Divide and conquer

Program – 2

Aim:

To find the majority element in an array nums of size n, where the majority element is the one that appears more than ⌊n/2⌋ times using the Boyer-Moore Voting Algorithm.

Input:

- An integer nnn, representing the size of the array.
- An array nums[n] of integers.

Code:

```
#include <stdio.h>




int majorityElement(int nums[], int n) {

   int candidate = 0, count = 0;




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

      if (count == 0) {

         candidate = nums[i];

      }

      if (nums[i] == candidate) {

         count++;

      } else {

         count--;

      }

   }



   return candidate;
```

```
}

int main() {
    int n;


    scanf("%d", &n);

    int nums[n];


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


    int majority = majorityElement(nums, n);
    printf("%d\n", majority);

    return 0;
}
```

Output:

	Input	Expected	Got	
✔	3 3 2 3	3	3	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.