CISC 235 Assignment #1

Instructions

This assignment is due at 4:30 p.m. on Monday, January 31st. Please email your submission to the Teaching Assistant assigned to your lab section (Lab A (Monday): 8jjl1@queensu.ca, Lab B (Friday): zi@acm.org).

Do not compress the files before attaching them to your email (please include each of the files as a separate attachment to the email).

The labs provide an opportunity for you to work on assignments with the supervision of our Teaching Assistants.

Part 1: BigInteger

Read the following web page:
http://www.leepoint.net/notes-java/data/numbers/60factorial.html

Copy and paste the factorial example into your programming environment, then run it.

You will use BigInteger to complete Part 2 of this assignment. More information on BigInteger is available here: http://java.sun.com/j2se/1.4.2/docs/api/java/math/BigInteger.html.

Part 2 (5 Marks): Recursive GCD using BigInteger

Write a recursive Java method to calculate the greatest common divisor (GCD) (http://en.wikipedia.org/wiki/Greatest_common_divisor) of two numeric values.

Calculate and display the GCD for 9384041234832094823094898 2306 and 3033092482498243802 4988.

Save this program to a file named “BigIntegerGCD.java”.

Part 3 (15 Marks): Recursive GCD using Linked Lists

Modify Part 2 to store numbers as linked lists instead of using BigInteger. Each digit in a number should be stored in a different node in a linked list. You can use a singly-linked list or a doubly-linked list. You must create a class to implement linked lists, similar to the one in the slides for week 2.

This class will need to have several methods. It will need a method for displaying the contents of a linked list as a number. It will also need at least one method that accepts two linked lists and returns another linked list that contains the result of a math operation. **Clarification: Do not use BigInteger.**

Save this program to a file named “LinkedListGCD.java”. If this program includes multiple files, please provide any necessary instructions in your submission email.

REVISION: As an alternative to implementing GCD, you will receive full marks if you implement a method that simply multiplies the two values, stored in the linked lists, and displays the result (no recursion required). However, students who implement GCD will also receive 1% of their participation grade for their submission :)

Part 4 (5 Marks): Efficiency Analysis

Write a short answer (bullet points are fine) to each of the following questions. **Please include three examples in each of your answers.**

A. When might it make sense to store a number in a linked list?

B. Why might it not make sense to store a number in a linked list?

C. How could your Part 3 submission be modified to reduce its running time?

Save this description to a file named “EfficiencyAnalysis.doc”.