```c
/* Majority.c */
#include <stdio.h>

//选出具备条件的众数候选者，运用减而治之的策略
int majEleCandidate(int *a, int n)
{
    int maj, i = 0, c = 0;
    for (; i < n; i ++)
    {
        if (0 == c)
        {
            maj = a[i];
            c = 1;
        }
        else
        {
            maj == a[i] ? c ++ : c --;
        }
    }
    return maj;
}

//验证候选者是否为众数
int majEleCheck(int *a, int n, int maj)
{
    int occurrence = 0;
    int i = 0;
    for (; i < n; i ++)
    {
        if (a[i] == maj)
        {
            occurrence ++;
        }
    }
    return (occurrence << 1) > n;
}

//此函数的时间复杂度为O(n)
//如果用分治法求众数时间复杂度至少为O(nlogn)
int majority(int *a, int n, int *maj)
{
    *maj = majEleCandidate(a, n);
    return majEleCheck(a, n, *maj);
}

int main()
{
    int n;
    scanf("%d", &n);
    int a[n];
    int i = 0;
    for (; i < n; i ++)
    {
        scanf("%d", &a[i]);
    }
    int maj;
    if (majority(a, n, &maj))
    {
        printf("majority = %d\n", maj);
    }
    else
    {
        printf("majority = NULL\n");
    }
    return 0;
}
```