DSA For MATH.

Bitwise Operators + Number Systems

- When we write int a=10 ; (in any language) internally whatever the things that you're doing, all the servers that're running, etc, so internally it's all a bunch of 0, 1.

- Computers understands only binary language (0 & 1)

- we don't code in 0 & 1 because it will becomes very complex. so that's why we have programming languages

- so that's how numbers are stored internally this is known as no system

- There are various no. system. base 10, base 8, base 16, base 2.

Q. What "base" Represents ?

→ Base basically means How many numbers do we have in the particular base concept.

* We have been studying since we're kids :- decimal no system (base 10)

• Base 2 : It means binary no. system. This is what is computer understands.

Now we can imagine 0 as false and 1 as true

* Bitwise Operators :

1] AND :-

(in words) : Here let's say if you're 2 no. all of them should be true

Truth table :-

a	b	a AND b
0	0	0
0	1	0
1	0	0
1	1	1 (all are true)

Page No.
Date
YOUVA

Conditions :

AND conditions means that all the numbers or input should be true, then only the entire expression will be true.

If one of the input is false then the output will be false only.

If we are trying of output of A & A . B & C then all the inputs should be true or equal to one (1)

In computer science we can say that 1 represents true and 0 represents false.

Electronic :-

If 1 & 1 all the gate will be open so there will be flow of current If 1 & 0 or any thing other it won't, AND gate is a basic logic gate.

Note: ① When you AND 1 with any number the digits will remains the same.

(Q) E.g 1 1 0 0 1 — same
1 1 1 1 1
1 1 0 0 1 — same

2] OR :-

(In words :- It is opposite of AND ga
if any one of the input is
true then the entire expression is
true.

Truth table :-

a	b	a OR b
0	0	0
1	0	1
0	1	1
1	1	1

Condition :

Even if both the inputs are true output will be always positive.

Very Imp

Page No.:	YOUVA
Date:	

3] XOR (⊕) : (If and only if)

(In words) :- It is exclusive OR Here if you've two no. only one of them should be true, and anything else will false as a output.

Truth table :

a	b	a ⊕ b
0	0	0
0	1	1
1	0	1
1	1	0

Condition :

If more than 2 no then, the add no. should be true.

Observation :

1] $a \oplus 1 = \bar{a}$.

Note: $\bar{a}$:- It is basically a complementary bar that means of opposite of that no.

e.g. ① $0 \oplus 1 = 1$

② $1 \oplus 1 = 0$

So, anything we X-OR with 1 you will get the answer opposite of that

2) $q \wedge 0 = q$

3) $q \wedge q = 0$

e.g $32 \wedge 32 = 0$

If X-OR the same input the output will always be zero

4) Complement ($\sim$)

Suppose : $q = 10110$

$\overline{q} = 01001$

A	B	C	D	o/p
0	0	0	0	01
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7

8421

8421.

A	B	C	D	
1	0	0	0	8 -
1	0	0	1	9
1	0	1	0	10 → A
1	0	1	1	11 → B
1	1	0	0	12 → C
1	1	0	1	13 → D
1	1	1	0	14 → E
1	1	1	1	15 → F

** Number System :-

1] Decimal :- 0 to 9

and it is base 10.

Base 10 means there are 10 no.s which we can use to represents any no. in decimal form.

E.g

$(357)_{10}$, $(10)_{10}$

2] Binary :- 0 & 1

And its of base 2

E.g

① $(10)_{10} \longrightarrow (1010)_2$

② $(7)_{10} \longrightarrow (0111)_2$

3] Octal :- 0 to 7 base 10

Now comparing decimal & octal.
In octal we can not 8 & 9 because we can only use the no.s from 0 to 7.

Decimal : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 ....

octal : 0, 1, 2, 3, 4, 5, 6, 7, 10, 11, 12, 13, 14, 15, 16, 17, 20..

So 8 in decimal is 10 in octal & 9 in decimal is 11 in octal

4) Hexddenimal: 0-9 & A-F base 16

e.g $(10)_{10} = (A)_{16}$
$(12)_{10} = (C)_{16}$

* Conversions:

Note:

① Decimal to base 2.

Q. Convert $(17)_{10}$ to base 2

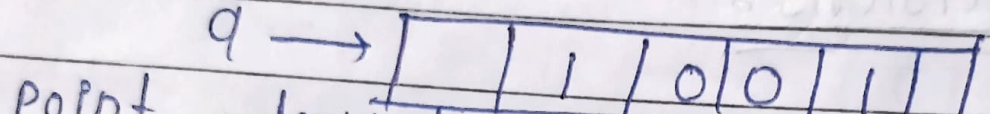
① keep dividing by base take remainder write in opposite

2	17	
2	8	1
2	4	0
2	2	0
	1	0

$(10001)_2$

$\therefore (17)_{10} = (10001)_2$

So, when you write int a = 17 compiler will store it like (1001) each ram has their individual calls.
So in ram it will be stored something like



& a will point towards it the call.

Q. $(17)_{10} = (?)_8$

8	17
	2 1

$(21)_8$

$\therefore (17)_{10} = (21)_8$

(2) Base b to decimal.

Q. $(10001)_2 \rightarrow (?)_{10}$

Step:

(1) Multiply & add the power of base with digits.

1	0	0	0	1
2^4	2^3	2^2	2^1	2^0
$2^4 \times 1$	$2^3 \times 0$	$2^2 \times 0$	$2^1 \times 0$	$2^0 \times 1$

$= 16 + 0 + 0 + 0 + 1$

$= \underline{17}$

$\therefore (10001)_2 = (17)_{10}$

Q. $(21)_8 = (?)_{10}$

2	1
8^1	8^0

$8^1 \times 2 + 8^0 \times 1$

$= 16 + 1$

$= 17$

$\therefore (21)_8 = (17)_{10}$

—o—

If you want to double the no just << by 1

continuing with operator

(5) left shift operator (<<):
It shifts all the bits towards the left side (one by one)

e.g $(10)_{10} = (1010)_2$

Q. 10 << 1 — { It basically means that shift i/p by 1.

Step I: Convert the internally it into binary no.

∴ 1010 << 1

Step II: Okay shift it towards the left one by one.

∴ 1 0 1 0 << 1 1 0 1 0 0

Step III. After shift the to the left as we will require one entire no. so whatever we need an extra no. in left shift operator we add 0. as above mention

1 0 1 0 0

$2^4 + 2^3 + 2^2 + 2^1 + 2^0$

1 1 1 1 1

16 x 1 + 0 + 4 x 1 + 0 + 0

= 16 + 4

= 20

> Hence, any no [a<<1 = 2a] means it going to double that no.

General point: If you left shift a no. "a" "b" time it's going to multiply $a \times 2^b$.

Imp = [a<<b $a \times 2^b$]

[a<<b = 2a]

⑥ Right shift operator :- ">>"

It is opposite of left shift Operator.

- It means the given I/P towards the right come by on.

Q. 0 0 1 1 0 0 1 >> 1

Ignore

"This question basically says that move the given I/P by 1, and by moving them one by one towards the right, we're to discovered the

∴ 0 0 1 1 0 0 1 ☒ >> 1

= 0 0 1 1 0 0

Note:

- Similarly like in decimal no. system when the no like (000123)
  = (123)10
- Here the leading zero are ignored
- this is same for all of the no. system even through the zero from left hand side will be ignore, the values will remained unchanged but we can't do that from right side as it will change the whole value

∴ =1100

Note: In binary 1+1 = 10 & 1 is carry so ans = 0,

Eg
```
  10
+ 11
----
 100
```

Note: In the right shift operator were dividing the no by 2.

Imp → $q >> b = \frac{q}{2^b}$ — (general point)

=

* Working:

Q. given a no n find if it is odd or even

Note:
① When you AND 1 with any no. the digits remain the same.
② we know that any no whatever calculation we do internally it will be calculated as a binary no even if you do substraction addition etc.

E.g

$12 + 7 \rightarrow 1100 + 0111 = 10000$

$1100 + 0111 = 10011$

$(19)_{10} \rightarrow (10011)_2$

1 0 0 1 1

$2^4 + 2^3 + 2^2 + 2^1 + 2^0$

$2^4 \times 1 + 0 + 0 + 2 \times 1 + 1 \times 1$

$= 16 + 2 + 1$

$= 19$

Note: Every no e.g :- 100 ID. leaving this, every other is a power of 2

1 → odd
0 → even

Page No:
Date:
YOUVA

Hence this is the no. whether it determines because we know that leave the last one entire no. will be even (always) why? because all of these are powers of 2

the last no will always be a 0 and any thing positive no. raceise to 0 is 1.

Hence, if this particular no is 1 it means that the answer to everything is +1

If 2° place == 1 → no. is odd
otherwise no. is even

find out what last digit is if it 1 then the no. is odd & if it is 0 then the no is even.

Q And the no. & get the last digit

Use

```
  1 0 0 1 0 1
& 0 0 0 0 0 1
-------------
  0 0 0 0 0 1   ---(1)
```

n & 1 == 1 → odd

Sum up :-

n & 1 == 1 ⟶ odd : even (else)

Code

```
public class EvenOdd {
    public static void main(String[] args){
        int n = 67;
        System.out.println(isOdd(n));
    }
    private static boolean isOdd(int n){
        return (n & 1) == 1;
    }
}
```

Note:

0 1 1 0 0 0 [1] → It is known as LSB (least significant bit)

It is called LSB

19. You're given an array of numbers and in that array every number appear twice only one number appears once. You're to find that no.

arr = [2, 3, 4, 1, 2, 3, 6, 4] (6 is the one)

So How will you can find this?

Note: Bit wise operators likes in normal maths operator they also follows the associative properties.

① E.g :- $(5 * 3) * (5 * 4)$

$= 5 * 5 * 3 * 4$ or
$5 * 4 * 3 * 5$ etc.

so, the order basically does not matter.

② $2 \wedge 3 \wedge 3 \wedge 4 = 2 \wedge 4 \wedge 3 \wedge 3$ etc

we wanna do it in constantine & in one single pass.

Note:

As we know any no ^ with the same no $a \wedge a = 0$ & $0 \wedge a = a$

so, if ^ the entire array, I know the all the duplicates will lead to zero.

Ans :- XOR all the numbers.

* Time complexity :- O(n)
* space complexity : O(1)

Q) arr = [-2, 3, 2, 4, -5, 5, -4]
ans = 3

Note:- ORDER does not matters

code

Ans
```
public class unique {
    public static void main(String[] args)
{
    int[] arr = {2,3,-3,-2,6,5,-5};
      System.out.println(unique(arr));
}
private static int unique(int[] arr){
        int ans = 0;
        for( int i : arr){
            ans ^= i;
        }
        return ans;
    }
}
```

[3] → doubt

Q. find ith bit of the number?

8 7 6 5 4 3 2 1
0 1 0 1 1 0 1 1 0
↓
0

Ans: 0 we know that we were able to find How to find LSB or right most bit we just And it or ⊥ .. So And the particular digit with 1 it will give us, that no, but And will be 0.

∴ Ans :-

1 0 1 1 0 1 1 0
0 0 0 1 0 0 0 0 → {mask}

mask

0 0 0 1 0 0 0 0

Q. How to get 00010000?

Note: Mask is a seperate no. or entity you've that allows us to get our answer related to it.

n = mask with n-1 zeros.

Q How do we do that?

do we need n-1 zeros in the right hand side

Q. what do we need to do in order to move the 1 towards the left hand side, get know zeros and one Here 1 towards the left hand side ~~get zeros on~~ left ~~sh~~ shift (1 << (n-1)

∴ 1<<4 = 10000

Ans :- n & 1 (1<< (n-1))

Q. Set the ith bit

Set means :- turn it do 1

So if ith bit 0 make it 1
if ith bit 1 remain 1 only

1 0 1 0 1 1 0

let's say you want to set the 4th bit

Note: 0 OR 1 = 1

∴ 1 0 1 0 1 1 0 —
0 0 0 1 0 0 0 — (mask)

Q. Reset ith bit

10110

Reset means : if it's 1 make it 0
if it's 0 remain 0

Note: If we and any no with 1 we'll get the no itself but if we and any no. with 0 we'll get 0 only

0 & 1 = 0

```
 1 0 1 1 0
&1 0 1 1 1   — {mask}
-----------
 1 0 1 1 0
```

Q. So how did get that mask ?

Ans: By Complement

mask = ~(1 << (n-1))

[3] - doubt

Page No.: YOUVA
Date:

Q. Find the position of their eight most set bit?

E.g :-

8	2	6	5	4	3	2	1
1	0	1	1	0	1	1	0
						Ans	

Ans:-2

looking from the eight hand side which is the set bit?
the first 1 that occurs from the eight handside.

Ans:-2

1 0 1 1 0 1 1 0

If were writing this no. in the form of 8 bit we so this one right most bit

h = a | b

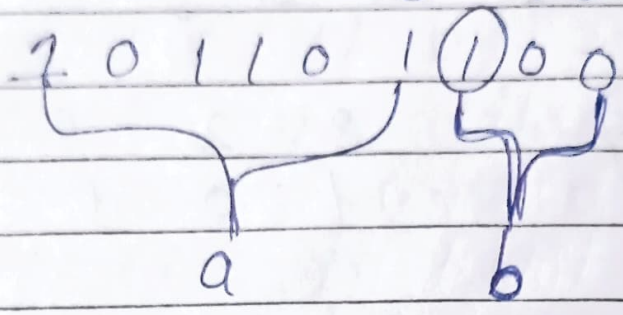
or

1 0 1 1 0 1 1 0 0

a b Ans:-4

$h = q \perp b$: $q = 10110$

$b = 00$



As we know 100 is our answer to we're not touching them And we have to create 10110 all of them ~~and~~ ~~we have to~~ as 0

we need make something like $\bar{q} \perp b = ?$ & then find these

int terms of formula it is equal to $-h$.

So, let's say

$$-h = \bar{q} \perp b$$

Ans :- h & $(-h)$ $\because q$ & $\bar{q} = 0$