

**Aim:**

**N-queens** problem is the problem of placing **N** queens on an **N x N** square chessboard (grid) such that no two queens attack each other.

Given an integer **N** return the count of all distinct solutions to the **N queens problem**.

Each solution contains a distinct board configuration of the N queens.

**You can see two possibilities for N = 4:**

	0	1	2	3
0		Q1		
1				Q2
2	Q3			
3			Q4	

  

	0	1	2	3
0			Q1	
1	Q2			
2				Q3
3		Q4		

You can see in 4 x 4 chessboard their are 2 possibilities where Queens are placed in safe states

**Sample test case 1:****Input:**

4 // First line of input is the size of the Chess board (square grid

**Output:**

2 // No of ways possible to place queens in safe state

**Sample test case 1:****Input:**

2 // First line of input is the size of the Chess board (square grid

**Output:**

0 // No of ways possible to place queens in safe state

You just have to complete the function **nQueen()** with parameter passed n represents the n x n size of the chess board and function returns the total no of solutions feasible to place queens in safe state, and if the their solution is no solution exists then return 0.

**Constraints to be followed:**

$2 \leq n \leq 10$

**Instructions:** To run your custom test cases strictly follow the input and output layout as mentioned in the visible sample test cases.

**Source Code:**

nQueenProblem.c

```
/* Write complete code for the required soluton */
#include <stdio.h>
#include <stdbool.h>
#define MAX 15
```

```
int count=0;
int board[MAX];

bool isSafe(int row,int col){
    for(int i=0;i<row;i++){
        if(board[i]==col|| board[i]-i==col-row || board[i]+i==col+row){
            return false;
        }
    }
    return true;
}

void solveQueen(int row, int n){
    if(row==n){
        count++;
        return;
    }
    for(int col=0;col<n;col++){
        if(isSafe(row,col)){
            board[row]=col;
            solveQueen(row+1,n);
        }
    }
}

int main(){
    int n;
    scanf("%d",&n);
    count=0;
    solveQueen(0,n);
    printf("%d\n",count);
    return 0;
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
2
0

Test Case - 2
User Output
3
0