Danny Tan

1. Int compare() {

return !(x^y);

}

1. void set\_bits(unsigned int x, unsigned int l, unsigned int r) {

x |= (4,294,967,295 << l) & (4,294,967,295 >> (31-r));

}

1. It counts the number of 1s in the binary bit.
2. It will be 2N because 2^N \* 2^N will gives 2^(2N) and adding 2^N will not increase the bit so it will just be 2N.
3. a. 2^(exponent-1)-1 = 2^(3-1)-1 = 3

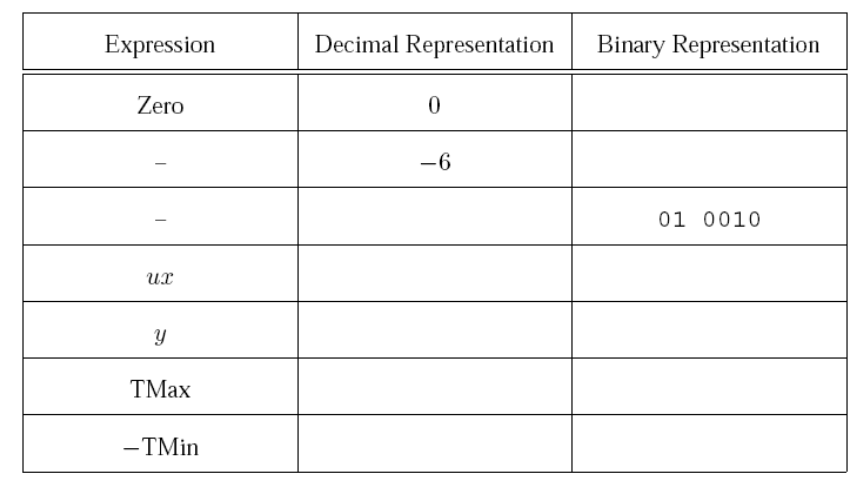
b. The lowest number will be 00001 where 0 is the sign, 001 is the exponent, and 1 is the fractional. So e will be 1-3 = -2 M will be 0.5 and the number will be 2^-2\*1\*0.5 = 0.125

c. The largest number will be 01101 where 1 is the sign, 110 is the exponent, and 1 is the fractional. So e will be 6 – 3 = 3 and m will be 1.5 and the answer will be 2^3\*1.5\*1 = 12.

1. a. y = (x << 3) - x

b. y = (x << 5) – (x << 2) - x

c. y = (x << 6) + (x << 1) + x



10 0000

-32

01 1111

31

101

-3

10 1111

47

18

11 1010

00 0000

1. a. 32

b. 1 << 5

c. x |= 0x00000020