/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Online Java Compiler.

Code, Compile, Run and Debug java program online.

Write your code in this editor and press "Run" button to execute it.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

import java.util.\* ;

public class Main

{

public static void main(String[] args) {

int n = 4 ;

boolean board[][] = new boolean[4][4] ;

System.out.println(queens(board,0));

}

public static int queens(boolean board[][],int row)

{

if(row==board.length)

{

display(board) ;

return 1 ;

}

int count = 0 ;

for(int col=0;col<board.length;col++)

{

if(isSafe(board,row,col))

{

board[row][col] = true ;

count = count + queens(board,row+1) ;

board[row][col] = false ;

}

}

return count ;

}

public static boolean isSafe(boolean board[][],int row,int col)

{ // vertical row check

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

{

if(board[i][col])

{

return false ;

}

}

// diagonal left check

int maxleft = Math.min(row,col) ;

for(int i=1;i<=maxleft;i++)

{

if(board[row-i][col-i])

{

return false ;

}

}

// diagonal right check ;

int maxright = Math.min(row,board.length-col-1) ;

for(int i=1;i<=maxright;i++)

{

if(board[row-i][col+i])

{

return false ;

}

}

return true ;

}

public static void display(boolean board[][])

{

for(boolean[] arr: board)

{

for(boolean element:arr)

{

if(element)

{

System.out.print("Q ");

}

else

{

System.out.print("X ");

}

}

System.out.println();

}

System.out.println();

}

}