Bitwise Operators

* Bit Manipulation :-

Operators

1) And operator

Q) Given a number, find if it's even or odd?
(Later)

* when you & 1 with any number, digits remain the same.

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

Same

* OR operator

* XOR (^) (If and only if)

exclusive OR

a	b	a^b
0	0	0
0	1	1
1	0	1
1	1	0

Observation : If XOR any number with 1 $a \wedge 1 = \bar{a}$ (a complement)

$a \wedge 0 = a$

$a \wedge a = 0$ (32 XOR 32 = 0)

* compliment operator (~)

$a = 10110$ $\bar{a} = 01001$

* Number System :-

1) Decimal → 0, 1, 2, . . . 9 Base 10

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

2) Binary Numbers → 0 & 1 , Base:2

$(10)_{10} = (1010)_2$

$(7)_{10} = (111)_2$

3) octal ⟶ 0, 1, 2, 3, 4, 5, 6, 7 Base : 8

0, 1, 2, 3, 4, 5, 6, 7, 10, 11, 12, 13, 14, 15, 16, 17, 20, 21

$(9)_{10} = (11)_8$

4) Hexadecimal

0 . . . 9 and A . F

Total = 16 numbers Base : 16

$(10)_{10} = (A)_{16}$

---

* Conversions :-

① Decimal to base b

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

keep dividing by base, take remainders, write in opposite

$$
\begin{array}{r|l}
2 & 17 \\
2 & 8 \quad - \ 1 \\
2 & 4 \quad - \ 0 \\
2 & 2 \quad - \ 0 \\
 & 1 \quad - \ 0
\end{array}
$$

reverse order

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

$$(17)_{10} = (\ ?\ )_8$$

$$
\begin{array}{r|l}
8 & 17 \\
 & 2 \quad - \ 1
\end{array}
$$

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

2) Convert any base b to decimal.

$$(10001)_2 = (\ ?\ )_{10}$$

Steps: multiply and add the power of base with digits.

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

$$= 16 + 1 = 17$$

$$(10001)_2 = (17)_{10} \longrightarrow \text{Ans}$$

8) $(21)_8 = (\,?\,)_{10}$

$= 2 * 8^1 + 1 * 8^0$

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

5) Left shift operators ($<<$)

$(10)_{10} = (1010)_2$

$$
\begin{array}{r|ll}
2 & 10 & \\
2 & 5 & - \; 0 \\
2 & 2 & - \; 1 \\
 & 1 & - \; 0
\end{array}
$$

$10 << 1$

10 left shift 1

step:

$1010 << 1$

10100

$1*2^4 + 0*2^3 + 1*2^2 + 0*2^1 + *2^0 = 20$

$$\boxed{\therefore \; a << 1 = 2a}$$

General point:

$a << b = a * 2^b$

10 left shit 1 $= 10 * 2^1 = 20$

6) Right Shift operator >>

Q: $0011001 >> 1 \Rightarrow 001100$

Ignore this part

$\downarrow$

1100

$(000011234)_{10} = (11234)_{10}$

$\hookrightarrow$ All the zeroes are ignored for all number systems.

$$\boxed{a >> b = a/2^b}$$

Q) Given a number find it is odd or even

→ Point: every number is calculated in binary form internally

011011010001101
least significant bit

$12 + 7 \Rightarrow$

$$\begin{array}{r} 1100 \\ +0111 \\ \hline 10011 \end{array}$$

$(19)_{10} = (10011)_2$

Note: $\underbrace{1001}_{\text{this will always be even}}1$ → leaving this, every other is a power of 2

Hence: if 2^0 place $== 1$ number is odd
otherwise number is even.

$$
\begin{array}{r}
100101 \\
\& \; 000001 \\
\hline
000001
\end{array}
\Rightarrow 1, \text{ Hence odd}
$$

So to sum up: n and $1 == 1 \Rightarrow$ odd
else even

Program:

```
public static void main (String[], args){
    int n = 69;
    System.out.println(isOdd(n));
}
static boolean isOdd (int n){
    return (n&1)==1;
}
}
```

Q) VIP

Every number in the array is appearing twice but there is a number which is not repeated find the answer arr = [2, 3, 4, 1, 2, 1, 3, 6, 4]

Ans

XOR property is

$a \wedge a = 0$

$a \wedge 0 = a$

$a \wedge 1 = \bar{a}$

Now use the XOR property in above array.

arr = [2, 3, 4, 1, 2, 1, 3, 6, 4]

$a \wedge 0 = a \Rightarrow$ Ans

Complexity O(N)

Space complexity is constant

ex: arr = [-2, 3, 2, 4, -5, 5, -4]

ans

Program:

```
public static void main(String[] args){
    int[] arr = {2,3,4,1,2,1,3,6,4};
    System.out.println(findUnique(arr));
```

}

```
static int findUnique(int[] arr){
    int unique = 0;
    for(int n : arr){
        unique ^= n;
    }
    return unique;
}
```

unique → This is a variable that will store the XOR operation.

Q: Find i^{th} bit of a number.

8	7	6	5	4	3	2	1
1	0	1	1	0	1	1	0

→ & this particular digit with 1

→ ? find

```
  10110110
& 00010000
  --------
  00010000 → Ans
```

This is called a mask

$n \Rightarrow$ mask with $(n-1)$ zeroes.

So to have 1 with 0000 in the R.H.S

left shift the 1 by 4

$1 << 4 \rightarrow 10000$

$1 << (n-1)$

Ans: $n \,\&\, (1 << n-1)$

Similar Question set the i^{th} bit turn it to 1.

$$\begin{array}{rccccccc} & 1 & 0 & 1 & \overset{4th}{0} & 1 & 1 & 0 \\ \text{OR} & 0 & 0 & 0 & 1 & 0 & 0 & 0 \rightarrow \text{mask} \\ \hline & 1 & 0 & 1 & 1 & 1 & 1 & 0 \;\checkmark \end{array}$$

Q: Reset i^{th} bit

$$\begin{array}{rccccccc} & 1 & 0 & \overset{5th\ bit}{1} & 0 & 1 & 1 & 0 \\ \& & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ \hline & 1 & 0 & 0 & 0 & 1 & 1 & 0 \end{array}$$

$1 \rightarrow 0$

$0 \rightarrow 0$

How to get this mask

To get this mask left shift it by $(n-1)$ times and take complement of it

mask: $\overline{1<<(n-1)}$ or $!(1<<(n-1))$

Q) Find the position of the right most set bit

Ex: $\underbrace{101101}_{a}1\underbrace{0}_{b}$ Ans = 2 [1 that occur first from right most]

$N = a1b$ $a = 101101$

$b = 0$

$-N = \bar{a}1b$ → How? :?!

Ans = N and $(-N)$

Negative of a number in binary form:

1 byte = 8 bits

10 in 1 byte?

0	0	0	0	1	0	1	0

most significant bit

least sign. bit

$-10 = ?$

positive or

Tells us number is negative

$1 \rightarrow -ve$

$0 \rightarrow +ve$

Steps: ① Take complement of a number
② Add 1 to it

Two's compliment method.

$(10)_{10} = (00001010)_2$

① (11110101) And the add 1

② (11110101)
$+ 00000001$

$(11110110)_2 \Rightarrow (-10)_{10}$

This -ve of 10 in 1 byte.

why Tho (?!)

$10 \overline{)\,10110111}$

Discarded

$$
\begin{array}{r}
1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \\
-\ \ 0\ 0\ 0\ 0\ 1\ 0\ 1\ 0 \\
\hline
\end{array}
$$

what's thus

100000000

$\Downarrow$

$\underline{11111111+1}$

$(1000)_2 = 111+1$

$(8)_{10} = 7+1$

$(10000)_2 = 1111+1$

$(16)_{10} = 15+1$

$$
\begin{array}{r}
1111 \\
+0001 \\
\hline
(10000)_2
\end{array}
$$

Now, $11111111+1-00001010$

$= \underbrace{11111111-00001010+1}_{\text{complement of a number}}$

$$
\begin{array}{r}
11111111 \\
-\ 00001010 \\
\hline
11110101
\end{array}
$$

→ complement

Range of a numbers.

Actual No. stored in

1) 1 byte · | 0,1 | 0,1 | 0,1 | 0,1 | 0,1 | 0,1 | 0,1 | 0,1 |

Total = 2*2*2*2*2*2*2*2

Total no of unique number it can store in 1 byte = 2^8 = 256 in 1 byte

Actual No stored in 7 bits

Total = 2^7 = +128 to -128

numbers in positive

numbers in negative

0 is there is a well

then, - 128 to 127

Range formula for n bits

$$\boxed{-2^{n-1} \text{ to } 2^{n-1} - 1}$$

Q: arr = [2, 2, 3, 2, 7, 7, 8, 7, 8, 8]

						1	0

0 1 → 1
1 0 → 2
1 1 → 3
1 0 0 → 4

3	3	6	3

If 3 was not in array

1 0
0 1 1
1 0
1 1 1
1 1 1
1 0 0 0
1 1 1
1 0 0 0
1 0 0 0

	3	3	7	4

%3

0 0 1 1 = 3

Everything appearing 3 times there set bit will also appear 3 times. Hence, the set bits will be % of 3 means remainder will be 0 and if remainder is 1 it is not a multiple of 3

Code it yourself.

Q: Amazon Question:

Find the n^{th} magic number

$$1 = \overset{5^3}{0}\ \overset{5^2}{0}\ \overset{5^1}{1} \rightarrow 5$$

$$2 = \overset{5^3}{0}\ \overset{5^2}{1}\ \overset{5^1}{0} \rightarrow 25$$

$$3 = \overset{5^3}{0}\ \overset{5^2}{1}\ \overset{5^1}{1} \rightarrow 30$$

$$4 = \overset{5^3}{1}\ \overset{5^2}{0}\ \overset{5^1}{0} \rightarrow 125$$

$$5 = 1\ 0\ 1 \rightarrow 130$$

|

:

n^{th} magic number.

If $n = 6 \Rightarrow \underbrace{11}\,0 \rightarrow$ Ignore

loop [$n \,\&\, 1 \Rightarrow$ This will give me last digit in binary number
$n >> 1$
Right shift

$$0 \times 5^1 + 1 \times 5^2 + 1 \times 5^3 = 150$$

Code! public static void main(String [ ] args){

```
int n = 5;
int ans = 0;
int base = 5;
while(n>0){
    int last = n & 1;
    n = n >> 1;

    ans = ans + last * base;
    base = base * 5;
}
System.out.println(ans);
}
}
```

Q: Find number of digits in base b

$(6)_{10} = 1$

$(6)_{10} = (110)_2 = 3$

Formula? $\log = x$

$\log_b a$

$a = b^x$

Similarly, $\log_2 6 = x$

$6 = 2^x$

Example: $\log_2 10 = 3.32$

$10 = 2^{3.22}$ → int + 1 = no. of digits

Formula: no. of digits in base b of number $n = \text{int}(\log_b n) + 1$

No. of digits in n $= \text{int}(\log_2 10) + 1$

$= 3 + 1 = 4$

$= (10)_{10} = 1010$

int ans = (int)(Math.log(n) / Math.log(b)) + 1;

Code:-

```
public static void main(String[] args){
    int n = 10;
```

```
int base = 2;

int ans = (int)(Math.log(n) /
    Math.log(base)) + 1;

Sout(ans);
}
} complexity : log(n)
```

$(10)_{10} \Rightarrow (12)_8$

0 1 2 3 4 5 6 7 8 9 10

0 1 2 3 4 5 6 7 10 11 12

octal → 0 to 7 Not 8 & 9

Pascals Triangle :-

1					
1	1				
1	2	1			
1	3	3	1		
1	4	6	4	1	
1	5	10	10	5	1

Q: Find the sum of n^{th} row:

Ans. sum of each row =

$$^nC_0 + {^nC_1} + {^nC_2} + {^nC_3} + \dots {^nC_n} = 2^n$$

For n^{t} row, sum $= 2^{n-1}$ so $(n-1)$
starts from index zero

Ans $= 1 << n-1 = 1 * 2^{n-1}$

Code Yourself.

Q: You are given a number find out if its power of 2 or not.

→ 100000 , 100010

$100000000 = (\underbrace{11111111}_{n-1}) + 1$

$$
\begin{array}{r}
100000000 \\
\& \; 011111111 \\
\hline
000000000
\end{array}
$$

Not a power of 2,

$$
\begin{array}{r}
100010 \\
\& \; 011111 \\
\hline
000010
\end{array}
$$

$\underbrace{11111}_{(n-1)} + 1$

Ans $\Rightarrow$ If n & n-1 == 0, then it is power of 2. (Code done in Mac)

Q) Calculate a^b

$3^6 \Rightarrow 3 \times 3 \times 3 \times 3 \times 3 \times 3$

$3^6 = 3^{2+4} = 3^2 \times 3^4$

$3^6 = 3^{110}$

ans = 1

base = 3

base = 9

n = 110

n >> 1 $\Rightarrow$ 0

n = 11 >> 1 $\Rightarrow$ 1 Right shift

n = 1 >> 1 = 1

ans = ans * base

$3^{110} \Rightarrow 3^4 * 1 \; 3^2 * 1 * 3^0 * 1$ → 1

Important to understand the question.

$O(\log(b))$

Doubt

$n = 110$

$n \& 1 \Rightarrow 0$

$n >> 1 \Rightarrow 11 \& 1 \Rightarrow 1$

$n = 11 >> 1 \Rightarrow 1 \& 1 = 1$

$n = 1 >> 1 = 0$

Decrease the Power first: 110 then : 11 and then: 1

Q: Given a number n, find the number of set bits in it.

$n = 9$

$n = 1001$ Ans = 2

```
while (n>0)
if (n & 1 == 1)
        Count++;
    n = n>>1;
```

or

$n \text{ and } (-n) = 0001$

$n - [n \text{ and } (-n)] = 1000$ count = 1

$n - 1$

$n = 1001$

$\&\ n-1 = 1000$

1000 → count 1

8 & 7

1000

$\&\ 0111$

0000 → count 2

(Code in Mac)

Number of set bits = number of iterations.

Q) Find XOR of no's from 0 to a.

A	B	XOR
0	0	0
0	1	1
1	0	1
1	1	0

a	XOR from 0 to A
0	0 = 0
1	0^1 → 00 XOR 01 = 01 = 1
2	0^1^2 = 3 → 01 XOR 10 = 11 = 3
3	0^1^2^3 = 0 → 11 XOR 11 = 00 = 0
4	0^1^2^3^4 = 4 → 0 XOR 4 = 4
5	0^1^2^3^4^5 = 4 XOR 5 = 1
6	0^1^2^3^4^5^6 = 5 XOR 6 = 7
7	6^7 = 0
8	8 = 8

011 XOR 100 = 111

XOR 111 = 111

$2|4$: $2|2 - 0$, $1 - 0$

$2|5$: $2|2 - 1$, 1 0

$2|6$: $2|3 - 0$, $2|1 - 1$

$2|100$: $2|50 - 0$, $2|25 - 1$, $2|12 - 0$, $2|6 - 0$

$$\begin{array}{r} 001 \\ \text{XOR } 110 \\ \hline 111 \end{array}$$

$$\begin{array}{r} \text{XOR } 101 \\ \hline 001 \end{array}$$

$$2 \mid 8,\quad 2 \mid 4 - 0,\quad 2 \mid 2 - 0,\quad 1 - 0$$

$$2 \mid 3 - 0,\quad 2 \mid 1 - 1$$

$$2 \mid 6,\quad 2 \mid 3 - 0,\quad 1 - 1 \quad 110$$

$$2 \mid 7 - 1,\quad 2 \mid 3 - 1,\quad 1$$

$$7 \to 0111 + 1 \Rightarrow 1000$$

$$\begin{array}{r} 8 \to 1000 \\ \hline 1111 \end{array}$$

$$\begin{array}{r} 1000 \\ \text{XOR } 1000 \\ \hline 0000 \end{array}$$

$$2 \mid 7,\quad 2 \mid 3 - 1,\quad 1 - 1 \qquad 2 \mid 6,\quad 2 \mid 3 - 0,\quad 1 - 1$$

$$\begin{array}{r} 111 \\ \text{XOR } 110 \\ \hline 0001 \\ 1000 \\ \hline 1001 \end{array} \qquad \begin{array}{r} 111 \\ \text{XOR } 110 \end{array}$$

If $a \% 4 = 0$ → a

$a \% 4 = 1$ → 1

$a \% 4 = 2$ → $a+1$

$a \% 4 = 3$ → 0

$$0 \to a$$

Q: XOR of all numbers between a & b

$a = 3$ & $b = 9$

$3 \wedge 4 \wedge 5 \wedge 6 \wedge 7 \wedge 8 \wedge 9$

$0 \wedge 1 \wedge 2 \wedge 3 \wedge 4 \wedge 5 \wedge 6 \wedge 7 \wedge 8 \wedge 9$

These are the extras

This is $0 \to (a-1)$

Ans $\Rightarrow$ $\boxed{f(b) \wedge f(a-1)}$

$f(x) \longrightarrow$ XOR of $0 \to x$

Watch the video for 3 last question again!!

*8) Flipping an image :- (Google Question)

$$\begin{matrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{matrix} \longrightarrow \begin{matrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{matrix}$$

Invert it

$0 \wedge 1 = 1$

$1 \wedge 1 = 0$

$$\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{matrix}$$

Ans

Code:

```
public int[][] flipAndInvertImage(
            int[][] image){
    for (int[] row : image){
        // reverse this array
        for(int i=0; i<(image[0].length+1)/2;
                i++){
            //swap
            int temp= row[i]^1 ;
            row[i] = row(image[0].length-i-1)^1;
            row[image[0].length -i-1] = temp;
        }
    }
    return image
}
```