Today's agenda

- Binary number system
- operators
- Problems
- Constraints

↳ decimal no. system

↳ {0-9} digits

00 10 20 90

01 11 21 91

02 12 22 92

⋮ 13 23 ⋮

09 19 29 ~~~~ 99

→ binary no. system

↳ {0, 1}

0000 → 0 0010 → 2 0100 → 4 1000

0001 → 1 0011 → 3 0101 → 5 1001

0110 → 6 1010

0111 → 7 1011

1100

1101

1110

1111

Conversion

ex: 30

2	30	0
2	15	1
2	7	1
2	3	1
2	1	1
	0	

$(11110)_2$

Quiz:

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

$\rightarrow (101101)_2$

Ly binary no. to decimal no.

$2^4 * 3$ $2^3 * 1$ $2^2 * 1$ $2^1 * 0$ $2^0 * 1$ ← bit index

ex: $(1\ 0\ 1\ 0\ 1)_2$

0: unset bit / off bit

1: set bit / on bit

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

$1 + 0 + 4 + 0 + 16 = 21$

Quiz:

6	5	4	3	2	1	0
1	0	1	1	0	1	0

$2^6 + 2^4 + 2^3 + 2^1 = 90$

$64 + 16 + 8 + 2$

ex: $(1\ 0\ 1\ 2\ 0)_2$ → invalid input

!!

4. Add binary numbers

```
  +1  +1
  2   6   8
  3   5   7
-------------
  6   2   5
```

0 + 0 → 0

0 + 1 → 1

1 + 0 → 1

1 + 1 → 10

1 + 1 + 1 → 11

①

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


Quiz

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

int → ≈ $-2*10^9$ to $2*10^9$ → -2^{31} to $2^{31}-1$ → 32 bits

long → ≈ $-2*10^{18}$ to $2*10^{18}$ → -2^{63} to $2^{63}-1$ → 64 bits

32 bits

$\pm \left[\frac{0/1}{\frac{1}{2}} * \frac{0/1}{\frac{1}{2}} * \frac{0/1}{\frac{1}{2}} \; \frac{}{\frac{1}{31}} \; \cdots \; \frac{0/1}{\frac{1}{2}} * \frac{0/1}{\frac{1}{2}} \right]$ (31, 30, 29, …, 1, 0)

-ve $(2^{31}) \rightarrow (-1 \rightarrow -2^{31})$ ↳ 2^{31}

↳ +ve $(2^{31}) \rightarrow (1 \rightarrow 2^{31}) \rightarrow (0 \rightarrow 2^{31}-1)$

Bitwise operators: { and, or, xor, leftshift, rightshift }

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

0 dominant

1 dominant

same same
Puppy Shame

→ &&
↓
To combine multiple cond^n

```
if( n==3 && n==2){
}
```

false dominant

| 1 dominant
↓
Bitwise operations

(i) 23 & 10

```
        4 3 2 1 0
23:     1 0 1 1 1
10:     0 1 0 1 0
        ---------
        0 0 0 1 0  → 2^1
```

Quiz:

```
20:     1 0 1 0 0
10:     0 1 0 1 0
        ---------
        0 0 0 0 0  → 0
```

(ii) 23 | 10

```
        4 3 2 1 0
23:     1 0 1 1 1
10:     0 1 0 1 0
        ---------
        1 1 1 1 1  → 31
```

Quiz:

```
        4 3 2 1 0
20:     1 0 1 0 0
10:     0 1 0 1 0
        ---------
        1 1 1 1 0  → 2^1 + 2^2 + 2^3 + 2^4 = 30
```

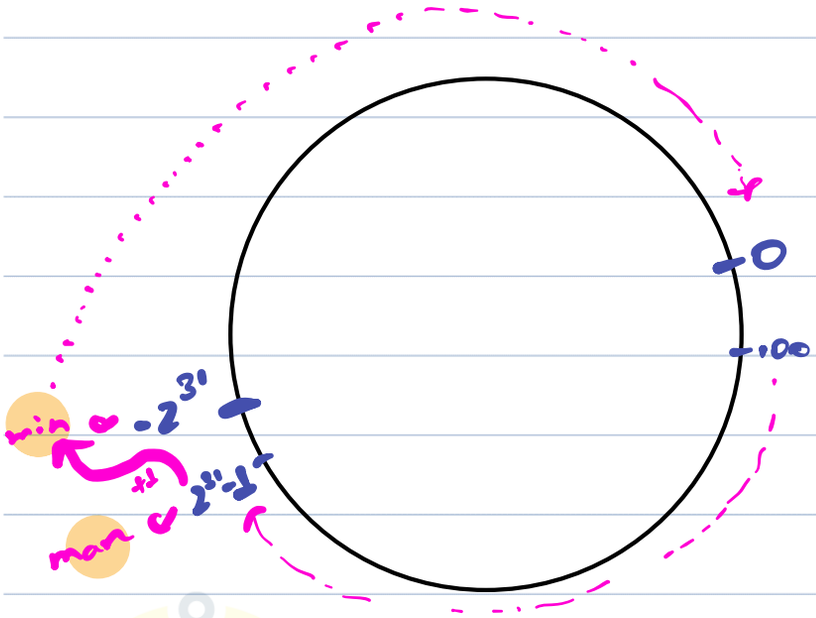
(iii) 23 ^ 10

```
        4 3 2 1 0
23:     1 0 1 1 1
10:     0 1 0 1 0
        ---------
        1 1 1 0 1  → 2^0 + 2^2 + 2^3 + 2^4 = 29
```

Quiz:

```
        4 3 2 1 0
20:     1 0 1 0 0
15:     0 1 1 1 1
        ---------
        1 1 0 1 1  → 2^0 + 2^1 + 2^3 + 2^4 = 27
```

int → 32 bits → $\{-2^{31}$ to $2^{31}-1\}$



int n: $2^{31}-1+1$

-2^{31}

int n: $2^{31}-1+100$

$-2^{31}+1$

$-2^{31}+99$

Break till 9:35 Pm

Q) you have been given a positive no., identify whether the number is even or odd.

Ex: N = 8 → even

N = 7 → odd

Note: use of mathematical operator {+, -, *, /, % & mod is not allowed}

N = 10 : 1 0 1 0

+1 0 0 0 1

N = 11 : 1 0 1 1

+1 0 0 0 1

N = 12 : 1 1 0 0

+1 0 0 0 1

N = 13 : 1 1 0 1

for even no. 0th bit index is always 0

for odd no. 0th bit index is always 1

How to check last bit?

int N = 10:

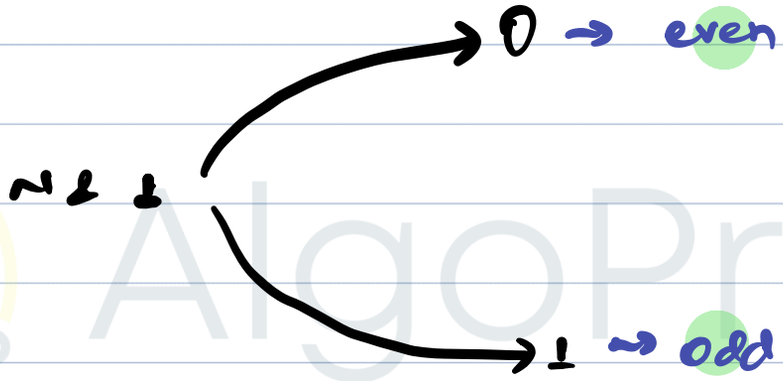
```
  3 2 1 0
  1 0 1 0
& 0 0 0 1
-------
  0 0 0 0 → 0
```

N & 1 = 0

N = 11:

```
  3 2 1 0
  1 0 1 1
& 0 0 0 1
-------
  0 0 0 1 → 1
```

N & 1 = 1




```
void checkevenodd(int n){
    if(n & 1 == 0){
        s.o.p("even");
    }
    else{
        s.o.p("odd");
    }
}
```

Properties

(i) Commutative Property

$a \& b = b \& a$

$a | b = b | a$

$a \hat{} b = b \hat{} a$

(ii) Associative Property

$a \& b \& c \rightarrow (a \& b) \& c = a \& (b \& c)$

$a | b | c \rightarrow (a | b) | c = a | (b | c)$

$a \hat{} b \hat{} c \rightarrow (a \hat{} b) \hat{} c = a \hat{} (b \hat{} c)$

(iii)

(a) $N \& N = N$

$N \rightarrow 101$
$\&$
$N \rightarrow 101$
101

$N \rightarrow 101$
$\&$
$0 \rightarrow 000$
000

(b) $N \& 0 = 0$

(c) $N | 0 = N$

$N \rightarrow 101$
$|$
$0 \rightarrow 000$
101

$N \rightarrow 101$
$|$
$N \rightarrow 101$
101

(d) $N | N = N$

(e) $N \hat{} 0 = N$

$N \rightarrow 101$
$\hat{}$
$0 \rightarrow 000$
101

$N \rightarrow 101$
$\hat{}$
$N \rightarrow 101$
000

(f) $N \hat{} N = 0$

Q) Given arr[N], every element appears twice except for one element which appears once, find that unique element.

Eg: arr[7]: { 6 8 8 7 7 10 6 } → 10

arr[5]: { 2 1 9 2 9 } → 1

arr[7]: { 6 8 12 7 7 10 6 } → invalid input

Idea1

↳ nested loop

i:=0 → j:=[0,1,2,3,4,5,6] i:=1 → j:=[0,1,2,3,4,5,6]

arr[7]: { 6 8 8 7 7 10 6 }

T.C: O(N²)

Idea2

arr[7]: { 6^8^8^7^7^10^6 }

inc. order → { 6^6^7^7^8^8^10 }

temp = 6^6 = 0^7 = 7^7 = 0^8 = 8^8 = 0^10 = 10

↳ you don't need to generate inc. order.

temp = 6^8 = 14^6 = 8^7 = 15^8 = 7^10 = 13^10

arr[7]: { X X X X X 10 X }

i: 0 1 2 3 4 5 6

14 8 15 7 13

6: 0 1 1 0

8: 1 0 0 0

1 1 1 0 → 14

14: 1 1 1 0

6: 0 1 1 0

1 0 0 0 : 8

8: 1 0 0 0

7: 0 1 1 1

1 1 1 1 → 15

15: 1 1 1 1

8: 1 0 0 0

0 1 1 1 → 7

7: 0 1 1 1

10: 1 0 1 0

1 1 0 1 → 13

13: 1 1 0 1

7: 0 1 1 1

1 0 1 0 → 10

Pseudo code

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

T.C: O(n)

S.C: O(1)

Left Shift (<<)

a = 45 :

6	5	4	3	2	1	0
0	1	0	1	1	0	1

Qui2 :

1 << n

Ls right shift (>>) → O(1)

	4	3	2	1	0
a : 10 :	0	1	0	1	0

Constraints