Started on	Thursday, 29 August 2024, 10:59 AM
State	Finished
Completed on	Thursday, 5 September 2024, 10:57 AM
Time taken	6 days 23 hours
Marks	1.00/1.00
Grade	**10.00** out of 10.00 (**100**%)

Question **1**

Correct

Mark 1.00 out of 1.00

Given an array `nums` of size `n`, return *the majority element*.

The majority element is the element that appears more than `⌊n / 2⌋` times. You may assume that the majority element always exists in the array.

Example 1:

```
Input: nums = [3,2,3]
Output: 3
```

Example 2:

```
Input: nums = [2,2,1,1,1,2,2]
Output: 2
```

Constraints:

- `n == nums.length`
- `1 <= n <= 5 * 10^4`
- `-2^31 <= nums[i] <= 2^31 - 1`

For example:

Input	Result
3 3 2 3	3
7 2 2 1 1 1 2 2	2

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>

int count(int arr[], int left, int right, int num){
    int count=0;
    for(int i=left ; i<=right ; i++){
        if(arr[i]==num)
            count++;
    }
    return count;
}

int majority(int arr[], int left, int right){
    if(left == right)
        return arr[left];

    int mid = (left+right)/2;
    int leftmajor = majority(arr, left, mid);
    int rightmajor = majority(arr, mid+1 , right);

    if(leftmajor == rightmajor)
        return leftmajor;

    int leftcount = count(arr, left , right, leftmajor);
    int rightcount = count(arr, left, right, rightmajor);
    int n = right - left + 1;

    if(leftcount > n/2)
        return leftmajor;
    if(rightcount > n/2)
        return rightmajor;
    return -1;
}

int main(){
```

```
35      int n;
36      scanf("%d",&n);
37      int arr[n];
38      for(int i=0 ; i<n ; i++)
39          scanf("%d",&arr[i]);
40
41      printf("%d",majority(arr, 0 , n-1));
42  }
```

	Input	Expected	Got	
✔	3 3 2 3	3	3	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

◄ 1-Number of Zeros in a Given Array

Jump to...

3-Finding Floor Value ►