

Introduction to Bit Manipulation

Binary Number Convention (1's and 2's complement)

$$\text{Ex:- } (7)_{10} \rightarrow (111)_2$$

Binary representation of 7

2	7
2	3
2	1

Binary representation of 7

$$(13)_{10} \rightarrow (1101)_2$$

Binary representation of 13

2	13	0	1
2	6	0	0
2	3	1	0
2	1	0	0

$$(111)_2 \rightarrow ()_{10}$$

Binary to Decimal conversion

$$1 \times 2^0 + 1 \times 2^1 + 1 \times 2^2 = 1 + 2 + 4 = 7$$

$$(1101)_2 \rightarrow ()_{10}$$

Binary to Decimal conversion

$$1 \times 2^0 + 0 \times 2^1 + 1 \times 2^2 + 1 \times 2^3 = 1 + 0 + 4 + 8 = 13$$

decimal to binary Code:

```

String count2Binary(int n)
{
    string res = "";
    while (n != 0) {
        if (n % 2 == 1) res += '1';
        else res += '0';
        n /= 2;
    }
    res += '\0';
    reverse(res);
    return res;
}
    
```

Binary to Decimal code:

```

int count2Binary(string x)
{
    int len = x.length();
    int p2 = 1;
    int num = 0;
    for (int i = len - 1; i >= 0; i--) {
        if (x[i] == '1') {
            num = num + p2;
        }
        p2 = p2 * 2;
    }
    return num;
}
    
```

int can store 32 bit

long long " " 64 bit

int, & (2^13 - 2 from 0) minimum value 30000

$\begin{array}{r} 0 \\ \boxed{1\ 1\ 1\ 1\ 1\ 1\ 1\ 1} \\ \text{28bit} \end{array}$ 4bit

1's complement

$$(13) \rightarrow (1\ 1\ 0\ 1)_2$$

(13) \rightarrow flip 0

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

flip + 0FF

81

2's complement

1. 1's complement

(13)₁₀ \rightarrow 1's complement

$$(13)_{10} \rightarrow (0\ 0\ 1\ 0)_2$$

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

Operations \rightarrow AND, OR, XOR, SHIFT, NOT

(united) group of binary bits

AND operation :-

1 & 1 = 1

all true \rightarrow true

else 1 false \rightarrow false

(111 & 101) = 101

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

$10 = 1000_2$
 $01 = 0010_2$
 $01 = 0001_2$

OR operation :-

1 true \rightarrow true

all false \rightarrow false

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

$10 = 1000_2$
 $01 = 0010_2$
 $11 = 1011_2$

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

$10 = 1000_2$
 $01 = 0010_2$
 $11 = 1001_2$

XOR operation :-

no of 1's \rightarrow odd \rightarrow 1
 \rightarrow even \rightarrow 0

Right Shift :-

$$13 \gg 1 \left(\frac{13}{2^1} \right) = 6$$

$$13 \gg 2 \quad \left(\frac{13}{4}\right) = 3$$

$$\begin{array}{r} \text{---} \\ 1 \quad 1 \quad 0 \quad x \quad 1 \quad x \\ \text{---} \\ 0 \quad 0 \quad - \end{array} \quad \begin{array}{r} \text{---} \\ 13 \\ \text{---} \\ 3 \end{array}$$

$$\left(\begin{array}{cc} 13 & 8 \\ 0 & 1 \end{array} \right) \xrightarrow{\text{R}_1 - 13\text{R}_2} \left(\begin{array}{cc} 0 & 8 \\ 0 & 1 \end{array} \right) \xrightarrow{\text{R}_1 \leftrightarrow \text{R}_2} \left(\begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array} \right)$$

$$m \gg k = \frac{m}{2k}$$

$$13 \rightarrow (1101)_2 \rightarrow 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

$$13 \gg 1 \rightarrow (0110)_2 \rightarrow 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 \\ = 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 \times \underline{\underline{0}}$$

NOTE OF OPERATOR'S

$$n = 13$$

$$m = -13$$

Sign
"Positive")

1 negative 100% 1 mm

final answer is $\boxed{-13}$

longest Int:-

0 1 1 1 1 ... 1

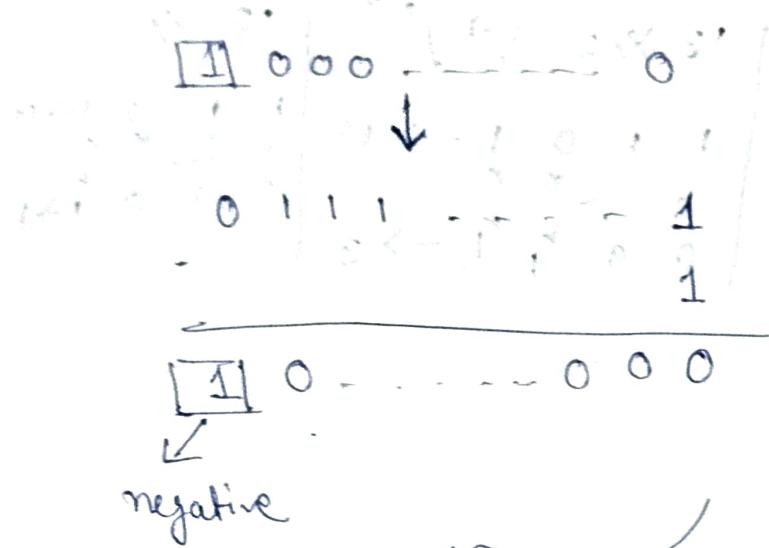
sign

$$= 2^{30} + 2^{29} + 2^{28} + \dots + 2^0$$

$$= (2^{31} - 1) \text{ INT_MAX}$$

Smallest :-

-2^{31}



$$1000\ldots000 = -2^{31} \text{ (INT: MIN)}$$

left shift :-

$$13 \ll 1$$

$$\begin{array}{r} 4\ 3\ 2\ 1\ 0 \\ 0\ 1\ 1\ 0\ 1 \end{array}$$

$$\rightarrow 13 : 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

$$\begin{array}{r} 1\ 110\ 100 \\ \hline 1\ 110\ 100 \end{array}$$

$$\text{num} \ll K \rightarrow \text{num} \times 2^K$$

$(2^{31}-1) \ll 1$ means overflow

NOT OPERATOR (\sim) :

5)

0 \sim (010) 011001 \rightarrow 5

11. [11(010)] + 1001 0d

-ve
overflows to 0
 $[11(010)] + 1001 \rightarrow$ 2's comp

1 \sim (010) 001000 0

0 - - - 001000 $\xrightarrow{\text{flip}}$
+1

10 - - - 001000 $\xrightarrow{\text{add}}$ 06

sign

(-6)

Num = -6

$x = \sim(-6)$

0.00 - - - 0110 (+6) 0 1 00

1 1 1 - - - 1 001 0 0 1 0

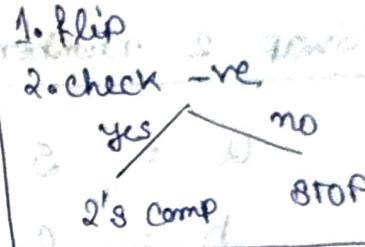
+ 1

$\xrightarrow{\text{add}}$ 1 1 1 - - - 1 010 $\rightarrow (-6) \gg 1$



Pos

00 - - - 0101 \rightarrow 5



Swap 2 numbers :-

$$\begin{array}{l} a = 5 \\ b = 6 \end{array}$$

$$\left| \begin{array}{l} a = a \uparrow b (5^6) \\ b = a \uparrow b [(5^6)^5] \\ a = a \uparrow b [(5^6)^5] \\ \quad \quad \quad = 6 \end{array} \right.$$

Swap 1 successfully

check if the i th bit set or not?

Ex:- $N = 13 \quad i = 2$

$$\Rightarrow (1 \ll 2) = 100$$

$$\begin{array}{r} 1 \ 1 \ 0 \ 1 \\ 80 \ 1 \ 0 \ 0 \\ \hline 0 \ 1 \ 0 \ 0 \end{array} \rightarrow ! = 0 \text{ means set}$$

if ($N \& (1 \ll i)$) $! = 0$; means set
else
not set;

Ex:- $N = 13 \quad i = 1$

$$\begin{array}{r} 3 \ 2 \ 1 \ 0 \\ 1 \ 1 \ 0 \ 1 \\ \hline 80 \ 0 \ 1 \ 0 \\ \hline 0 \ 0 \ 0 \ 0 \end{array} \Rightarrow (1 \ll 1) = 10$$

= 0 not set

Other method

$$N = 13 \quad i = 2$$

$$\begin{array}{r}
 1 \ 1 \ 0 \ 1 \\
 0 \ 0 \ 1 \ 1 \\
 \hline
 0 \ 0 \ 0 \ 1
 \end{array}
 \begin{array}{l}
 \text{right shift} \\
 (13 \gg 2) \\
 (N \gg 1)
 \end{array}$$

~~AND 1~~

$$\begin{array}{r}
 0 \ 1 \ 0 \ 0 \\
 \hline
 0 \ 1 \ 0 \ 0
 \end{array}
 \begin{array}{l}
 \text{!=0} \\
 \text{(set)}
 \end{array}$$

$$N = 13 \quad i = 1 \quad (i >> 1) \rightarrow 8, 1 \quad 8, 1 \neq 0 \rightarrow \text{set}$$

$$\begin{array}{r}
 1 \ 1 \ 0 \ 1 \\
 0 \ 1 \ 1 \ 0 \\
 \hline
 0 \ 0 \ 0 \ 1
 \end{array}
 \begin{array}{l}
 \text{right shift} \\
 (0) \\
 \text{AND 1}
 \end{array}$$

$$\begin{array}{r}
 0 \ 0 \ 0 \ 0 \\
 \hline
 0 \ 0 \ 0 \ 0
 \end{array}
 \begin{array}{l}
 \rightarrow = 0 \quad (\text{not set})
 \end{array}$$

if $[(N \gg i) \& 1 \neq 0]$ means set

else

not set

Set ith bit

$$N = 9 \quad i = 2$$

$$(1 \ll 2) = 100$$

$$\begin{array}{r}
 1 \ 0 \ 0 \ 1 \\
 1 \ 0 \ 1 \ 0 \ 0 \\
 \hline
 1 \ 1 \ 0 \ 1
 \end{array}
 \begin{array}{l}
 \text{OR} \\
 \text{set 2bit}
 \end{array}$$

Condition:-
 $N \& (1 \ll i) \neq 0$

$$N = 13 \quad i = 2 \quad (1 \ll 2) = 100$$

$$\begin{array}{r}
 1 \ 1 \ 0 \ 1 \\
 1 \ 0 \ 1 \ 0 \ 0 \\
 \hline
 1 \ 1 \ 0 \ 1 \rightarrow 13
 \end{array}
 \boxed{N \& (1 \ll i)}$$

Clear ith bit

$$N = 13 \quad i = 2 \quad \sim(1 \ll 2) = \sim(0100)$$

$$\therefore \sim(0100) = 1011$$

$$\begin{array}{r}
 1 \ 1 \ 0 \ 1 \\
 1 \ 0 \ 1 \ 1 \\
 \hline
 1 \ 0 \ 0 \quad \text{(padding) } 0 \ 0 \ 0 \ 0 \ 0 \ 0
 \end{array}$$

→ clear

Condition :-

$$\boxed{N \& (\sim(1 \ll i))}$$

Toggle ith bit :-

$$\text{Ex: } N = 13 \quad i = 2 \quad (1 \ll 2) = 100$$

$$\begin{array}{r}
 1 \ 1 \ 0 \ 1 \\
 1 \ 0 \ 1 \ 0 \ 0 \\
 \hline
 1 \ 0 \ 0 \quad \text{1} \quad \text{padding}
 \end{array}$$

→ toggle

$$(101101)$$

Ex:- $i = 1$ for $N = 13$

$(1 \ll i) = 10$

$\begin{array}{r} 1101 \\ \times 0010 \\ \hline 110100 \end{array}$

\rightarrow Toggle

Condition :- $N \wedge (1 \ll i) = 0$ (i.e. 0 is result)

\leftarrow To remove last

Remove the last set bit (Rightmost) :

$N = 16$ (if $N = 13$ then $N = 40$ then \leftarrow)

$\begin{array}{r} 10000 \\ \times 111 \\ \hline 00000 \end{array}$

\leftarrow 15 \leftarrow 12 \leftarrow 39

\leftarrow 00000 \leftarrow 00000 \leftarrow 00000

Condition :- $[N \wedge (N-1)] = 0$ (i.e. 0 is result)

\leftarrow 1000000 (i.e. 0 is result)

\leftarrow 1000000 (i.e. 0 is result)

$N > R = 10$

If num is a power of 2 or not :-

$$N = 16$$

$$\begin{array}{r} 10000 \\ 801111 \\ \hline 00000 \end{array}$$

$$N = 13$$

$$\begin{array}{r} 01101 \\ 801100 \\ \hline 01100 \end{array}$$

$$N = 32$$

$$\begin{array}{r} 100000 \\ 801111 \\ \hline 000000 \end{array}$$

Condition $[N \& (N-1) = 0] \rightarrow$ power of 2

else

not power of 2

Count the number of set bit

int countsetbit(int n)

{ int cnt = 0;

while ($n > 1$)

{ if ($(n \% 2 == 1)$) cnt += 1;

or cnt += n % 1

$n = n / 2$; $\rightarrow n = n \gg 1$

}

if ($n == 1$) cnt += 1;

return cnt;

another approach

$$2^n = 1 \ll n$$

last bit of odd numbers always 1
last bit of even numbers always 0

Ex:- $13 \rightarrow 1101 \} \text{odd}$
 $11 \rightarrow 1011 \} \text{odd}$
 $12 \rightarrow 1100 \} \text{even}$
 $8 \rightarrow 1000 \} \text{even}$

check even or odd is

$$13 \rightarrow \begin{array}{r} 1101 \\ -8000 \\ \hline 0001 \end{array} \quad (N \& 1) = 1 \rightarrow \text{odd}$$

$\downarrow 0 \rightarrow \text{Ans 1}$

$$11 \rightarrow \begin{array}{r} 1011 \\ -8000 \\ \hline 0011 \end{array} \quad ((N \& 1) = 1 \rightarrow \text{odd})$$

$$12 \rightarrow \begin{array}{r} 1100 \\ -8000 \\ \hline 0000 \end{array} \quad (N \& 1) = 0 \rightarrow \text{even}$$

$$8 \rightarrow \begin{array}{r} 1000 \\ -8000 \\ \hline 0000 \end{array} = 0$$

Divided by 2: $\frac{1}{2} \times 100 = 50$

$N = 13$

12

$$Ans = 6(n/i)$$

$$N = 12$$

$$i = 2$$

$$Ans = 6 \left(\frac{m}{i} \right) + 1 + 3 + \dots + p$$

$$\begin{array}{r} 13 \uparrow 1 \quad 1 \quad 0 \quad 1 \quad 13 \uparrow 1 \quad 1 \quad 0 \quad 0 \\ \textcircled{6} \uparrow 0 \quad 1 \quad 1 \quad 0 \quad \textcircled{6} \uparrow 0 \quad 1 \quad 1 \quad 0 \end{array}$$

Condition :- $N = (N \gg 1)$

Count number of set bit

```
int cnt = 0;
```

While ($N \neq 0$)

$$\{ N = (N \& (N - 1)), \\ \text{cont}++;$$

en ();

newer cut

$$N = 180 \quad 0 \quad 0$$

18A 1 A 1

1

$$\begin{array}{r} 8 \quad 1 \quad 1 \quad 0 \quad 0 \\ \hline & & & & \rightarrow 12 \end{array}$$

1 1 0 0

ent =

— 10 —

$$\frac{8}{1} \quad 0 \quad 1 \quad 1 \rightarrow 11$$

1 0 0 0

cnt = 2

— 1 —

8 7 7

0000

1000 int 23

Minimum bitflips to convert a number

start = 10

1010

goal = 7

0111

1) first \oplus XOR start and goal.

$$\begin{array}{r} 1010 \\ \oplus 0111 \\ \hline 1101 \end{array}$$

$$\begin{array}{ccccccc} & 1 & 0 & 1 & 0 & 0 & 0 \\ \oplus & 0 & 1 & 1 & 1 & 0 & 0 \\ \hline & 1 & 1 & 0 & 1 & 0 & 0 \end{array}$$

2) 11010 \rightarrow check how many bit set this is the min numbers flips to convert a number

$$[s_0, s_1] \leftarrow 0, 1 \text{ and}$$

$$s_0 \leftarrow [s_0, s_1] \leftarrow 1, 1 \text{ and } 1.$$

Code :-

```
int minbitflip(int start, int goal)
{
    int num = start ^ goal; // O(1). Because num = 01111111
    int count = 0; // O(1) num = 01111111 and
    while (num != 0) { // O(1)
        if (num > 0) { // O(1)
            num = num & (num - 1); // O(1)
            count++; // O(1)
        }
    }
    return count; // O(1)
}
```

TC $\Rightarrow \log_2(\text{start} \wedge \text{goal})$

SC $\Rightarrow O(1)$

Power Set (Prints all subsets)

nums = [1 2 3] n=3

	2	1	0	1 1 1 0	0 1 3 1
0 →	0	0	0	→ []	1 1 1 0
1 →	0	0	1	→ [1]	1 1 1 0
2 →	0	1	0	→ [2]	1 1 1 0
3 →	0	1	1	→ [1, 2]	1 1 1 0
4 →	1	0	0	→ [3]	0 0 1 1
5 →	1	0	1	→ [1, 3]	0 0 1 1
6 →	1	1	0	→ [2, 3]	0 0 1 1
7 →	1	1	1	→ [1, 2, 3]	0 0 1 1

Subset

Code :-

```

subset = (2^n) = 1 << n
arr = [ ]
for (num = 0 → subset-1)
    {
        list = []
        for (i=0 → n-1)
            {
                if (num & (1 << i))
                    list.add(arr[i])
            }
        arr.add(list)
    }
    reharr arr;

```

$2^0 = 1$
 $2^1 = 2$
 $2^2 = 4$

~~Gx~~ :-

<u>subset = 0</u>	0 0 0	0 0 0	0 0 0
	8 0	8 0	8 0
	0 0 0	0 0 0	0 0 0
	0 0 0	0 0 0	0 0 0
	0 0 0	0 0 0	0 0 0
<u>subset = 1</u>	0 0 1	0 0 1	0 0 1
	8 0	8 0	8 0
	0 0 0	0 0 0	0 0 0
	0 0 0	0 0 0	0 0 0
	0 0 0	0 0 0	0 0 0
<u>subset = 2</u>	0 0 0	0 0 1	0 0 1
	8 0	8 0	8 0
	0 0 0	0 0 0	0 0 0
	0 0 0	0 0 0	0 0 0
	0 0 0	0 0 0	0 0 0
<u>subset = 3</u>	0 1 1	0 1 1	0 1 1
	8 0	8 0	8 0
	0 0 0	0 0 0	0 0 0
	0 0 0	0 0 0	0 0 0
	0 0 0	0 0 0	0 0 0
<u>subset = 4</u>	1 0 0	1 0 0	1 0 0
	8 0	8 0	8 0
	0 0 0	0 0 0	0 0 0
	0 0 0	0 0 0	0 0 0
	0 0 0	0 0 0	0 0 0

Example :- $\mu_{\text{num}} = 6$ (0.110)

$$\begin{array}{l} \text{2 iteration: } 1(0001) \\ (1\ll1) = 0010 \end{array} \quad \left| \begin{array}{l} \text{num 8 } (1\ll1) \\ = 0110 \& 0010 \\ 0010 \rightarrow \text{1 bit element [2]} \end{array} \right.$$

$$\begin{array}{l} \text{Iteration 2 (0010)} \\ (1 \ll 2) = 0100 \end{array} \left| \begin{array}{l} \text{num } 8(1 \ll 2) \\ = 0110 \& 0100 \\ = 0100 \longrightarrow \text{2 bit element [3]} \end{array} \right.$$

Single number - II

Input = nums = [5, 5, 5, 2, 4, 4, 1] 0 1 2 3 4
 Output : 2 0 1 2 3 4 5 6 7 8 9 10 11

all numbers appears twice except for one element.

Approach :-

1) Sort all elements Ex: [2, 4, 4, 5, 5, 5]

2) $\boxed{2 \ 4 \ 4} \ 4 \ 5 \ 5 \ 5 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0$

i

Check if diff element

return $[i-1]$

Ex:-

arr = [1, 1, 1, 2, 2, 2, 3, 4, 4, 4]

$\boxed{1 \ 1 \ 1}$

i

2 2 2 3 4 4 4 0 0 0 0 0 0

i

check

i

check

diff element

return $[i-1]$

Time complexity O(n^2) as we have to check every element with every other element (n > 1).
 Space complexity O(1) as we are traversing the array only.

(n^2) * n = n^3 / (n^2) * n = n^3

∴ Time complexity = O(n^3)

P.T.O

(n^2) * n = n^3 / (n^2) * n = n^3

∴ Space complexity = O(1)

∴ Space complexity = O(1)

Code:

```
int findSingleNumber(vector<int>& nums) {
    sort(nums.begin(), nums.end());
    for(int i=1; i<nums.size(); i+=3) {
        if(nums[i] != nums[i-1]) {
            return nums[i-1];
        }
    }
    return nums[nums.size()-1];
```

(i-1) \rightarrow 0, 3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48, 51, 54, 57, 60, 63, 66, 69, 72, 75, 78, 81, 84, 87, 90, 93, 96, 99, 102, 105, 108, 111, 114, 117, 120, 123, 126, 129, 132, 135, 138, 141, 144, 147, 150, 153, 156, 159, 162, 165, 168, 171, 174, 177, 180, 183, 186, 189, 192, 195, 198, 201, 204, 207, 210, 213, 216, 219, 222, 225, 228, 231, 234, 237, 240, 243, 246, 249, 252, 255, 258, 261, 264, 267, 270, 273, 276, 279, 282, 285, 288, 291, 294, 297, 299, 302, 305, 308, 311, 314, 317, 320, 323, 326, 329, 332, 335, 338, 341, 344, 347, 350, 353, 356, 359, 362, 365, 368, 371, 374, 377, 380, 383, 386, 389, 392, 395, 398, 401, 404, 407, 410, 413, 416, 419, 422, 425, 428, 431, 434, 437, 440, 443, 446, 449, 452, 455, 458, 461, 464, 467, 470, 473, 476, 479, 482, 485, 488, 491, 494, 497, 499, 502, 505, 508, 511, 514, 517, 520, 523, 526, 529, 532, 535, 538, 541, 544, 547, 550, 553, 556, 559, 562, 565, 568, 571, 574, 577, 580, 583, 586, 589, 592, 595, 598, 601, 604, 607, 610, 613, 616, 619, 622, 625, 628, 631, 634, 637, 640, 643, 646, 649, 652, 655, 658, 661, 664, 667, 670, 673, 676, 679, 682, 685, 688, 691, 694, 697, 699, 702, 705, 708, 711, 714, 717, 720, 723, 726, 729, 732, 735, 738, 741, 744, 747, 750, 753, 756, 759, 762, 765, 768, 771, 774, 777, 780, 783, 786, 789, 792, 795, 798, 801, 804, 807, 810, 813, 816, 819, 822, 825, 828, 831, 834, 837, 840, 843, 846, 849, 852, 855, 858, 861, 864, 867, 870, 873, 876, 879, 882, 885, 888, 891, 894, 897, 899, 902, 905, 908, 911, 914, 917, 920, 923, 926, 929, 932, 935, 938, 941, 944, 947, 950, 953, 956, 959, 962, 965, 968, 971, 974, 977, 980, 983, 986, 989, 992, 995, 998, 1001, 1004, 1007, 1010, 1013, 1016, 1019, 1022, 1025, 1028, 1031, 1034, 1037, 1040, 1043, 1046, 1049, 1052, 1055, 1058, 1061, 1064, 1067, 1070, 1073, 1076, 1079, 1082, 1085, 1088, 1091, 1094, 1097, 1100, 1103, 1106, 1109, 1112, 1115, 1118, 1121, 1124, 1127, 1130, 1133, 1136, 1139, 1142, 1145, 1148, 1151, 1154, 1157, 1160, 1163, 1166, 1169, 1172, 1175, 1178, 1181, 1184, 1187, 1190, 1193, 1196, 1199, 1202, 1205, 1208, 1211, 1214, 1217, 1220, 1223, 1226, 1229, 1232, 1235, 1238, 1241, 1244, 1247, 1250, 1253, 1256, 1259, 1262, 1265, 1268, 1271, 1274, 1277, 1280, 1283, 1286, 1289, 1292, 1295, 1298, 1301, 1304, 1307, 1310, 1313, 1316, 1319, 1322, 1325, 1328, 1331, 1334, 1337, 1340, 1343, 1346, 1349, 1352, 1355, 1358, 1361, 1364, 1367, 1370, 1373, 1376, 1379, 1382, 1385, 1388, 1391, 1394, 1397, 1399, 1402, 1405, 1408, 1411, 1414, 1417, 1420, 1423, 1426, 1429, 1432, 1435, 1438, 1441, 1444, 1447, 1450, 1453, 1456, 1459, 1462, 1465, 1468, 1471, 1474, 1477, 1480, 1483, 1486, 1489, 1492, 1495, 1498, 1501, 1504, 1507, 1510, 1513, 1516, 1519, 1522, 1525, 1528, 1531, 1534, 1537, 1540, 1543, 1546, 1549, 1552, 1555, 1558, 1561, 1564, 1567, 1570, 1573, 1576, 1579, 1582, 1585, 1588, 1591, 1594, 1597, 1599, 1602, 1605, 1608, 1611, 1614, 1617, 1620, 1623, 1626, 1629, 1632, 1635, 1638, 1641, 1644, 1647, 1650, 1653, 1656, 1659, 1662, 1665, 1668, 1671, 1674, 1677, 1680, 1683, 1686, 1689, 1692, 1695, 1698, 1701, 1704, 1707, 1710, 1713, 1716, 1719, 1722, 1725, 1728, 1731, 1734, 1737, 1740, 1743, 1746, 1749, 1752, 1755, 1758, 1761, 1764, 1767, 1770, 1773, 1776, 1779, 1782, 1785, 1788, 1791, 1794, 1797, 1799, 1802, 1805, 1808, 1811, 1814, 1817, 1820, 1823, 1826, 1829, 1832, 1835, 1838, 1841, 1844, 1847, 1850, 1853, 1856, 1859, 1862, 1865, 1868, 1871, 1874, 1877, 1880, 1883, 1886, 1889, 1892, 1895, 1898, 1901, 1904, 1907, 1910, 1913, 1916, 1919, 1922, 1925, 1928, 1931, 1934, 1937, 1940, 1943, 1946, 1949, 1952, 1955, 1958, 1961, 1964, 1967, 1970, 1973, 1976, 1979, 1982, 1985, 1988, 1991, 1994, 1997, 1999, 2002, 2005, 2008, 2011, 2014, 2017, 2020, 2023, 2026, 2029, 2032, 2035, 2038, 2041, 2044, 2047, 2050, 2053, 2056, 2059, 2062, 2065, 2068, 2071, 2074, 2077, 2080, 2083, 2086, 2089, 2092, 2095, 2098, 2101, 2104, 2107, 2110, 2113, 2116, 2119, 2122, 2125, 2128, 2131, 2134, 2137, 2140, 2143, 2146, 2149, 2152, 2155, 2158, 2161, 2164, 2167, 2170, 2173, 2176, 2179, 2182, 2185, 2188, 2191, 2194, 2197, 2199, 2202, 2205, 2208, 2211, 2214, 2217, 2220, 2223, 2226, 2229, 2232, 2235, 2238, 2241, 2244, 2247, 2250, 2253, 2256, 2259, 2262, 2265, 2268, 2271, 2274, 2277, 2280, 2283, 2286, 2289, 2292, 2295, 2298, 2301, 2304, 2307, 2310, 2313, 2316, 2319, 2322, 2325, 2328, 2331, 2334, 2337, 2340, 2343, 2346, 2349, 2352, 2355, 2358, 2361, 2364, 2367, 2370, 2373, 2376, 2379, 2382, 2385, 2388, 2391, 2394, 2397, 2399, 2402, 2405, 2408, 2411, 2414, 2417, 2420, 2423, 2426, 2429, 2432, 2435, 2438, 2441, 2444, 2447, 2450, 2453, 2456, 2459, 2462, 2465, 2468, 2471, 2474, 2477, 2480, 2483, 2486, 2489, 2492, 2495, 2498, 2501, 2504, 2507, 2510, 2513, 2516, 2519, 2522, 2525, 2528, 2531, 2534, 2537, 2540, 2543, 2546, 2549, 2552, 2555, 2558, 2561, 2564, 2567, 2570, 2573, 2576, 2579, 2582, 2585, 2588, 2591, 2594, 2597, 2599, 2602, 2605, 2608, 2611, 2614, 2617, 2620, 2623, 2626, 2629, 2632, 2635, 2638, 2641, 2644, 2647, 2650, 2653, 2656, 2659, 2662, 2665, 2668, 2671, 2674, 2677, 2680, 2683, 2686, 2689, 2692, 2695, 2698, 2701, 2704, 2707, 2710, 2713, 2716, 2719, 2722, 2725, 2728, 2731, 2734, 2737, 2740, 2743, 2746, 2749, 2752, 2755, 2758, 2761, 2764, 2767, 2770, 2773, 2776, 2779, 2782, 2785, 2788, 2791, 2794, 2797, 2799, 2802, 2805, 2808, 2811, 2814, 2817, 2820, 2823, 2826, 2829, 2832, 2835, 2838, 2841, 2844, 2847, 2850, 2853, 2856, 2859, 2862, 2865, 2868, 2871, 2874, 2877, 2880, 2883, 2886, 2889, 2892, 2895, 2898, 2901, 2904, 2907, 2910, 2913, 2916, 2919, 2922, 2925, 2928, 2931, 2934, 2937, 2940, 2943, 2946, 2949, 2952, 2955, 2958, 2961, 2964, 2967, 2970, 2973, 2976, 2979, 2982, 2985, 2988, 2991, 2994, 2997, 2999, 3002, 3005, 3008, 3011, 3014, 3017, 3020, 3023, 3026, 3029, 3032, 3035, 3038, 3041, 3044, 3047, 3050, 3053, 3056, 3059, 3062, 3065, 3068, 3071, 3074, 3077, 3080, 3083, 3086, 3089, 3092, 3095, 3098, 3101, 3104, 3107, 3110, 3113, 3116, 3119, 3122, 3125, 3128, 3131, 3134, 3137, 3140, 3143, 3146, 3149, 3152, 3155, 3158, 3161, 3164, 3167, 3170, 3173, 3176, 3179, 3182, 3185, 3188, 3191, 3194, 3197, 3199, 3202, 3205, 3208, 3211, 3214, 3217, 3220, 3223, 3226, 3229, 3232, 3235, 3238, 3241, 3244, 3247, 3250, 3253, 3256, 3259, 3262, 3265, 3268, 3271, 3274, 3277, 3280, 3283, 3286, 3289, 3292, 3295, 3298, 3301, 3304, 3307, 3310, 3313, 3316, 3319, 3322, 3325, 3328, 3331, 3334, 3337, 3340, 3343, 3346, 3349, 3352, 3355, 3358, 3361, 3364, 3367, 3370, 3373, 3376, 3379, 3382, 3385, 3388, 3391, 3394, 3397, 3399, 3402, 3405, 3408, 3411, 3414, 3417, 3420, 3423, 3426, 3429, 3432, 3435, 3438, 3441, 3444, 3447, 3450, 3453, 3456, 3459, 3462, 3465, 3468, 3471, 3474, 3477, 3480, 3483, 3486, 3489, 3492, 3495, 3498, 3501, 3504, 3507, 3510, 3513, 3516, 3519, 3522, 3525, 3528, 3531, 3534, 3537, 3540, 3543, 3546, 3549, 3552, 3555, 3558, 3561, 3564, 3567, 3570, 3573, 3576, 3579, 3582, 3585, 3588, 3591, 3594, 3597, 3599, 3602, 3605, 3608, 3611, 3614, 3617, 3620, 3623, 3626, 3629, 3632, 3635, 3638, 3641, 3644, 3647, 3650, 3653, 3656, 3659, 3662, 3665, 3668, 3671, 3674, 3677, 3680, 3683, 3686, 3689, 3692, 3695, 3698, 3701, 3704, 3707, 3710, 3713, 3716, 3719, 3722, 3725, 3728, 3731, 3734, 3737, 3740, 3743, 3746, 3749, 3752, 3755, 3758, 3761, 3764, 3767, 3770, 3773, 3776, 3779, 3782, 3785, 3788, 3791, 3794, 3797, 3799, 3802, 3805, 3808, 3811, 3814, 3817, 3820, 3823, 3826, 3829, 3832, 3835, 3838, 3841, 3844, 3847, 3850, 3853, 3856, 3859, 3862, 3865, 3868, 3871, 3874, 3877, 3880, 3883, 3886, 3889, 3892, 3895, 3898, 3901, 3904, 3907, 3910, 3913, 3916, 3919, 3922, 3925, 3928, 3931, 3934, 3937, 3940, 3943, 3946, 3949, 3952, 3955, 3958, 3961, 3964, 3967, 3970, 3973, 3976, 3979, 3982, 3985, 3988, 3991, 3994, 3997, 3999, 4002, 4005, 4008, 4011, 4014, 4017, 4020, 4023, 4026, 4029, 4032, 4035, 4038, 4041, 4044, 4047, 4050, 4053, 4056, 4059, 4062, 4065, 4068, 4071, 4074, 4077, 4080, 4083, 4086, 4089, 4092, 4095, 4098, 4101, 4104, 4107, 4110, 4113, 4116, 4119, 4122, 4125, 4128, 4131, 4134, 4137, 4140, 4143, 4146, 4149, 4152, 4155, 4158, 4161, 4164, 4167, 4170, 4173, 4176, 4179, 4182, 4185, 4188, 4191, 4194, 4197, 4199, 4202, 4205, 4208, 4211, 4214, 4217, 4220, 4223, 4226, 4229, 4232, 4235, 4238, 4241, 4244, 4247, 4250, 4253, 4256, 4259, 4262, 4265, 4268, 4271, 4274, 4277, 4280, 4283, 4286, 4289, 4292, 4295, 4298, 4301, 4304, 4307, 4310, 4313, 4316, 4319, 4322, 4325, 4328, 4331, 4334, 4337, 4340, 4343, 4346, 4349, 4352, 4355, 4358, 4361, 4364, 4367, 4370, 4373, 4376, 4379, 4382, 4385, 4388, 4391, 4394, 4397, 4399, 4402, 4405, 4408, 4411, 4414, 4417, 4420, 4423, 4426, 4429, 4432, 4435, 4438, 4441, 4444, 4447, 4450, 4453, 4456, 4459, 4462, 4465, 4468, 4471, 4474, 4477, 4480, 4483, 4486, 4489, 4492, 4495, 4498, 4501, 4504, 4507, 4510, 4513, 4516, 4519, 4522, 4525, 4528, 4531, 4534, 4537, 4540, 4543, 4546, 4549, 4552, 4555, 4558, 4561, 4564, 4567, 4570, 4573, 4576, 4579, 4582, 4585, 4588, 4591, 4594, 4597, 4599, 4602, 4605, 4608, 4611, 4614, 4617, 4620, 4623, 4626, 4629, 4632, 4635, 4638, 4641, 4644, 4647, 4650, 4653, 4656, 4659, 4662, 4665, 4668, 4671, 4674, 4677, 4680, 4683, 4686, 4689, 4692, 4695, 4698, 4701, 4704, 4707, 4710, 4713, 4716, 4719, 4722, 4725, 4728, 4731, 4734, 4737, 4740, 4743, 4746, 4749, 4752, 4755, 4758, 4761, 4764, 4767, 4770, 4773, 4776, 4779, 4782, 4785, 4788, 4791, 4794, 4797, 4799, 4802, 4805, 4808, 4811, 4814, 4817, 4820, 4823, 4826, 4829, 4832, 4835, 4838, 4841, 4844, 4847, 4850, 4853, 4856, 4859, 4862, 4865, 4868, 4871, 4874, 4877, 4880, 4883, 4886, 4889, 4892, 4895, 4898, 4901, 4904, 4907, 4910, 4913, 4916, 4919, 4922, 4925, 4928, 4931, 4934, 4937, 4940, 4943, 4946, 4949, 4952, 4955, 4958, 4961, 4964, 4967, 4970, 4973, 4976, 4979, 4982, 4985, 4988, 4991, 4994, 4997, 4999, 5002, 5005, 5008, 5011, 5014, 5017, 5020, 5023, 5026, 5029, 5032, 5035, 5038, 5041, 5044, 5047, 5050, 5053, 5056, 5059, 5062, 5065, 5068, 5071, 5074, 5077, 5080, 5083, 5086, 5089, 5092, 5095, 5098, 5101, 5104, 5107, 5110, 5113, 5116, 5119, 5122, 5125, 5128, 5131, 5134, 5137, 5140, 5143, 5146, 5149, 5152, 5155, 5158, 5161, 5164, 5167, 5170, 5173, 5176, 5179, 5182, 5185, 5188, 5191, 5194, 5197, 5199, 5202, 5205, 5208, 5211, 5214, 5217, 5220, 5223, 5226, 5229, 5232, 5235, 5238, 5241, 5244, 5247, 5250, 5253, 5256, 5259, 5262, 5265, 5268, 5271, 5274, 5277, 5280, 5283, 5286, 5289, 5292, 5295, 5298, 5301, 5304, 5307, 5310, 5313, 5316, 5319, 5322, 5325, 5328, 5331, 5334, 5337, 5340, 5343, 5346, 5349, 5352, 5355, 5358, 5361, 5364, 5367, 5370, 5373, 5376, 5379, 5382, 5385, 5388, 5391, 5394, 5397, 5399, 5402, 5405, 5408, 5411, 5414, 5417, 5420, 5423, 5426, 5429, 5432, 5435, 5438, 5441, 5444, 5447, 5450, 5453, 5456, 5459, 5462, 5465, 5468, 5471, 5474, 5477, 5480, 5483, 5486, 5489, 5492, 5495, 5498, 5501, 5504, 5507, 5510, 5513, 5516, 5519, 5522, 5525, 5528, 5531, 5534, 5537, 5540, 5543, 5546, 5549, 5552, 5555, 5558, 5561, 5564, 5567, 5570, 5573, 5576, 5579, 5582, 5585, 5588, 5591, 5594, 5597, 5599, 5602, 5605, 5608, 5611, 5614, 5617, 5620, 5623, 5626, 5629, 5632, 5635, 5638, 5641, 5644, 5647, 5650, 5653, 5656, 5659, 56

Single - 1

nums = [4, 1, 2, 1, 2] int ans = 0

while is once? Ans = 4

Approach = $4 \wedge 1 \wedge 2 \wedge 1 \wedge 2$

$$= 4$$

Code :-

```
int XOR = 0
```

```
for (int i = 0; i < n - 1)
```

```
{ XOR = XOR ^ num[i]; }
```

return XOR

TC $\rightarrow O(N)$

SC $\rightarrow O(1)$

Ques. Given an array of size n. Find the number which appears only once.

Solution:-

1. Using Hash Table

2. Using Hash Map

3. Using Bit Manipulation

4. Using Sorting

5. Using Divide and Conquer

6. Using Linear Time Complexity

7. Using Constant Space Complexity

8. Using Constant Time Complexity

9. Using Constant Space Complexity

10. Using Constant Time Complexity



Find Single Number - II (Hindi)

Problem Statement = Given an array of integers in which exactly two elements appear only once and all the other elements appear exactly twice. Return the two elements that appear only once. You can return them in any order.

Approach :-

- ① First XOR all elements
- ② Identify the rightmost bit
- ③ Convert all numbers into two buckets according to rightmost bit
- ④ Then XOR two buckets' all numbers and find two unique numbers.

Ex:- $\text{nums} = [2, 4, 2, 14, 3, 7, 7, 3]$

$$\begin{aligned}\text{XOR} &= \underline{\underline{2}}^{\wedge} \underline{\underline{4}}^{\wedge} \underline{\underline{2}}^{\wedge} \underline{\underline{14}}^{\wedge} \underline{\underline{3}}^{\wedge} \underline{\underline{7}}^{\wedge} \underline{\underline{7}}^{\wedge} \underline{\underline{3}} \\ &= 14^{\wedge} 4 = 1110^{\wedge} 0100 = 1010\end{aligned}$$

Identify rightmost set bit :- $\text{num} \wedge [\text{num} \wedge (\text{num}-1)]$

$$\text{num} = 1010$$

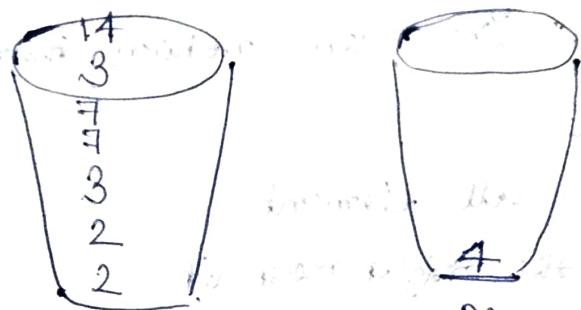
$$\begin{array}{r} \text{num} - 1 = 1001 \\ \hline 1000 \end{array}$$

$$\begin{array}{r} 1010 \\ \hline \wedge 0010 \end{array}$$

→ Isolate the rightmost bit.

Convert all bit numbers in binary and according to rightmost bit set, comment into two bucket of all numbers.

2	\rightarrow	0	0	1	0
4	\rightarrow	0	1	0	0
2	\rightarrow	0	0	1	0
14	\rightarrow	1	1	<u>1</u>	0
3	\rightarrow	0	0	1	1
7	\rightarrow	0	1	1	1
7	\rightarrow	0	1	1	1
3	\rightarrow	0	0	1	1



B1
stone which
lightmost &
is set

A hand-drawn diagram of a cylinder. The top and bottom are represented by two circles. A vertical line connects them. The number '4' is written at the bottom center of the cylinder.

B2
stone which
rightmost bit
is not set bit

then XOR all which is in B1 and XOR which is in B2

$$B_1 = 2^1 2^1 3^1 7^1 7^1 3^1 14^1 \quad 0.14285714285714285714285714285714$$

$$B_2 = 4 \text{ (approximate value)} = 46.6 \text{ (6.1 - 130) } \frac{\text{m}}{\text{s}^2}$$

Code :-

```
vector<int> singleNumber(vector<int> &nums){  
    int xornm = 0;  
    for(int i=0; i<nums.size(); i++){  
        xornm = xornm ^ num[i]; // XOR all elements  
    }  
    if(xornm == INT_MIN){  
        rightmost = INT_MIN;  
    }  
    else{  
        rightmost = (xornm & (xornm-1)) ^ xornm; // find right most bit  
    }  
    int bucket1 = 0;  
    int bucket2 = 0;  
    for(int i=0; i<nums.size(); i++)  
    {  
        if(num[i] & rightmost){  
            bucket1 = num[i] ^ bucket1; // divides into two bucket  
        }  
        else{  
            bucket2 = num[i] ^ bucket2;  
        }  
    }  
    return {bucket1, bucket2};  
}
```

XOR of numbers in a
given range

$N = 1$	$\rightarrow 1$	$(N \oplus 1) = 1$	Ans
$N = 2$	$\rightarrow 3$	$(N \oplus 2) = 3$	$N+1$
$N = 3$	$\rightarrow 0$	$(N \oplus 3) = 0$	0
$N = 4$	$\rightarrow 4$	$(N \oplus 4) = 0$	N

$N = 5$	$\rightarrow 1$	Statement 1
$N = 6$	$\rightarrow 7$	Statement 2
$N = 7$	$\rightarrow 0$	Statement 3
$N = 8$	$\rightarrow 8$	Statement 4

$$N = 9 \rightarrow 1$$

(0 ~ Standard Ans)
(0 ~ Standard Ans)

Approach :-

start point = 3

end point = 5

and calculate standard answer by formula

$$\text{XORTILLN}(3-1) = 1^2$$

$$\text{XORTILLN}(5) = 1^1 2^1 3^1 4^1 5^1$$

$$\text{Ans} = \text{XORTILLN}(5) \wedge \text{XORTILLN}(3-1)$$

$$= 1^1 2^1 3^1 4^1 5^1 1^1 2^1$$

$$= 3^1 4^1 5^1 = 2$$

(Ans with standard method)

Code :-

```

int XORHLLN(int n){
    if (n%4 == 1) return 1;
    if (n%4 == 2) return int(1);
    if (n%4 == 3) return 0;
    return n;
}

```

```

int findRangeXOR(int l, int r){
    return XORHLLN(r+1) ^ XORHLLN(l);
}

```

Divide 2 numbers without multiplication and division

Dry Run :- dividend = 10

divisor = 3
 $n = 10$: 6 iterations (from $10 >= 3$ to $10 < 3$)
 $(d = 3)$: 3rd iteration (from $10 >= 3$ to $10 < 3$)
ans = 0

$\Rightarrow 10 >= 3 \rightarrow \text{cont} = 0$
 $10 >= (3 \ll 1)$
 $\Rightarrow 10 >= 3 \times 2^1$
 $10 >= 6$

$\boxed{\text{cont} = 1}$

ans = $0 + (1 \ll 1)$
 $= 0 + 2^1 = 2$

$n = n - d * (1 \ll 1)$
 $= 10 + 3 \times 2^1$
 $= 10 + 6 = 16$

$$4 >= 3 \quad \text{cont} = 0$$

$$4 >= (3 << 1)$$

$$= 3 \times 2$$

$$4 >= 6$$

$$\text{cont} = 0$$

$$\text{ans} = 2 + (1 << 0)$$

$$= 2 + 2^0$$

$$= 2 + 1 = 3$$

$$n = n - d \cdot (1 << 0)$$

$$= 4 - 3 \cdot 1$$

$$= 1$$

thus the remainder is defined
 $\text{ans} = 3$ satisfies both conditions

Code :-

```
int divide (int dividend, int divisor) {
```

```
    if (dividend == divisor) return 1;
```

```
    if (dividend == INT_MIN && divisor == -1)
```

```
        return INT_MAX;
```

```
    if (divisor == 1) return dividend;
```

```
    bool ispositive = true;
```

```
    if (dividend >= 0 && divisor < 0)
```

```
        ispositive = false;
```

```
    else if (dividend <= 0 && divisor > 0)
```

```
        ispositive = false;
```

```
    else ispositive = true;
```

```
long n = dividend;
tag
long d = divisor;
n = abs(n);
d = abs(d);
int ans = 0;

while (n >= d){
    int count = 0;
    while (n >= (d << (count + 1)))
        { count++;
    }

    ans += (1 << count);
    n -= d * (1 << count);

}

// overflow
if (ans == (1 << 31) && ispositive)
    return INT_MAX;
if (ans == (1 << 31) && !positive)
    return INT_MIN;

ispositive ? ans : -1 * ans;
```