1. Decimal no. system

base : 10 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

0	10
1	11
2	12
3	13
4	14
5	15
6	16
7	17
8	18
9	19

2. Binary no. system

Base : 2 digit → 0, 1

00 → 0	10 → 2	100 → 4
01 → 1	11 → 3	101 → 5

Conversions

1. Decimal to binary

$(27)_{10} = (11011)_2$

2	27	
2	13	1
2	6	1
2	3	0
2	1	1
	0	1

$(28)_{10} = (11100)_2$

2	28	
2	14	0
2	7	0
2	3	1
2	1	1
	0	1

2. Binary to decimal

$(1101)_2 = (13)_{10}$

1 1 0 1

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

$8 + 4 + 0 + 1 = 13$

384

$3\times10^2 + 8\times10^1 + 4\times10^0$

$300 + 80 + 4 = 384$

1101

MSD ... LSD

$(1\ 0\ 1\ 0)_2 = (10)_{10}$

1 0 1 0

2^3 2^1

$= 8 + 2 = 10$

quiz

Binary of 25

$(25)_{10} \rightarrow (11001)_2$

2	25	
2	12	1
2	6	0
2	3	0
2	1	1
	0	1

$(10110)_2 \rightarrow (22)_{10}$

1 0 1 1 0

16 4 2

Addition of two binary numbers

```
  1  1  1
  3  8  4  9
+    1  7  6
-----------
  4  0  2  5
```

15

```
  1  1  1  1
     1  0  1  1
+    0  1  1  1
--------------
  1  0  0  1  0
```

11
+7

18

0 + 0 = 0

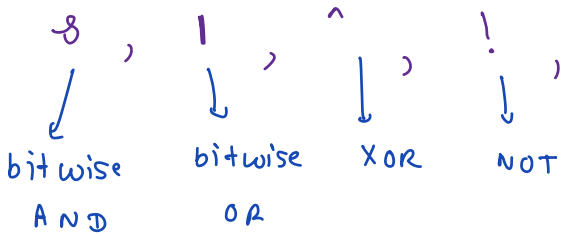
0 + 1 = 1

1 + 0 = 1

1 + 1 = 10

```
  1  1  1
  1  0  1  1  0
+    1  0  1  1
-----------------
  1  0  0  0  0  1
```

Bitwise operators



leftshift, rightshift

& → 0 is dominating

| → 1 is dominating

^ → same same
puppy shame

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

1) 10 & 12

```
  1 0 1 0
& 1 1 0 0
---------
  1 0 0 0   ⟶ 8
```

```
System.out.println(10 & 12); // 8
```

2) 10 | 12

```
  1 0 1 0
| 1 1 0 0   (OR)
---------
  1 1 1 0   ⟶ 14
```

```
System.out.println(10 | 12); // 14
```

3) 10 ^ 12

```
  1 0 1 0
^ 1 1 0 0
---------
  0 1 1 0   ⟶ 6
```

```
System.out.println(10 ^ 12); // 6
```

Properties of bitwise operators

a and b are two no.

1) Commutative

a & b = b & a

a | b = b | a

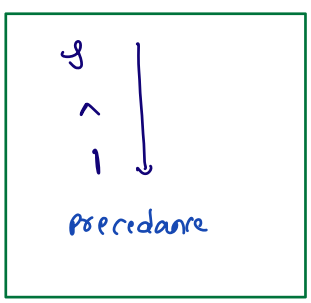
a ^ b = b ^ a

2) Associative

a & b & c = (a & b) & c = a & (b & c)

a | b | c = (a | b) | c = a | (b | c)

a ^ b ^ c = (a ^ b) ^ c = a ^ (b ^ c)



3) n & 0 = 0

n & n = n

n | 0 = n

n | n = n

n ^ 0 = n

n ^ n = 0

Q-1 Given a int no., check whether it is even or odd.

Note: we can't use +, -, *, /, %.

10 → 1010

13 → 1101

```
if (LSD == 0) {
        even
}
else {
        odd
}
```

```
void solve(int n) {
    if ((n & 1) == 1) {
        sopln("odd");
    }
    else {
        sopln("Even");
    }
}
```

n = 13

```
  1101
& 0001
------
  0001
```

o/p: odd

n = 18

```
  10010
& 00001
-------
  00000
```

o/p: even

Q.2 **Single element**

Given an A[], every value is coming except for a no. that is coming once. Find the single no.

TC: O(n)

SC: O(1)

A = 5 2 3 2 **8** 5 3

ans = 5 ^ 2 ^ 3 ^ 2 ^ 8 ^ 5 ^ 3

= (5 ^ 5) ^ (2 ^ 2) ^ (3 ^ 3) ^ 8 = **8**

(0) (0) (0)

A = 3 5 3 **8** 5

```
int singleElement(int[] A) {
    int ans = 0;
    for(int i=0; i<A.length; i++) {
        ans = ans ^ A[i];
    }
    return ans;
}
```

```
  0 0 0
^ 0 1 1
-------
  0 1 1
^ 1 0 1
-------
  1 1 0
^ 0 1 1
-------
  0 1 0 1
^ 1 0 0 0
---------
  1 1 0 1
^ 0 1 0 1
---------
  1 0 0 0
```

Left shift <<

with every left shift no. is getting twiced.

a = 13 1 1 0 1 → 13

a << 1 1 1 0 1 0 → 26

a << 2 1 1 0 1 0 0 → 52

a << 1 → 2a

a << 2 → 4a

a << 3 → 8a

.
.
.
.

a << N → $2^N * a$

1 << N = 2^N

↓

O(1)

Rightshift >>

with every rightshift no. is getting halved.

a = 13 → 1 1 0 1 → 13

a >>1 → 1 1 0 → 6

a >>2 → 1 1 → 3

$a >> 1 = a / 2$

$a >> 2 = a / 4$

$a >> 3 = a / 8$

⋮

$a >> N = a / 2^N$

Doubts

ans %. ($10^9 + 7$)

```
int m = (int) (Math.pow(10, 9) + 7);
```

```
//a, b, c → int
ans = a + b + c;
return ans % m;
```

i) long variables

ii) apply mod in correct manner

$(a + b) \% m =$ $(a \% m + b \% m) \% m$

$(a - b) \% m =$ $(a \% m - b \% m + m) \% m$

$a = 5 \quad b = 2$

$(5 \% m + 2 \% m) \% m$

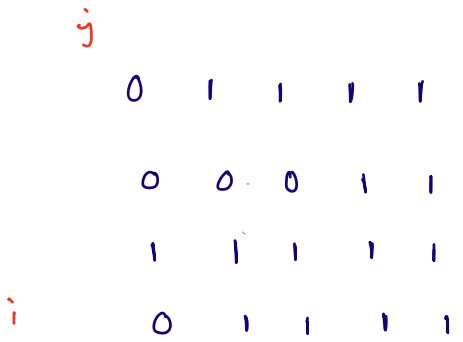
$(5 + 2) \% m = 7$

$(5 \% m - 2 \% m + m) \% m$

$(5 - 2 + m) \% m$

$(3 + m) \% m = 3$

```
int j = m-1 ;
for ( i → 0 to n-1 )
    while (A[i][j] == 1 && j>=0){
        j-- ;
        ans=i ;
    }
}
```



ans = ~~0~~ 2

ans = 0