# Number Complement -

$5 \rightarrow$ 0000　00000　00000　000101

$\sim 5$ 11111　11111　11111　111010

interchange 0 and 1

& mask $\rightarrow$ 00000　00000　00000　000

to change 1 to 0.

last set bit ka count pta krna h

Mask kaise milega?

000⟨101⟩

↑↑↑

0000 -- ---- 000 ⟨111⟩

$n = 0$; 00000 ----- 00 << Left shift once, you get one '1' at end

000 -- ~ 001 << Repeat left shift.

200 --- .0011

⟨while (mask < num)⟩ →

① $5 \rightarrow$ 000 ------- 00101

$\sim 5 \rightarrow$ ⟨111 --- ---- 11010⟩

mask $\rightarrow$ ⟨000 ----- 00⟩ 111 & ⟩

000 -- -- 00 ---- 010

→ ②

Code: →

```
int findComplement (int num)
{
    int mask = 0;
    while (mask < num)
        mask = (mask << 1) | 1;
    }
    int ans = (~num) & mask;
    return ans;
}
```

## # Power of Two Submissions-

### 1st Approach-

$n = 16 \rightarrow 2^4 \rightarrow 2 \times 2 \times 2 \times 2 \rightarrow$ true
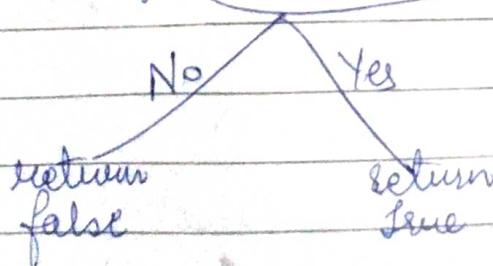$\quad 15 \rightarrow 2^x$ X $\rightarrow$ false.

$\qquad\qquad\qquad\qquad$ set bit

$2^0 \rightarrow 1 \rightarrow 0000|\boxed{1}|$
$2^1 \rightarrow 2 \rightarrow 000\boxed{1}0$
$2^2 \rightarrow 4 \rightarrow 0000\boxed{1}00$
$2^3 \rightarrow 8 \rightarrow 0000\boxed{1}000$
$2^4 \rightarrow 16 \rightarrow 0000\boxed{1}0000$
$2^5 \rightarrow 32 \rightarrow 0000\boxed{1}00000$

I/P num → count no. of set bit

$\qquad\qquad$ ↓ if (count == 1)

Also, if $(n <= 0)$ $\qquad$ No $\qquad$ Yes
return false. $\qquad\qquad$

$\qquad\qquad$ return $\qquad\qquad$ return
$\qquad\qquad$ false $\qquad\qquad$ true

# How to count set bits?

$$5 \rightarrow 0000\ 000\ 10\ \boxed{1}$$

isko &1 kro then

if &1 → 1 → set bit

&1 → 0 → not set bit

to check next bit —

>>

$$0000\ 00\ 1\boxed{0} \quad \&1$$

>>

$$00\ 0\ 00\ 00\boxed{1} \quad \&1$$

>>

- - - - - - - continue

until no. is 0.

## 2nd Approach —

$$n = 2^1$$
$$= 2^2$$
$$= 2^3 \rightarrow 8 \rightarrow 1000$$

let $n = 2^x \rightarrow 1000$

$(n-1)$ is always $\rightarrow \dfrac{0111}{0000}$

if $(n)\ \&\ (n-1)$

↳ → 0 → then n is $2^n$ → true

↳ → 1 → false

```
if (n <= 0)
  return false;
int ans = n & (n-1);
return (ans == 0);
```

# Approach without converting to Binary −

```
                            int p=0;
(b)     while(n < num)
        {
            n = pow(2,i)
            if(n == ans)
                return true;
            i++;
        }
```

H/W → 1) Complete previous pattern assignment.
      2) Try Reverse integer Question.

```
int reverse(int x)
{
    int ans =0;
    while(x! = 0)
    {
        int digit = x % 10;
        if((ans > INT_MAX/10) || (ans > INT_MIN/10)
        {
            return 0;
        }
        ans = (ans *10) + digit;
        x = x/10;
    }
    return ans;
}
}
```