| &   ^   ~

<< left   >> right

$1000_2 = 2^3$

$a = 10$: 0 0 0 0 1 0 1 0 $\Rightarrow 10 \quad 10 \times 2^0$

$a << 1$: 0 0 0 1 0 1 0 0 $\Rightarrow 20 \quad 10 \times 2^1$

$a << 2$: 0 0 1 0 1 0 0 0 $\Rightarrow 40 \quad 10 \times 2^2$

$a << 3$: 0 1 0 1 0 0 0 0 $\Rightarrow 80 \quad 10 \times 2^3 \quad 2^4 + 2^2$

$2^5 + 2^3 = 2(2^4 + 2^3)$

$a << 4$: 1 0 1 0 0 0 0 0 $\Rightarrow 160 \quad 10 \times 2^4$

$a << 5$: 0 1 0 0 0 0 0 0 $\Rightarrow 64$ (overflow)

$$a << n = a \times 2^n$$

(Assuming no overflow)

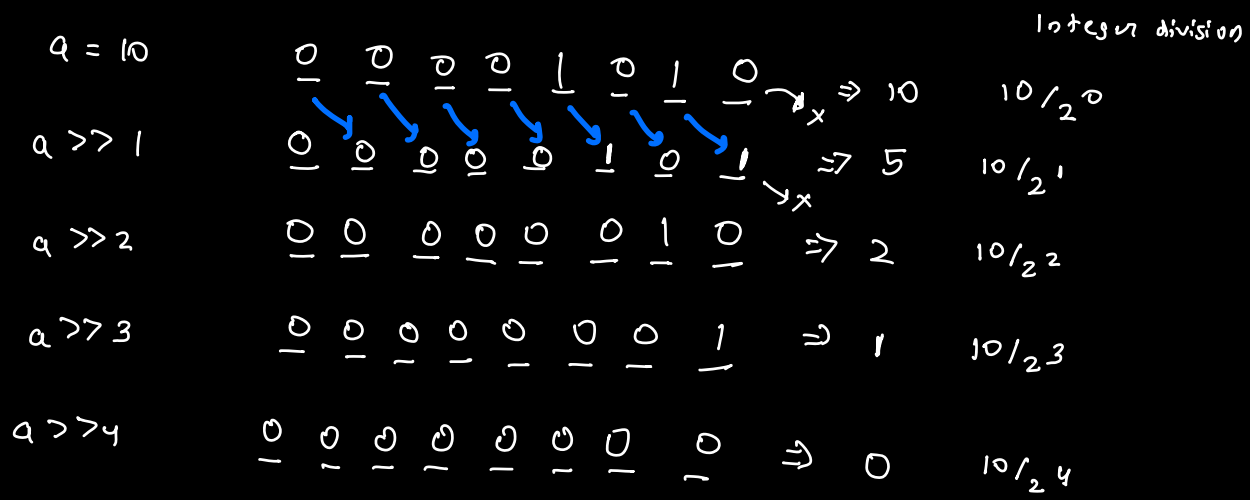
$a = 1$

$1 << 1 = 1 \times 2^1 = 2$

$1 << 2 = 1 \times 2^2 = 2^2$

$1 << 3 = 1 \times 2^3 = 2^3$

$$1 << n = 2^n$$



$$a >> n = \frac{a}{2^n}$$

Quiz 1 $15 << 2$

0 0 0 0 1 1 1 1

0 0 1 1 1 1 0 0 ⇒ 60

32 bit

$$15 << 2 = 15 \times 2^2 = 60$$

Quiz 2 $29 >> 2$

$$= \frac{29}{2^2} = 7$$

Quiz 3

$2^n = ?$

$a << n = a \times 2^n$

$2 << n = 2 \times 2^n = 2^{n+1}$

$1 << n = 1 \times 2^n = 2^n$ ✓

Quiz 4

$5^n = ?$

$5 * (1 << n) = 5 \times 2^n$ ✗

$5 << n = 5 \times 2^n$ ✗

$5 << (n-1) = 5 \times 2^{n-1}$ ✗

Q. Given 2 positive no. N & i. Check if i^{th} bit in N is set or unset.

1 → set
0 → unset

N = 4
i = 1

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

return false

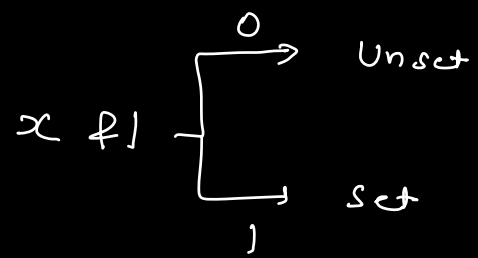
N = 10

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

i = 2 → false

i = 3 → True

Check if the rightmost bit is set / unset



N : 0 0 0 0 1 0 1 0 (bits 7 6 5 4 3 2 1 0)

i = 3

N >> 1 : 0 0 0 0 0 1 0 1

N >> 2 : 0 0 0 0 0 0 1 0

N >> 3 : 0 0 0 0 0 0 0 1

$$(N >> i) \,\&\, 1$$

```
boolean checkbit(N, i)
{
    if ((N>>i) & 1 == 0)
        return False
    else
        return True
}
```

N : 0 0 0 0 1 0 1 0 (bits 7 6 5 4 3 2 1 0)

0 0 0 0 0 0 0 1

&

0 0 0 0 0 0 0 1

---

0 0 0 0 0 0 0 1

```
  00001010
& 00001000
----------
  00001000
```

```
N & (1<<i)
```

0 → unset

2^i → set

```
boolean checkbit(N, i)
{
    if (N & (1<<i) == 0)
        return False
    else
        return True
}
```

```
    7 6 5 4 3 2 1 0
N = 1 0 1 0 0 1 0 0
&   0 0 0 0 0 1 0 0
-----------------
    0 0 0 0 0 1 0 0
```

Q. Given two number N, i. Set the i^{th} bit.

N = 4
i = 3

```
7 6 5 4 3 2 1 0
0 0 0 0 0 1 0 0
```

00001100 = 12 Ans.

0 | 1 = 1

1 | 1 = 1

```
  00001101
| 00001000
----------
  00001101
```

```
int setbit ( N, j)
{
    return  N | (1<<j)
}
```

```
int setbit ( N, j)
{
    if ( checkbit(N,j) == true)
                return N
    else

                return N + (1<<j)
}
```

Q. Toggle i^{th} bit of N

0 ^ 1 → 1

1 ^ 1 → 0

return N ^ (1<<i)

N = 10
i = 3

```
  00001010
^ 00001000
------------
  00000010
```

Tower Research

Q. Given a +ve no N. Toggle all the bits starting from the right until rightmost set bit.

N = 20

```
  10100
  10011
```

N = 24

```
  11000
  10111
```

32 bits → int

```
for (i=0 ; i<32 ; i++)
{
    if ( checkbit (N, i) == false)
        setbit (N, i)
    else
    {
        toggle (N, i)
        Break
    }
}
```

1000
0999

1000
↓ -1
0111

N	(N-1)
3 : 011	010 = 2
15 : 1111	1110 = 14
6 : 110	101 = 5
8 : 1000	0111 = 7
20 : 10100	10011 = 19
24 : 11000	10111 = 23

```
int toggleuntilrightmost(N)
{
    return N-1
}
```

Google, Amazon

Q. Given +ve N. Count of no. of set bits of N

N = 10 → 1010 => 2

N = 8 → 1000 => 1

N = 15 → 1111 => 4

```
int countSetBits(int N)
{
    int cnt = 0
    for(int i=0; i<32; i++)
    {
        if(checkbit(N, i) == true)
            cnt++
    }
    return cnt
}
```

TC: O(1)

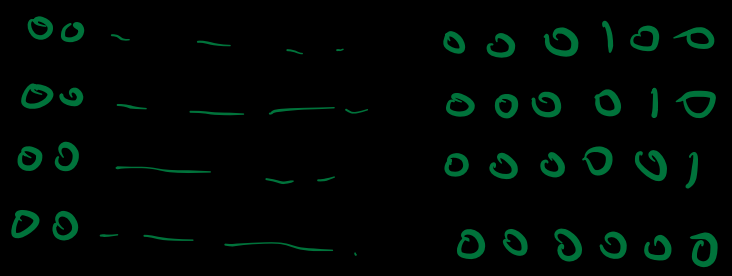
```
int countSetBits(int N)
{
    int cnt = 0
    while(N > 0)
    {
        if(N & 1 == 1)
            cnt++
        N = N >> 1
    }
    return cnt
}
```

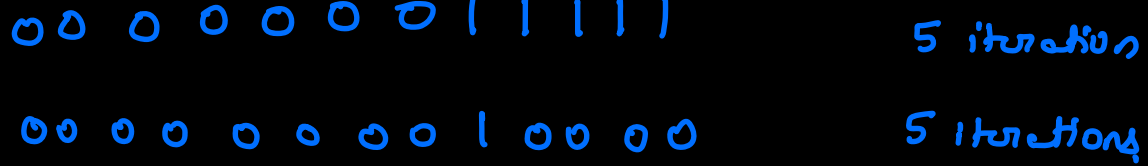
TC: $\log N \leq 32$

00000000000010000

00000000000001000

$N = 10^9$





N

3: 011

15: 1111

6: 110

8: 1000

20: 10100

24: 11000

(N-1)

010 = 2

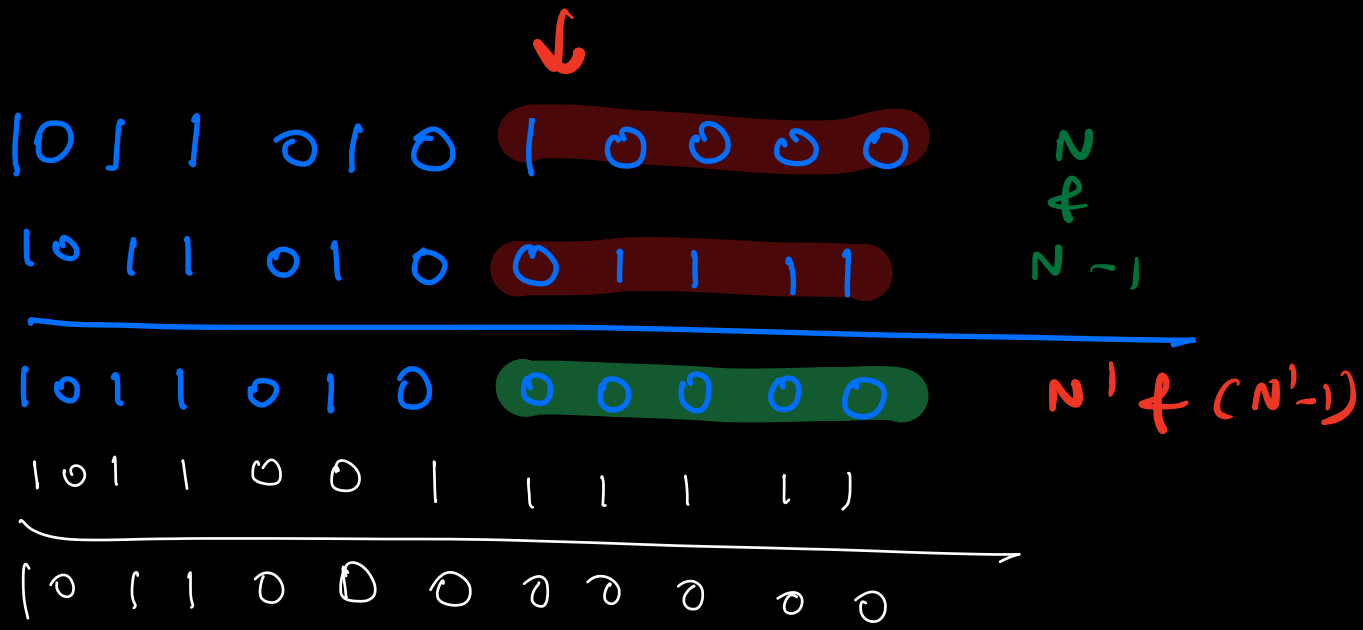
1110 = 14

101 = 5

0111 = 7

10011 = 19

10111 = 23



$N \;\&\; (N-1) \Rightarrow$ rightmost set bit will be unset

```
int countSetBits(int N)
                 16
{
    int cnt = 0
    while(N > 0)
    {
        N = N & (N-1)
        cnt++
    }
    return cnt
}
```

TL: $O(\log N)$

iteration

10000 → 1

1111 → 5

1111111111111111 → 32

$N \leq 10^9$

N = 16

No of bits $\log N$

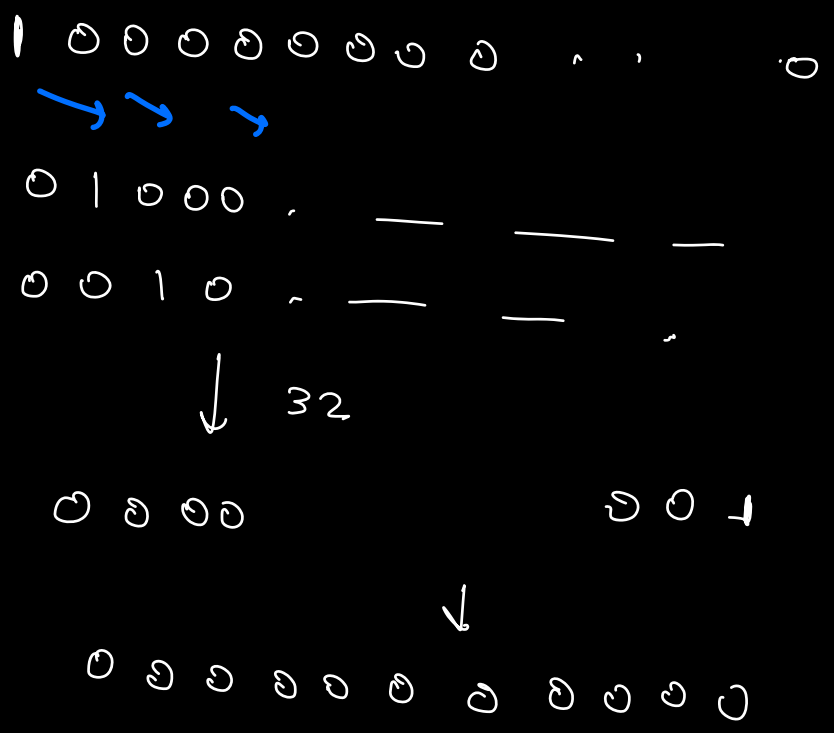
10000

N = 1024

10000000000 → 11

Doubts.

$N = 2^{32}$



$$\begin{array}{r} 1\ 1\ 0 \\ \wedge\ 1\ 1\ 0 \\ \hline 0\ 0\ 0 \end{array}$$

```
int countSetBits(int N)
{
    int cnt = 0
    while(N > 0)
    {
        N = N & (N-1)
        cnt++
    }
    return cnt
}
```

N = 21

```
10101
 ↓
N = 10100
 ↓
10000
 ↓
00000
```

cnt = 0 → 1 → 2 → 3