

C 기초 프로그래밍 교육

Week 2. variables, input, branch

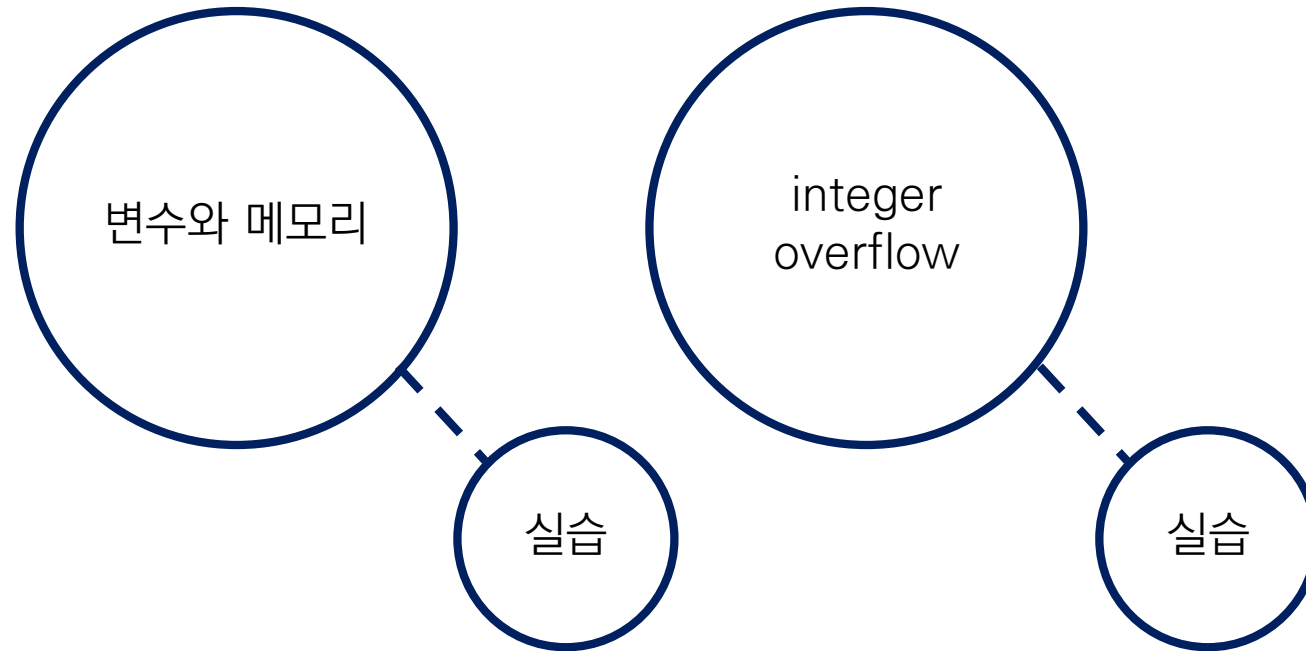
| 교육 구성

INDEX

-
1. 변수와 메모리
 2. 입력문
 3. 제어문 (if)
-

1. 변수와 메모리

with
format string



변수 선언과 연산자

“ 변수(變數, variable) ”

- 변할 수 있는 값
- 데이터를 저장하고 이를 연산하기 위해 필요
- 적절한 변수 이름을 사용함으로써 코드의 가독성을 높일 수 있다.

변수 만드는 법

〈자료형〉 〈변수 이름〉 = 〈초기 값〉;

```
#include <stdio.h>

int main()
{
    int number = 3;
}
```

변수 선언과 연산자

〈 자주 사용되는 자료형 〉

자료형	표현	크기
int	정수	4 bytes
float	소수	4 bytes
double	소수	8 bytes
char	문자	1 bytes

변수 선언과 연산자

〈 연산자 〉

연산자	역할	예시
%	나머지	7 % 3 == 1
=	대입	int a = 3;
==	등위	if(a == 3)
!=	Not 등위	if(a != 4)
>	좌측이 우측보다 큼	if(a > 3)
<	우측이 좌측보다 큼	if(a < 3)
--	1 감소	a--;
++	1 증가	a++;

변수 선언과 연산자

〈 format string (형식 지정자) 〉

출력에 변수를 넣고 싶다!

JAVA

```
System.out.println("변수 : " + 변수);
```

C

```
printf("변수 : %d", 변수);
```



이것이
형식 지정자!(format string)

변수 선언과 연산자

〈 format string (형식 지정자) 〉

Format String	설명
%d	정수를 나타냄
%c	문자를 나타냄
%s	문자열을 나타냄
%x	16진수 표현으로 출력

더해서 출력하기

- 소스 만들기 : vi [이름].c
- 저장하기 : 커맨드 모드 -> :wq
- 컴파일 : gcc -o [이름] [이름].c
- 실행 : ./[이름]

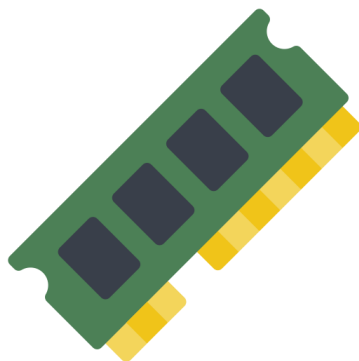
```
#include <stdio.h>

int main() {
    int a = 3;
    int b = 4;
    int c = a + b;

    printf( );
}
```

```
minibee@argos-edu:~/cedu/week2$ ./prac1
7
```

변수 선언 시 메모리



메모리?

- 우리가 여기서 말할 메모리는 RAM(Random Access Memory)
- 프로그램이 실행되는 동안 **임시**로 어떤 저장 공간이 필요할 때 사용
- (중요) 우리가 변수를 선언하면 메모리 어딘가에 값을 저장한다!

변수 선언 시 메모리

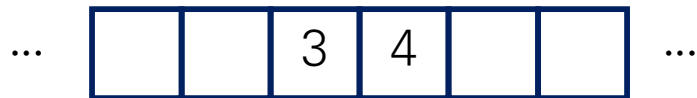
```
#include <stdio.h>

int main() {
    int a = 3;
    int b = 4;
    int c = a + b;

    printf("%d\n", c);
}
```



(메모리 어딘가)



(메모리 어딘가)

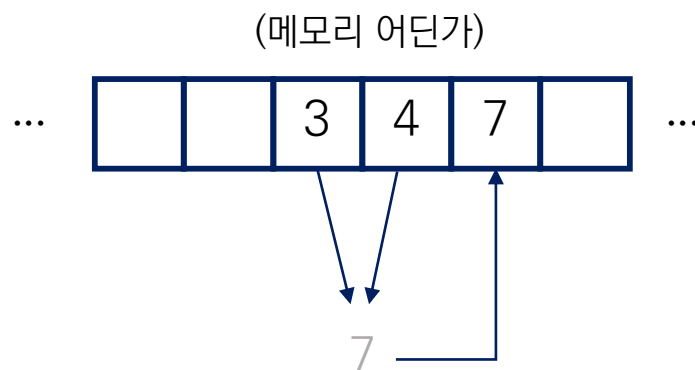
변수 선언 시 메모리 어딘가에 그 값을 저장한다!

변수 선언 시 메모리

```
#include <stdio.h>

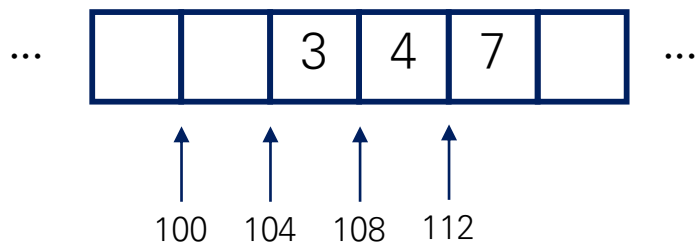
int main() {
    int a = 3;
    int b = 4;
    int c = a + b;

    printf("%d\n", c);
}
```



필요할 때 복사해서 쓴다!

변수 선언 시 메모리



메모리에도 집주소가 있다.

* 변수 집 주소 보는 법

&변수

변수 선언 시 메모리

```
#include <stdio.h>
int main()
{
    int a = 3;
    printf("%p\n", &a);
}
```

```
minibeef@argos-edu:~/cedu/week2$ ./prac2
0x7ffffae792b94
```

연속된 변수의 메모리 주소

- 소스 만들기 : vi [이름].c
- 저장하기 : 커맨드 모드 -> :wq
- 컴파일 : gcc -o [이름] [이름].c
- 실행 : ./[이름]

```
#include <stdio.h>

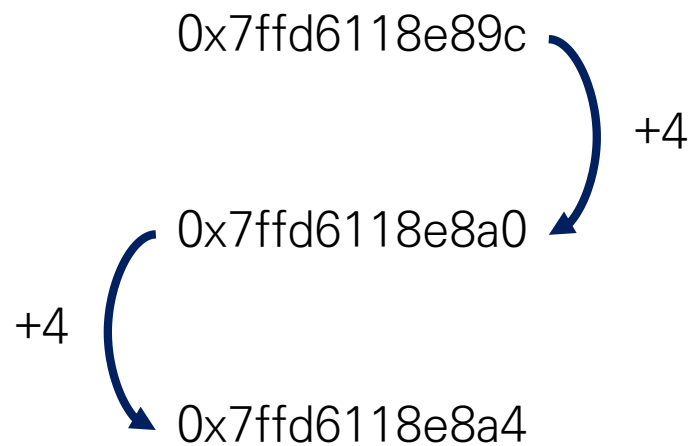
int main()
{
    int a = 3;
    int b = 4;
    int c = 5;

    printf( );
}
```

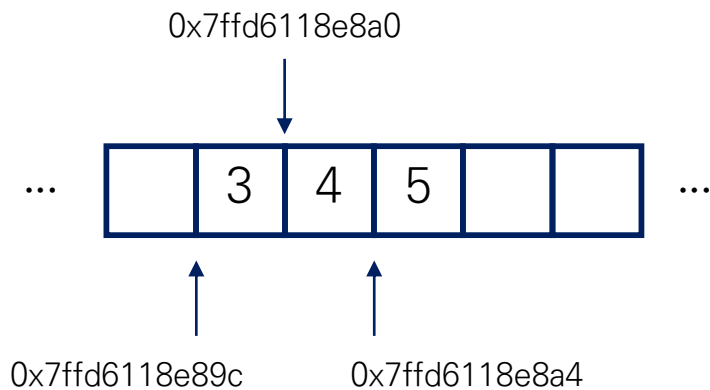
연속된 변수 a, b, c를 선언하고 이들의 주소 차이를 생각해봅시다.

연속된 변수의 메모리 주소

```
minibee@argos-edu:~/cedu/week2$ ./prac2  
0x7ffd6118e89c 0x7ffd6118e8a0 0x7ffd6118e8a4
```



연속된 변수의 메모리 주소



int형 변수 크기 : 4 byte

Integer Overflow

Integer Overflow

정수

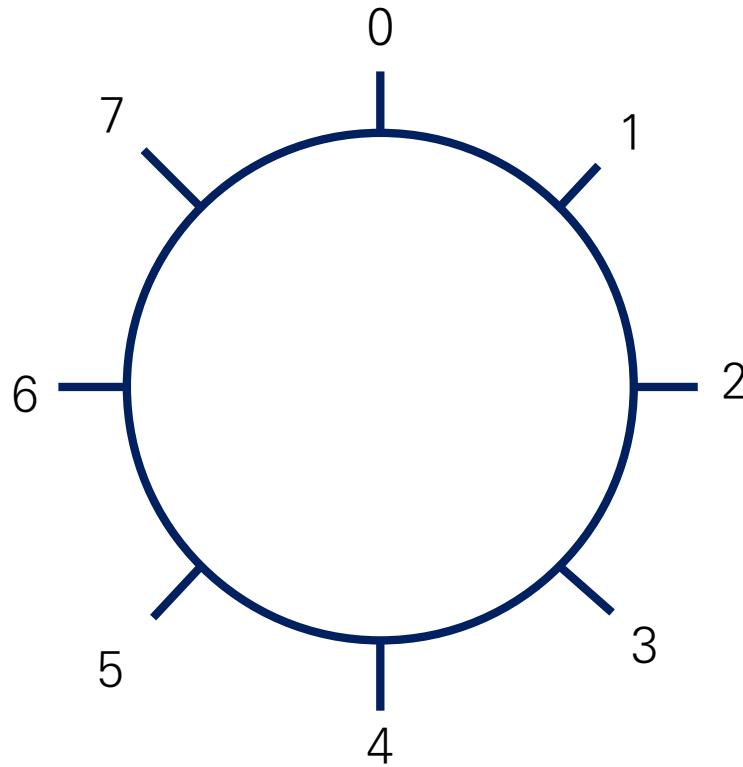
흘러 넘침

정수를 담을 수 있는 크기를 넘어서면 음수나 아주 작은 수로 바뀌는 버그

2,147,483,647

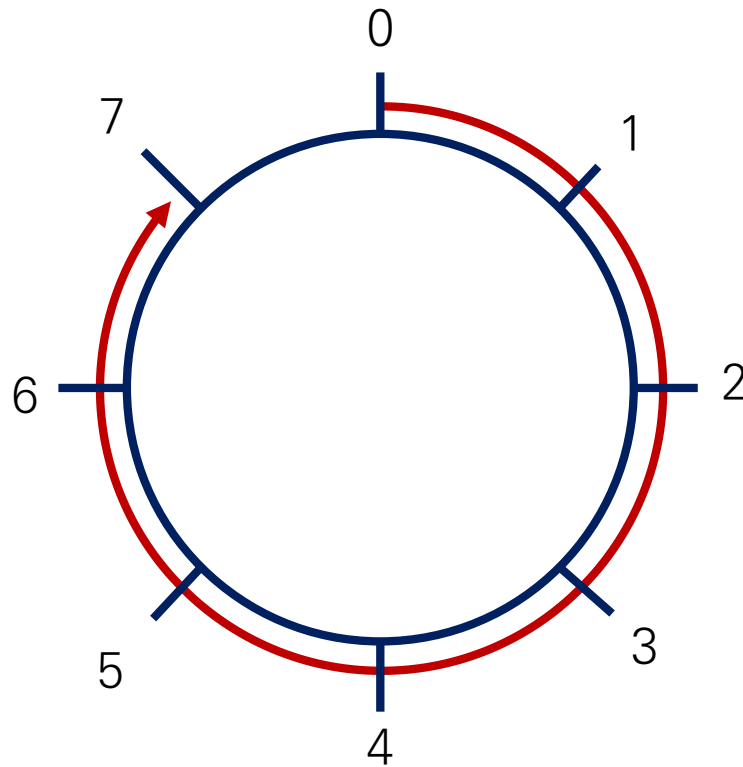
Integer Overflow

아래와 같이 0~7을 나타내는 수 체계가 있다고 하자



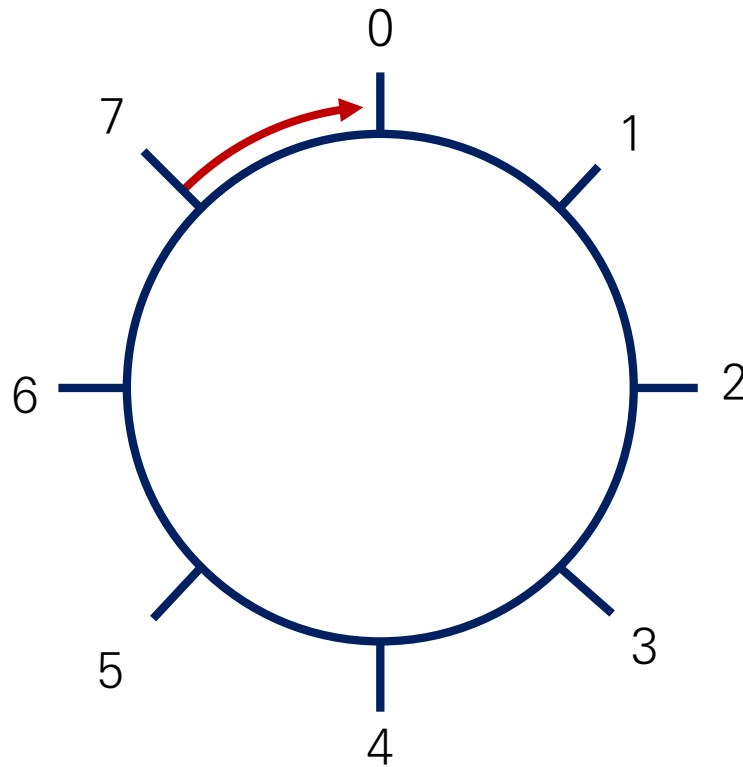
Integer Overflow

0에서 부터 7만큼 이동하면 7이다.

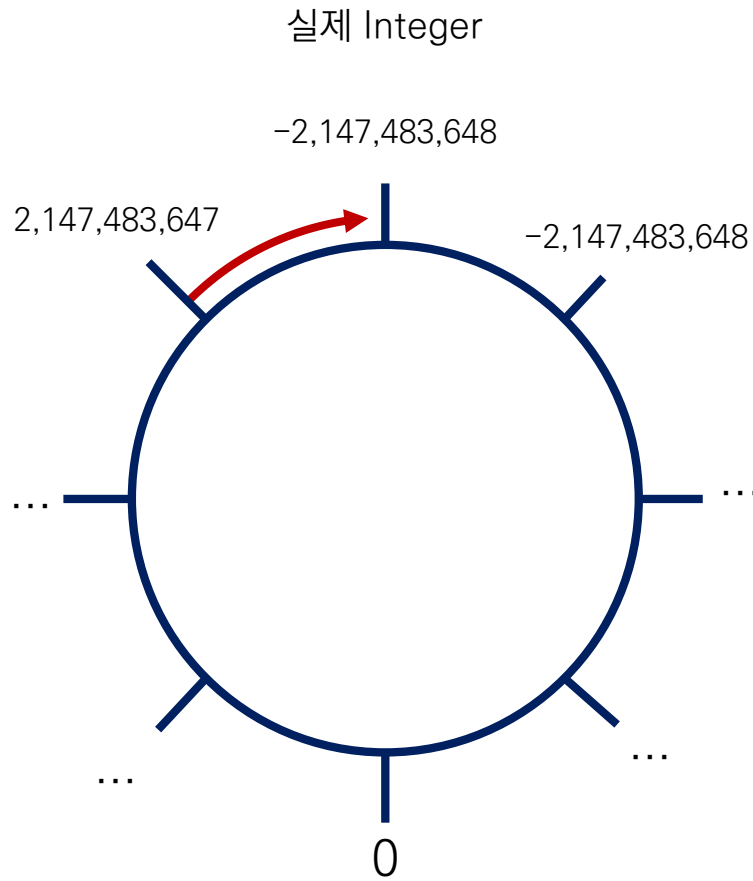


Integer Overflow

7(최대치)로 부터 1만큼 이동하면 최솟값인 0으로 바뀐다!



Integer Overflow



Integer Overflow 발생시키기

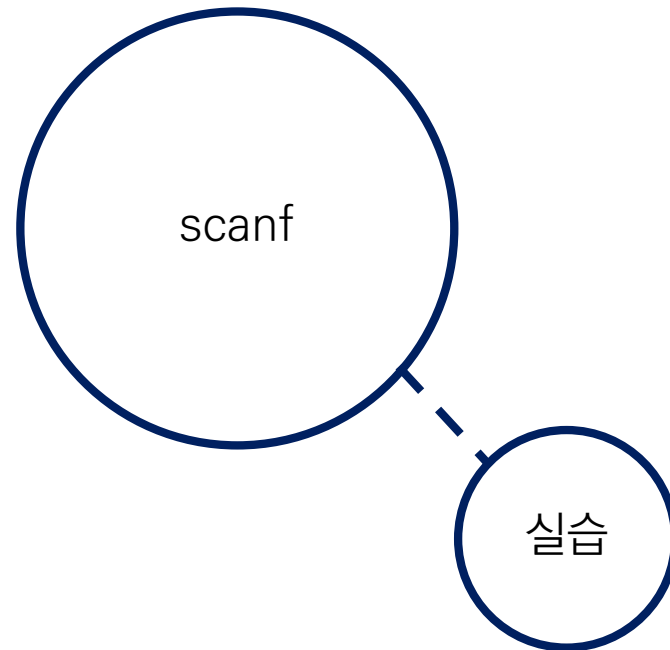
```
#include <stdio.h>

int main()
{
    int a = 2147483647;
    printf("int max : %d\n", a);

    a++;
    printf("int max + 1 : %d\n", a);
}
```

```
minibee@argos-edu:~/cedu/week2$ ./prac3
int max : 2147483647
int max + 1 : -2147483648
```

2. 입력



scanf

scanf(포맷 스트링, 변수 주소)

```
#include <stdio.h>

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

scanf

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



Q. 근데 왜 주소를 써야 하나요?

A. Call by Value!

scanf

Call by Value (값에 의한 호출)

C에서는 어떤 값을 가져다 쓸 때 원본이 아닌, 복사본을 가져와 연산한다.

```
int a;  
  
int b = a + 3;  
  
printf("%d", b);
```

scanf

즉 입력한 값을 변수에 저장해야 하는데, 복사본에 저장을 하면 의미가 없다.
→ 변수 주소에 가서 직접 저장

scanf("%d", &a);

수업한 내용을 필기하고
사진을 찍어 저장했다고 해보자

필기를 보고싶다면? 갤러리에 들어가서 사진을 보면 된다(복사본).

필기를 수정해야 한다면? 직접 필기 노트가 있는 곳으로 가서 수정해야 한다(원본).

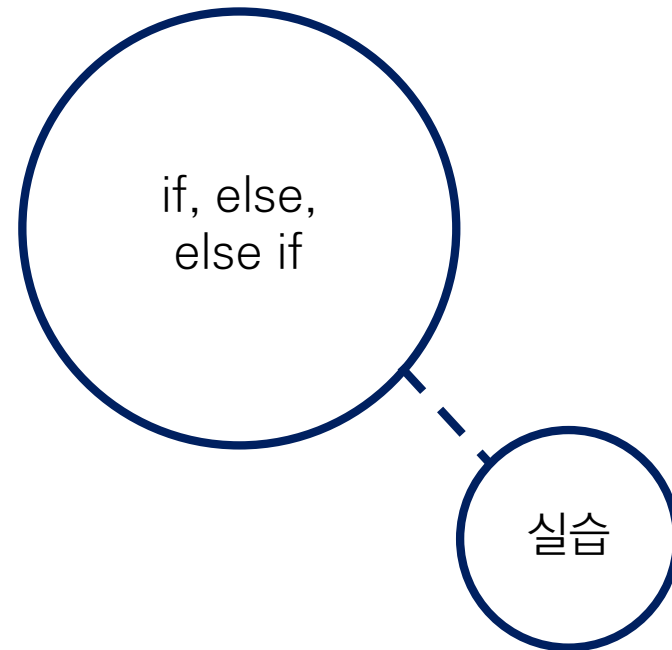
입력 받은 값을 2배로

```
#include <stdio.h>

int main()
{
    int a;
    [REDACTED]
    printf("%d\n", a * 2);
}
```

```
minibee@argos-edu:~/cedu/week2$ ./scan
4
8
```

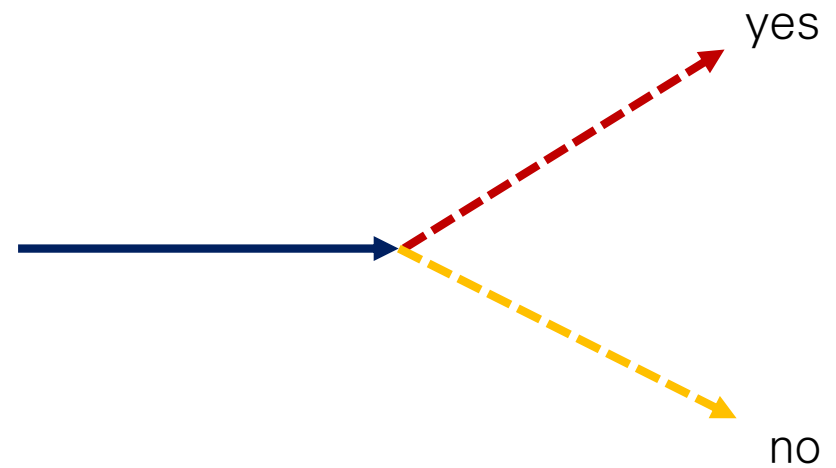
3. 제어문



if, else, else if

제어문?

프로그램의 흐름을 경우에 따라 여러 갈래로 나누고 싶을 때



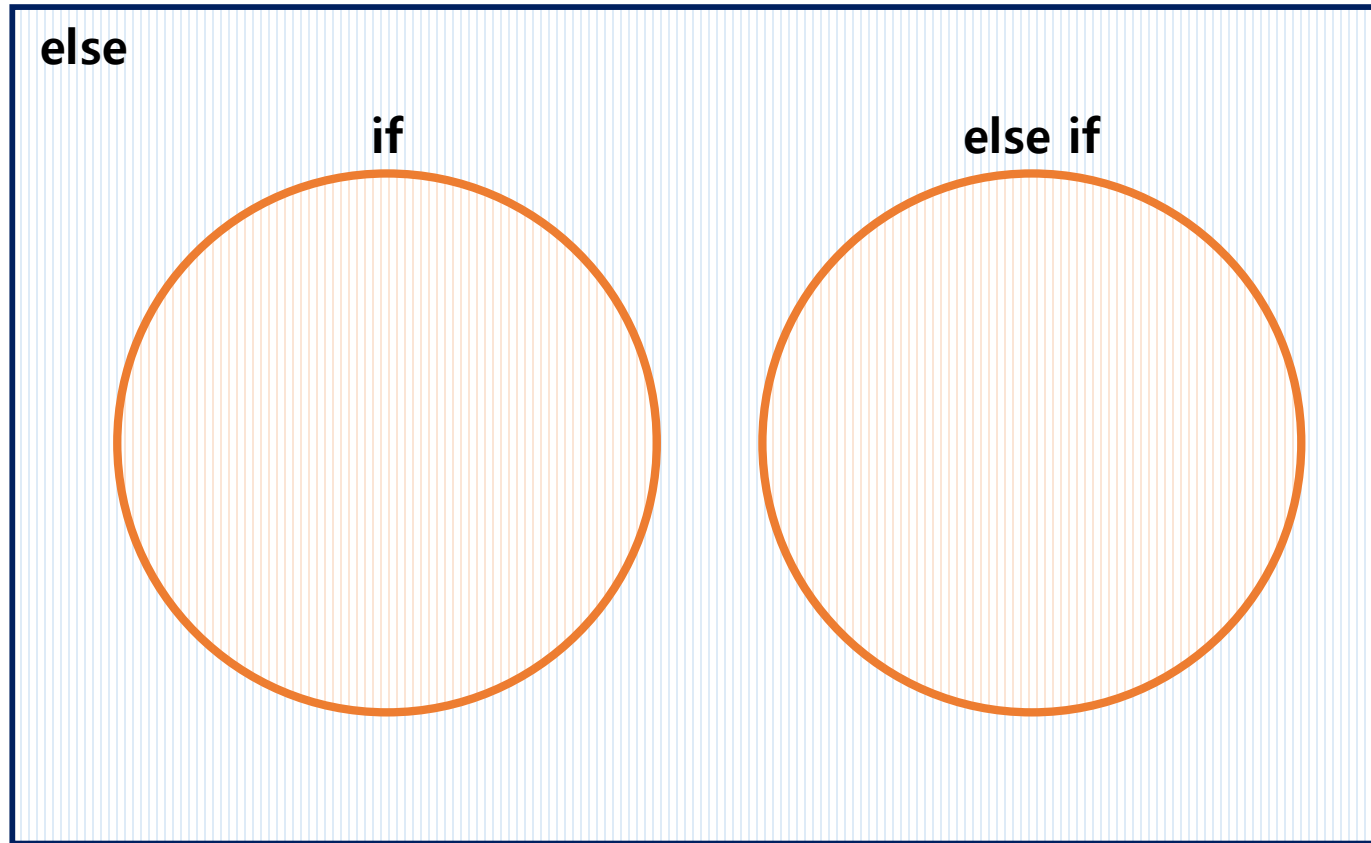
if, else, else if

```
if(조건) {  
  
} else if(조건) {  
  
} else if(조건) {  
  
} else {  
  
}
```

조건이 참이면 중괄호 안을 실행

else는 조건 이외 상황에 실행

if, else, else if



if, else, else if

```
#include <stdio.h>
int main()
{
    int a;
    scanf("%d", &a);

    if(a > 10) {
        printf("Larger Than 10\n");
    } else if(a == 10) {
        printf("Equal 10\n");
    } else {
        printf("Smaller Than 10\n");
    }
}
```

```
minibeef@argos-edu:~/cedu/week2$ ./if
9
Smaller Than 10
minibeef@argos-edu:~/cedu/week2$ ./if
10
Equal 10
minibeef@argos-edu:~/cedu/week2$ ./if
11
Larger Than 10
minibeef@argos-edu:~/cedu/week2$
```

if, else, else if

자주 하는 실수?

```
#include <stdio.h>
int main()
{
    int a;
    scanf("%d", &a);

    if(a > 10) {
        printf("Larger Than 10\n");
    } else if(a == 10) {
        printf("Equal 10\n");
    } else {
        printf("Smaller Than 10\n");
    }
}
```

```
#include <stdio.h>
int main()
{
    int a;
    scanf("%d", &a);

    if(a > 10) {
        printf("Larger Than 10\n");
    } else if(a = 10) {
        printf("Equal 10\n");
    } else {
        printf("Smaller Than 10\n");
    }
}
```

if, else, else if

자주 하는 실수?

```
minibeef@argos-edu:~/cedu/week2$ ./if
9
Equal 10
minibeef@argos-edu:~/cedu/week2$ ./if
10
Equal 10
minibeef@argos-edu:~/cedu/week2$ ./if
11
Larger Than 10
```

```
#include <stdio.h>

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

    if(a > 10) {
        printf("Larger Than 10\n");
    } else if(a = 10) {
        printf("Equal 10\n");
    } else {
        printf("Smaller Than 10\n");
    }
}
```

if, else, else if

자주 하는 실수?

if (a = 10) 괄호 안쪽을 수행

if (10) 0 이외의 모든 숫자는 참(True)

제어문 실습

음수? 양수?

```
#include <stdio.h>
int main()
{
    int a;
    scanf("%d", &a);

    {
        puts("a는 음수입니다.");
    }
    {
        puts("a는 양수입니다.");
    }
}
```

Hint : <, =, >

과제 1 - 주거 침입

```
#include <stdio.h>

int main()
{
    int a;
    int b = 10;
    scanf("%d",         );
    printf("%d\n", b);
}
```

이 위치에 &b를 사용하지 않고
B의 값을 바꾸는 방법에 대해 생각해보기.

정확한 코드를 모르겠다면 아이디어만 제시하세요

```
minibee@argos-edu:~/cedu/week2/hw$ ./hw1
12345
12345
```

과제 2 - My First Hacking

```
#include <stdio.h>

int main()
{
    unsigned int money = 0;
    int salary = 0;

    printf("\nhello, Your Money is %d$\n\n", money);

    printf("===== PAYDAY =====\n");
    printf("BOSS : How much do you want to get paid?\n");
    printf("Me : ");
    scanf("%d", &salary);

    if(salary > 100) {
        printf("BOSS : You're fired!\n");
    } else {
        printf("BOSS : Sure, Good choice\n");
        money += (unsigned int) salary;
    }

    printf("\nYour Money is %u$\n", money);
    if(money > 10000) {
        printf("You Win!\n");
    } else {
        printf("You Lose!\n");
    }
}
```

기다리고 기다리던 월급날이 되었다! 사장님께 돈을 얼마나 달라고 할까?

1. 처음 돈은 0\$ 이다.
2. 입력을 통해 내가 원하는 만큼 월급을 제시할 수 있다.
3. 제시한 금액이 100\$를 넘어가면 해고 당하고, 돈은 하나도 못 받는다.
4. 하지만 내 잔고에는 10000\$ 이상이 있어야 승리한다.

* Type Casting : 변수의 자료형을 바꾸는 것

* unsigned int : 0~4,294,967,295

과제 2 - My First Hacking

```
minibeef@cargos-edu:~/cedu/week2/hw$ ./hw2  
  
hello, Your Money is 0$  
  
===== PAYDAY =====  
BOSS : How much do you want to get paid?  
Me :   
BOSS : Sure, Good choice  
  
Your Money is   
You Win!
```

끝