Jude Balasingam
100504990

LAB 3 Sudoku Solution Validator Report

Files:
sudoku.c
sudoku.h
sudoku_check.c
sudoku_check.h
Makefile
puzzle.txt

How to run:
make –f Makefile
./sudoku.exe

sudoku.c

First a puzzle is read into a 2 dimensional array from a text file. The 2d array is of a structure type which contains an int variable **value** and an array of possible values, **pos_val**. The array **pos_val** would only be set if **value** for that position is empty (equal to 0). If this is the case, then **pos_val** is set to {45,1,2,3,4,5,6,7,8,9}. The first value of **pos_val** is the sum of the possible values for that position on the puzzle. Next the program will go there each position on the puzzle and remove the digits from **pos_val** based on if they already exist in the same row, column, and sub-grid. Each position will have 3 threads doing this at a time. Threads will be run for 1 whole row at a time, so the maximum threads that will be running at a single time would be 27 (3 threads x 9 cols/row). If the value at a position is already filled, in, no threads would be created for that position. Eventually **pos_val** for each location will only have 1 value and that gets stored in **value**. Once we have the solution, the program outputs the solution to a text file.

sudoku_check.c

This is used to check that the solution is valid. 27 threads may run at single time: 9 for each row, 9 for each column, and 9 for each sub-grid. If a value is empty (equal to 0), it will output a message saying which location in the 2d array. When a value exists twice in the same row, column, or sub-grid, a message will be output to state this too. If the solution is correct, then only a success message will be output.