Computer Science 1 - Program 4
KnightsRegistrar (Stacks & Queues) – Grading Criteria
Total: 100 Points

Late Penalty: 10% for up to 24 hours late, and 25% for up to 48 hours late.

Implementation Restrictions (25 pts)

Read from and then print to a file (5 pts)
Structs to support queues and stacks are declared (5 pts)
Appropriate functions to operate on the queue are declared and work properly (5 pts)
Appropriate functions to operate on the stack are declared and work properly (5 pts)
At least one stack and three queues were used in the program (5 pts)

Execution Points (60 pts)

10 points for not crashing. (No matter how incomplete the program is, if it doesn't crash it earns these 10 points. If it does crash, no matter how complete the program is it loses these 10 points.)

Remaining 50 points:
The test file is very large. You will NEED to use Winmerge (a file comparison tool) for this program. Open two files in Winmerge: the correct output and the student's output. If Winmerge says the two files are identical, the student gets the full 50 points. Otherwise, you need to determine how many lines are wrong. Once Winmerge opens two files, and assuming they are NOT identical, Winmerge shows the # of lines that are different in the lower right corner. Additionally, there are 20,159 lines in the correct output (this includes three blank lines at the end). So if the # of lines different is 2197, divide that number by 20159 to get the % of lines different. 2197/20159 = 10.9% different, or 89.1% the same. So award 89.1% of the 50 remaining execution points, which equates to 45 pts (round UP).

Note:
Unfortunately, this is NOT the best way to grade this program, because one small mistake in the output can trickle down and mess up all the rest. Here's what you need to do:

IF the program runs and IF the file size of the students output is remotely similar to the file size of the correct output (1081 KB), then their program is semi-working, but it is doing things out of order. We don't want them to lose 30 of the 40 points just because the output was sooo far off. In this case, look at the code, IN DETAIL (maybe taking 30 minutes for one student), and try to determine how well you think they did. Study the ORDER that they processed things and award points BACK based on how you deem appropriate. So if they got 80% of the output wrong, resulting in only 10 of the 50 points, you may so choose, for example, to award back 30 additional points (or more)

based on what you see in the code, resulting in 40 of the 50 of the remaining points, or a total of 50/60 execution points.

General Rule: BE NICE in giving Execution points!

If the program doesn't compile, do the following:

Try to fix it for 5 minutes. If you can get it to compile, take off points for the errors (you decide how many) and then grade the running program. If you can't, award at most 20 of the execution points by looking at the code.

Style Points (15 pts)

Header comment w/name, program, date – 4 pts
Appropriate variable names – 2 pts
Appropriate use of white space – 2 pts
Appropriate indenting – 2 pts
Comments in code – 5 pts

NOTES:

1. **If the program does not use stacks and queues, they get NO credit.**

2. If the program does not compile, spend, at most, five minutes to see if you can fix the complication errors. If you can fix them quickly, grade using the above criteria and then deduct 30 points from the grade. If you cannot fix the compilation errors within five minutes, award a maximum of 50 out of a 100, but adjust the score based on what you can tell was done in the code.

3. If the program does compile but crashes, award at most 70 points based on what you see in the code.