Revision

$(45)_{10} = (101101)_2$

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

Bitwise operators :   &, |, ^, ~, <<, >>

sopln (13 & 10) ; 8

```
  1 1 0 1      → 13
AND 1 0 1 0    → 10
  _____
  1 0 0 0      → 8
```

sopln (13 | 10) ; 15

```
  1 1 0 1      → 13
OR 1 0 1 0     → 10
  _____
  1 1 1 1      → 15
```

sopln (13 ^ 10); 7

$$1101 \rightarrow 13$$
$$\text{XOR} \quad 1010 \rightarrow 10$$
$$\overline{0111} \rightarrow 7$$

sopln (15 << 3); 120

$$a << N = 2^N * a$$

$$1 << N = 2^N$$

sopln (200 >> 4); 12

$$a >> N = a \mid 2^N$$

every bitwise operator in working in O(1)

Q-1 Given an integer n and index i, find if ith indexed bit in n is on/set or off?

on/set → 1

off → 0

n = 13, i = 2
```
  3 2 1 0
  1 1 0 1        true
```

n = 45, i = 3
```
  5 4 3 2 1 0
  1 0 1 1 0 1        true
```

n = 45, i = 4
```
  5 4 3 2 1 0
  1 0 1 1 0 1        false
```

idea 1: Convert n into binary form and the ith bit.
(todo)

idea 2: using AND operator

n = 45, i = 3

```
            5 4 3 2 1 0
45 :        1 0 1 1 0 1
1 << 3 :  8 0 0 1 0 0 0
          _____
            0 0 1 0 0 0
```

n = 45, i = 4

```
        5 4 3 2 1 0
45:     1 0 1 1 0 1
1<<4:   0 1 0 0 0 0
       ─────────────
        0 0 0 0 0 0
```

boolean checkbit (int n, int i) {

   // check whether ith bit in n is on or not

   if ((n & (1<<i)) == 0) {

      return false;

   }

   else {

      return true;

   }

}

TC: O(1)

1) n = 10 , i = 2

10 & (1<<2) ⇒ 0

```
        1 0 1 0
   &    0 1 0 0
       ─────────
        0 0 0 0
```

2) n = 13 , i = 3

13 & (1<<3) = 8

```
        1 1 0 1
   &    1 0 0 0
       ─────────
        1 0 0 0
```

Q-2 Given an integer number n, calculate total no. of set bits.

n	ans
45 (101101)	4
15 (001111)	4
12 (001100)	2
1 (000001)	1

idea1 : conversion from decimal to binary.

$n = 13$

```
int   solve(int n) {
    int count = 0;
    while (n > 0) {
        int rem = n %. 2;
        n = n / 2;
        if (rem == 1) {
            count++;
        }
    }
    return count;
}
```

n	rem	Count
13	1	1
6	0	1
3	1	2
1	1	3
0		

$13 \rightarrow 1101$

int: 4 bytes   (1 byte = 8bits)

int : 32 bits

code is working on bits of integer, that's why max

iterations: 32

itr: $\log_2 N$

TC: $O(\log_2 N)$

idea2 :   int → 32 bit   go on every bit   i → 0 to 31

and then checkbit (n,i).


```
int solve (int n) {
    int count = 0;
    for (int i = 0; i <= 31; i++) {
        if (checkbit (n,i) == true) {
            count ++;
        }
    }
    return count;
}
```

itr: 32 times

TC: O(1)

i) with OR

```
          5 4 3 2 1 0                          5 4 3 2 1 0
45        1 0 1 1 0 1          45             1 0 1 1 0 1

1<<3   OR 0 0 1 0 0 0          1<<4    OR 0 1 0 0 0 0
          _____                         _____
          1 0 1 1 0 1                         1 1 1 1 0 1
```

conclusion :   n | (1 < < i)   set the bit at ith idx.

```
                                        n = 10      i = 2

int  setBit (int n, int i) {            1 0 1 0

   n = n | (1<<i) ;                  OR  0 1 0 0
                                        _____
   return n;                            1 1 1 0

}
                                        n = 10      i = 1

                                        1 0 1 0

                                    OR  0 0 1 0
                                        _____
                                        1 0 1 0
```

i) with XOR

45   5 4 3 2 1 0
101101

$1 << 3$ XOR 001000
_____
100101

45   5 4 3 2 1 0
101101

$1 << 4$ XOR 010000
_____
111101

$$n \wedge (1 << i) \implies \text{flip } i^{th} \text{ bit}$$

1) $n \, \& \, (1 << i)$ →   check bit at i in n

2) $n \mid (1 << i)$ →   set bit at i in n

3) $n \wedge (1 << i)$ →   flip bit at i in n

Negative numbers

8 bit :

$$\boxed{-}\ \underset{7}{-}\ \underset{6}{-}\ \underset{5}{-}\ \underset{4}{-}\ \underset{3}{-}\ \underset{2}{-}\ \underset{1}{-}\ \underset{0}{-}$$

↳ signed bit

(for -ve no. sign bit = 1)

(for +ve no. sign bit = 0)

10 :   0 0 0 0 1 0 1 0

-10 : $\boxed{1}$ 0 0 0 1 0 1 0   ⟹ −128 + 8 + 2 = −118 ✗

⬇   ⬇   ⬇

−128 +   8 + 2

$$-a = \sim a + 1$$

-10

a   0 0 0 0 1 0 1 0
(10)

∼a   1 1 1 1 0 1 0 1

+   1
─────────────────────────

sign bit ← $\boxed{1}$ 1 1 1 0 1 1 0

↙ ⬇ ⬇ ⬇   ⬇ ⬇

−128 + 64 + 32 + 16 + 4 + 2 ⟹ −10

-45

$\downarrow$

$-a$

a   0 0 1 0 1 1 0 1

$\sim a$ 1 1 0 1 0 0 1 0

$+$ 1

sign $\leftarrow$ 1 1 0 1 0 0 1 1
bit

1 1 0 1 0 0 1 1

$-128 + 64 + 16 + 2 + 1 = -45$

Ranges

8 bit : $\boxed{\dfrac{0}{7}}$ $\dfrac{1}{6}$ $\dfrac{1}{5}$ $\dfrac{1}{4}$ $\dfrac{1}{3}$ $\dfrac{1}{2}$ $\dfrac{1}{1}$ $\dfrac{1}{0}$

$max = 2^0 + 2^1 + 2^2 + 2^3 + 2^4 + 2^5 + 2^6$

$Sn = \dfrac{a(r^t - 1)}{r - 1}$

$a = 1$
$r = 2$
$t = 7$

$= \dfrac{1(2^7 - 1)}{1} = 2^7 - 1$

$$\frac{1}{7} \quad \frac{0}{6} \quad \frac{0}{5} \quad \frac{0}{4} \quad \frac{0}{3} \quad \frac{0}{2} \quad \frac{0}{1} \quad \frac{0}{0}$$

$$\text{min} = -2^7$$

$$\text{max} = \boxed{2^7 - 1}$$

$$\text{min} = \boxed{-2^7}$$

32 bit :

$$\boxed{\frac{-}{31}} \quad \frac{-}{30} \quad \frac{-}{29} \quad \frac{-}{28} \quad \cdots \cdots \cdots \quad \frac{-}{5} \quad \frac{-}{4} \quad \frac{-}{3} \quad \frac{-}{2} \quad \frac{-}{1} \quad \frac{-}{0}$$

max

$$\boxed{\frac{0}{31}} \quad \frac{1}{30} \quad \frac{1}{29} \quad \frac{1}{28} \quad \cdots \cdots \cdots \quad \frac{1}{5} \quad \frac{1}{4} \quad \frac{1}{3} \quad \frac{1}{2} \quad \frac{1}{1} \quad \frac{1}{0}$$

$$\text{max} = 2^0 + 2^1 + 2^2 + \cdots \cdots + 2^{30}$$

$$\text{max} = \frac{1(2^{31} - 1)}{\underline{1}} = 2^{31} - 1$$

$$a = 1$$
$$r = 2$$
$$t = 31$$

min

$$\boxed{\underset{31}{1}} \quad \underset{30}{0} \quad \underset{29}{0} \quad \underset{28}{0} \ldots \ldots \ldots \underset{5}{0} \quad \underset{4}{0} \quad \underset{3}{0} \quad \underset{2}{0} \quad \underset{1}{0} \quad \underset{0}{0}$$

$$\text{min} = -2^{31}$$

$$\begin{array}{|c|}
\hline
\text{max} = 2^{31} - 1 \\
\\
\text{min} = -2^{31} \\
\hline
\end{array} \qquad \Rightarrow \quad \text{int (32 bits)}$$

$$A = \begin{bmatrix} 5 & 2 \\ 9 & 1 \\ 4 & 3 \end{bmatrix} \qquad B = 2$$

$$ans = \begin{bmatrix} 10 & 4 \\ 18 & 2 \\ 8 & 6 \end{bmatrix}$$