在N\*N的方格棋盘放置了N个皇后，使得它们不相互攻击（即任意2个皇后不允许处在同一排，同一列，也不允许处在与棋盘边框成45角的斜线上。   
你的任务是，对于给定的N，求出有多少种合法的放置方法。

**Input**

共有若干行，每行一个正整数N≤10，表示棋盘和皇后的数量；如果N=0，表示结束。

**Output**

共有若干行，每行一个正整数，表示对应输入行的皇后的不同放置数量。

**Sample Input**

1

8

5

0

**Sample Output**

1

92

10

#include<bits/stdc++.h>

using namespace std;

int sum,n;

int column[20];

int a[20];

void dfs(int x)//行

{

int i,j;

if(x>n)

{

sum++;

return;

}

for(i=1;i<=n;i++)//列

{

column[x]=i;//x行放棋子的列数是i

int sf(0);

for(j=1;j<x;j++)//看之前的行有没有放棋子的列是i的，如果有就不行了，或者斜行不行

if(column[j]==i||abs(j-x)==abs(column[j]-column[x]))

{

sf=1;

break;

}

if(sf==0)

dfs(x+1);

//之所以没有column[x]=0类似这么处理是因为会被后来的数据覆盖

}

}

int main()

{

// freopen("input.txt", "r", stdin);

n=11;

for(n=1;n<=11;n++)

{

sum=0;

dfs(1);

a[n]=sum;

}

while(cin>>n&&n)

cout<<a[n]<<endl;

return 0;

}

//之所以要在dfs中有一层j循环来扫前面看这一列是否被占用（完全可以开一个数组，只要这列被占了就赋值1，否则为0就行了）只为了判断斜行