**组合数**

|  |
| --- |
| Recipient |
|  |
| 题目描述 快递小哥每天都辛苦的送快递，今天他需要送N份快递给N个收件人，第i份快递需要送给第i个收件人。请问其中发生恰好K个送错了的情况数是多少？ 输入 存在多样例。每行输入两个整数N和K，1≤N≤1000,0≤K≤N。如果两个都为0，则表示输入结束，这个样例不需要处理。 输出 每行输出一个样例的结果，因为数值会比较大，所有结果需要对109+7取模。 样例输入 1 1  2 1  3 2  1000 1000  0 0 样例输出 0  0  3  37043040  解释：  组合数和错排这里的解法都是用到的递推**组合数：**递推公式是C(n,m)=C(n-1,m-1)+C(n,m-1);因为对于m个物体中的一个物体，要么被选，要么不被选，两种可能，所以如果被选了，就只需要从剩下的m-1个物体中选n-1个即C(n-1,m-1),  如果没有被选，就需要从剩下的m-1个物体中选n个即C(n,m-1),然后相加，即C(n,m)=C(n-1,m-1)+C(n,m-1)。**错排公式：**递推公式为f(n)=(n-1)\*(f(n-1)+f(n-2));因为对于n个物体错排，如果第一个物体错排的是第k个物体(n-1)种选择，那么分析第k个物体，如果第k个物体的位置错排的是第1个物体那么剩下的n-2个物体再进行错排就可以了即f(n-2)  如果第k个物体的位置错排的不是第1个物体而是w那么就相当于把第k个物体扔了直接再第一个物体下面排w即f(n-1)然后相加。即f(n)=(n-1)\*(f(n-1)+f(n-2))。  代码如下 |

#include<iostream>

#include<cstdio>

#include<cstring>

using namespace std;

const int mod =1000000007;

long long c[1005][1005],f[1005];

void init1()

{

for(int i=0;i<1005;i++){

c[i][i]=c[0][i]=1;

}

for(int i=1;i<1005;i++){

for(int j=i+1;j<1005;j++){

c[i][j]=(c[i-1][j-1]+c[i][j-1])%mod;

}

}

}

void init2(){

f[0]=1;

f[1]=0;

f[2]=1;

for(int i=3;i<1005;i++){

f[i]=((i-1)\*(f[i-2]+f[i-1]))%mod;

}

}

int main(){

int n,k;

init1();

init2();

while(~scanf("%d%d",&n,&k),n+k){;

printf("%I64d\n",(c[n-k][n]\*f[k])%mod);

}

return 0;

}

**组合数：**

**1.卡特兰数**

令h(0)=1,h(1)=1，catalan数满足递推式[1]  ：

**h(n)= h(0)\*h(n-1)+h(1)\*h(n-2) + ... + h(n-1)h(0) (n>=2)**

例如：h(2)=h(0)\*h(1)+h(1)\*h(0)=1\*1+1\*1=2

h(3)=h(0)\*h(2)+h(1)\*h(1)+h(2)\*h(0)=1\*2+1\*1+2\*1=5

另类递推式[2]  ：

**h(n)=h(n-1)\*(4\*n-2)/(n+1);**

递推关系的解为：

**h(n)=C(2n,n)/(n+1) (n=0,1,2,...)**

递推关系的另类解为：

**h(n)=c(2n,n)-c(2n,n-1)(n=0,1,2,...)**

**卡特兰数程序：**

#include<stdio.h>

#define n 20

main()

{

long long a[20]={0,0,0,0,0,0,0,0,0,0};

int i,j;

a[0]=1;

a[1]=1;

for(i=2;i<n;i++)

for(j=0;j<n;j++)

if(i>j)

a[i]+=a[j]\*a[i-j-1];

for(i=0;i<n;i++)

printf("%lld\n",a[i]);

}

**序列：1 1 2 5 14 42 132 429**

**1430 4862 16796**

**Problem 2020 组合**

### Time Limit: 1000 mSec    Memory Limit : 32768 KB

## IMG_257Problem Description

给出组合数C(n,m), 表示从n个元素中选出m个元素的方案数。例如C(5,2) = 10, C(4,2) = 6.可是当n,m比较大的时候，C(n,m)很大！于是xiaobo希望你输出 C(n,m) mod p的值！

## IMG_258Input

输入数据第一行是一个正整数T,表示数据组数 (T <= 100) 接下来是T组数据，每组数据有3个正整数 n, m, p (1 <= m <= n <= 10^9, m <= 10^4, m < p < 10^9, p是素数)

## IMG_259Output

对于每组数据，输出一个正整数，表示C(n,m) mod p的结果。

## IMG_260Sample Input

2

5 2 3

5 2 61

## IMG_261Sample Output

1

10

## IMG_262Source

FOJ有奖月赛-2011年04月（校赛热身赛）

#include <iostream>

#include <string.h>

#include <stdio.h>

using namespace std;

typedef long long LL;

LL n,m,p;

LL quick\_mod(LL a, LL b)

{

LL ans = 1;

a %= p;

while(b)

{

if(b & 1)

{

ans = ans \* a % p;

b--;

}

b >>= 1;

a = a \* a % p;

}

return ans;

}

LL C(LL n, LL m)

{

if(m > n) return 0;

LL ans = 1;

for(int i=1; i<=m; i++)

{

LL a = (n + i - m) % p;

LL b = i % p;

ans = ans \* (a \* quick\_mod(b, p-2) % p) % p;

}

return ans;

}

LL Lucas(LL n, LL m)

{

if(m == 0) return 1;

return C(n % p, m % p) \* Lucas(n / p, m / p) % p;

}

int main()

{

int T;

scanf("%d", &T);

while(T--)

{

scanf("%I64d%I64d%I64d", &n, &m, &p);

printf("%I64d\n", Lucas(n,m));

}

return 0;

}