Bit Manipulation I

Bit Operations

1 → true, set bit

0 → false, unset bit

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

A	~A (NOT)
0	1
1	0

---

45 | 10

45 → 1 0 1 1 0 1

10 → 0 0 1 0 1 0

OR

1 0 1 1 1 1

5 4 3 2 1 0

$2^5 + 2^3 + 2^2 + 2^1 + 2^0 = 47$

Properties

A = 5 (101)

1. $A | 0 = A$

2. $A | A = A$

3. $A \wedge 0 = A$

```
 101
^000
 101
```

4. $A \wedge A = 0$

```
 101
^101
 000
```

5. $A \& 0 = 0$

6. $A \& A = A$

7. Odd / Even

4 → 100
6 → 110
8 → 1000
10 → 1010

Even → 0

5 → 101
7 → 111
9 → 1001
11 → 1011

Odd → 1

check last bit →

A & 1		
	→ 0	even no.
	→ 1	odd no.

A = 5

```
 101
&001
 001
```

A = 20

```
 10100
&00001
 00000
```

8. Commutative Property

$A \& B = B \& A$

$A | B = B | A$

$A \wedge B = B \wedge A$

9. Associative Property

$(A \& B) \& C = A \& (B \& C)$

$(A | B) | C = A | (B | C)$

$(A \wedge B) \wedge C = A \wedge (B \wedge C)$

Question

$a \wedge b \wedge a \wedge d \wedge b \Rightarrow (a \wedge a) \wedge (b \wedge b) \wedge d$

$\Rightarrow (0 \wedge 0) \wedge d$

$\Rightarrow 0 \wedge d \Rightarrow d$

Question

$1 \wedge 3 \wedge 5 \wedge 3 \wedge 2 \wedge 1 \wedge 5 \Rightarrow (1 \wedge 1) \wedge (3 \wedge 3) \wedge (5 \wedge 5) \wedge 2$

$= 0 \wedge 0 \wedge 0 \wedge 2 = 2$

Q Given an integer array where every no. occurs twice except one number. find that unique no.

Soln → ans = $\forall i, \wedge a[i]$

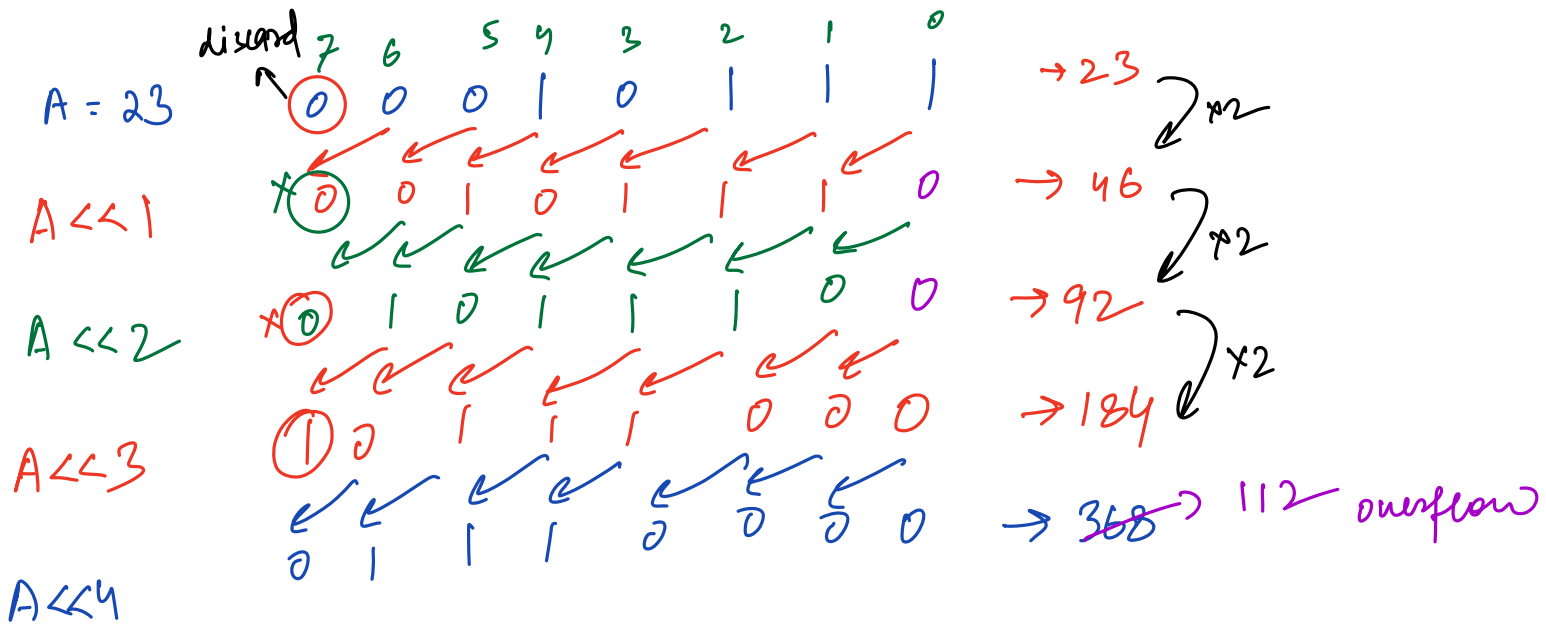
```
ans = a[0]
for (i=1 to n-1) {
    ans ^= a[i]
}
return ans
```

TC : O(N)

SC : O(1)

Left Shift (<<)

for explaining → 8 bit system → 0 - 255



$2^6 + 2^5 + 2^4 = 64 + 32 + 16$

$= 112$

$N << 1 = N * 2$
$N << K = N * 2^K$

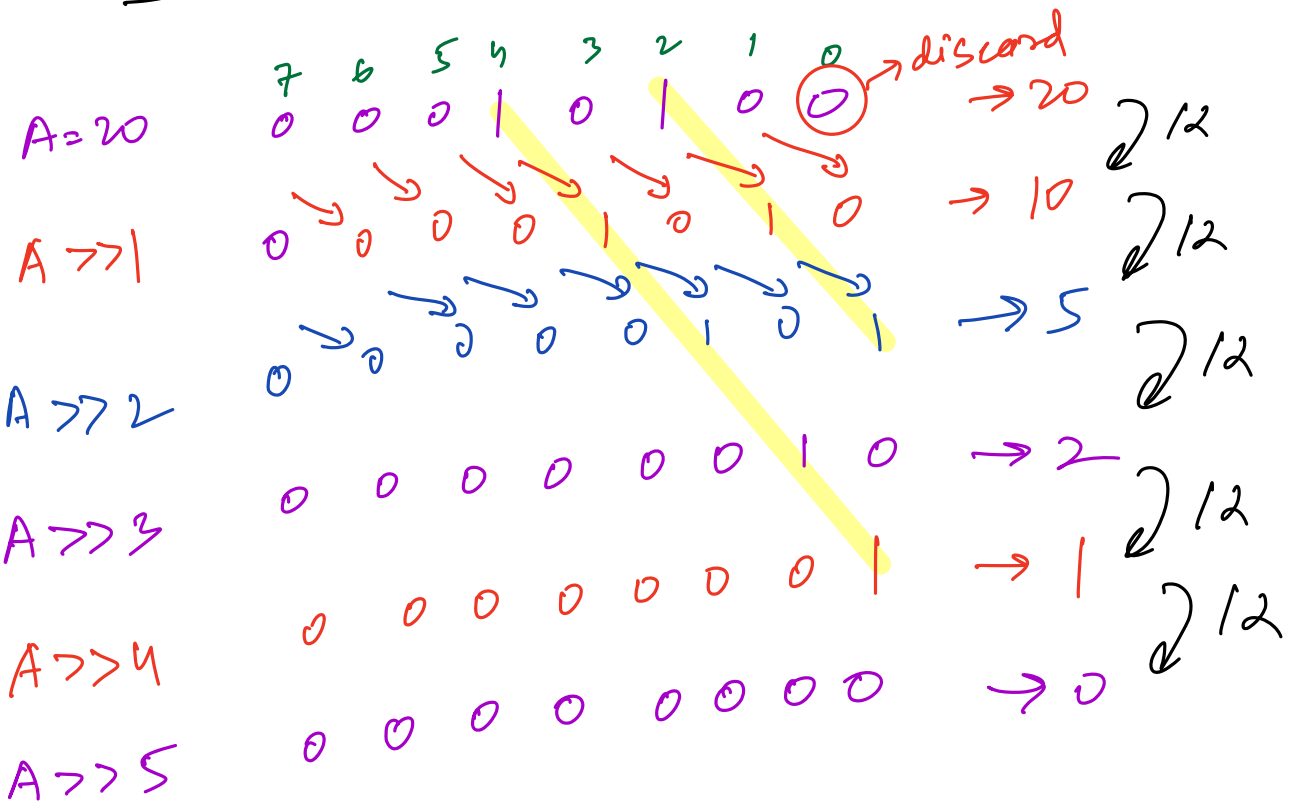
$1 << n = 2^n$

take care of overflow

$x * 2 > INT\_MAX$

$x > \frac{INT\_MAX}{2}$ ⇒ overflow

Right Shift (>>)



$N >> 1 = N/2$
$N >> K = N/2^K$

no overflow

$1 << 3 \Rightarrow 2^3 = 8$

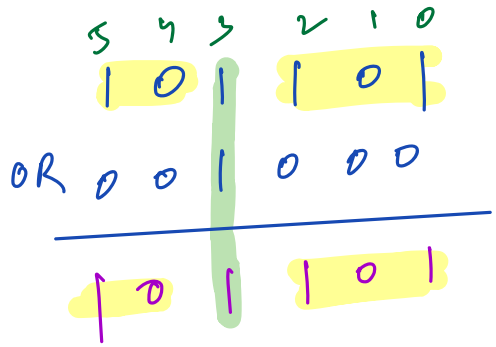
Power of left shift operator

$(1<<K) = 2^K$ — 000100000 (K^{th} bit) (only K^{th} bit is set)

1. OR — $N | (1<<K)$ → set K^{th} bit of N

N = 45

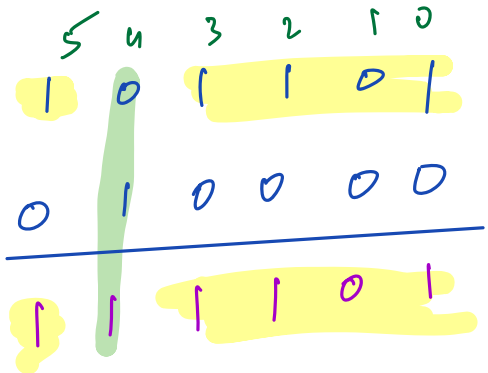
K = 3



→ 45 (K^th bit is already 1 ⇒ no change)

N = 45

K = 4



→ 61 (else → N+2^K)

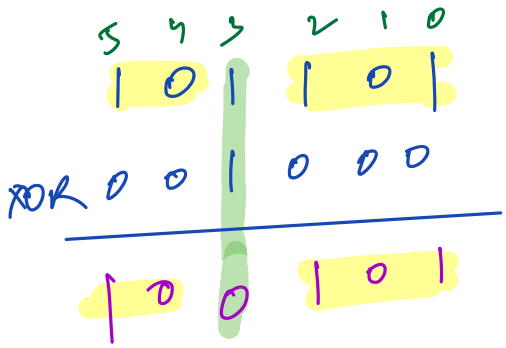
2. XOR — $N \wedge (1<<K)$ → toggle K^{th} bit

1 → 0

0 → 1

N = 45

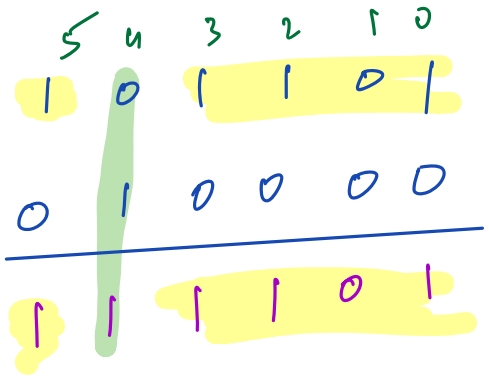
K = 3



→ 37 (already 1 ⇒ N-2^K)

N = 45

K = 4



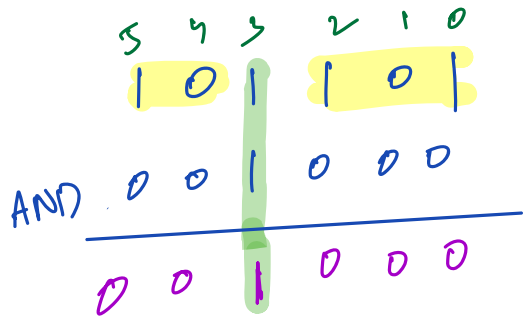
→ 61 (else ⇒ $N+2^K$)

3. AND $N\&(1<<K)$ → 2^K (K^{th} bit is set)

→ 0 (K^{th} bit is unset)

N = 45

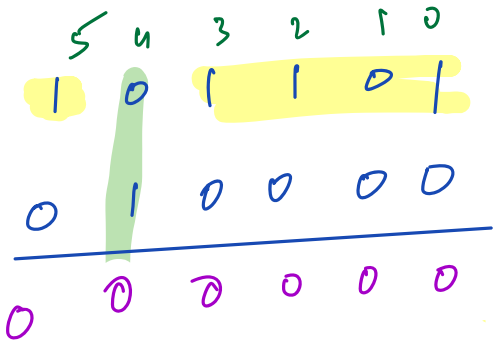
K = 3



→ 2^3 (2^K) [K^{th} bit is 1 ⇒ 2^K]

N = 45

K = 4



→ 0 [else ⇒ 0]

Question 1 for any number N,

a. check if K^{th} bit is set ?

$x = N\&(1<<K)$

if ($x>0$) ⇒ K^{th} bit is set

else ⇒ not set

```
def checkBit(N, K) {
    if((N & (1<<K)) > 0)
        return true;
    return false
}
```

TC: O(1)
SC: O(1)

Question 2

Count the # sets bits in N.
= ↳ integer → 32 bits

```
count=0
for(i: 0 to 31) {
    if(N & (1<<i) > 0)
        ++count
}
return count;
```

TC: O(1)
SC: O(1)

```
count=0
while(N>0) {
    if(N & 1 == 1) count++
    N = N >> 1    // N = N/2
}
```

TC: O(log N)
SC: O(1)

return count

$N \to \frac{N}{2} \to \frac{N}{2^2} \to \dots \to \frac{N}{2^K} = 1 \quad \Rightarrow 2^K = N$

$K = \log_2 N$

Question 3

Unset i^{th} bit of a number, if it is set.

eg $N = 12$ (1100) $\quad i = 2$

$\Rightarrow 1000 = 8$

XOR $\Rightarrow$ $0 \to 1$, $1 \to 0$

```
if (N & (1<<K) == 0)   // unset
                       // do nothing
    return N

else                   // set,
    return N ^ (1<<K)  // do toggle
```

Question 4 Create a binary number with specific pattern.

The pattern is : A 0's followed by B 1's followed by C 0's.

$A, B, C <= 20$

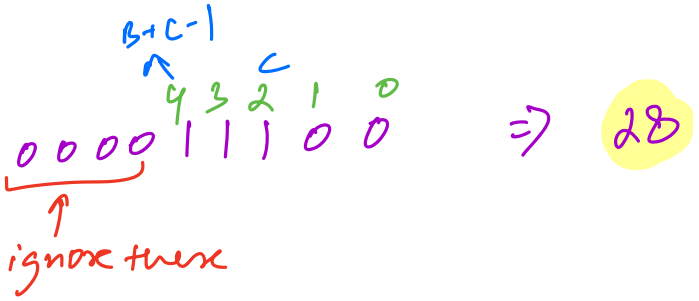
A, B, C are input.

Return the decimal value of answer.

Eg A = 4

B = 3

C = 2



⇒ 28

1. Ignore first A 0's.

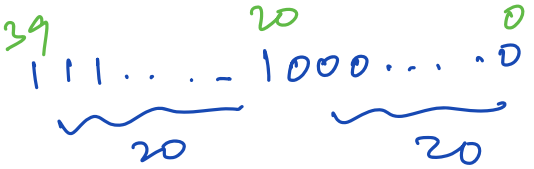
```
long ans=0
for(i=C to B+C-1) {
    ans |= (1<<i)
}
return
```

TC: O(B)

SC: O(1)

Since A, B, C <= 20

max ans ? ⇒



$2^{39} + 2^{38} + \ldots + 2^{20}$

use long

[int ~ 10^9 ~ 2^{30}]

[long ~ 10^{18} ~ 2^{60}]

$2^3 - 1 = 7 = 111$

$2^4 - 1 = 15 = 1111$

$2^B - 1 = \underbrace{11\ldots1}_{B \text{ times}}$

(1<<B) - 1

!!

((1<<B) - 1) << C

$\Rightarrow$ $\underbrace{11111}_{B}\underbrace{000\ldots0}_{C}$