Today's Content :-

→ Number System basics

→ Binary to decimal
→ Add 2 binary no's.
→ Bitwise operators.

Quote :-

It's not the will to win that matters, everyone has that.
It's the will to prepare to win that matters.

Number System basics :-

↳ Decimal No. System → [0-9] → [10]

7 3 4 : $7*100 + 30 + 4$

10^2 10^1 10^0

6 5 9 4 : $6*1000 + 5*100 + 90 + 4$

10^3 10^2 10^1 10^0

2 45 : $2*100 + 40 + 5$

Other number systems :-

binary , ternary ,

hexa

Octal

Number system → [0-7] → [8]

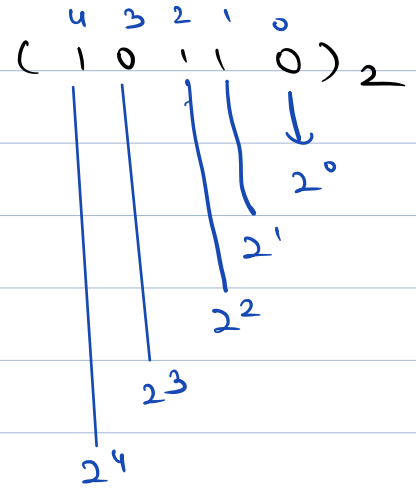
Octal $(125)_8 \rightarrow 1*8^2 + 2*8^1 + 5*8^0$

8^2 8^1 8^0

$\Rightarrow 85$

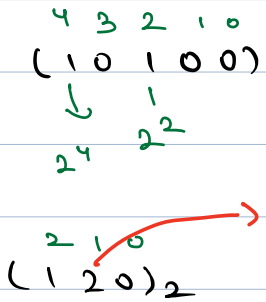
Binary → every digit [0 - 1]

→ base [2].

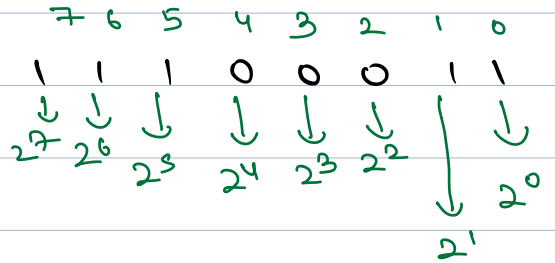


$= 2^4 + 0 + 2^2 + 2^1 + 0 \Rightarrow 22.$

e.g) $(1\,0\,1\,0\,0)_2 \rightarrow 2^4 + 2^2 \Rightarrow 20$



Invalid binary representation

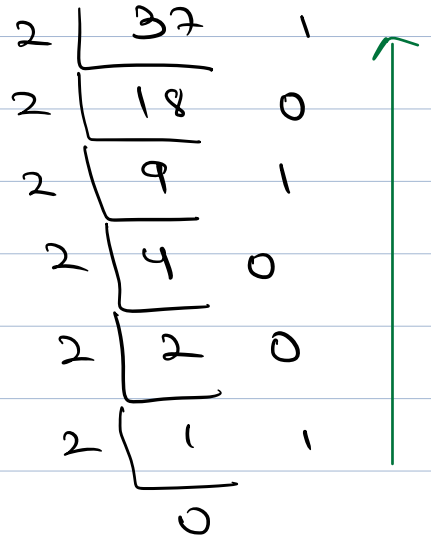


$\Rightarrow 2^7 + 2^6 + 2^5 + 2^1 + 2^0$

$\Rightarrow 128 + 64 + 32 + 2 + 1$

$\Rightarrow 227$

Decimal to Binary.

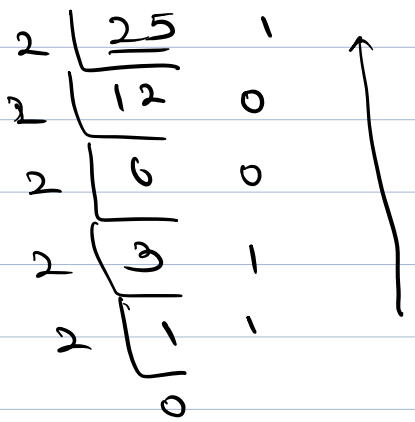


5 4 3 2 1 0
→ 1 0 0 1 0 1

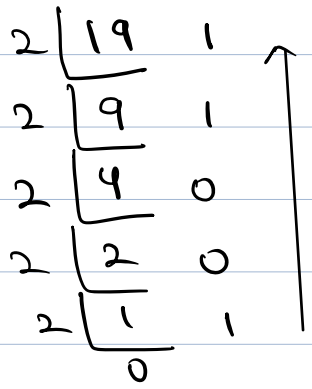
2^5 2^4 2^3 2^2 2^1 2^0

$\Rightarrow 2^5 + 2^2 + 2^0$

$\Rightarrow 37.$



$(1\ 1\ 0\ 0\ 1)_2$



$\rightarrow (1\ 0\ 0\ 1\ 1)_2$

$(0\ 2\ 1\ 0\ 1)_3 \rightarrow$

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

$= 0 + 54 + 9 + 0 + 1 \Rightarrow 64$

$(1\ 3\ 1)_8 \rightarrow 1*8^2 + 3*8^1 + 1*8^0$

$\Rightarrow 64 + 24 + 1$

89

$(9\ A)_{16} \rightarrow 9*16^1 + (A)*16^0$ (A = 10)

$\rightarrow 16*9 + 10*1$

$\Rightarrow 154$

Add 2 Decimal numbers :-

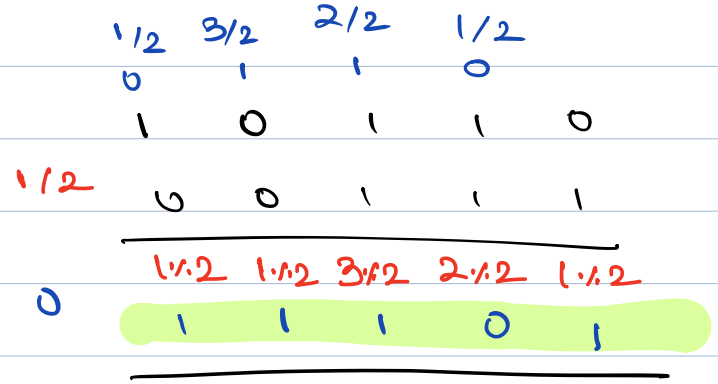
	1	1	
	7	8	9
	1	4	2
0	9	3	1

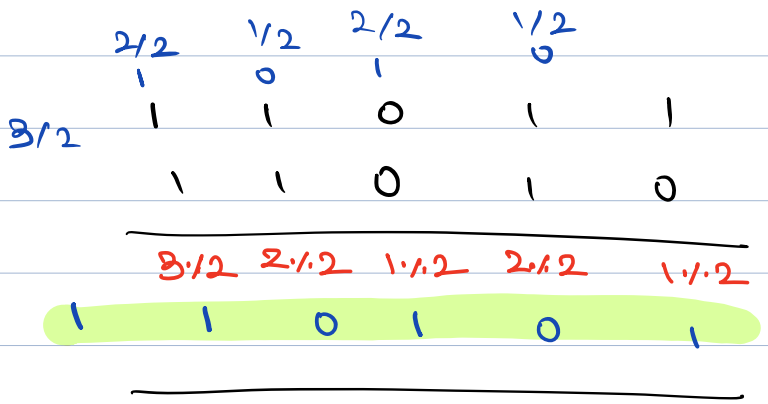
	12/10	8/10	12/10	
11/10	7	8	3	9
	3	9	4	8
	11%10	17%10	8%10	
1	1	7	8	7

d = 8%10,

c = 5/10

Add 2 binary NO's, d = 8½
c = 8/2





∴ why binary :-

→ > : ~ : 1
< = ~ : 0

int x = 25
↳ machine stores it in binary.

x = 1 1 0 0 1

print(x)

Bitwise operators (&, |, ^, ~, <<, >>)

And, or, xor, invert, left shift, right shift

Truth table :-

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

xor :- Addition without carry

```
  0      0      1      1   <- don't consider carry
  0      1      0      1
 ---    ---    ---    ---
  0      1      1      0
```

2 / 2 = 1, 2 % 2 = 0

Some basic problems on Bitwise

a = 29, b = 19.

a : 1 1 1 0 1

b : 1 0 0 1 1

print(a & b) 1 0 0 0 1 → 17

print(a | b) 1 1 1 1 1 → 31

print(a ^ b) 0 1 1 1 0 → 14

a : 1 1 0 1

b : 1 0 1 0

a & b 1 0 0 0

a | b 1 1 1 1

a ^ b 0 1 1 1

Properties :-

a = 10 : 1 0 1 0

& 1 : 0 0 0 1

print(a&1) 0 0 0 0 → 0

a = 11 : 1 0 1 1

1 0 0 0 1

print(a&1) 0 0 0 1 → 1

a = 14 : 1 1 1 0

1 0 0 0 1

print(a&1) 0 0 0 0 → 0

a = 13 : 1 1 0 1

1 0 0 0 1

print(a&1) : 0 0 0 1 → 1

// Observation

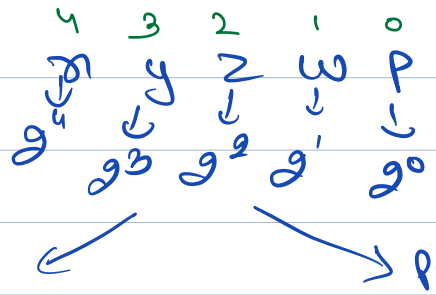
print(10&1) = 0

print(14&1) = 0

print(11&1) = 1

print(13&1) = 1

```
if(a&1==0){
  a is even
}
if(a&1==1){
  a is odd
}
```



p = 0

$(x*2^4 + y*2^3 + z*2^2 + w*2^1 + 0)$

p = 1

$x*2^4 + y*2^3 + z*2^2 + w*2^1 + 1$

Basic properties :-

a & 1 → even : 0, odd : 1

a & 0 :- 0

a & a :- a

a | 1 = Todo

a | 0 = a

a | a = a

a ^ 0 = a

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

a ^ a = 0

a ^ 1 → even : a+1, odd : a-1

a = 1 0 1 0 0 → 20
^ 1 = 0 0 0 0 1
1 0 1 0 1 → 21

a = 1 1 0 0 1 → 25
1 = 0 0 0 0 1
1 1 0 0 0 → 24

10:36 → 10:46 pm :-

1/2 0/2 1/2
0 0 0

1 1 0 0 0 → 24
0 0 0 0 1 → 1

1+1=2 1+1=2 0+0=2 0+0=2 1+1=2
1 1 0 0 1

After learning Bit Basics,

Students and me:-

Bitwise properties :-

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

$a \& b = b \& a$

$a | b = b | a$

$a \wedge b = b \wedge a$

Commutative

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

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

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

Associative.

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

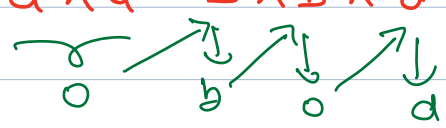
$(a | b | c) = (c | b | a) = (b | a | c)$

$(a \wedge b \wedge c) = (c \wedge b \wedge a) = (b \wedge a \wedge c)$

Ques 5 numbers :-

$a \wedge b \wedge a \wedge d \wedge b =$

$a \wedge a \wedge b \wedge b \wedge d \rightarrow d$



Ques) 7 numbers :-

$e \wedge f \wedge a \wedge f \wedge e \wedge g \wedge a = g$

Ques Given N array elements, every element repeats twice except 1, find unique element.

ar[5] = { 6, 9, 6, 10, 9 } → 10

ar[7] = { 12 9 12 8 7 9 8 } → 7

idea 1 :-

1) for every element, iterate on array & get its frequency.

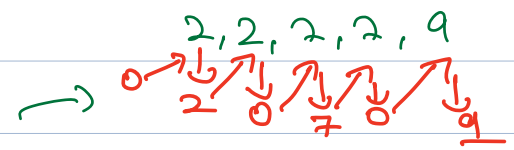
```
for (i=0; i<N; i++) {
    for (j= ; j< ; j++) {
```

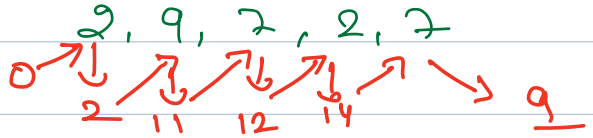
=

```
    }
}
```

T.C → O(N²)

S.C → O(1)





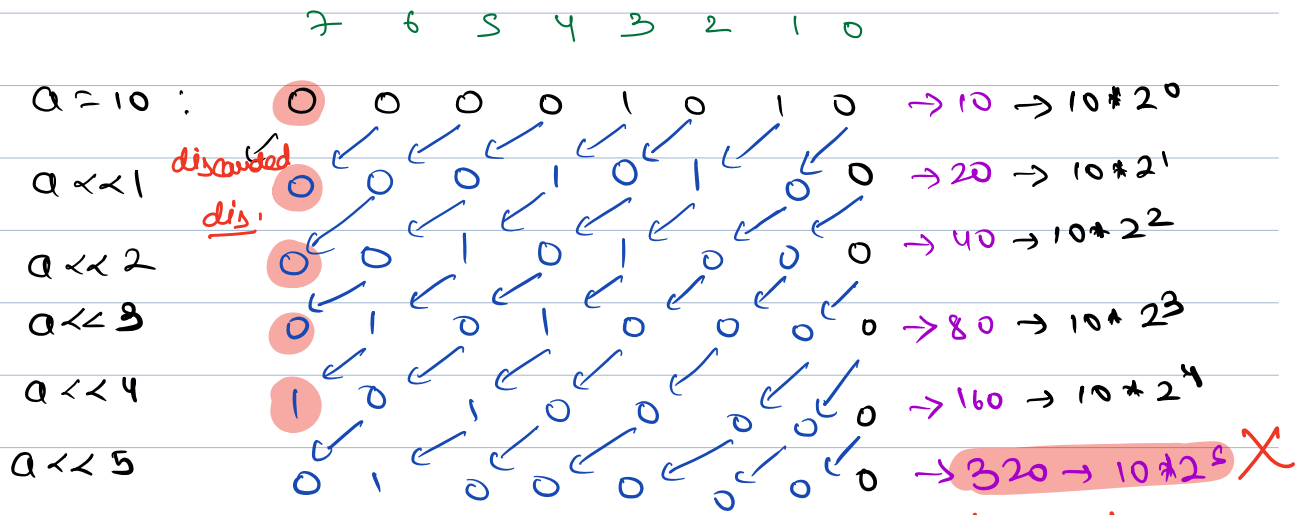
idea 2 :-

```
ans = 0
i=0; i<N; i++){
    ans = ans ^ arr[i]
}
return ans;
```

T.C → O(N), S.C → O(1)

// left shift :-

8 bit number



we are losing data

query/ies.

a << 1 ⇒ a * 2^1

a << 2 ⇒ a * 2^2

a << 3 ⇒ a * 2^3

a << N ⇒ a * 2^N