我的算法思想：设F(X,Y)是将X个苹果放入Y个盘子里。分析：当X<Y时，即比如说将5个苹果放进8个盘子里，根据题意，和将5个苹果放进5个盘子是一样的结果（5，1，1和1，5，1 是同一种分法，且盘子相同）。

继续分析，利用动态规划的思想，假如将7个苹果放进3个盘子里，可以分2种情况考虑：1，空着一个盘子不放，即将7个苹果放进2个盘子里；2，先每个盘子均放进一个苹果，再按照本规则继续放下去。

于是有递推公式：***F(X,Y)=F(X,Y-1)+F(X-Y,Y)***

以7，3为例，**F(7,3)=F(7,2)+F(4,3)=F(7,1)+F(5,2)+F(4,2)+F(1,3)=F(7,1)+F(5,1)+F(3,2)。。。。。。**

**易知，F(X,1)，F(1,Y),F(0,Y),F(X,0)都等于1；**

解题思路：笔者对组合数学并不了解，见别人都说是组合数学题，那就归到这一类吧，这题的思想是找递归关系，我们不妨令f(m,n)表示m个苹果放到n个盘子里有多少种放法，下面对不同的情况给予讨论:

(1):当盘子数为1的时候，只有一种放法就是把所有苹果放到一个盘子里。

(2):当苹果数为1的时候，也只有一种放法，注意题目中说明，盘子之间并无顺序,所以不管这个苹果放在哪个盘子里，结果都算一个。

(3):当m<n时，因为此时最多只能放到m个盘子中去（一个里放一个），实际上就相当于把m个苹果放到m个盘子里一样，也就是f(m,m);

(4):当m==n时,此时分两种情况讨论，一种是一个盘子里放一个，只是一种，第二种是，至少有一个盘子里不放苹果这就相当于是f(m,m-1);

(5):当m>n时，也分两种情况讨论，一种是至少有一个盘子里不放苹果，这样子就相当于f(m,n-1),第二种是，先取出n个苹果一个盘子里放一个，再将剩下的m-n个苹果放到n个盘子里去，即f(m-n,n);

综上所述:

得到递归表达式：

f(m,n)=1 当 m=1或n=1；

f(m,n)=f(m,m) 当m<n;

f(m,n)=1+f(m,m-1) 当m=n;

f(m,n)=f(m-n,n)+f(m,n-1);

在递归的过程中采用记忆化搜索可以减少不必要的时间，算过的东西就不要再算了。

#include <iostream>

#include <cstdio>

#include <cstring>

#include <cmath>

#include <algorithm>

using namespace std;

int cnt;

int dfs(int m,int n)

{

if(m==0||n==1)return 1;

else

{

if(m<n)return dfs(m,m);

else return dfs(m,n-1)+dfs(m-n,n);

}

}

int main()

{

int T,m,n;

cin>>T;

while(T--)

{

cnt=0;

cin>>m>>n;

cout<<dfs(m,n)<<endl;

}

return 0;

}