INTRODUCTION TO BIT MANIPULATION

✯ Decimal to Binary Conversion

Divide number repeatedly by 2 and concatenate all remainders from last to first.

Pseudocode :

```
convertToBinary(N) {
    res = ""
    while (N != 1) {
        if (N % 2 == 1) {
            res += '1'
        } else {
            res += '0'
        }
        N = N/2
    }
    reverse(res)
    return res
}
```

✯ Convert Binary to Decimal

Multiply each digit with 2 raised to its weight and add them.

Pseudocode :

```
convertToDecimal(N) {
    l = N.size();
    p2 = 1;
```

```
    num = 0 ;
    for(i = l-1 → 0) {
        if (N[i] == '1') {
            num += p2 ;
        }
        p2 = p2 × 2 ;
    }
    return num ;
}
```

✯ Binary Operators

AND : 1 , 1 → 1
0 , 1 → 0
0 , 0 → 0
1 , 0 → 0

OR : 1 , 1 → 1
0 , 1 → 1
1 , 0 → 1
0 , 0 → 0

XOR : 1 , 1 → 0
0 , 0 → 0
1 , 0 → 1
0 , 1 → 1

RIGHT SHIFT : Shift all binary digits to right by once.

Eg : 0000 ···· 01101 >> 1
= 0000 ···· 00110

In decimal, this is just division by 2^K. K shifts to the right means division by 2^K in decimal.

LEFT SHIFT: Exact Opposite of Right Shift. Each shift to the left means a multiplication by 2 this time. Since everything moves 1 place to the left, the rightmost or the most significant bit gets filled with 0.

☆ How is Negative Numbers Stored

In a 32 bit integer, Least Significant Bit is responsible for telling sign. 0 means positive and 1 means negative

So to store $-x$ as an integer, we take 2's complement of x.

☆ NOT Operator

Reverses all bits (0's to 1's and 1's to 0)

Might also cause the sign bit to change, so we take a 2's complement if its a negative