공통점은 input과 output의 합이 1이라는 점이다
이를 활용하여 x + y = 1 즉, y = 1 - x 함수를 떠올릴 수 있다
물론, (1, 0)과 (0, 1)의 좌표를 지나는 함수를 구해도 된다
하지만, if-else가 보기 편하다(가독성이 높다)
취향에 따라 선택하자

1일차 내용 정리

[2641] A7: [SCSA 3-7]
2,3,5의 배수 판단하기

Language: C

```c
#include <stdio.h>

int compare(int num)
{
    if(num % 2 == 0){
        return 2;
    }else if(num % 3 == 0){
        return 3;
    }else if(num % 5 == 0){
        return 5;
    }else{
        return 0;
    }
}

void main(void)
{
    int num;
    scanf("%d", &num);
    printf("%d\n", compare(num));
}
```

여러가지 방법이 있겠지만, 가독성 높은 코드를 구성해보았다
물론, 해당 문제 조건에 공통배수는 입력되지 않는다고 가정했으므로
if문 여러개로 구성해도 좋다

```
if(num % 2 == 0) return 2;
if(num % 3 == 0) return 3;
if(num % 5 == 0) return 5;
return 0;
```

1일차 내용 정리

[2643] A8: [SCSA 3-16]
3의 배수 또는 5의 배수 찾는 함수

Language: C

```c
#include <stdio.h>

int f1(int num)
{
    if(num % 3 == 0 || num % 5 == 0){
        return 1;
    }else{
        return 0;
    }
    // return num % 3 == 0 || num % 5 == 0;
}

void main(void)
{
    int num;
    scanf("%d", &num);
    printf("%d\n", f1(num));
}
```

OR 연산자를 활용할 수 있는지,
3의 배수와 5의 배수를 판별할 수 있는지에 관한 문제이다
if-else로 코드를 엮어도 좋고, 한 줄로 return해도 좋다

정답의 return !(num % 3) || !(num % 5);
경우는 '=='부분에서 비교 연산할 때 사용되는 시간을 줄이기 위해 구성되었다

1일차 내용 정리

[1348] B3: [SCSA 기본C 워크샵 문제]

넘어온 정수가 3과 5의 공배수인지 판별하는 Common_Calc 함수를 구현하시오

Language: C

```c
#include <stdio.h>

char Common_Calc(int num);

int main(void)
{
    //함수작성
    int num;

    scanf("%d", &num);
    printf("%c", Common_Calc(num));
    return 0;
}

char Common_Calc(int num)
{
    //함수작성
    if(num % 3 == 0 && num % 5 == 0){
        return 'O'
    }else{
        return 'X'
    }


}
```

AND 연산자를 활용할 수 있는지,

공배수의 정의에 대해 알고 있는지에 관한 문제이다

3과 5의 공배수가 15임을 알고

num % 15 == 0으로 구성할 수도 있지만, 컴퓨터에게 계산을 시키자

Logic상 3과 5대신 다른 값이 들어올 가능성이 있다면 따로 판별하는 걸 권장한다