**CS 406 Homework 2**

**Veli Can Erdem**

The code initializes colors\_global array first, an array denoting the color of a vertex, to zero which is a null color. There is also a global changedArr array, which will be mentioned.

In the parallel region, forbidden\_color array is initialized for the greedy coloring approach (it's size is 10,000, which seems arbitrary but it is enough since it is a function of maximum degree of a vertex, which is lower). Then for each vertex, colors are assigned parallel in the for loop.

Next, we check whether a color conflict has occurred in a loop, due to parallelly assigned colors. Every thread checks whether there has been a color conflict, then if there was, the changedArr[threadID] becomes true. Then an omp single construct is used for checking whether the loop should continue to run for all threads, checking if changedArr[threadID] has a *true* boolean.

The program was able to get x1.3 speedup for 2 threads and 4 threads and 2.3x speedup for 8 threads.