SCALER

Bit Manipulation - 1

TABLE OF CONTENTS

Bit-wise Operators : &, |, ^, ~, <<, >>

same same puppy shame

a	b	a&b	a|b	a^b	~a
0	0	0	0	0	1
0	1	0	1	1	1
1	0	0	1	1	0
1	1	1	1	0	0

Basic Properties

$a \& 1 == 1 \rightarrow$ a is odd
$a \& 1 == 0 \rightarrow$ a is even.

1. **Even / Odd Number** →

11	→	1	0	1	1
&		0	0	0	1
		0	0	0	1

$\boxed{11 \& 1 \rightarrow 1}$

10	→	1	0	1	0
&		0	0	0	1
		0	0	0	0

$\boxed{10 \& 1 \rightarrow 0}$

$a \& 1$
- $\xrightarrow{=1}$ 0^{th} bit is 1 → a is odd
- $\xrightarrow{=0}$ 0^{th} bit is 0 → a is even

2. **A & 0** → 0

a	→	1	0	1	1	0	1	1
0	→	0	0	0	0	0	0	0
		0	0	0	0	0	0	0

&

3. **A & A** → A

```
   1 0 1 1 0 0 1
&  1 0 1 1 0 0 1
   -------------
   1 0 1 1 0 0 1
```

4. **A | 0** → A

```
A →  1 0 1 1 0 0 1
0 →  0 0 0 0 0 0 0
     -------------
     1 0 1 1 0 0 1
```

5. **A | A** → A

A → 1 0 1 1 0 0 1

A → 1 0 1 1 0 0 1

A → 1 0 1 1 0 0 1

6. **A ^ 0** → A

A → 1 0 1 1 0 0 1

0 → 0 0 0 0 0 0 0 ^

1 0 1 1 0 0 1

7. **A ^ A** $\rightarrow 0$

$$\begin{array}{lccccccc} A = & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ A = & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ \hline & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \;\; \wedge$$

Xor of two same values will result in 0.

Commutative Property →

$A \& B = B \& A$

$A | B = B | A$

$A ^\wedge B = B ^\wedge A$

Associative Property →

$(A \& B) \& C = A \& (B \& C)$

$(A | B) | C = A | (B | C)$

$(A ^\wedge B) ^\wedge C = A ^\wedge (B ^\wedge C)$

< Question- 1 > : Evaluate the expression: **a ^ b ^ a ^ d ^ b**

$$\underbrace{a \wedge b \wedge a \wedge d \wedge b}$$

$$\downarrow$$

$$\underbrace{(a \wedge a)} \wedge \underbrace{(b \wedge b)} \wedge d$$

$$\downarrow \quad \downarrow$$

$$\underbrace{0 \wedge 0} \wedge d$$

$$\downarrow$$

$$\underbrace{0 \wedge d}$$

$$\downarrow$$

$$\underline{\underline{d}}$$

< Question- 2 > : Evaluate the expression: **1 ^ 3 ^ 5 ^ 3 ^ 2 ^ 1 ^ 5**

$$\underbrace{1 \wedge 3 \wedge 5 \wedge 3 \wedge 2 \wedge 1 \wedge 5}$$

$$\downarrow$$

$$1 \wedge 1 \wedge 3 \wedge 3 \wedge 5 \wedge 5 \wedge 2$$

$$\downarrow$$

$$0 \wedge 2$$

$$\downarrow$$

$$2$$

< Question > : Given arr[N] where every element is present twice except one unique element. Find that unique element.

arr[] → [2 7 2 7 4 6 4]

idea → take xor of all the elements.

code →

```
ans = 0
for( i = 0; i < N; i++) {
        ans = ans ^ arr[i];
}
return ans;
```

ans = ~~0~~
~~2~~
~~5~~
~~7~~
~~0~~
~~4~~
~~2~~
6

$\boxed{\begin{array}{l} T.C \to O(N) \\ S.C \to O(1) \end{array}}$

Assumption → 1 int → 8 bits.

Left Shift Operator (<<)

a =	0	0	0	0	1	0	1	0	→	10
a << 1	0	0	0	1	0	1	0	0	→	20
a << 2	0	0	1	0	1	0	0	0	→	40
a << 3	0	1	0	1	0	0	0	0	→	80
a << 4	1	0	1	0	0	0	0	0	→	160
a << 5	0	1	0	0	0	0	0	0	→	~~320~~

×2 ×2 ×2 ×2 ×2

64

overflow.

The maximum no. that can be stored in 8 bits → 255

∴ It is not possible to store 320 in 8-bits.

$a << n \Rightarrow a \times 2^n$ [careful with overflow]

$1 << n \Rightarrow 2^n$

Right Shift Operator (>>)

a =	0	0	0	0	1	0	1	0	→ 10
a >> 1	0	0	0	0	0	1	0	1	→ 5
a >> 2	0	0	0	0	0	0	1	0	→ 2
a >> 3	0	0	0	0	0	0	0	1	→ 1
a >> 4	0	0	0	0	0	0	0	0	→ 0

x

$$a >> n \Rightarrow \frac{a}{2^n}$$

$$1 >> n = \frac{1}{2^n}.$$

Power of Left Shift Operator

set → 1
unset → 0

1. OR Operator → set the i^{th} - bit

	4	3	2	1	0
N →	1	0	1	1	0
1<<3 →	0	1	0	0	0
	1	1	1	1	0

	4	3	2	1	0
N →	1	1	0	1	0
1<<3 →	0	1	0	0	0
	1	1	0	1	0

$$N = N \mid (1 \ll i)$$ ⇒ setting i^{th} - bit in N.

2. AND Operator → check for i^{th} bit → set/unset

	5	4	3	2	1	0
N →	1	0	1	1	0	1
1<<3 →	0	0	1	0	0	0
	0	0	1	0	0	0

&

	5	4	3	2	1	0
N →	1	0	1	1	0	1
1<<4 →	0	1	0	0	0	0
	0	0	0	0	0	0

&

$N \,\&\, (1 \ll i)$
- $== 0$ → i^{th} bit is unset in N
- $!= 0$ → i^{th} bit is set in N

3. XOR Operator → toggle i^{th} bit

$$
\begin{array}{lccccccc}
 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\
N \rightarrow & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\
1<<3 \rightarrow & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
\hline
 & 1 & 0 & 1 & 0 & 0 & 0 & 1
\end{array}
\quad \wedge
$$

$$
\begin{array}{lccccccc}
 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\
N \rightarrow & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\
1<<2 \rightarrow & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
\hline
 & 1 & 0 & 1 & 1 & 1 & 0 & 1
\end{array}
\quad \wedge
$$

$\boxed{N = N \wedge (1<<i)}$ → i^{th} bit will be toggled

→ How to unset i^{th} bit of N ?

```
if( (N & (1<<i)) == 0 ){
        //ith bit is 0
}
else{
        N = (N ^ (1<<i));
}
```

1 int → 4B → 32 bits.

< Question > : Given an integer N. Count the set-bits in N.

Example :

N = 12 →	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
	31																						9	8	7	6	5	4	3	2	1	0

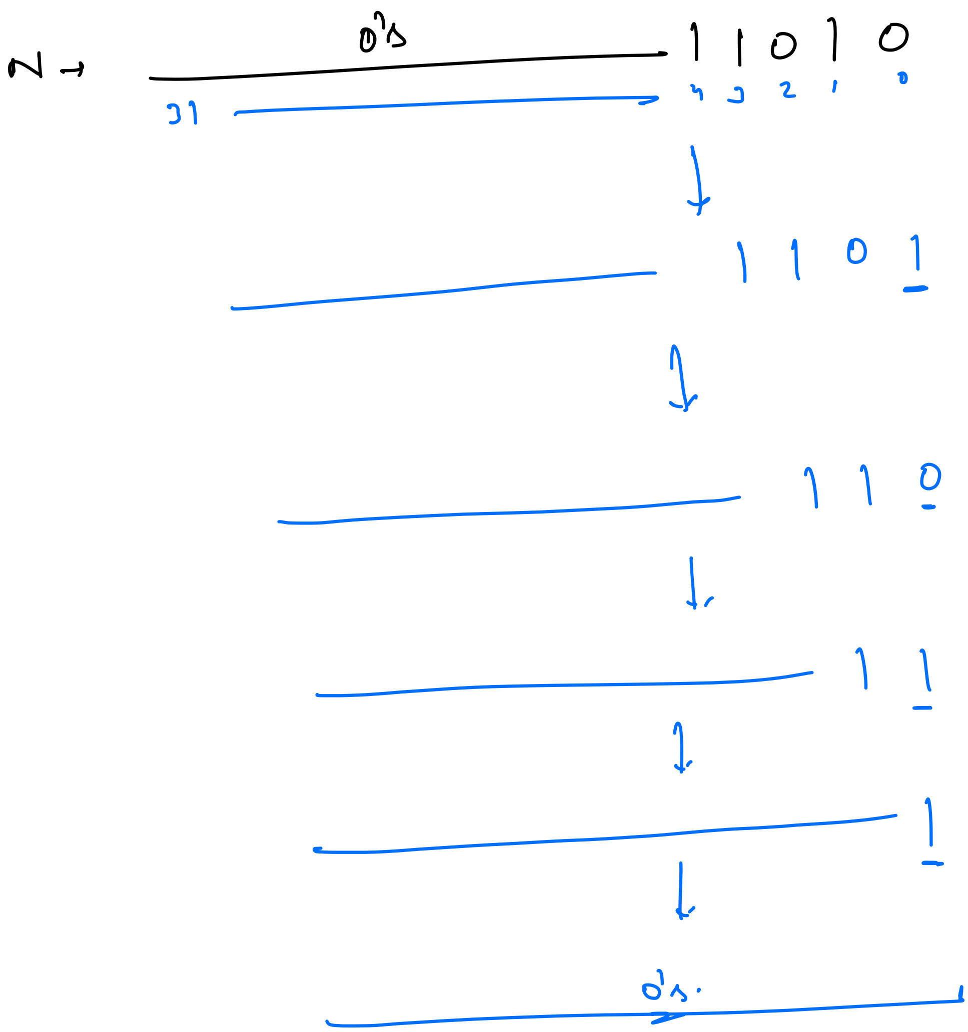
code →

```
count = 0;

for( i = 0;   i < 32;   i++){
|     if( (N & (1<<i))  != 0 ) { // ith bit is set
|     |        count++;
|     }
}

return  count;
```

$$\begin{bmatrix} T.C \rightarrow O(1) \\ S.C \rightarrow O(1) \end{bmatrix}$$




```
count = 0
while( N != 0){
    if( (N & 1) == 1){
        count++;
    }
    N = (N >> 1);   // N = N/2
}
```

Both are same

At max 32 iterations

T.C → O(1)
S.C → O(1)

< Question > : Given three integer - A, B, C

$$\underbrace{\text{0's}}_{A}\ \underbrace{\text{1's}}_{B}\ \underbrace{\text{0's}}_{C} \rightarrow \text{Decimal no.}$$

↳ set bits from C to B+C-1

Example : A = 4, B = 3, C = 2 1 ≤ A, B, C ≤ 20

0 0 0 0 1 1 1 0 0 → 28

Eg - A = 5, B = 4, C = 3 → 120

0's — 0 0 0 0 0 1 1 1 1 0 0 0
11 10 9 8 7 6 5 4 3 2 1 0

A = 5, B = 4, C = 3

code :-

```
ans = 0;
for( i = C; i < B+C ; i++) {
    ans = (ans | (1<<i));
}
return ans;
```

0 — 1 1 1 1 0 0 0
31 — 6 5 4 3 2 1 0

[T.C → O(B)
S.C → O(1)]

$$\underbrace{\qquad\qquad\qquad}_{A}^{1's}\ \underbrace{\qquad\qquad}_{B}^{0's}\ \underbrace{\qquad}_{C}^{1's}$$

A = 4, B = 2, C = 3

$\overbrace{\qquad\qquad}^{0's}$ 1 1 1 1 0 0 1 1 1

8 7 6 5 4 3 2 1 0

```
for( i=0; i<C; i++){
|       //set the bit
}

for( i = B+C ; i< B+C+A; i++){
|       //set the bit
}
```