摩尔投票法找众数

```
class Solution {
public:
    int majorityElement(vector<int>& nums) {
        int x = 0, votes = 0, count = 0;
        for(int num : nums){
            if(votes == 0) x = num;
            votes += num == x ? 1 : -1;
        }
        // 验证 x 是否为众数
        for(int num : nums)
            if(num == x) count++;
        return count > nums.size() / 2 ? x : 0; // 当无众数时返回 0
    }
};
```

计算排列数和组合数

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

//计算阶乘
/*
注意：0！= 1
*/
int factorial(int n)
{
    int fc=1;
    for(int i=1;i<=n;++i) fc *= i;
    return fc;
}

//计算组合数
/*从n个不同元素中取出m(m≤n)个元素的所有组合的个数，叫做n个不同元素中取出m个元素的组合数。用符号 c(n,m) 表示。
组合数公式：c(n,m)=n!/(m! * (n-m)!)
性质：c(n,m) = c(n,m-n)
递推公式：c(n,m) = c(n-1,m-1) + c(n-1,m)
*/
int combo(int n,int m)
{
    int com=factorial(n)/(factorial(m)*factorial(n-m));
    return com;
}

//计算排列数
/*从n个不同的元素中任取m(m≤n)个元素的所有排列的个数，叫做从n个不同的元素中取出m(m≤n)个元素的排列数。
排列与元素的顺序有关，组合与顺序无关。
排列数公式：A(n,m) = n*(n-1)*...*(n-m+1)=n!/(n-m)!
```

```
排列数公式中总共有m项乘积。
*/
int permutation(int n,int m)
{
    int perm=factorial(n)/factorial(n-m);
    return perm;
}

int main(int argc,char **argv)
{
    int m,n;
    while(cin>>n>>m){
        cout<<"c(n,m)= "<<combo(n,m)<<endl;
        cout<<"A(n,m)= "<<permutation(n,m)<<endl;
    }
    return 0;
}
```

二进制中1的个数

```
class Solution {
public:
    int hammingWeight(uint32_t n) {
        int ret = 0;
        for (int i = 0; i < 32; i++) {
            if (n & (1 << i)) {
                ret++;
            }
        }
        return ret;
    }
};


class Solution {
public:
    int hammingWeight(uint32_t n) {

        int count=0;
        while (n){
            count+=n&1;
            n>>=1;
        }
        return count;

    }
};
// n&(n-1) 解析： 二进制数字 nn 最右边的 11 变成 00 ，其余不变。

class Solution {
    public:
    int hammingWeight(uint32_t n) {
        int res = 0;
        while(n != 0) {
            res++;
```

```
            n &= n - 1;
        }
        return res;
    }
}
```