```c
#include <stdio.h>

// Do binary search to find the earliest occurrence of query in
// the array arr sorted in non-decreasing order
int binSearchEarliest(int *arr, int sta, int fin, int query){
    if(fin == sta){
        if(arr[sta] == query)
            return sta;
        else
            return -1;
    }
    int center = sta + (fin - sta + 1)/2;
    if(arr[center] == query){ // Found at least one occurrences
        // Any earlier occurrences can only be to the left (sorted array)
        int prev = binSearchEarliest(arr, sta, center - 1, query);
        if(prev == -1)
            return center;
        else
            return prev;
    }else if(arr[center] < query){ // Only hope is to search the right
        return binSearchEarliest(arr, center, fin, query);
    }else if(arr[center] > query){ // Only hope is to search the left
        return binSearchEarliest(arr, sta, center - 1, query);
    }
    return -1;
}

// Do binary search to find the last occurrence of query in
// the array arr sorted in non-decreasing order
int binSearchLast(int *arr, int sta, int fin, int query){
    if(fin == sta){
        if(arr[sta] == query)
            return sta;
        else
            return -1;
    }
    int center = sta + (fin - sta + 1)/2;
    if(arr[center] == query){ // Found at least one occurrences
        // Any later occurrences can only be to the right (sorted array)
        int next = binSearchLast(arr, center, fin, query);
        if(next == -1)
            return center;
        else
            return next;
    }else if(arr[center] < query){ // Only hope is to search the right
        return binSearchLast(arr, center, fin, query);
    }else if(arr[center] > query){ // Only hope is to search the left
        return binSearchLast(arr, sta, center - 1, query);
    }
    return -1;
}

// How many times does the query number q occur in the sorted array a?
int numOccurrences(int *arr, int n, int q){
    // Find the earliest occurrence of q
    int start = binSearchEarliest(arr, 0, n - 1, q);
    if(start < 0) // Not found at all
        return 0;
    return binSearchLast(arr, 0, n - 1, q) - start + 1;
}

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

    // How many times does the first element appear?
    int c0 = numOccurrences(arr, n, arr[0]);

    // How many times does the middle element appear?
    int cmid = numOccurrences(arr, n, arr[n/2]);

    // If any element is appearing a majority number of times,
    // it has to be the middle element. If the middle element does not
    // appear a majority number of times, no other element can.
    if(cmid >= n/2 + 1)
        printf("YES\n%d",cmid);
    else
        printf("NO\n%d", c0);
}
```