2-Majority Element

Aim:

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 \* 104
* -231 <= nums[i] <= 231 - 1

**For example:**

| **Input** | **Result** |
| --- | --- |
| 3  3 2 3 | 3 |
| 7  2 2 1 1 1 2 2 | 2 |

Answer:(penalty regime: 0 %)

Algorithm:

 Read the integer n and the array a of size n.

 Set k = a[0] as the candidate element.

 Use a recursive function count to find the number of occurrences of k in the array a.

 If the count of k is greater than n / 2, print k.

 Otherwise, iterate through the array and print the first element

Code:

#include<stdio.h>

int c=0;

int count(int a[],int l,int r,int k)

{

if(l<r)

{

int mid =(l+r-1)/2;

if(a[mid]==k)

{

c++;

}

count(a,l,mid,k);

count(a,mid+1,r,k);

}

return c;

}

int main()

{

int n;

scanf("%d",&n);

int a[n];

for(int i=0;i<n;i++)

{

scanf("%d",&a[i]);

}

int k=a[0];

if(count(a,0,n,k)>n/2)

printf("%d",k);

else

{

for(int i=0;i<n/2;i++)

{

if(a[i]!=k)

{

printf("%d",a[i]);

break;

}

}

}

}

Output:

|  | **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- | --- |
|  | 3  3 2 3 | 3 | 3 |  |

Passed all tests!

**Correct**

Marks for this submission: 1.00/1.00.

Result:

The expected output was obtained