

```

1  /* Majority.c */
2  #include <stdio.h>
3
4  //选出具备条件的众数候选者，运用减而治之的策略
5  int majEleCandidate(int *a, int n)
6  {
7      int maj, i = 0, c = 0;
8      for (; i < n; i++)
9      {
10         if (0 == c)
11         {
12             maj = a[i];
13             c = 1;
14         }
15         else
16         {
17             maj == a[i] ? c++ : c--;
18         }
19     }
20     return maj;
21 }
22
23 //验证候选者是否为众数
24 int majEleCheck(int *a, int n, int maj)
25 {
26     int occurrence = 0;
27     int i = 0;
28     for (; i < n; i++)
29     {
30         if (a[i] == maj)
31         {
32             occurrence++;
33         }
34     }
35     return (occurrence << 1) > n;
36 }
37
38 //此函数的时间复杂度为O(n)
39 //如果用分治法求众数时间复杂度至少为O(nlogn)
40 int majority(int *a, int n, int *maj)
41 {
42     *maj = majEleCandidate(a, n);
43     return majEleCheck(a, n, *maj);
44 }
45
46 int main()
47 {
48     int n;
49     scanf("%d", &n);
50     int a[n];
51     int i = 0;
52     for (; i < n; i++)
53     {
54         scanf("%d", &a[i]);
55     }
56     int maj;
57     if (majority(a, n, &maj))
58     {
59         printf("majority = %d\n", maj);
60     }
61     else
62     {
63         printf("majority = NULL\n");
64     }
65     return 0;
66 }

```