A murder happened in the hotel. As the best detective in the town, you should examine all the N rooms of the hotel immediately. However, all the doors of the rooms are locked, and the keys are just locked in the rooms, what a trap! You know that there is exactly one key in each room, and all the possible distributions are of equal possibility. For example, if N = 3, there are 6 possible distributions, the possibility of each is 1/6. For convenience, we number the rooms from 1 to N, and the key for Room 1 is numbered Key 1, the key for Room 2 is Key 2, etc.   
To examine all the rooms, you have to destroy some doors by force. But you don’t want to destroy too many, so you take the following strategy: At first, you have no keys in hand, so you randomly destroy a locked door, get into the room, examine it and fetch the key in it. Then maybe you can open another room with the new key, examine it and get the second key. Repeat this until you can’t open any new rooms. If there are still rooms un-examined, you have to randomly pick another unopened door to destroy by force, then repeat the procedure above, until all the rooms are examined.   
Now you are only allowed to destroy at most K doors by force. What’s more, there lives a Very Important Person in Room 1. You are not allowed to destroy the doors of Room 1, that is, the only way to examine Room 1 is opening it with the corresponding key. You want to know what is the possibility of that you can examine all the rooms finally.

Input

The first line of the input contains an integer T (T ≤ 200), indicating the number of test cases. Then T cases follow. Each case contains a line with two numbers N and K. (1 < N ≤ 20, 1 ≤ K < N)

Output

Output one line for each case, indicating the corresponding possibility. Four digits after decimal point are preserved by rounding.

Sample Input

3

3 1

3 2

4 2

Sample Output

0.3333

0.6667

0.6250

Hint

Sample Explanation

When N = 3, there are 6 possible distributions of keys:

Room 1 Room 2 Room 3 Destroy Times

#1 Key 1 Key 2 Key 3 Impossible

#2 Key 1 Key 3 Key 2 Impossible

#3 Key 2 Key 1 Key 3 Two

#4 Key 3 Key 2 Key 1 Two

#5 Key 2 Key 3 Key 1 One

#6 Key 3 Key 1 Key 2 One

In the first two distributions, because Key 1 is locked in Room 1 itself and you can’t destroy Room 1, it is impossible to open Room 1.

In the third and forth distributions, you have to destroy Room 2 and 3 both

题意：有n个锁着的房间和对应n扇门的n把钥匙，每个房间内有一把钥匙。你可以破坏一扇门，取出其中的钥匙，然后用取出钥匙打开另一扇门（如果取出的钥匙能打开房门则接着打开，取出其中钥匙，如此往复，若打不开则继续破坏一扇门）。最多可以破坏k（k<=n）扇门，但是编号为1的门只能用钥匙打开。求能打开所有门（被破坏或是被钥匙打开）的概率。

解题思路：钥匙和门的关系是成环状的，打开一个门之后，该环内的所有房间都可以进入，怎么说呢，就拿Hint里的#6来举例，Room1 Room2 Room3是在一个环当中的，假设我破坏了Room3，那么我取出Room3内的钥匙Key2就可以打开Room2，而Room2里有钥匙Key1，那我们又可以打开Room1。

因此，该题就转化成了求N个房间形成1~K个环有多少种可能，然后除以总的分配方案数即为题目要我们求的概率。

首先，总的分配方案数是比较好求的，N的全排列N！种，因为N<=20，有可能超int型范围，所以\_\_int64或long long是必不可少的

其次就是求N个房间成i个环的种类数了，而第一类斯特林数S(N,K)=S(N-1,K-1)+(N-1)\*S(N-1,k)恰恰就是求N个元素形成K个非空循环排列的方法数

讲到这里，你或许会有点疑问，为什么这公式和百度百科上的公式不一样，因为百度百科给出的其实是第二类斯特林数公式

剩下的就是枚举形成的环，但是要排除掉编号为1的房间独立成环的可能

S(N,M)-S(N-1,M-1)，表示N个元素形成M个环，减去1独自成环，即剩下的N-1个元素形成M-1个环，算得的结果便是所求值

算是第一类斯特灵数的模板题吧

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

#include<queue>

#include<math.h>

#include<vector>

#include<map>

#include<set>

#include<stdlib.h>

#include<cmath>

#include<string>

#include<algorithm>

#include<iostream>

#define exp 1e-10

using namespace std;

const int N=21;

long long f[N],s[N][N];

int main()

{

int T,n,k,i,j;

f[0]=1;

for(i=1;i<N;i++)

f[i]=f[i-1]\*i; //求阶乘

for(i=1;i<N;i++)

{

s[i][i]=1;

for(j=1;j<i;j++)

s[i][j]=s[i-1][j-1]+(i-1)\*s[i-1][j];

}

scanf("%d",&T);

while(T--)

{

long long ans=0;

scanf("%d%d",&n,&k);

for(i=1;i<=k;i++)

ans+=s[n][i]-s[n-1][i-1];

printf("%.4f\n",(double)ans/f[n]);

}

}