2.Medium\03.Majority_element.cpp

```cpp
/*
QUESTION:-
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
*/

/*
APROACH:-
-> Initialize two variables: candidate and count. Set candidate to the first element of the array, and count to 1.
-> Iterate through the array starting from the second element:
    If the current element is equal to the candidate, increment the count by 1.
    If the current element is different from the candidate, decrement the count by 1.
    If the count becomes 0, update the candidate to the current element and set the count to 1 again.
-> After the iteration, the candidate variable will hold the majority element.
Return the candidate as the result.
*/

// CODE:-
int majorityElement(vector<int> &nums)
{
    int candidate = nums[0];
    int vote = 1;
    for (int i = 1; i < nums.size(); i++)
    {
        if (vote <= 0)
            candidate = nums[i];
        if (nums[i] == candidate)
            vote++;
        else
            vote--;
    }
    return candidate;
}

// TIME COMPLEXITY = O(N)
// SPACE COMPLEXITY = O(0)
```