**COUNT N-QUEEN**

You are given an empty chess board of size N\*N. Find the number of ways to place N queens on the board, such that no two queens can kill each other in one move. A queen can move vertically, horizontally and diagonally.

**Input Format:**

A single integer N, denoting the size of chess board.

**Constraints:**

1<=N<=11

**Output Format**

A single integer denoting the count of solutions.

**Sample Input**

4

**Sample Output**

2

Program-

#include<iostream>

using namespace std;

int count=0;

bool issafe(int i,int n,int arr[][11],int j){

for(int row=0;row<i;row++){

if(arr[row][j]==1){

return false;

}

}

int x=i;

int y=j;

while(x>=0 && y>=0){

if(arr[x][y]==1){

return false;

}

x--;

y--;

}

x=i;

y=j;

while(x>=0 && y<n){

if(arr[x][y]==1){

return false;

}

x--;

y++;

}

return true;

}

bool nqueen(int arr[][11],int i,int n){

//base case

if(i==n){

count=count+1;

}

//recursiv case

for(int j=0;j<n;j++){

if(issafe(i,n,arr,j)){

arr[i][j]=1;

int nextqueenrakhpaye=nqueen(arr,i+1,n);

if(nextqueenrakhpaye){

return true;

}

arr[i][j]=0;

}

}

return false;

}

int main(){

int n;

cin>>n;

int arr[11][11]={0};

nqueen(arr,0,n);

cout<<count;

return 0;

}