CS 201 - Data Structures and Discrete Mathematics I – Spring 2004
Programming Assignment 1: Recursion

General information:

Deadline: 11:59pm Mar 31, 2004

The purpose of this homework is to practice Recursion. You are required to use JAVA to code; no other programming languages are acceptable. This homework is not a group project and everybody should work on it individually. We will run all the programs using MOSS, which will give us an indication whether any two or more programs are too similar to have been written independently.

Problems

1. (20 points) Write a java class named **my_WordReverse** using recursion. It takes a sentence and returns the sentence in reverse order. For example, given "welcome to recursion world" as input, your program should print out "world recursion to welcome". Note that space is the only separator.

2. (30 points) Write a java class named **my_makeChange** that reads an amount in cents and calculates the number of unique ways of making change (with coins) for that amount and print them out. You must check the input amount is positive. If not, print an error and stop. Here is a sample format:

Enter amount of change desired (in cents): 10
Amount of change desired is 10 cents.

Number	Halves	Quarters	Dimes	Nickels	Pennies
========	========	==========	=======	=========	=========
1	0	0	1	0	0
2	0	0	0	2	0
3	0	0	0	1	5
4	0	0	0	0	10

Instructions

1. Your programs should run like:
    Problem 1:
        Enter string:
        welcome to recursion world
        Reverse string:
        world recursion to welcome

    Problem 2:
        Enter amount of change desired (in cents):
        10
        Amount of change desired is 10 cents.

Number	Halves	Quarters	Dimes	Nickels	Pennies
========	========	==========	=======	=========	=========
1	0	0	1	0	0
2	0	0	0	2	0
3	0	0	0	1	5
4	0	0	0	0	10

2. **What to turn in**: You should turn in two java files, one is named **my_WordReverse.java**, and the other is named **my_makeChange.java**. You should write main function within each java file. So that after compiling them, we can run it directly.

3. **How to turn in**: login to bert.cs.uic.edu, get into your working directory and run **turnin**.

      "turnin -c cs201 -p project1 my_WordReverse.java my_makeChange.java "

  You may run **turnin** as many times as you want (only the last one is kept).

  You can use "turnin -c cs201 -p project1 –v" to check whether you have turned in successfully.
For more help, you can type

      turnin -h

or

      man turnin

4. You MUST make sure that your program can compile using "javac" and run using "java" on bert.cs.uic.edu.

      javac **my_WordReverse.java**

  If no compiling error, you will see my_WordReverse.class created in your directory. Use

      java my_WordReverse

  to run your program.

5. If you use some IDE, like Jbuilder, you must make sure that your code can be compiled and run in bert without any additional packages.

6. Remember to comment your code.

7. Zero mark will be given if your program does not compile, your program gets into an infinite loop (does not terminate) or you did not turn in a program.

8. You MUST use recursion for both problems. Zero mark will be given for any question that does not use recursion even if your program for the question works perfectly fine.