# 증감 연산자의 위치에 따른 결과값 차이

a++과 같이 증감연산자가 뒤에 붙은 경우에는 변수를 수식에 먼저 적용한 뒤 1을 더하고, ++a 같이 증감연산자가 먼저 나온 경우에는 변수에 1을 먼저 더한 뒤 수식에 적용한다.

a=10

b = ++a; a는 11, b는 11

b = a++; a는 11, b는 10 -----🡪 a가 1이 더해지기전 수식에 적용됐으니 a와 b값이 다르다.

b = a--; a는 9, b는 10 -----🡪 a가 1을 빼기 전에 수식에 적용됐으므로 a와 b값이 다르다.

b = --a; a는 9, b는 9

#증감 연산자 혼자 해보기 4-1 예측

a=5, b=10

a는 초기에 5값이 들어갔지만 a=b++로 인해 10이 되었고, d=++a, a=b + ++d로 인해 다시 23으로 변합니다.

b는 초기에 10값이 들어갔지만 a=b++로 인해 11이 되었고 이후 수식의 영향을 받지 않았기에 결과값도 11로 나왔습니다.

d는 d=++a로 인해 11이 되었고, a=b + ++d 의 식으로 인해 12의 결과값을 출력하였습니다.

#혼자 해보기 4-2

<코딩 중 오류를 찾기 위해 서로 고민한 내용이 있어 추가로 작성하였습니다.>

#define \_CRT\_SECURE\_NO\_VARNINGS  
#include <stdio.h>  
  
int main(void)  
{  
int a, b, c;  
  
printf("삼각형의 밑변과 높이를 입력하시오\n");  
scanf("%d %d", &a &b);  
  
c = a \* b / 2;  
  
printf("삼각형의 넓이는 %d입니다\n", c);  
  
  
return 0;  
  
}

A: 어디가 틀렸는지 한 번 봐주실 수 있나요

B: scanf 괄호 안에 , 가 빠진 것 같은데요

A: 수정해도 오류가 나오는 것 같아요

B: 첫 문장에 철자가 틀렸네요

A: 맞는 것 같아요 실행됩니다

#예제 4-9 변수의 특정 위치 비트 값 알아내기

#define \_CRT\_SECURE\_NO\_WANRNINGS

#include <stdio.h>

int main(void)

{

unsinged char lastmask = 1;

unsigned char firstmask = 128;

unsigned char input;

printf("값을 입력하시오.\n");

scanf("%hhd", $input);

printf("입력한 값 %d의 첫 번째 비트는 %d이다.\n, input, (input & firstmask) > 127);

printf("입력한 값 %d의 마지막 비트는 %d이다.\n", input, input & lastmask);

return 0;

}

<ppt와 책의 내용이 달라 책을 기준으로 대화를 하였습니다>

A: 코딩 내용이 첫 번째와 마지막 비트를 찾는 것 같은데 14번째 문장에서 127보다 크다는 기호는 왜 작성된 걸까요?

B: firstmask가 128이기 때문에 표현해 준 것 아닐까요?

A: 그렇다면 굳이 이 표현을 쓰지 않아도 출력 값에는 이상 없을 것 같지 않나요.

B: 한 번 출력해볼까요. 없어도 출력값은 똑같이 나오네요. 지장 없는 것 같아요. 근데 char 문자열이라 %c를 써도 된다고 알고 있는데 여기서는 hhd를 쓰네요? 그리고 %c를 쓰면 값이 다르게 나오는데 이유가 뭘까요?

A: