(200 pts) due: November 23, 2013 11:59pm

Important Notes

- **This assignment is to be done on your own.** If you need help, see the instructor or TA.
- Please start the assignment as soon as possible and get your questions answered early.
- Read through this specification completely before you start.
- Some aspects of this specification are subject to change, in response to issues detected by students or the course staff.

1 Description

The goal of this assignment is to work on linked lists problems.

1.1 Part I (200pts)

Given a singly-linked list defined as below:

```
/**
 * Definition for singly-linked list.
 * public class ListNode {
 *     int val;
 *     ListNode next;
 *     ListNode(int x) {
 *         val = x;
 *         next = null;
 *     }
 * }
 */
```

you need to implement the following methods without changing the definition of ListNode:

1. Given a linked list and a value x, partition it such that all nodes less than x come before nodes greater than or equal to x. You should preserve the original relative order of the nodes in each of the two partitions. For example,

   Given $1 \to 4 \to 3 \to 2 \to 5 \to 2$ and $x = 3$, return $1 \to 2 \to 2 \to 4 \to 3 \to 5$.

2. Given a list, rotate the list to the right by k places, where k is non-negative.

   For example:

   Given $1 \to 2 \to 3 \to 4 \to 5 \to NULL$ and $k = 2$, return $4 \to 5 \to 1 \to 2 \to 3 \to NULL$.

3. Given a linked list, swap every two adjacent nodes and return its head.

   For example, Given $1 \to 2 \to 3 \to 4$, you should return the list as $2 \to 1 \to 4 \to 3$.

   Your algorithm should use only constant space. You may not modify the values in the list, only nodes itself can be changed.

4. Given a linked list, remove the n^{th} node from the end of list and return its head.

   For example,

   Given linked list: $1 \to 2 \to 3 \to 4 \to 5$, and $n = 2$.

   After removing the third node from the end, the linked list becomes $1 \to 2 \to 4 \to 5$.

   You can assume that n is always valid. Bonus points if (a) you can do this in one pass, and/or (b) you can give more than one solution.

The interface of the required methods are defined as below:

```
public class SinglyLinkListUtil {
   ...
   public ListNode partition(ListNode head, int x) {
      ...
   }

   public ListNode rotateRight(ListNode head, int n) {
      ...
   }

   public ListNode swapPairs(ListNode head) {
      ...
   }

   public ListNode removeNthFromEnd(ListNode head, int n) {
      ...
   }
}
```

1.2 Part II: Extra Points

- You will receive extra points if you provide more than one solution to any problem(s) in Part I.
- A *singly linked matrix* is two-dimensional linked list using a singly linked approach. Each link (except those on the top row and left side) is pointed to by the link directly above it and by the link on its left. You can start at the upper-left link and navigate to, say, the link on the third row and fifth column by following the pointers down two rows and right four columns. You need to implement constructor given the number of rows and the number of columns and a method that sets a value to a given cell.

  The interface of the required methods are defined as below:

```
public class SinglyLinkedMatrix {
  private Node head = null;
  private int row = 0;
  private int col = 0;

  private class Node {
    public int data;
    public Node nextCol;
    public Node nextRow;

    public Node(int data, Node nextCol, Node nextRow) {
      this.data = data;
      this.nextCol = nextCol;
      this.nextRow = nextRow;
    }
  }

  ...
   public SinglyLinkedMatrix(int row, int col) { //constructor
      ...
   }

   public boolean set(int row, int col, int val) {
       //returns true if a value is successfully set
       //returns false if the given row and col number is out of range.
      ...
   }

   @Override
   public String toString() {
     ...
   }
}
```

You are welcome to provide more methods such as inserting a row/column etc.

2 Submission

Provide working code for the class and method required for this assignment (50pts each problem in Part I). You are strongly encouraged to write unit tests as you develop your solution, since this can help you to check the correctness of your methods. However, you do not have to turn in any test code. Turn in a zip file named LastnameFirstname-Assignment6.zip, containing all your source code. The package name for the assignment must be edu.brynmawr.cs206.assignment6. Include the Javadoc tag @author in each class source file. **Do not turn in class files**.