Project 3 Write-Up:

Design & Implementation:

The Sudoku solution is checked using 11 threads – 1 to check all the columns, 1 to check all the rows and 9 to check each of the sub-grids.

To check for columns, an array of ints with 9 slots is declared and initialized to all 0 to keep track of values in the grid for each column. As the algorithm iterates through each column, the appropriate slot for the number in a specific grid position in the column is updated (e.g., if the value at a grid position is 1, then index 0 in the array of ints is incremented by one). Once the column have been iterated, the array keeping track of the number of occurrences of digits 1 through 9 is iterated and each index is checked for missing values. If the value at a specific index is 0, then then a message is outputted to the command that that value is missing from the column that it is in.

A similar procedure is used to check for rows.

Finally, to check for each sub-grid, a struct list is created that contains fields for row and column. This struct is then given an alias parameters. For each of the 9 threads that are created, the starting row and column are passed into the function as a pointer using the above struct. Then for each sub-grid, an int array with 9 slots is declared and initialized to all 0 as the check for columns and rows. However, rather than iterating an entire row or column, the 3x3 sub-grid specified by the column and row passed into the thread is examined instead. Following the iteration of the sub-grid, the array is checked similar to the check for columns and rows, and the appropriate message is outputted to the terminal if a value is missing.

At any point, if a value is missing, a global variable tracking the validity of the sudoku solution is update, and before the program terminates, this variable is checked and depending on the value, a message indicating whether the solution is valid or invalid is ouputted to the terminal.