

小数据直接循环

```
for(int i=0; i< N ;i ++)  
    for(int j =0; j<= i ;j++)  
        if(!j) f[i][j] =1;  
        else f[i][j] =f[i-1][j] +f[i-1][j-1];
```

大数据用卢卡斯定理

```
#include<iostream>  
#include<algorithm>  
  
using namespace std;  
  
typedef long long LL;  
  
int qmi(int a,int k,int p)  
{  
    int res = 1;  
    while(k)  
    {  
        if(k&1)res = (LL)res*a%p;  
        a = (LL)a*a%p;  
        k>>=1;  
    }  
    return res;  
}  
  
int C(int a,int b,int p)//自变量类型int  
{  
    if(b>a)return 0;//漏了边界条件  
    int res = 1;  
    //  $a!/(b!(a-b)!)$  =  $(a-b+1)*\dots*a / b!$  分子有b项  
    for(int i=1,j=a;i<=b;i++,j--)//i<=b而不是<  
    {  
        res = (LL)res*j%p;  
        res = (LL)res*qmi(i,p-2,p)%p;  
    }  
    return res;
```

```
}
//对公式敲
int lucas(LL a,LL b,int p)
{
    if(a<p && b<p)return C(a,b,p);//lucas递归终点是 $C_{bk}^{ak}$ 
    return (LL)C(a%p,b%p,p)*lucas(a/p,b/p,p)%p;//a%p后肯定是<p的,所以可以用C(),
    但a/p后不一定<p 所以用lucas继续递归
}

int main()
{
    LL a,b;
    int p;
    cin >> a >> b >> p;
    cout << lucas(a,b,p) << endl;
    return 0;
}
```