[CSED211] 컴퓨터 SW시스템 개론  
Lab Session Howework #2

학번 : 20180551

학과 : 컴퓨터공학과

이름 : 이준석

<과제에 사용하게 될 수업 부분>

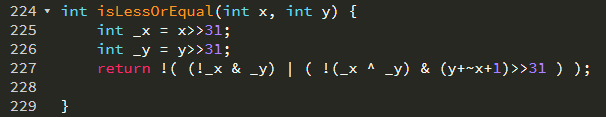
1. 정수형
2. 실수형

* 실수형은 실수를 (-1)^S \* M \* 2^E라고 표현하고, 거기에서 S, M, E을 이진수로 변환하여 32비트에 저장한다. S는 부호를 나타내는 0과 1이다. 그러므로 한 비트로 할당된다. M의 소수점 아래의 부분을 23비트까지만 가져와서(만약 23비트 초과의 수라면 rounding을 진행한다.) frac 부분에 저장한다. 마지막으로 E = exp – bias(단, bias = 2^(k-1) -1) 라는 식을 통해 exp를 구해서 8비트의 exp 부분에 저장한다.
* Exp가 전부 0이고 frac도 전부 0이면 그 값은 0이다.(sign bit에 따라 +와 - 2개가 존재한다.)
* Exp가 전부 0이고 frac은 0이 아니면 0에 아주 가까운, denormalized value가 된다.
* Exp가 전부 1이고 frac은 0이면 무한대를 나타낸다. ( 이것도 +와 -가 있다.)
* Exp가 전부 1이고 frac은 0이 아니면 Not a Number, 즉 수가 아니게 된다.

**Problem 1**

매개변수가 x, y의 순서로 주어진다고 할 때,

x <= y 을 만족하면 1 을 반환하고 아니라면 0을 반환하는 함수를 구현하는 문제이다.



x와 y의 msb를 고려하기 위해 오른쪽으로 31칸 이동시켰다. (line 225~226)

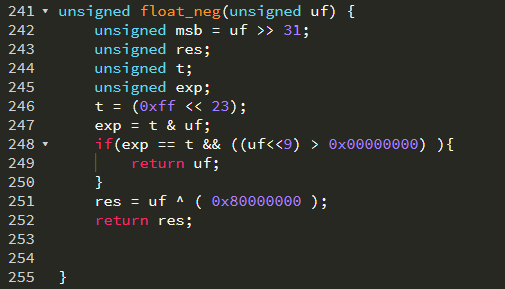
(x <= y)는 (x > y)의 부정과 같다는 것을 이용하였다.

‘x가 양수이고(\_x가 000….) y가 음수이다(\_y가 111….) 혹은 서로 부호가 같고 y에서 x를 뺀 값이 음수이다(1111….)’ 라는 조건을 만족하는 것이 x > y이다. 그것의 부정은 바로 x <= y이다.

**Problem 2**

절대값은 같고 부호는 반대인 float을 구하는 문제이다.

NaN이라면 입력된 값을 그대로 반환한다.



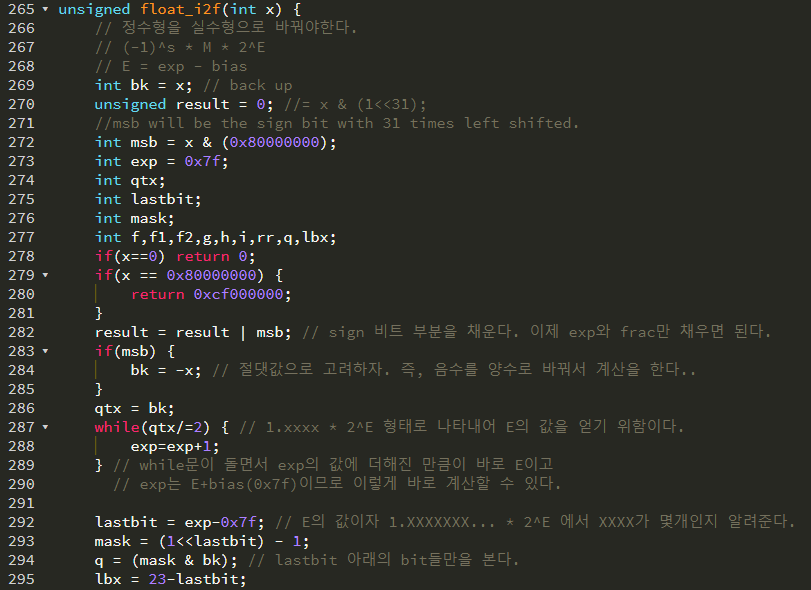
exp 부분이 11111111 이고 frac부분이 0보다 크다면 그것이 바로 NaN이다. 그 때 입력을 그대로 반환한다. (line 248~250)

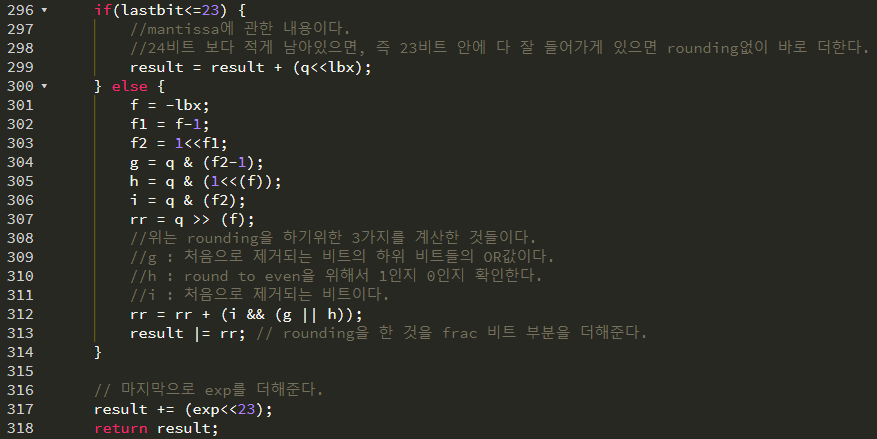
NaN이 아니라면 가장 상위의 sign bit만 변환하면 된다. (line 251)

**Problem 3**

Int형을 float형으로 변환하는 함수를 구현하는 문제이다.

int형의 수를 float형으로 변환하기 위해 (-1)^S \* M \* 2^E에서의 S, M, E를 찾아야 한다.





예외 처리를 한다. 0은 그대로 0으로 반환하고 tmin과 tmax의 절댓값 차이로 인해 표현이 불가능한 tmin을 예외적으로 따로 연산하여 반환한다. (line 278~281)

1000…. 과의 & 연산을 통해 msb(S값)를 구하고 그것을 0000….에 | 연산으로 더해준다.(line 272, line 282)

원래 수의 절대값을 취한다. (line 283~285)

절대값을 취한 수를 2로 나누며 exp의 값을 구한다. 한 번 나눠질 때마다 exp를 0x7F(bias값)부터 하나씩 증가시키면 구할 수 있다. (line 273, line 287~289)

매개변수로 입력받은 수를 이진수로 표현했을 때, 소수점 아래의 비트가 23개 이하라면 rounding없이 바로 frac으로 더해주고 23개 초과라면 rounding을 해준 뒤에 frac으로 더해준다.(line 296~314)

마지막으로 exp를 더해주면 float형으로의 변환이 완료된다. (line 317)

**Problem 4**

어떤 float값을 2배해서 반환하는 함수를 구현하는 문제이다.

값을 두 배로 만들기 위해서는

1. Exp 부분이 000… 인 경우(zero 혹은 denormalized)는 frac을 왼쪽으로 한 비트 이동시킨다. ( frac을 한 비트 왼쪽으로 이동시키면 M의 값이 2배가 되므로 가능하다. )
2. Normalized value는 exp에서 1을 더해주면 된다. ( E = exp – bias 이므로 exp가 1 증가하면 E도 1증가한다. 그러면 (-1)^s \* M \* 2^E에서 결국 2배가 되는 것이다.)

Exp가 11111111인 경우는 원래 값을 그대로 반환한다.

