**题目描述 Description**

将整数n分成k份，且每份不能为空，任意两种划分方案不能相同(不考虑顺序)。  
例如：n=7，k=3，下面三种划分方案被认为是相同的。  
1 1 5

1 5 1

5 1 1  
问有多少种不同的分法。

**输入描述 Input Description**

输入：n，k (6<n<=200，2<=k<=6)

**输出描述 Output Description**

输出：一个整数，即不同的分法。

**样例输入 Sample Input**

 7 3

**样例输出 Sample Output**

4

**数据范围及提示 Data Size & Hint**

 {四种分法为：1，1，5;1，2，4;1，3，3;2，2，3;}

f[i,j]表示将i分成j份的方案数。

N数分K分,若最小值为1,则1占一份,剩余N-1数分K-1份；  
若最小值不为1,则i-j,还可以分成j分.  
i-j 就相当于把每份（即j份）都减去1  
比如 f[7][3] 中的 f[4][3] 为 1 1 2 对应 f[7][3] 中的 2 2 3

f[i][j]表示将i划分成j份的方法数。   
f[i][j]=f[i-j][j]+f[i-1][j-1];

1.f[i-j][j]:最小的一份>=2的方案总数。   
所以所有数>=2，有限制，   
但此方案数等于把每份都减1的方案数，   
等效之后就没有限制了。

2.f[i-1][j-1]最小的一份=1的方案总数。

动态规划的递推形式还可以写递归

#include<bits/stdc++.h>

using namespace std;

int doit(int n,int k)

{

if(n<k)return 0;

if(n==k||n<=1||k<=1)return 1;

return doit(n-1,k-1)+doit(n-k,k);

}

int main()

{

int n,k;

cin>>n>>k;

cout<<doit(n,k);

}