CS1083 Assignment # 9 - Winter 2023

Due: Wednesday, 20 November before 4:30pm in the Desire2Learn dropbox. (See submission instructions below).

The purpose of this lab is to introduce you to working with linked lists, including using recursion.

Character Counts

The number of occurrences of each letter appearing in a statement is recorded in a file where each pair is the character and its count. For the statement "keep calm and code on" the following character and count pairs are:

```
a 2
c 2
d 2
e 3
k 1
l 1
m 1
n 2
o 2
p 1
```

Write a class called `CharList.java` which contains a definition for a linked list of characters and their count. The class should contain an inner class `CharNode` that holds information for a single node in the list (a node has a character, a count, and a reference to the next node) and contains methods to perform operations on the character data represented as a linked list. You must write the following six methods:

- `insertValue(char valueIn, int countIn):` Inserts character and count into the list in alphabetical order of the character value ('a' would be at the head of the list). If the character already exists in the list, then adds the count to the existing node.
- `charCount():` Returns the character count of all characters in the list.

- `readList(String filename):` Reads a file containing character and count pairs and inserts these pairs into a linked list. Makes use of the `insertValue()` method.
- `mergeLists(CharList other):` Merges an other list with this list and overwrites this list with the result. Makes use of the `insertValue()` method.
- `printRec1(Node front):` recursively prints the nodes in order, from first node to last.
- `printRecBackwards1(Node front):` recursively prints the nodes in reverse order, from last node to first.

Next, write a driver called `TestCharList.java` that reads two file names, opens those files, and reads in a String, character by character, and stores them in two separate `CharLists.` Print each after they are read in.

Next, merges the two lists using your `mergeLists()` method and prints the result forwards and backwards.

Two test files have been provided for you, which you can use to test your implementation. Include one other pair of files, so that your testing covers all cases, including inserting at the start and end of the list, and adding to an existing node.

The provided test case includes the files `test1a.in` ("keep calm and code on"):

```
a 2
c 2
d 2
e 3
k 1
l 1
m 1
n 2
o 2
p 1
```

and `test1b.in` ("hello world"):

```
d 1
e 1
h 1
l 3
o 2
r 1
w 1
```

Running the test driver with these files gives the following output:

```
Total character count: 27
a: 2
c: 2
d: 3
e: 4
h: 1
k: 1
l: 4
m: 1
n: 2
o: 4
p: 1
r: 1
w: 1
now backwards:
w: 1
r: 1
p: 1
o: 4
n: 2
m: 1
l: 4
k: 1
h: 1
e: 4
d: 3
c: 2
a: 2
```

Your electronic submission (submitted via Desire2Learn) will consist of two files. Name your files YourName-fileName.extension, e.g. JohnSmith-as9.zip, JohnSmith-as9.pdf:

1. A single pdf file containing a listing of the code for your program.
2. A zip file containing all your Java classes and the test files used.