Student name :	Carter Hawks
Student email :	ckh170000@utdallas.edu
Class name :	2336.001_F18
Submitted on :	Sep 16, 2018 07:34 am

solution.cpp

```
// As a preface, this problem has deleted itself from codezinger 3 times and counting, so I have rewritten it c
// Hopefully it does not submit empty...

/* Problem Analysis
 * Given an array of integers, we have to find the highest appearing number and then determine if that value is
 * In other words, if that value appears more than (N/2) times.
 *
 */

/* Problem Design
 * 1. We first have to determine what value appears the most in the array. This can be done by iterating throug
 * element of the array, and comparing it to every other element of the array, while counting up the number of
 * appearances.
 * 2. Keep track of the index with the most appearances along the way.
 * 3. Once the value that appears the most has been determined, we must decide whether or not that value appear
 * majority of the time. This can be done by seeing if the appearances of that value is greater than (N/2) wher
 * array length.
 * 4. Return the value if it is the majority element, otherwise return 0.
 *
 */

#include<iostream>
#include<cstdio>
using namespace std;

 /* Function to find the candidate for Majority */
int findCandidate(int a[], int N) {
        int maxCount = 0;
        int maxIndex = -1;

        // for every value in array a
        for(int i = 0; i < N; i++){
                // determine how many duplicate values that index has.

                // count up by comparing value a[i] to every other value in the array.
                int count = 0;
                for(int j = 0; j < N; j++){
                        if(a[i] == a[j]){
                                count++;
                        }
                }

                // if the count of this value is greater than any previous count, set it to the be current rei
                if(count > maxCount){
                        maxCount = count;
                        maxIndex = i;
                }
        }
        return a[maxIndex];
}

/* Function to check if the candidate occurs more than n/2 times */
bool isMajority(int a[], int cand, int N) {

        int count = 0;
        for(int i = 0; i < N; i++){
```

```
        if(a[i] == cand){
                count++;
        }
    }
        if(count > N/2){
                return true;
        } else {
                return false;
        }
}

// highest level function - connects the other two functions together.
int findMajority(int array[], int N){
    int candidate = findCandidate(array, N);
    if(isMajority(array, candidate, N)){
        return candidate;
    } else {
        return 0;
    }
}

//Your program will be evaluated by this main method and several test cases.
int main(){
    int N;
    cin >> N;
    int A[N];
    for(int i=0; i<N ; i++){
        cin >> A[i];
    }
    cout << findMajority(A,N);
}
```

Name

Custom test case

Input

7 1 2 2 1 2 2 1

Output (Lines:2)

2

Expected Output (Lines:0)

Status

NA

Name

Custom test case

Input

12 2 3 4 3 2 4 3 2 4 2 3 4

Output (Lines:2)

0

Expected Output (Lines:0)

Status

NA

Name

Custom test case

Input
10 1 2 3 4 5 6 7 8 9 0

Output (Lines:2)
0

Expected Output (Lines:0)

Status
NA

Name
Default

Input
5 1 2 1 2 2

Output (Lines:2)
2

Expected Output (Lines:1)
2

Status
Pass
