Bit Manipulation

#include<bitset>

To print a number in binary form : cout<<bitset<32>(x);

Bit-Hacks:

1. Add 1 to a given integer

-~x

Since –x=~x+1;

-~x=x+1;

2. Swapping of two numbers using XOR

If(x!=y)

{

X=x^y;

Y=x^y;

X=x^y;

}

3.Turn off k’th bit

n&~(1<<(k-1))

4.Turn on k’th bit

n|(1<<(k-1)

5.Check k’th bit set or not

n&(1<<(k-1)) = if it non zero then set otherwise not set

6.Toggle k’th bit

n^(1<<(k-1))

7.To unset the rightmost bit in ‘n’

n&(n-1)

8.Find the position of rightmost set bit

n=n^(n&(n-1))

while(n)

{ n=n>>1; pos++;}

9.Convert uppercase to lowercase

Take OR with space

ch | ‘ ‘

10. Convert lowercase to uppercase

ch&(‘\_’)

11.Invert alphabet’s case : Take XOR with space

12.Find letter’s position

‘A’&31 returns 1

* Problem -1 :

Number of bits to flip to change one number to another

Take xor of these two numbers and then find total number of 1’s in its binary form.

\_builtin\_popcount(n) =>returns total no. of set bits

Link : https://medium.com/techie-delight/bit-manipulation-interview-questions-and-practice-problems-27c0e71412e7