1315：集合的划分

题目描述

设S是一个具有n个元素的集合，S＝□a1，a2，……，an□现将S划分成k个满足下列条件的子集合S1，S2，……，Sk，且满足：

1. Si≠∅

2. Si∩Sj＝∅　　　(1≤i，j≤k，i≠j)

3. S1∪S2∪S3∪…∪Sk＝S

则称S1，S2，……，Sk是集合S的一个划分。它相当于把S集合中的n个元素a1，a2，……，an放入k个(0＜k≤n＜30)无标号的盒子中，使得没有一个盒子为空。请你确定n个元素a1，a2，……，an 放入k个无标号盒子中去的划分数S(n,k)。

输入

给出n和k。

输出

n个元素a1，a2，……，an 放入k个无标号盒子中去的划分数S(n,k)。

输入样例

10 6

输出样例

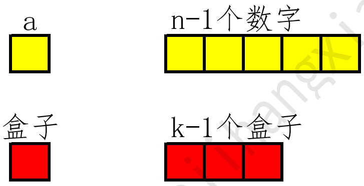
22827

解析

我们设计算划分数的函数为Count(n,k)

考虑一般情况，对于任意的含有n个元素的集合S，放入k个无标号的盒子中去，对于任意一个元素 a，只可能有以下两种情况：

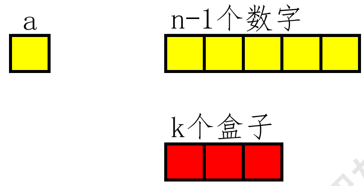
1、a单独在一个盒子里，其他n-1个元素放k-1个盒子，那么问题就转化成为了其他n-1个元素放入k-1个盒子共有多少种方法，即Count(n-1,k-1)，如下图所示：



2、a与其他的元素一起构成一个子集，分两步：

第一步：先把剩下的n-1个元素(除去a)放进k个盒子，即Count(n-1, k)；

第二步：再把a放进k个盒子，每放入一个盒子，就产生了一种全新的方法，那么总的方法数就等于k*Count(n-1,k)，如下图所示：



最终的方案数就是这两个方案的总数，即：

Count(n,k) = Count(n-1,k-1) + k*Count(n-1,k)

这时，我们还需要考虑两个极端情况：

1、由于不允许存在空集，因此，当数字的个数与盒子的数量相同时，分配的方案只有1种，即每个数字独立一个盒子。

2、当只有一个盒子的时候，方案也只有一种，就是大家装在一起嘛。

3、当盒子数量小于待分配的数字数量时，或者不存在可使用的盒子时，不存在合理的分配方案。

编码

```cpp
#include <bits/stdc++.h>

using namespace std;

//存在乘法，使用longlong 类型
long long count(int n, int k) {
    //当可分配的盒子与目标数字的数量相同或者只有一个盒子时，方案只有一种
    if (n == k || k == 1)
        return 1;
    //当待分配的数字小于盒子数量或者不存在盒子时，不存在方案
    if (n < k || k == 0)
        return 0;
    //a独立时的数量                        a与其他数字在一起时的数量
    return count(n - 1, k - 1) + k * count(n - 1, k);
}

int n, k;

int main() {
    cin >> n >> k;
    cout << count(n, k);
    return 0;
}
```