```
/*****************************************************************************************************
*****************
Programmed by: Luis Barquero
Purpose: Program will read in a text file containing an unsolved Sudoku problem and it will
recursively solve it.
 * RecursiveSolver.cpp
 *
 * This is the recursive solver for the SUDOKU game

 *****************************************************************************************************
 *****************
 */

#include "RecursiveSolver.h"
#include <iostream>
#include <time.h>
#include <omp.h>

using namespace std;

RecursiveSolver::RecursiveSolver() {
}

RecursiveSolver::~RecursiveSolver() {
}

/*
 * Solve the game using recursive function
 */
void RecursiveSolver::solve(int count) {
    // Write your own recursive solver here!
    int tn = 2; //the number of threads
    omp_set_num_threads(tn); //you can also set the number of threads here besides the
    environment variable
    int depth;
    if(count == grid.get_slot_number())
    {
        solved = true;
        grid.print_grid();
    }

    else
    {
        Slot s;
        //if(count < 10)
        //{
            #pragma omp parallel for default(shared)
            for(int i = 1; i <= 9; i++)
            {
                s = grid.get_empty_slot(count);
                {
                    if(!solved)
                    {
                        grid.set_digit(s.row, s.column, i);
                        if(check(s))
                        {
                            solve(count + 1);
                        }
                    }
                }
            }
        //}
        if(!solved)
            grid.set_digit(s.row, s.column, 0);
    }
```

```
61  }
62
```