排列组合

排列的定义：

从 n 个不同的元素中，任取 $m(m \leq n)$ 个不同的元素按照一定的顺序排成一列。用 $A(n,m)$或 A_n^m表示。

计算公式：$A_n^m = n(n-1)(n-2)\ldots(n-m+1) = \frac{n!}{(n-m)!}$

规定$0! = 1$

第二类斯特林数

第二类斯特林数表示将 n 个不同的元素拆分成 m 个集合的方案数，记为 $S(n,m)$ 或 $\left\{\begin{matrix} n \\ m \end{matrix}\right\}$

问题描述：

将 n 个不同的球放入 m 个无差别的盒子中，要求盒子非空，有几种方案？

其方案数公式：

$S(n,m) = \frac{1}{m!}\sum_{k=0}^{m}(-1)^k C(m,k)(m-k)^k$

递推式：

假设要把 $n+1$ 个元素分为 m 个集合，则分析如下：

(1) 如果 n 个元素构成了 $m-1$ 个集合，那么第 $n+1$ 个元素单独构成一个集合。方案数 $S(n,m-1)$

(2) 如果 n 个元素构成了 m 个集合，那么将第 $n+1$ 个元素插入到任何一个集合，方案数为 $m * S(n-m)$

综合以上情况可以得到递推式：

$S(n+1,m) = S(n,m-1) + m * S(n,m)$

$S(n,m) = S(n-1,m-1) + m * S(n-1,m)$

性质：

(1) $S(n,0) = 0^n$ (2) $S(n,1) = 1$ (3) $S(n,n) = 1$ (4) $S(n,2) = 2^{n-1} - 1$

(5) $S(n,n-1) = C(n,2)$ (6) $S(n,n-2) = C(n,3) + 3 \cdot C(n,4)$

(7) $S(n,3) = \frac{1}{2}(3^{n-1}+1) - 2^{n-1}$ (8) $S(n,n-3) = C(n,4) + 10 \cdot C(n,5) + 15 \cdot C(n,6)$

(9) $\sum_{k=0}^{n} S(n,k) = B_n$ B_n是贝尔数

C++：高精度第二类斯特林数

```
#include <bits/stdc++.h>
using namespace std;
const int L = 1001;//调整字符串长度，本题1000足矣
string add(string a,string b)//只限两个非负整数相加
{
    string ans;
    int na[L]={0},nb[L]={0};
    int la=a.size(),lb=b.size();
    for(int i=0;i<la;i++) na[la-1-i]=a[i]-'0';
    for(int i=0;i<lb;i++) nb[lb-1-i]=b[i]-'0';
    int lmax=la>lb?la:lb;
    for(int i=0;i<lmax;i++) na[i]+=nb[i],na[i+1]+=na[i]/10,na[i]%=10;
    if(na[lmax]) lmax++;
    for(int i=lmax-1;i>=0;i--) ans+=na[i]+'0';
    return ans;
}
int na[L];
string mul(string a,int b)//高精度a乘单精度b模板
{
    string ans;
    int La=a.size();
    fill(na,na+L,0);
    for(int i=La-1;i>=0;i--) na[La-i-1]=a[i]-'0';
    int w=0;
    for(int i=0;i<La;i++) na[i]=na[i]*b+w,w=na[i]/10,na[i]=na[i]%10;
    while(w) na[La++]=w%10,w/=10;
    La--;
    while(La>=0) ans+=na[La--]+'0';
    return ans;
}
int n, m;
string f[101][101];
int main(){
    for ( int i = 1; i <= 100; i++ )
    f[i][1] = "1";//初始化，一个盒子（m=1）的时候只有一种放法
    for ( int i = 2; i <= 100; i++ )
    for ( int j = 1; j <= i; j++ )
    f[i][j] = add ( f[i-1][j-1], mul ( f[i-1][j], j ) );//带上高精度运算的状态转移
    while ( cin >> n >> m ){
        if ( n < m ) printf ( "0\n" );//特判输出0
        else cout << f[n][m] << endl;//输出每个n，m对应的答案f[n][m]
    }
    return 0;//华丽落幕
}
```

Python：第二类斯特林数

```python
S = [[0 for i in range(105)] for j in range(105)]
n = m = 100
for i in range(1, n):
    S[i][i] = 1
    for j in range(1, i):
        S[i][j] = S[i-1][j]*j + S[i-1][j-1]
while True:
    n, m = map(int, input().split())
    print(S[n][m],end = "\n")
```