

C교육 2주차

201702075 조수환





INDEX



001/	1주차 과제 풀이
002/	자료형과 변수
003/	입출력
004/	조건문과 반복문
005/	과제

C교육 2주차

1주차 과제 풀이

- 과제 1
- 과제 2
- 과제 3



1주차 과제 풀이

과제 1



A. mkdir 명령어를 사용하여 C-education 디렉토리 생성 후 이동하여
다시 mkdir 명령어를 사용해 practice, homework 디렉토리 생성

1. C-education 디렉토리 생성

```
sh2358@edu-argos:~$ mkdir C-education
sh2358@edu-argos:~$ ls
201702075 C-education
```

2. C-education 디렉토리로 이동

```
sh2358@edu-argos:~$ cd C-education/
```

3. practice, homework 디렉토리 생성

```
sh2358@edu-argos:~/C-education$ mkdir practice homework
sh2358@edu-argos:~/C-education$ ls
homework practice
```

1주차 과제 풀이

과제 2



A. mv 명령어(or cp 명령어)를 통해 print.c 파일을 C-education의 homework 디렉토리로 옮긴다.

1. mv or cp 명령어를 통해 ~/C-education/homework로 print.c 파일 이동

```
sh2358@edu-argos:~/201702075$ mv print.c ../C-education/homework/
```

2. homework 디렉토리로 이동 후 ls 명령어로 파일 이동 확인

```
sh2358@edu-argos:~/201702075$ cd ../C-education/homework/  
sh2358@edu-argos:~/C-education/homework$ ls  
print.c
```

1주차 과제 풀이

과제 3



A. rm 명령어를 통해서 학번 디렉토리를 삭제한다.

디렉토리 삭제에는 -r 옵션을 붙인다.

```
sh2358@edu-argos:~$ ls
201702075  C-education
sh2358@edu-argos:~$ rm -r 201702075/
sh2358@edu-argos:~$ ls
C-education
```

C교육 2주차

자료형과 변수

- 자주 쓰는 자료형
- 연산자
- 변수와 메모리
- Overflow





자료형	표현	크기
int	정수	4byte
float	실수(소수)	4byte
double	실수(소수)	8byte
char	문자	1byte

4가지 자료형 외에도 다양한 자료형이 있습니다.

다른 자료형들이 궁금하다면 구글을 참고합시다!

자료형과 변수

연산자



연산자	역할	예시
<code>+, -, *, /</code>	사칙연산	<code>7 + 3 == 10</code>
<code>%</code>	모듈러(나머지) 연산	<code>7 % 3 == 1</code> , (몫 2, 나머지 1)
<code>=</code>	대입	<code>int a = 3</code> , (변수 a에 값 3을 대입)
<code>==</code>	등위	<code>if(a == 3)</code> , (a가 3과 같다면)
<code>!=</code>	Not 등위	<code>if(a != 3)</code> , (a가 3과 같지 않다면)
<code><, ></code>	비교연산자(미만, 초과)	<code>if(a < 3)</code> , (a가 3 미만이라면)
<code><=, >=</code>	비교연산자(이하, 이상)	<code>if(a <= 3)</code> , (a가 3 이하라면)
<code>--</code>	-1	<code>a-- == a - 1</code>
<code>++</code>	+1	<code>a++ == a + 1</code>



변수와 메모리

변수(variable)

- 변할수 있는 값
- 값을 저장하고 연산하기 위해 필요함
- 변수 선언 : [자료형] [변수명]
- 변수명은 항상 직관적으로!

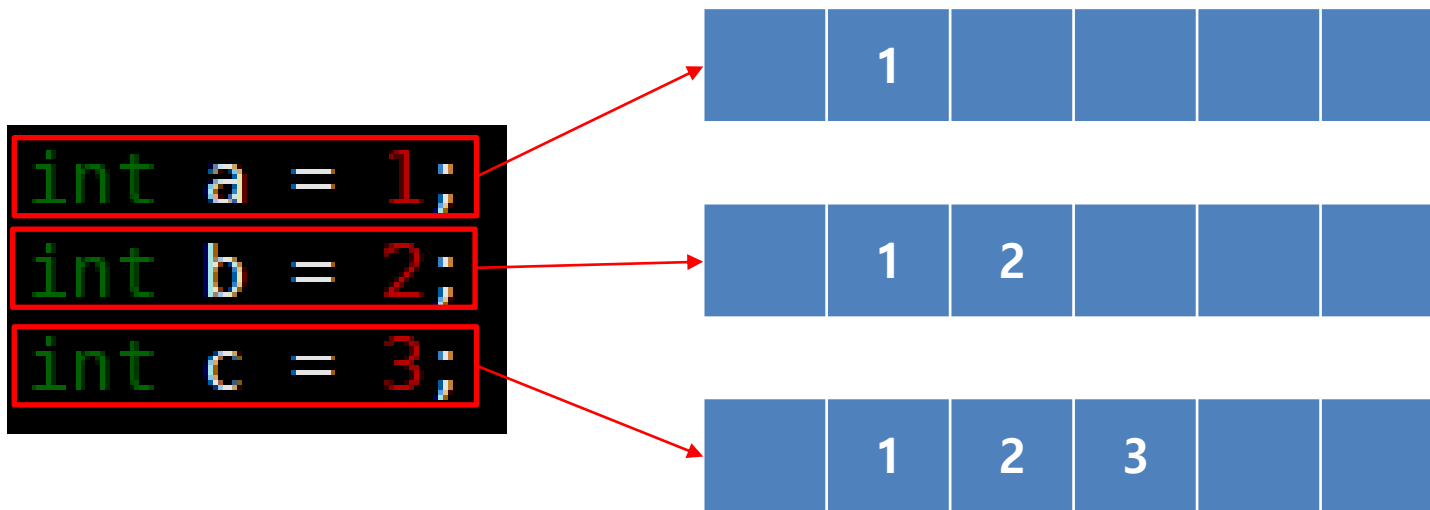
```
int a;
```

[자료형(int)] [변수명(a)]

```
int a, b, c;
```

같은 자료형으로 여러 변수 선언 가능

선언된 변수는 메모리 어딘가에 저장된다.

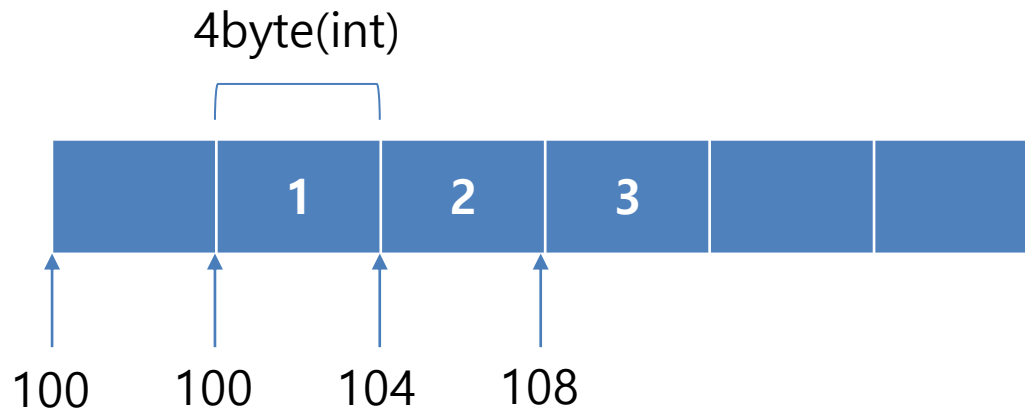


나중에 필요할 때, 값을 복사해서 사용함



각각의 변수들은 고유한 메모리 주소를 가진다.

또한 자료형의 크기에 맞는 공간을 할당받는다.



만약 자료형의 표현범위를 넘는다면?

자료형과 변수 Overflow



Overflow

- 문자 그대로 흘러넘친다는 뜻
- 그 대상에 따라 integer overflow, stack overflow 등 다양하다.

```
int max = 2147483647;  
printf("max : %d\n", max);  
printf("max + 1 : %d\n", max + 1);  
return 0;
```



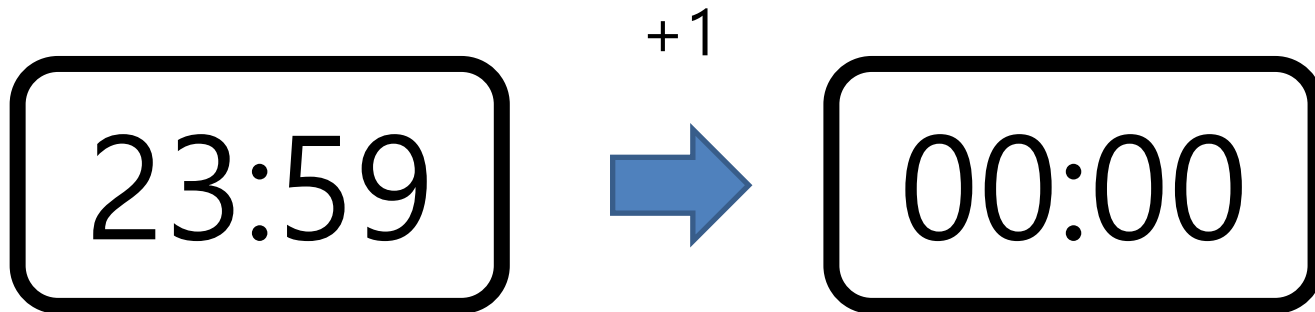
```
sh2358@edu-argos:~/C-education/practice$ ./overflow  
max : 2147483647  
max + 1 : -2147483648
```

max + 1은 당연히..

???

Integer Overflow

정수의 표현 범위를 넘어서면 아주 작은 수로 바뀌는 현상



00:00 ~ 23:59 까지 표현 가능한 전자시계의 예시이다.

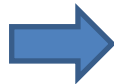
표현범위가 정해진 정수형 변수 또한 같은 현상을 겪는다.



int형 변수의 표현 범위는 -2147483648 ~ 2147483647이므로

max + 1의 값이 -2147483648로 표시되었던 것

```
int min = -2147483648;  
printf("min : %d\n", min);  
printf("min - 1 : %d\n", min - 1);  
return 0;
```



```
sh2358@edu-argos:~/C-education/practice$ ./overflow  
min : -2147483648  
min - 1 : 2147483647
```

위의 예시처럼 반대의 경우도 가능하다.

00:00의 1분전이 23:59 인것 처럼



* 눈보라 게임사의 모 게임 랭크 점수 실화 *

 5,000점 랭킹 1위 (상위 0.01%)		
승률	K/D	플레이 시간
0%	1.03 : 1	30시간
1승 2무 233패	5 / 5	236게임

1점에서 패배하자 0점이나 음수가 아닌 최대 점수 5000점이 됨

C교육 2주차

입출력

- Format string
- 입출력 함수
- 실습





Format string(형식 지정자) : 입출력의 형식을 지정해주는 문자

Format string	설명
%d	정수를 나타냄
%f	실수를 나타냄
%c	문자를 나타냄
%s	문자열을 나타냄
%x	16진수로 표현

이외에도 다른 format string이 있음.

각각의 자료형에 맞게 사용!

printf()

- C의 표준 입출력 함수 중 하나
- 여러 종류의 데이터를 형식에 맞게 출력 해준다
- 변수 출력 시 : printf("[형식 지정자]", [변수명])
- 아래의 예시처럼 출력하고자 하는 문자열과 변수를 함께 출력 가능

형식지정자 %d (정수형)

변수명 max (int)

```
int max = 2147483647;  
printf("max : %d\n", max);  
printf("max + 1 : %d\n", max + 1);  
return 0;
```



scanf()

- C의 표준 입출력 함수 중 하나
- 여러 종류의 데이터를 형식에 맞게 입력 받을 수 있다.
- 사용법 : scanf("[형식 지정자]", &[변수명])

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

&[변수명]는 변수의 주소를 의미한다.

```
scanf("%d %d", &b, &c);
```

이런식으로 여러 개의 변수도 받을 수 있다

```
printf("a = %d\n", a);
```

 → 애는 그냥 변수를 쓰는데

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

 → 애는 왜 변수의 주소를 쓸까?

A. C에서는 어떤 값을 불러올 때 원본이 아닌 복사본을 가져온다.

```
printf("a = %d\n", a);
```

 원본이나 복사본이나 출력에 영향 X

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

 변수의 값 = 입력받은 값이 되어야 하므로
변수의 주소(원본)를 기입한다.

정수 a , b , c 를 입력받아 $a + b - c$ 의 값을 형식에 맞게 출력하시오

```
#include <stdio.h>

int main(void) {
    int a, b, c;
    printf("정수 a, b, c의 값을 입력하시오 : ");
    /*

        write your code

    */
}
```

```
sh2358@edu-argos:~/C-education/practice$ gcc -o practice1 practice1.c
sh2358@edu-argos:~/C-education/practice$ ./practice1
정수 a, b, c의 값을 입력하시오 : 2 3 4
a + b - c = 2 + 3 - 4 = 1
```

막히는게 있다면 저나 튜터에게 질문해주세요

C교육 2주차

조건문과 반복문

- 조건문
- 반복문
- 실습



조건문



if, else, else if

- 가장 보편적인 조건문 문법
- 조건이 참일 때, 중괄호 안의 코드를 실행한다.
- 사용법 : if([조건]) {[코드]} else {[코드]}

```
int num;
scanf("%d", &num);
if(num < 0) {
    printf("Negative Number\n");
} else if(num == 0) {
    printf("Zero\n");
} else {
    printf("Positive Number\n");
}
return 0;
```

```
sh2358@edu-argos:~/C-education/practice$ ./sign
-4
Negative Number
```

```
sh2358@edu-argos:~/C-education/practice$ ./sign
0
Zero
```

```
sh2358@edu-argos:~/C-education/practice$ ./sign
3
Positive Number
```




조건문과 반복문

조건문

switch/case문

- 선택지가 여러 개일 때 적합한 조건문
- 사용법 : switch([변수]) {

case [상수값]: [코드]

default [코드]}

```
sh2358@edu-argos:~/C-education/practice$ ./switch
0
grade A
```

```
sh2358@edu-argos:~/C-education/practice$ ./switch
4
grade F
```

```
sh2358@edu-argos:~/C-education/practice$ ./switch
-5
Not Valid Number
```

```
int absence;
scanf("%d", &absence);
switch(absence) {
    case 0:
        printf("grade A\n");
        break;
    case 1:
        printf("grade B\n");
        break;
    case 2:
        printf("grade C\n");
        break;
    case 3:
        printf("grade D\n");
        break;
    case 4:
        printf("grade F\n");
        break;
    default:
        printf("Not Valid Number\n");
}
```

결석 횟수에 따른 학점 출력



입력받은 정수가 짝수인지 홀수인지 출력하시오.

```
#include <stdio.h>

int main(void) {
    int num;
    scanf("%d", &num);
    /*
        write your code
    */
}
```

```
sh2358@edu-argos:~/C-education/practice$ ./practice2
3
Odd Number
```

```
sh2358@edu-argos:~/C-education/practice$ ./practice2
2
Even Number
```

조건문과 반복문

반복문

for문

- 반복 횟수가 명확할 때 주로 쓰는 반복문 문법
- 사용예시 : `for(int i = 0 ; i < 10 ; i++) {[코드]}`

```
int fact = 1;
for(int i=1 ; i<=10 ; i++) {
    fact *= i;
}
printf("10! = %d\n", fact);
return 0;
```

```
sh2358@edu-argos:~/C-education/practice$ ./factorial
10! = 3628800
```

조건문과 반복문

반복문

while문

- 조건에 따라 반복 횟수를 결정 해야 할 때 주로 사용
- 사용예시 : while([조건]) {[코드]} *조건이 true일때 반복

```
int num = 2;
while(num <= 100) {
    printf("%d ", num);
    num *= 2;
}
printf("\n");
return 0;
```

100 이하의 2의 n제곱 출력

```
sh2358@edu-argos:~/C-education/practice$ ./expo
2 4 8 16 32 64
```

조건문과 반복문

반복문



등차수열 a_n 의 초항, 공차, n 을 입력받고 $a_1 \dots a_n$ 을 출력하시오

```
#include <stdio.h>

int main(void) {
    int init, comm, n;
    scanf("%d %d %d", &init, &comm, &n);
    /*
        write your code
    */
}
```

```
sh2358@edu-argos:~/C-education/practice$ ./practice3
1 3 10
1 4 7 10 13 16 19 22 25 28
```

$$a_n = a_1 + (n - 1)d, \quad a_1 = 1, \quad d = 3, \quad n = 10$$

C교육 2주차

과제

- 임금 횡령(필수)
- 복수(필수)
- 별 찍기(도전)
- 약수의 개수(도전)





악덕사장 철수는 사내 복지 기금을 명목으로 직원들의 임금을 횡령하려고 한다.

철수는 다음 조건에 맞게 기금을 제외하고 급여를 지급하는 프로그램을 만든다.

- 3개월치의 월급을 입력받는다.
- 월급은 최대 1000만원이고, 이를 넘게 기입하면 그 달의 월급은 없다.
- 급여에 따라서 300이상은 20%, 500이상은 30%, 나머지는 10%를 징수한다.
- 기금을 제외한 실제 급여의 합이 1500이상이면 추가로 10%를 징수한다.

입력받은 월급에 맞게, 직원의 실제 급여를 출력하시오



과제

임금 횡령(필수)

* 월급은 정수이며, 연산과정에서의 소수 자리는 무시한다. *

Ex) $15 * 0.5 = 7.5 \rightarrow 7$, 그냥 정수형 변수에 연산하면 알아서 이렇게 됩니당

* 입출력 예시 *

```
sh2358@edu-argos:~/C-education/homework$ ./homework1
2000000
4000000
7000000
9900000
```



$1800000 + 3200000$
 $+ 4900000 = 9900000$

```
sh2358@edu-argos:~/C-education/homework$ ./homework1
10000000
10000000
10000000
18900000
```



$7000000 * 3 = 21000000$,
 $21000000 * 0.9 = 18900000$



철수의 행태에 분노한 영수는 사장실에 잠입해서 프로그램의 코드를 알아냈다.

프로그램을 분석하여 임금을 마구잡이로 받아내보자.

- 과제 1에서 작성한 프로그램을 그대로 실행한다.
- 총 급여가 10억원 이상이 나오도록 숫자를 입력할 방법을 생각해본다.
- 10억원 이상이 나온다면, 프로그램을 어떻게 고쳐야 할지 생각하고 고쳐본다.

10억원 이상이 출력되었다면 결과화면과 수정된 코드를

이미 위와 같은 경우를 대비했다면 과제 1의 코드만 제출하시면 됩니다.

한 변의 길이가 n 인 삼각형 4개를 이용하여 마름모를 표현하려고 한다.

다음과 같은 모양이 나오도록 "*"을 출력하시오

```

3
*** ***
**  **
*   *

*   *
**  **
*** ***
  
```

```

5
***** *****
****  ****
***   ***
**    **
*     *

*     *
**    **
***   ***
****  ****
***** *****
  
```

```

8
***** *****
***** *****
*****  *****
*****   *****
*****    *****
*****     *****
*****      *****
*****       *****

*****      *****
*****     *****
*****    *****
*****   *****
*****  *****
***** *****
***** *****
***** *****
***** *****
  
```

Q & A

Thank You for Listening

