Bit manipulation

★ Basics of Bit Manipulation

- → Number System & Conversions
- → Addition of two binary Numbers
- → Bitwise operators
- → Negative Numbers
- → Range of no.

⇒ DOUBTS

GET READY WITH YOUR PEN & PAPER

1> Number System

Way of writing to express the number

A> Decimal no. System

↳ base = 10

↳ It is called base 10 because single digit in this no. System can be from 0 to 9

↳ The different position of each digit represents power of 10

Ex:- $[7\ 3\ 2]_{10}$

$\Rightarrow 7\times10^2 + 3\times10^1 + 2\times10^0$

$\Rightarrow 700 + 30 + 2$

$\Rightarrow 732$ Ans

Ex:- $[2\ 4\ 6\ 3]_{10}$

$\Rightarrow 2\times10^3 + 4\times10^2 + 6\times10^1 + 3\times10^0$

$\Rightarrow 2000 + 400 + 60 + 3$

$\Rightarrow 2463$

B:> Binary no System

↳ Base = 2

↳ Called base 2, because each digit in this no. system can be 0 or 1

↳ The position of each digit represents power of 2

Ex:- $[0\ 1\ 1\ 0]_2$ (positions: $0 \to 2^3$, $1 \to 2^2$, $1 \to 2^1$, $0 \to 2^0$)

$\Rightarrow 0\times2^3 + 1\times2^2 + 1\times2^1 + 0\times2^0$

$\Rightarrow 0 + 4 + 2 + 0$

$\Rightarrow 6$ Ans

Ex:- $[1\ 0\ 0\ 1]_2$ (positions: $1 \to 2^3$, $0 \to 2^2$, $0 \to 2^1$, $1 \to 2^0$)

$\Rightarrow 1\times2^3 + 1\times2^0$

$\Rightarrow 8 + 1$

$\Rightarrow 9$ Ans

Q why we need binary no. System?

↳ All Computers / Machines understand binaries.

Binary to Decimal no.

Ex $[1\ 0\ 0\ 0]_2 \Rightarrow 1 \times 2^3$

$\Rightarrow [8]_{10}$ Ans

$2^3\ \ 2^2\ \ 2^1\ \ 2^0$

Observation :- We need to start from rightmost.

Quiz! :- $[1\ 0\ 1\ 1\ 0\ 1\ 0]_2$ (positions 6 5 4 3 2 1 0)

$\Rightarrow 1 \times 2^6 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^1$

$\Rightarrow 64 + 16 + 8 + 2$

$\Rightarrow 90$ Ans

IDEA

- Extract the last digit 1 by 1
- Multiply the rem with correct base value
- Add it to the Ans.

PSEUDO CODE ⇒ H.W.

I Decimal to Binary

$[20]_{10} \rightarrow (10100)_2$

		↓ Rem
2	20	
2	10	0
2	5	0
2	2	1
2	1	0
	0	1

$* 10^0 \Rightarrow 0$

$* 10^1 \Rightarrow 0$

$* 10^2 \Rightarrow 100$

$* 10^3 \Rightarrow 0$

$* 10^4 \Rightarrow 10000$

10100

quiz 2 :- $[45]_{10} \rightarrow (\ ??\ )_2$

2	45	
2	22	1
2	11	0
2	5	1
2	2	1
2	1	0
	0	1

Ans : 101101

(II) → (A) ADDITION OF 2 Decimal NOs.

9/10	12/10	5/10	
0	(1)	(0)	0
0	7	8	2
0	1	4	3
0	9%10	12%10	5%10
	9	2	5

dig = Sum %10
Carry = Sum /10

← Ans

(B) Addition of 2 Binary Nos.

Sum binary

Sum	binary
0	00
1	01
2	10
3	11

carry ↑ digit ↑

Ex

2/2	2/2	2/2	
(1)	(1)	(1)	(0)
0	1	0	1
0	0	1	1
1%2	2%2	2%2	2%2
1	0	0	0

digit = Sum%2

Carry = Sum/2

Ans ⇒

Ex:-

1	0	0	1	0	0	0	
0	1	1	0	1	0	1	
0	1	0	0	1	1	0	
1	0	1	1	0	1	1	← Ans

IDEA :- 89 + 143

Sum/2 ↗ Sum %2 ↗

	n_1	n_2	$n_1\%10$ d_1	$n_2\%10$ d_2	carry	d_1+d_2+c Sum	S%10 rem	ans
/10 ↘	89	143	9	3	0	12	2	$2*10^0$
/10 ↘	8	14	8	4	12/10	13	3	$3*10^1$
/10 ↘	0	1	0	1	13/10	2	2	$2*10^2$
/10 ↘	0	0	0	0	2/10	0	0	$0*10^3$
								232

PSEUDO CODE ⇒ H.W

~~III~~ BITWISE OPERATORS

Used to perform operation on binary no. bit by bit.

0 → false / unset
1 → true / set

Bitwise operators

- → AND (&)
- → OR (|)
- → XOR (^)
- → NOT (!/~)

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

A & B = both the bits are Set then only resultant bit will be Set

A | B = Any one bit is Set then Ans bit will also be Set

A ^ B = Bit will be Set for opposite inputs bits

Eg 5 & 6 = 4

```
  5 - 1 0 1
& 6 - 1 1 0
-------------
    1 0 0
```

Eg 20 & 45 = 4

```
 20 : 0 1 0 1 0 0
& 45 : 1 0 1 1 0 1
-------------------
    0 0 0 1 0 0
```

Eg 92 ^ 154

```
  92 : 0 1 0 1 1 1 0 0
^ 154 : 1 0 0 1 1 0 1 0
-----------------------
    1 1 0 0 0 1 1 0
```

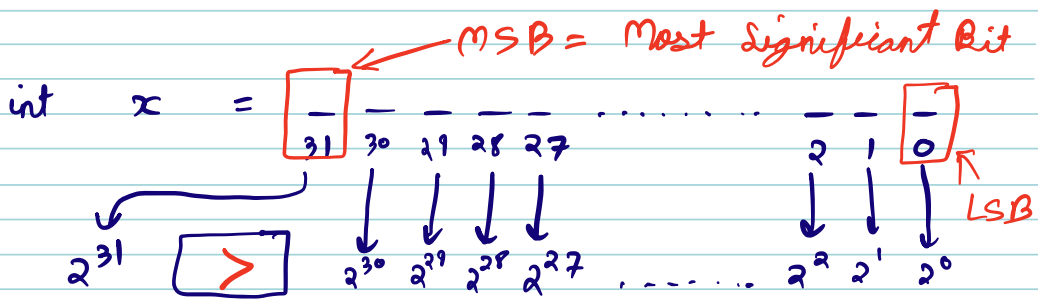
Quiz 4 :-

```
    20 :   0 1 0 1 0 0
∧   45 :   1 0 1 1 0 1
-----------------------
           1 1 1 0 0 1
```

10 : 45

IV Binary Representation of -ve nos.

$[-45]_{10} = (\ ??\ )_2$



$= 2^{30} + 2^{29} + \ldots\ldots\ldots + 2^2 + 2^1 + 2^0$

GP, where $a = 1$

$r = 2$

$n = 31$

Sum of GP $= \frac{a(r^n - 1)}{r - 1}$

$= \frac{1(2^{31} - 1)}{2 - 1}$

$= 2^{31} - 1$

Hence, MSB(2^{31}) > Sum of all lower terms

* MSB → 0, positive number
  MSB → 1, -ve number

$(-45)_{10} = (??)_2$

Assumption int ⟶ 8 bits → Just for Explanation

$(45)_{10} = 0\ 0\ 1\ 0\ 1\ 1\ 0\ 1$

★ -ve nos. are stored as 2's compliment.

Question :- How to find 2's Compliment ?

Ans ① 1's Compliment :- Toggle every bit

② Add 1 to it :- +1 to 1's compliment

③ we will get 2's Compliment.

2's Compliment = 1's Compliment + 1

$(45)_{10} = 0\ 0\ 1\ 0\ 1\ 1\ 0\ 1$

→ 1's compliment = $1\ 1\ 0\ 1\ 0\ 0\ 1\ 0$

→ +1 = $0\ 0\ 0\ 0\ 0\ 0\ 0\ 1$

→ 2's compliment = $1\ 1\ 0\ 1\ 0\ 0\ 1\ 1$

$\Rightarrow [-45]_{10}$

$\Rightarrow -2^7 + 2^6 + 2^4 + 2^1 + 2^0$

$\Rightarrow -128 + 64 + 16 + 2 + 1$

$\Rightarrow -128 + 83$

$\Rightarrow -45$ Hence proved.

Quiz 5 :- $(-3)_{10} = (??)_2$

Ans $(3)_{10} = 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1$

1's Compliment : $1\ 1\ 1\ 1\ 1\ 1\ 0\ 0$

$+\ 1 = 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1$

$[-3]_{10}$ = 2's Compliment = $1\ 1\ 1\ 1\ 1\ 1\ 0\ 1$

Quiz 6 :- $(-10)_{10} = (??)_2$

$(10)_{10} = 0\ 0\ 0\ 0\ 1\ 0\ 1\ 0$

1's Compliment = $1\ 1\ 1\ 1\ 0\ 1\ 0\ 1$

$+\ 1 = 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1$

$[-10]_2$ = 2's Compliment = $1\ 1\ 1\ 1\ 0\ 1\ 1\ 0$

V RANGES

① BYTE = 8 bits

Min Num = 1 0 0 0 0 0 0 0

$= -128$

$= -2^7$

Max Num = 0 1 1 1 1 1 1 1

$= 2^7 - 1$

$= 127$

-128 to 127

OR -2^7 to $2^7 - 1$

② INT = 32 bits.

Min Num = 1 0 0 0 ..... 0 0 0

$= -2^{31}$

Max Num = 0 1 1 1 ..... 1 1 1

$= 2^{31} - 1$

Range : -2^{31} to $2^{31} - 1$

$\approx -10^9$ to 10^9

③ Long = 64 bits

Range :- -2^{63} to $2^{63}-1$
$\approx -10^{18}$ to 10^{18}

Min Num = 1 0 0 0 ..... = -2^{63}

Max Num = 0 1 1 1 1 1 1 ... = $2^{63}-1$

BONUS KNOWLEDGE

* Importance of Constraints

⇒ Given two integers a & b. Return

a * b

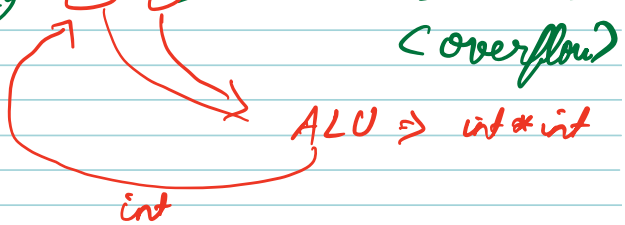
$a <= b <= 10^6$

① int ans = a * b; → Incorrect (overflow)
return ans;

② long ans = a * b; → Incorrect (overflow)
return ans;

ALU ⇒ int * int
↳ return int

③ long ans = long (a * b) ⟶ Incorrect (overflow)



④ long ans = (long) a * b ⟶ Correct

Question :- Given N, calculate the sum of array elements.

constraints :	$1 <= N <= 10^5$
	$1 <= A[i] <= 10^6$

PSEUDO CODE :-

```
int Sum = 0
for (i = 0; i < N; i++){
    Sum += A[i];
}
return Sum;
```

max Sum $\leq 10^{11}$

* To correct the code, we need to use Sum as long.