CS260 - Programming Assignment 0 - Recursion

The purpose of this assignment is to give you some practice working with recursion.

Overview

For this assignment you'll write a program to find solutions for the Knights Tour. using brute force (just trying all possible solutions)

The program should prompt the user for the following:

- a board size (square) from 3x3 to 7x7. (8x8's and above take too long using brute force)
- a starting position (row,col; 1,1 is the upper left corner)

Sample Output

It should show all the solutions. A sample run would look something like this:

```
Welcome to the Knights Tour solver!

Board size (3-7): 5
Starting position (row, col): 5,1

Thinking....

Solution #1:
23 10 19 14 25
18 05 24 09 20
11 22 13 04 15
06 17 02 21 08
01 12 07 16 03

...
```

Details

You can define a fixed size array and just use the portion you want. It should be initialized with a number that is not a valid move (like 0 or 99);

Each instance of the recursive function call needs it's own copy of the board-to-date. Since arrays are always passed by reference, and we need to pass by value (to get our own copy), the simplest solution is to define the board array inside a class (or structure) because classes can be passed by value.

Assuming you defined a board struct or class (with all public members) which contains a two-dimensional array representing the board and an integer representing the board size, the recursive function description and prototype will probably look like these:

```
//=============================================================
// Description:
//    Recursively solves the knights tour using brute force.
//    Prints any solutions if finds.
// Args:
//   board (I/O) - the board we're working with
//          (board with previous moves and size)
//   row (I) - the row we're going to attempt to place the knight on this
move.
//   col (I) - the column we're going to attempt place the knight on this
move.
//   currentMoveNumber (I) - the move we're making
//    (1=first placement, 16=last placement on 4x4 board)
//     Note: row and col may be invalid (<0 or >=boardsize)
//          or already taken (<currentMoveNum)
//  Returns:
//    The number of solutions the given board and move leads to
//=============================================================
int solveKnightsTour(Board board, int row, int col, int currentMoveNum=1);
```

Very Important Stuff

Program should be well written and function properly. The best one(s) gets a cookie. (note: image is for illustrative purposes only; actual cookie may look different and be considerably smaller).

All programs should follow the class's Coding Conventions
Submit the following:

- Your .cpp file
- The executable