CSC 4500, Spring 2019 **Homework #2 (HW2)** **Circles and Arrows, Part 2**

In HW1, you implemented a game we're calling "Circles and Arrows." In HW2, you will use your HW1 program, and add to it several new twists. Before working on HW2, review the details of the HW1 specifications. Those specifications apply to HW2, unless the HW2 specification explicitly says otherwise.

In HW1, you were to assume that the data in the file HW2infile.txt defined a circles and arrows system that was a strongly connected digraph. That is, there was a path of arrows between any two circles. In HW2, you can't assume that characteristic of the input circles and arrows. You must TEST that assumption. If the system of circles and arrows IS strongly connected, then your program should continue processing; but if the system is NOT connected, then your program should write an error message to the screen and to the output file that describes the problem, and then halt.

As before, make sure that the input file conforms to the specified format. If the input file deviates, print an error message to the screen and to the output file, and halt. In HW2, you may assume that the number of out arrows at any circle will not exceed 20.

The name of the output file for HW2 should be HW2lastnameOutfile.txt where lastname is your lastname.

In HW1, you simulated one game of circles and arrows. Assume that the input file correctly defines a connected system of circles and arrows. Although you will read in the input file information once, then you'll run that simulated game ten (10) times. Before each simulated game, reinitialize to the start of a game. Keep statistics on the 10 game simulations, and then print out to the screen and to the output file the following information:

- The average number of total checks per game
- The maximum number of total checks in a single game
- The minimum number of total checks in a single game

- The average number of checks on a single circle over all the games
- The minimum number of single circle checks
- The maximum number of single circle checks

In HW1, the maximum number of circles was 10. Increase that to 20.

It wasn't mentioned in the HW1 specification, but it is at least theoretically possible for the game to go on for a very long time because arrows are selected randomly. In order to avoid waiting too long for a game to continue, impose a limit of a million checks on any one game. (Think about this: is that a reasonable number?) If that limit is reached, stop the program with a sensible output that helps the user know what happened.

Be sure to revise and expand your opening comments and your internal comments to document the changed specifications. Be sure to test your program with different input files, both correct and incorrect input files.