LAB S11 – KNIGHT'S TOUR

A chess board consists of 64 squares arranged in an 8-by-8 grid. In the game of chess, the knight is the most interesting piece. While all other pieces may only move in straight lines, a knight always moves by traveling two spaces in any of the four primary directions (left, right, up, or down) and then one space in a perpendicular direction, forming the classic "L" shape.

Write a recursive method that will allow a knight to start in the top left corner of the board and travel to every square on the board (using only legal knight moves) without ever landing on the same square more than once; the knight may not leave the board.

Your output for this lab should be a table representing a chess board in which each square is labeled with the number of the step on which the knight touches it; the squares should be labeled from 0 to 63. The rows and columns of the table should be labeled 0 through 7. An incomplete solution is shown below; The number symbol (#) is used in place of the numbers 11 through 63.

```
    0   1   2   3   4   5   6   7
0   0   #   #   #   2   #   #   #
1   #   #   1   #   #   #   #   #
2   #   #   #   #   #   3   #   9
3   #   #   #   #   # 10   #   4
4   #   #   #   #   #   #   8   #
5   #   #   #   #   #   #   5   #
6   #   #   #   #   #   7   #   #
7   #   #   #   #   #   #   #   6
```

Criteria

1) You must use a recursive method to solve this problem. You may create as many non-recursive helper methods as you like.
2) The output must appear in the same format as the example above.
3) Pre and post conditions must be included for each method you write if you submit your lab through Canvas (labs shown to Holm in class do not require pre and post conditions).

Submission

1) Submit your code for this lab on a Googly Doc through Canvas by 11:59 pm Wednesday, March 18. Alternatively, you may show your lab to Holm in class before the due date. ***If you submit through Canvas, you must provide a sample output.***