Computer Science 1 - Program 3
KnightsRecurse (Recursion) – Grading Criteria
Assigned: 2/8/12
Due: 2/22/12 (Wednesday) by 11:55 pm (WebCourses time)

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

Implementation Restrictions and code points (15 pts)

- Uses WRAPPER functions as described in the writeup. (10 pts…2 pts each)
- Reads from input file and prints to output file (5 pts)

Execution Points (75 pts)

10 pts 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.)

There were five recursive problems, and each needs to be graded separately. **Note**: If one of the recursive functions fails, briefly check actual code and award partial credit as deemed appropriate.

The remaining 65 execution points are awarded as follows:

KnightsMultiply (10 pts)
10 test cases – each is worth 1 pt

KnightsFlip (10 pts)
2 test cases – 5 pts each

KnightsShape (15 pts)
5 test cases – 3 pts each

KnightsScotch (20 pts)
10 test cases in input file – each is worth 2 pts

KnightsDepot (10 pts)
10 test cases in input file – each is worth 1 pt

If a program crashes, resulting in no output, try to give it an input file with ONLY the KnightsMultiply. If that function runs and outputs correctly, awards points for KnightsMultiply. Then continue in this fashion, making the input file have only the KnightsFlip test cases, and so on. It may be the case that three or four of the functions work, but the program could crash simply due to one of them, resulting in no output altogether; so you may need to check the output of each function individually.

Style Points (10 pts)

- Header comment w/name, program, date – 2 pts
- Appropriate variable names – 1 pts
- Appropriate use of white space – 2 pts
- Appropriate indenting – 2 pts
- Comments in code – 3 pts

NOTES:

1. CHECK each of the FIVE actual functions; **make sure that recursion was used.** If recursion was NOT used to solve one of the functions, NO CREDIT should be given for the execution pts of that function. So if a program worked flawlessly and all functions were recursive with the exception of KnightsShape, for example, the student would lose ALL of the 15 points for KnightsShape.

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 crashes, award at most 70 points based on what you see in the code.