Today's Content

Set ith bit
unset ith bit -> Try your own.
check Bit
Count Bit
Negative numbers.
Ranges
Importance of Constraints
unset or continuous Bits in N

↓

If time allows.

Ques) Given N, i, check if ith bit in N is set or unset.

set → 1
unset → 0

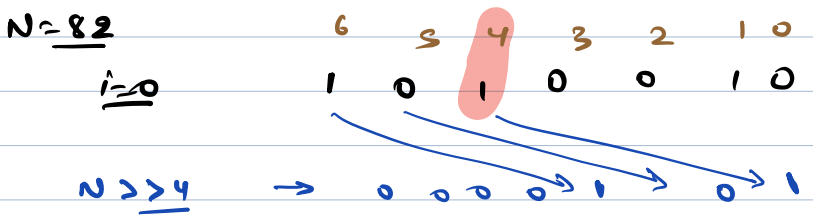
N = 21 : 1 0 1 0 1 → True.
i = 2

Brute force :- Convert to Binary, store it and then check.

if 0th bit is 0 → (N & 1) == 0

else 1

N = 82
i = 4



((N >> 4) & 1 == 0) → 4th bit was unset

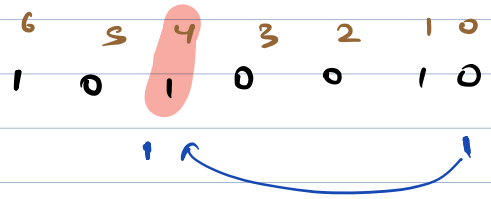
else it was set.

```
bool checkBit(int N, int i){
    if ((N>>i) & 1) == 0){
        return false
    } else {
        return True }
```

T.C → O(1).

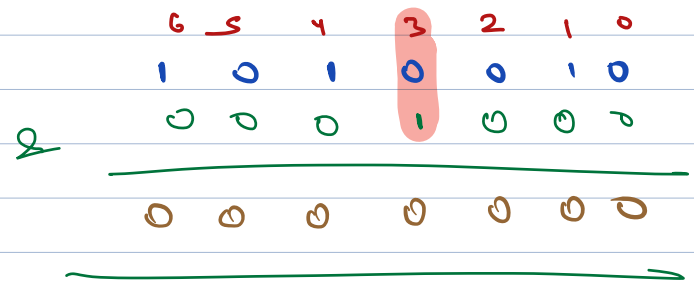
3

N=82

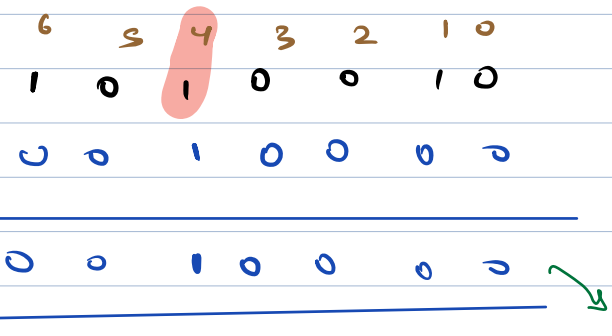


i=3 =>

1<<3



N=82



1<<4.

when, (N & 1<<i) = 0 (The bit was 0)

↳ (1<<i) (That bit was 1)

Count Set Bits :-

		output
N = 10,	1 0 1 0	: 2
N = 27,	1 1 0 1 1	: 4

Approach 1 :- Convert to Binary and count 1's.

Approach 2 :- 32 bits. [0 - 31]

```
int c = 0;
for (i = 0; i < 32; i++) {
    if (checkBit(N, i)) {
        c++
    }
}
return c;
```

T.C → O(1).
S.C → O(1).

Approach 3 :-

N = 45 c = 0.

5	4	3	2	1	0	
1	0	1	1	0	1	c++
0	1	0	1	1	0	
0	0	1	0	1	1	c++
0	0	0	1	0	1	c++
0	0	0	0	1	0	
0	0	0	0	0	1	c++
0	0	0	0	0	0	

c = 4 Ans.

```
int CountSetBits(int N){
    int c=0;
    while(N>0){
        if(N&1==1){
            c++;
        }
        N=N>>1 → ow;
    }
    return c;
}
```

T.C → (log N)
S.C → O(1).

* How -ve no's are stored.

8 bit numbers.

↑ Sign Bit

	7	6	5	4	3	2	1	0
+10 →	0	0	0	0	1	0	1	0
-10 →	1	0	0	0	1	0	1	0

	7	6	5	4	3	2	1	0
4 :	0	0	0	0	0	1	0	0
-4 :	1	0	0	0	0	1	0	0
+ 10 :	0	0	0	0	1	0	1	0
-14 :	1	0	0	0	1	1	1	0

1 more problem:-

1 0 0 0 0 0 0 0 0 → -0

0 0 0 0 0 0 0 0 0 → 0

2's complement :-

$-a$ = 2's complement of a.

= 1's complement of $a+1$.

$0 \rightleftharpoons 1$

-10 :- 2's complement of 10

= 1's complement of $10+1$

$= \sim(00001010) + 1$

$\rightarrow \quad 11110101$

$+ \quad \quad 1$

$11110110 \rightarrow -10$

$-2^7 \quad 2^6 \quad 2^5 \quad 2^4 \quad 2^3 \quad 2^2 \quad 2^1 \quad 2^0$

$7 \quad 6 \quad 5 \quad 4 \quad 3 \quad 2 \quad 1 \quad 0$

$1 \quad 1 \quad 1 \quad 1 \quad 0 \quad 1 \quad 1 \quad 0$

$\rightarrow \quad -2^7 + 2^6 + 2^5 + 2^4 + 2^2 + 2^1 \Rightarrow$

$-128 + 64 + 32 + 16 + 4 + 2$

$\Rightarrow \quad -10.$

$10 = \quad 0\ 0\ 0\ 0\ 1\ 0\ 1\ 0$

$+ \quad -4 = \quad 1\ 1\ 1\ 1\ 1\ 1\ 0\ 0$

$1\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0 \rightarrow 6$

$0\ 0\ 0\ 0\ 0\ 1\ 0\ 0$

$\rightarrow \quad 1\ 1\ 1\ 1\ 1\ 0\ 1\ 1$

$\rightarrow \quad \quad \quad \quad +1$

$1\ 1\ 1\ 1\ 1\ 1\ 0\ 0$

// 4 bit no

3	2	1	0
-2^3	2^2	2^1	2^0

8 bit no.

-2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0

→ Most significant bit

MSB value is -ve.

(*) Convert Binary to decimal :-

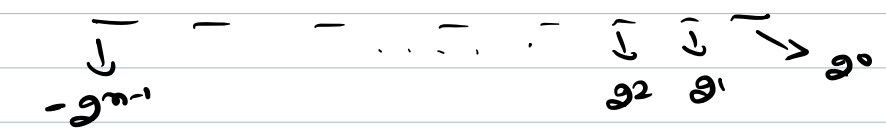
4 bit No

-2^3	2^2	2^1	2^0	
1	0	1	1	→ -5
1	0	1	0	→ -6
0	0	1	1	→ 3
1	0	0	0	→ -8
1	1	1	1	→ -1

(*) 8 bit numbers :-

-2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0		Decimal
0	0	0	1	0	1	0	1	→	21
1	0	0	1	0	1	0	1	→	-107
1	0	0	1	0	0	0	1	→	-111
0	0	0	1	0	0	0	1	→	17

n Bit numbers.



Max neg :- 1 0 0 0 0 . . . → -2^{n-1}

Max +ve :- 0 1 1 1 1 1 . . . 1 1 → $2^{n-1}-1$

2^{n-2} 2^1 2^0

s.p → $2^0 + 2^1 + 2^2 + \dots 2^{n-2}$

$a = 1,\ r = 2,$

terms = $n-1$

$\left(\frac{a(r^{n-1}-1)}{r-1}\right)$

$1\left(\frac{2^{n-1}-1}{2-1}\right) = 2^{n-1}-1$

Range of n bit no → $[-2^{n-1}$ to $2^{n-1}-1]$

	Bytes	bits	
byte/char	1	8	$[-128$ to $127]$
int	4	32	$[-2^{31}$ to $2^{31}-1]$
			close approx → $[-2*10^9$ to $2*10^9]$
long	8	64	$[-2^{63}$ to $2^{63}-1]$

close Approx $[-8*10^{18}$ to $8*10^{18}]$

$2^{10} = 1024 \simeq 10^3 \Rightarrow 2^{30} = 10^9 \rightarrow 2^{31} = 2*10^9$

$2^{10} \simeq 10^3 \rightarrow (2^{30})^2 \simeq (10^9)^2 \longrightarrow 2^{60} \overset{*2^3}{\simeq} 10^{18}$

$2^{63} \simeq 8*10^{18}$

2 bit numbers :-

-2^1	2^0	
0	0	→ 0
0	1	→ 1
1	0	→ -2
1	1	→ -1

3 bit nos :-

[min -4 , max 3]

-2^2	2^1	2^0	
0	0	0	→ 0
0	0	1	→ 1
0	1	0	→ 2
0	1	1	→ 3
1	0	0	→ -4
1	0	1	→ -3
1	1	0	→ -2
1	1	1	→ -1

Importance of Constraints :-

1) Given an array calc sum of it.

```
long sum (int arr[]) {
    long sum=0;
    for (i=0; i<n; i++) {
        sum= sum+ arr[i]
    }
    return sum
}
```

Constraints :-

$1 <= N <= 10^5$

$1 <= arr[i] <= 10^6$

$1 <= sum <= 10^{11}$

Constraints
↓
V.V. Important

→ TLE → TLE.

→ Datatypes → Error

Ques) Given 2 numbers a & b, return their prod.

```
prod (int a, int b) {
    int c = a * b
    long c = a * b
    long c = long (a * b)
    long c = (long) a * b;
    return c;
}
```

3

$1 <= a, b <= 10^6$

Take care :-

→ when we multiply int * int
or
long * long.

(*) Unsigned variables :- MSB weightage will be +ve.

C/ C++ / C#.

unsigned int x; → 32 bits → $(0, 2^{32}-1)$

↳ that x variable can't store a neg. no.

In java there is no such thing as unsigned.

Ques) Given x, y set x continuous bits & y unset bits.

x = 3, y = 2, 11100 → 28

x = 5, y = 3, 11111000

---

3 bit no.
111 → 7
1000 → 8 (1<<3)-1

4 bit no →
1111 → 15
10000 → 16 (1<<4)-1

11111 → (1<<5)-1

T.C →
↳ O(1)

((1<<n)-1)<<y

Q1 100 coins lying on a table.
10 of them are heads up 90 are on tails side.

split in 2 pile, such that there are same no of heads in each pile.

90 { heads → h, Tails → t }

heads → 90-h, tails → 10-h } flip } 10 coins,

It is allowed, you can flip all coins of one pile at most once.

$h = 90 - t \Rightarrow h + t = 90$.

100 → 10 h, 90 T

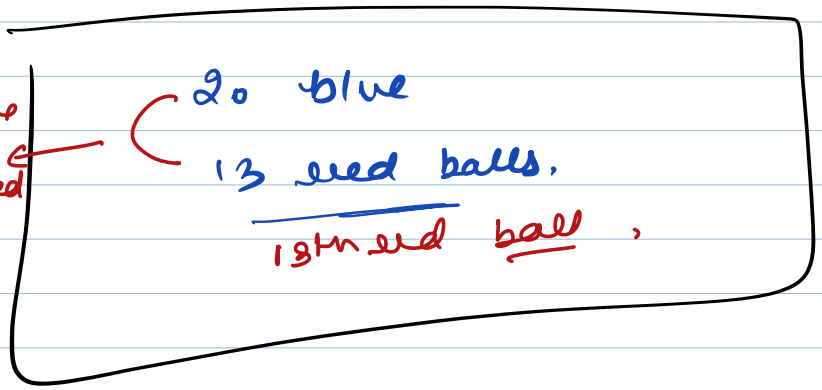
Part 1 → 1 heads, 89 tails

Part 2 → 9 heads, 6 tails

100 → 10 h, 90 T

Part 1 → 0 heads, 90 tails

10 heads, 0 tails.



Same color → blue ball, (diff → red ball.

blue

O O

↓

-1 blue

2 red balls

↓

-2 red

+1 blue

blue / red

-1 blue.