

题目描述：

1.众数是指一组数据中出现次数多的那个数，众数可以是多个

2.中位数是指把一组数据从小到大排列，最中间的那个数，如果这组数据的个数是奇数，那最中间那个就是中位数，如果这组数据的个数为偶数，那就把中间的两个数之和除以 2，所得的结果就是中位数

3.查找整型数组中元素的众数并组成一个新的数组，求新数组的中位数

输入描述：

输入一个一维整型数组，数组大小取值范围  $0 < N < 1000$ ，数组中每个元素取值范围  $0 < E < 1000$

输出描述：

输出众数组成的新数组的中位数

示例 1

输入：

10 11 21 19 21 17 21 16 21 18 15

输出：

21

说明：

示例 2

输入：

2 1 5 4 3 3 9 2 7 4 6 2 15 4 2 4

输出：

3

说明：

示例 3

输入：

5 1 5 3 5 2 5 5 7 6 7 3 7 11 7 55 7 9 98 9 17 9 15 9 9 1 39

输出：

7

```
#include <stdio.h>
```

```
int main() {
    int a;
    int b = 0;
    int input[1000];
    while (scanf("%d", &a) != EOF) { // 注意 while 处理多个 case
        // 64 位输出请用 printf("%lld") to
        input[b] = a;
        b++;
        // printf("%d\n", a + b);
    }
}
```

```
int table[1000] = {0};
b = 0;
while (input[b] > 0) {
    table[input[b]]++;
    b++;
}
int max = 0;
for(int i = 0; i < 1000; i++){
    if (table[i] > max) {
        max = table[i];
    }
}
int max_table[1000];
b = 0;
for(int i = 0; i < 1000; i++){
    if (table[i] == max) {
        max_table[b] = i;
        b++;
    }
}
```

```

    }
}
b = 0;
while (max_table[b] > 0){
    b++;
}
if (b%2 == 1) {
    printf("%d",max_table[b/2]);
}
else {
    // float out = ((float)(max_table[b/2]) + (float)(max_table[(b/2)-1]))/2;
    int out = ((max_table[b/2]) + (max_table[(b/2)-1]))/2;
    // printf("%d",max_table[0]);
    printf("%d",out);
}
// printf("%d",max);

return 0;
}

```