**CODE**

//n Queen

import java.util.Scanner;

public class Queens

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

System.out.println("Enter the value of n");

int n=sc.nextInt();

char board[][]=new char[n][n];

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

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

board[i][j]='-';

if(solveNQueens(board,0,n)) solution(board,n);

else System.out.println("No solution exists");

}

public static void solution(char board[][], int n)

{

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

{

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

System.out.print(" "+board[i][j]+" ");

System.out.println();

}

}

public static boolean isSafe(char board[][], int row, int column, int n)

{

int i,j;

for(i=0;i<column;i++)

{

if(board[row][i]=='Q') return false;

}

for(i=row,j=column; i>=0 && j>=0;i--,j--)

{

if(board[i][j]=='Q') return false;

}

for(i=row,j=column; i<n && j>=0;i++,j--)

{

if(board[i][j]=='Q') return false;

}

return true;

}

public static boolean solveNQueens(char board[][], int column, int n)

{

if(column>=n) return true;

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

{

if(isSafe(board,i,column,n))

{

board[i][column]='Q';

if(solveNQueens(board, column+1,n)) return true;

board[i][column]='-';

}

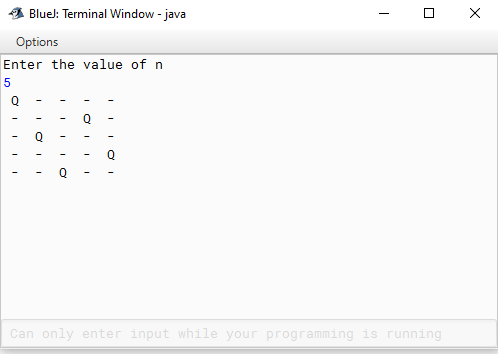
}

return false;

}

}

**OUTPUT**

****