

컴퓨터 구조 과제 1

컴퓨터 구조 3분반

소프트웨어학과

32170578 김산

<코드>

```
#include<stdio.h>
#include<limits.h>
int saturating_add(int x, int y){
    int sum = x+y;
    int w = (sizeof(int)<<3)-1; //int 형의 MSB를 확인하기 위한 shift 연산의 피연산자를 구함

    //x,y,x+y 각각의 MSB 반환
    int isX_negative = x >> w;
    int isY_negative = y >> w;
    int isSum_negative = sum >> w;

    //x+y가 양수 또는 음수에서 overflow 인지 판별 overflow 일때 1, overflow 가 아닐때 0 반환
    int isPositive_overflow = (~isX_negative) & (~isY_negative) & isSum_negative;
    int isNegative_overflow = isX_negative & isY_negative & (~isSum_negative);

    isPositive_overflow = isPositive_overflow >> w;
    isNegative_overflow = isNegative_overflow >> w;

    //오버플로일 때 INT_MAX 또는 INT_MIN을 반환, 오버플로가 아닐 때 합을 반환
    return (((~isPositive_overflow)&(~isNegative_overflow)&sum))
    + (isNegative_overflow & INT_MIN) + (isPositive_overflow & INT_MAX);
}
int main(){
    //test case 1 : positive overflow
    printf(" 2147383647 + 100 : %d\n", saturating_add(2147383647,100));
    //test case 2 : negative overflow
    printf("-2147383646 - 100 : %d\n", saturating_add(-2147383646,-100));
    //test case 3 : not overflow (pos + pos)
    printf(" 100 + 267          : %d\n", saturating_add(100,267));
    //test case 4 : not overflow (neg + neg)
    printf("-184 + -267         : %d\n", saturating_add(-184,-267));
    //test case 5 : not overflow (neg + pos)
    printf("-354 + 777          : %d\n", saturating_add(-354,777));

    return 0;
}
```

<실행결과>

```

  TERMINAL  PROBLEMS  OUTPUT  DEBUG CONSOLE

2147383647 + 100 : 2147383747
-2147383646 - 100 : -2147383746
100 + 267      : 367
-184 + -267    : -451
-354 + 777     : 423
[1] + Done                                           "/usr/bin/gdb" --int
san@DESKTOP-ORAM3EQ:/mnt/c/workspace$
```

실행환경

(WSL)Ubuntu - 20.04LTS

GNU gdb (Ubuntu 9.1-0ubuntu1) 9.1

<결과보고>

교재의 Bit-Level Integer Coding Rules에 따라 코드를 작성하였습니다.

두 2의 보수의 합을 연산하는 과정에서 overflow가 일어나더라도 overflow가 일어난 값을 반환하지 않고 부호에 따라 TMAX 또는 TMIN의 값을 반환해야 한다. 먼저 두 수를 x, y라 하고 두 수와 두수의 합의 MSB를 구한다. 만약 x의 MSB가 1 y의 MSB가 1 합의 MSB가 0이면 이는 negative overflow가 일어난 것으로 알 수 있다. 반대의 경우 x의 MSB가 0 y의 MSB가 0 합의 MSB가 1이면 positive overflow가 일어난 경우이다. 이를 이용한 비트 연산을 통해 positive overflow가 일어나면 INT_MAX를 반환하고 negative overflow가 일어나면 INT_MIN을 반환한다. overflow가 일어나지 않으면 두수의 합을 반환한다.