Started on	Tuesday, 1 October 2024, 1:59 PM
State	Finished
Completed on	Tuesday, 1 October 2024, 1:59 PM
Time taken	11 secs
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 %)

```
#include<stdio.h>
int major(int a[],int left,int right);
int count(int a[],int left,int right,int n);
int major(int a[],int left,int right)
{
    if(left==right)
    {
        return a[left];
    }
    int mid=(left+right)/2;
    int lm=major(a,left,mid);
    int rm=major(a,mid+1,right);
    if(lm==rm)
    {
        return lm;
    }
    int lc=count(a,left,right,lm);
    int rc=count(a,left,right,rm);
    return(lc>rc) ? lm:rm;

}
int count(int a[],int left,int right,int n)
{
    int c=0;
    for(int i=left;i<=right;i++)
   {
       if(a[i]==n)
       {
           c++;
       }

}
return c;
}
```

```
int main(){
    int n;
    scanf("%d",&n);
    int a[n];
    for(int i=0;i<n;i++)
    {
        scanf("%d",&a[i]);

    }
    int maj=major(a,0,n-1);
    printf("%d",maj);
}
```

	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 ►