**CS 2401 Assignment #6**

**Due Date:** Sunday, March 26, 11:59PM (See the syllabus for late policy)

**Objective:** The goal of this assignment is to practice recursion.

**Assignment:** This assignment is an extension of the previous lab assignment. The assignment requires writing recursive methods for some linked list operations. **The use of loops is strictly prohibited in this assignment.** That is, you cannot use for, while, and do-while in the recursive methods you will write. The only time you can use a loop is when you initiate values for a linked list.

Consider that you are given he following linked list node.

```
public class MyListOfInts {
  public int firstInt;
  public MyListOfInts restOfTheInts;

  public MyListOfInts(int f){
    firstInt=f;
  }

  public MyListOfInts(int f, MyListOfInts r){
    firstInt=f;
    restOfTheInts=r;
  }

}
```

Clearly, this class can be used to create a linked list. Write an additional class named `ListOperations` that will contain the following methods. Notice that you have to write only a few additional new methods (marked in blue) to the previous Lab assignment (Lab 5).

1. Write a recursive method named `printMyList(MyListOfInts m)` to print all the integers in the linked list `M`. Notice that `M` is the head of the linked list.
2. Write a recursive method named `sumOfMyList(MyListOfInts m)` that will sum up all the integers in the linked list `M` and return the summation value.
3. Write a recursive method named `maxOfMyList(MyListOfInts m)` that will return the largest number in a linked list.
4. Write a recursive method named `lengthOfMyList(MyListOfInts m)` that will compute and return the length of a given linked list `M`.
5. Write a recursive method named `reverseMyList (MyListOfInts m)` to reverse a linked list. Return the head of the reversed linked list.
6. **New operation:**
   Write a recursive method `removeANumberFromMyList(MyListOfInts m, int removee)` to remove the first occurrence of a specific number

`removee` from a linked list `M`. Return a new head. You must make sure that the linked list sent as a parameter during the first call remains intact after returning the new linked list.

7. **New operation:**
   Write a recursive method `insertANumberIntoMyList(MyListOfInts m, int insertee, int position)` to insert a number (`insertee`) in a specific `position` of a linked list. Positions start from 0 (that is, the position of the head of a linked list is 0).
8. **New operation:**
   Write a recursive method `reportOnValues(MyListOfInts m)` that will use each integer value in a linked list `M` to compute a recursive formula implemented in a method `specialRecursiveFunction(int x)`. The recursive function that `specialRecursiveFunction` implements is as follows.

$$f(x) = \begin{cases} 1 & x = 0 \\ 1 + f\left(\frac{x}{2}\right) & x \text{ is even} \\ 1 + f(x-1) & x \text{ is odd} \end{cases}$$

   where $x$ is a number in one node of a linked list.

A template for the `ListOperations` program is given below. You need to write inside the body of the methods. Modify the main method for testing purpose, as necessary.

```
public class ListOperations {

  public static void main(String[] args){
    MyListOfInts t=null;
    for (int i=0; i<5;i++){
      //int ran = (int) (100.0* Math.random());
      int ran = i+2;
      t=new MyListOfInts(ran, t);
    }

    System.out.println("All numbers in the list:");
    printMyList(t);
    System.out.println();

    System.out.println("Sum="+sumOfMyList(t));
    System.out.println("Max="+maxOfMyList(t));
    System.out.println("Length="+lengthOfMyList(t));

    t=reverseMyList(t);
    System.out.println("All numbers in the reversed list:");
    printMyList(t);
    System.out.println();
```

```
    System.out.println("Remove 3");
    MyListOfInts tt=removeANumberFromMyList(t, 3);
    System.out.println("All numbers in the new list:");
    printMyList(tt);
    System.out.println();

    System.out.println("All numbers in the previous list:");
    printMyList(t);
    System.out.println();

    System.out.println(
       "Insert a number in a position of the new list:");
    tt=insertANumberIntoMyList(tt, 20, 1);
    printMyList(tt);
    System.out.println();

    System.out.println(
      "Values obtained for the special recursive method:");
    reportOnValues(tt);

}


/* Write a recursive method to print all the numbers separated by spaces.
This method cannot contain any loop (that is, uses of for, while, do
while are prohibited).
*/
public static void printMyList(MyListOfInts m){

     /* Write your code here */
}

/* Write a recursive method to retrieve the sum of all the numbers in a list.
This method cannot contain any loop (that is, uses of for, while, do while
are prohibited).
*/
public static int sumOfMyList (MyListOfInts m){

     /* Write your code here */

}

/* Write a recursive method to retrieve the largest number from the list.
Assume that there is at least one number in the given list when the method
is called from the main function. This method cannot contain any loop (that
is, uses of for, while, do while are prohibited).
*/
public static int maxOfMyList (MyListOfInts m){

     /* Write your code here */
}

/* Write a recursive method to compute the length of a list.
This method cannot contain any loop (that is, uses of for, while, do while
are prohibited).
*/
public static int lengthOfMyList (MyListOfInts m){
     /* Write your code here */
}
```

```
    /* Write a recursive method named reverseMyList that will reverse a given
    linked list m. Return the head of the reversed linked list. It is fine
    if you need to modify the given linked list to obtain the reversed one. This
    method cannot contain any loop
    */
    public static MyListOfInts reverseMyList (MyListOfInts m){
         /* Write your code here */
    }


    /* Write a recursive method to remove the first occurrence of a specific
    number from a list. As an example, if your list initially contains
    20 5 10 3 5 1 and your removee is 5, the returned list should contain
    20 10 3 5 1 after the removal. Return a new head. You must make sure that
    the parameter list remains intact after returning the new list to the main
    method. This method cannot contain any loop.
    */
    public static MyListOfInts removeANumberFromMyList(
               MyListOfInts m, int removee){

         /* Write your code here */
    }

    /* Write a recursive method to insert a number (insertee) into a specific
    position of a list. Positions start from 0 (that is, the position of
    the head of a list is 0). This method cannot contain any loop (that is,
    uses of for, while, do-while are prohibited).
    */
    public static MyListOfInts insertANumberIntoMyList(MyListOfInts m,
            int insertee, int position){

         /* Write your code here */

    }

    /* Write a recursive method named reportOnValues that will use each value in
    a list to compute a recursive formula implemented in the method
    specialRecursiveFunction. This method cannot contain any loop.
    */

    public static void reportOnValues(MyListOfInts m){

         /* Write your code here. This method must call specialRecursiveFunction
          */
    }

    /*This method cannot contain any loop */
    public static double specialRecursiveFunction(int x){

         /* Write your code here */
    }

}
```

The terminal-output of the program above is as follows.

```
All numbers in the list:
6 5 4 3 2
Sum=20
Max=6
Length=5
All numbers in the reversed list:
2 3 4 5 6
Remove 3
All numbers in the new list:
2 4 5 6
All numbers in the previous list:
2 3 4 5 6
Insert a number in a position of the new list:
2 20 4 5 6
Values obtained for the special recursive method:
3.0
7.0
4.0
5.0
5.0
```

**Deliverables:** You are expected to submit two Java files (`MyListOfInts.java` and `ListOperations.java`) via Blackboard. You have to demo your programs within one week after the due date. Your demo will be based on your last submission in the Blackboard. Your TA will instruct you with further details.

**Please do not submit your Lab 5 assignment as your Lab 6 submission. Attempts with no effort to complete the new operations will receive zero credit in Lab 6.**