CS608 Programming Assignment 3

This assignment has two parts: Part 3A and Part 3B. If you successfully complete both, you will receive 15 points. If you successfully complete only one (either one), you will receive 10 points.

Programming Assignment 3A: Linked List

The programming assignment is to add two more methods to our implementation of linked list (see my notes): getLastElement() and searchFor(x). The method returns (an integer) the last element of the linked list. The method searchFor(x) searches the linked list for the specified element x and returns true it is found, otherwise returns false.

Now the programming assignment 3A:

Starting from the program in my notes, which implements a linked list of, add two more methods: getLastElement() and searchFor(x).

A text file called **dataForLinkedList.txt** contains the following:
First line: One line of integers ending in a zero.
Second line: an integer (call it x)
Third line: an integer (call it y)
Fourth line: an integer (call it z)

Here is a sample data file (actual file will be different):

```
37  82  19  15  92 33 23 56 78 34 23 90 55 24 53 75 86 35 89 0
50
66
75
```

Your program reads this text file with numbers and does the following:

1. Creates a linked list with numbers from the first line. The zero at the end of the line is given to help you stop reading. Don't include 0 in the linked list.
2. Display the linked list elements
3. Display the size of the linked list

4. Searches for x (the number on the second line) and prints if the number is found in the linked list or not.
5. Print the last element of the linked list
6. Insert integer y (third line) in the beginning of the list
7. Display the linked list again
8. Insert integer z (fourth line) at the end of the linked list
9. Display the linked list elements again

All output must contain appropriate comments.

Programming Assignment 3B: Generating large factorials

The assignment 3B is to write a Java program to compute the value of n!
The program will query the user for an integer, n, and return the exact value of that integer's factorial, n!. It will then ask whether the user wishes the value of another factorial. If so, the query, return, and ask cycle is repeated.

As you know, n! is the product 1 * 2 * 3 * .... * n.
For small n, it is easy to compute n!
1! = 1
2! = 1*2 = 2
3! = 1*2*3 = 6

It is important to observe that n! = (n-1)! * n
So 4! = 3! * 4 = 24
5! = 4! * 5 = 120
6! = 5! * 6 = 720
And so on.

Factorials grow very fast. The calculator fails to display even 20! exactly. Your program must be able to display n! exactly for reasonably large n.

In this week's folder on Black Board, I have provided a small paper called, *Multiple Precision Arithmetic: A Programming Assignment in CS2 Applying Linked Lists.* (It is available if you search on the Web.) Read this paper, which helps you to get started with program. The paper explains the algorithm in general terms (no Java).

Input: Ask the user for the value of n
Output: Display the exact value of n!

General instructions:

- If your program has several classes, include all of them in the same file and name your Java file
  CS6083Axxxxx.java (Assignment 3A) and CS6083Bxxxxx.java (assignment 3B), where xxxxx is your last name. **Example**: If your name is John Smith, name the file CS6083Asmith.java and CS6083Bsmith.java. **DO NOT SEND ZIP files.**

- Output must include: **Your name, course number and date (use Date class).** If any of the above items are missing, you will not receive full credit.

- Send your Java file as email attachment to CS608Assignment@gmail.com. Include your name and assignment number in the email subject.

Note: I will run your programs and grade them. If your programs do not compile (that is, show syntax errors, you will receive 0 for the programming assignment).