**Explanation done in meeting 1-**

**How to take infinite input**

If we don’t know how many integers are given for input

int x;

while(cin>>x)

{

}

If it is given that -1 is present at the last of input

while(true)

{

      int x;

      cin>>x;

      if(x==-1)

            break;

}

**Fast I/O**

ios\_base::sync\_with\_stdio(0);

cin.tie(0);

cout.tie(0);

**Prime number**

Segmented sieve

Given l, r as ranges and we have to find the prime number that lie between l and r

max value of r = 10^12

r-l <= 10^6

n= r-l+1;

a[n];

l, l+1, l+2, .. , r

0, 1, 2,       , n-1

for(int i=0;i<n;i++)

{

       x = i+l;

for(int j=0;j<v.size();j++)

{

       if(x%v[j] ==0)

      {

             f=0;

             break;

       }

}

}

**Bitwise operators**

Consider 2 integers a and b

a= 9 1001

B = 14 1110

**or**  (|) A|b =   1111 15

**And** (&) a&b =  1000 8

**Not** (~) A      = 0110   6

**Xor** (^) a^b  =   0111 7

**<<** (\*2) a<<1 10010

**>>** (/2) a>>1 0100

Some properties of these operators

1 | n = 1

0 | n = n

n | n = n

0 & n = 0

1 & n = n

n & n = n

1 ^ n = ~n

0 ^ n = ~n

n ^ n = 0

How to find whether a number is power of 2 or not

while(n%2 ==0)

{

N = n/2;

}

if(n==1)

Return true;

Else

Return false;

If we have to find this in O(1) constant time

N-1 0111111

N 1000000

N&(N-1) 0000000

N^(N-1) 1111111 problem is that we also have to find this number

n&(n-1) ==0

n^(n-1) == ()

To count number of set bits

cout<<\_\_builtin\_popcount(n);

Read more builtin function from gfg

Parity - count number of set bits in a number and find it’s parity accordingly

Even = 0

Odd  = 1

Some STL functions used in number theory

min(a,b)

min(a,min(b,c))

max(a,b)

\_\_gcd(a,b)

LCM = a\*b/\_\_gcd(a,b)