N-Queens

In the world of chess, there are numerous challenges and brain teasers. One such thought exercise is the n-queens problem. Imagine I give you an integer greater than 0 called n. You now magically have a chessboard of n x n dimension upon which you must place n queens. However, each queen must be placed such that no queen on the board is threatened by another.

# Requirements

You will write a recursive algorithm to solve the n-queens problem as described above.

1. Create a program that prompts the user to enter a positive integer value
2. Your program will display the following:
   1. The value of n
   2. The total number of solutions for the given value.
   3. Print all solutions for the given value along with the number of steps it took to find the solution.
      1. A step is defined as placing a queen on the board. For each solution output, the number of steps required to find that solution.

Here is an example of correct program output for an input of 4:

n=4

Total Solutions: 2

Solution 1: found in 50 steps

- Q - -

- - - Q

Q - - -

- - Q -

Solution 2: found in 100 steps

- - Q -

Q - - -

- - - Q

- Q - -

For an input value of 6…

n=6

Total Solutions: 4

Solution 1: found in 75 steps

[Continue to print each solution]

Upload a well-named archive (LastNameFI\_NQueens.zip) of your solution to the LMS.

# Rubric

**Automatic Zero:** You fail to deliver your code base to the LMS, you fail to pass-off with your instructor, or your application throws an exception during runtime.

(10 points) Recursive algorithm

(10 points) Correct solutions

(10 points) A reasonable number of steps to calculate