The program begins by opening a text file that is hard coded into the program. The text file contains a completed Sudoku board that needs to be checked for verification. As the program reads each number on the board, it inputs the numbers into a 2d array to be used throughout the program. This is the composition of the main thread. In total I ended up using 3 threads, other than the main thread, because I believed that if I could get the same work done with less threads, then I could save on how many resources I’m using.

The next thread is what controls the function for checking if a certain row is valid. It goes number by number, compares it to a previous number, and increments a counter for that number. If at any time that counter reaches 2, then it is known that the current row is invalid because there is a duplicate number.

The thread for the column checks is almost identical to the row checking thread. The thread for the square checking is a little more complicated. It starts by going to the first number of the first sub-square, begins comparing it to every other number in the sub-square and it also increments a counter for that number. However, once it has verified that none of the 8 other numbers match the first, it moves on to the second number and re-does the process over again. This happens for each of the 9 sub-squares.