

1.Easy\13.Find_element_present_only_once.cpp

```
1  /*
2  QUESTION:-
3  Given a non-empty array of integers nums, every element appears twice except for one. Find
   that single one.
4  You must implement a solution with a linear runtime complexity and use only constant extra
   space.
5
6
7  Example 1:
8  Input: nums = [2,2,1]
9  Output: 1
10
11 Example 2:
12 Input: nums = [4,1,2,1,2]
13 Output: 4
14 */
15
16 /*
17 APPROACH:-
18 -> We can use XOR operation as we know xor cancels out the same elements
19 -> Intial xr=0 then traverse the entire array and xor each element with xr
20 -> Since only one element is present once and all other are present twice so the remaining
   element would be the
21     one which is present only once cause all other gets cancels out
22 */
23
24 // CODE:-
25 int singleNumber(vector<int> &nums)
26 {
27     int xr = 0;
28     for (int i = 0; i < nums.size(); i++)
29     {
30         xr = nums[i] ^ xr;
31     }
32     return xr;
33 }
34
35 // TIME COMPLEXITY = O(N)
36 // SPACE COMPLEXITY = O(0)
```