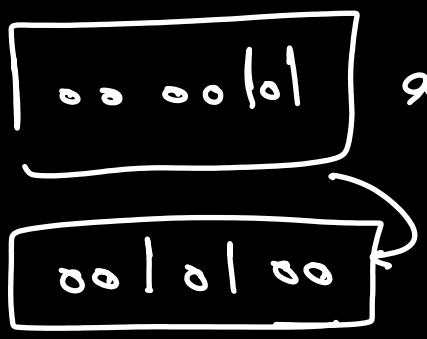
" Bit manipulation - techniques "

① Bitwise operators :→

* left Shift :-

ex :- $5 << 2 \Longrightarrow$ = multiply 2^2

↳ 5 in Binary



after shifting two times to left

* right shift :-

equal to devide by 2^x

00101 ⟶ 00001

the original number is 10 after right shift by one towards right means $10 / 2^{①}$ ← the no of shifted bits

---

* Check old or Even :-

if the last Bit is one means that the number is odd, else if the last number is 0 means the Num is even.

* I will Perform anding with 1 and if the output is one mean the number is odd, if it gives Zero means the number is even.

$$\begin{array}{r} 10\ 11\ 0\ 111 \\ 00\ 00\ 0\ 001 \\ \hline 00\ 00\ 0\ 001 \end{array} \quad \&$$

*Get ith Bit:—

5 ⟹ 0 0 0 0 1 0 1

and I want to get the third Bit, how to do?

$[5 \& (1 << i)] > 0$ ⟹ 1 / 0

(5)	0	0	0	0	1	0	1
(1)	0	0	0	0	0	0	1
1<<2	0	0	0	0	1	0	0
	0	0	0	0	1	0	0

← This number is greather than one, so the origin Bit is one not Zero.

---

*Set intBit:—

Simply by performing or we can set the ith Bit:

0 1 0 1 0 ← if I want this Bit to be one I will perform or with one

0 0 0 1 0 0

0 0 0 1 1 0

---

*Clear last i Bits:→

n=5

1 1 1 1 0 1 0 1 → 1 1 1 0 0 0 0 0

I want the last five Bits to equal 0

① mask = ~(0) << i ;

② then Perform and with the original number

```
111 00000
1111 0101 8
--------------
111 00000
```

\#

---

* Power of two:—

if (N & N-1) == 0, then N is Power of two

```
  1 0000   => 16
8
  0 1111   => 15
  ---------
  0 0000
```

=> then N which is 16 is a Power of two

---

* fast exponentiation:→

$a^5 = 3^5 = 3^{101}$

n = Power = 101

① last Bit is 1 or 0 ? this can be done by Performing (1&a).

② if it is 1 => (1*a)

③ a = a * a.

④ Shift the number towards right By one.

```cpp
int fastExpo(int a,int n){
    int ans = 1;

    while(n>0){
        int last_bit = (n&1);
        if(last_bit){
            ans = ans * a;
        }
        a = a*a;
        n = n>>1;

    }
    return ans;
}
```

1 0 1

$1*a * 1 \times a^4 = a^5 \Rightarrow 3^5$ # This is the answer.

*Decimal to Binary Format:→

I want to Convert 13 to Binary

			Modulus
①	2	13	①
	2	6	0
	2	3	1
	2	1	1
		0	

⇒ 1101

⇒ The logic will be getting the modulus then multiply it by 10 ① → 0 → ∞(+)

how to get the answer in int Format

1 0 1 1

$(1 \times 10^0) + (0 \times 10^1) + (1 \times 10^2) + (1 \times 10^3)$

$1 + 0 + 100 + 1000$

$= 1101$

#