CS1083
Assignment #9

Daniyal Khan
3765942

CharList.java:

```
import java.util.Scanner;
import java.io.*;

public class CharList {
    public CharNode head;

    public CharList() {
        this.head = null;
    }

    public void insertValue(char valueIn, int countIn) {
        // if list is empty create a new node
        if (head == null) {
            head = new CharNode(valueIn, countIn);
            return;
        }

        if(head.data == valueIn) {  // check if valueIn matches
head node
            head.count += countIn;
            return;
        }

        // if valueIn comes before the head alphabetically,
insert at the head
        if (valueIn < head.data) {
            CharNode newNode = new CharNode(valueIn, countIn);
            newNode.next = head;
            head = newNode;
            return;
        }

        CharNode current = head;
        while (current.next != null && current.next.data <
valueIn) {
            current = current.next;
        }

        // Check if we found a match in current.next
        if (current.next != null && current.next.data ==
valueIn) {
        current.next.count += countIn;
    } else {
        CharNode newNode = new CharNode(valueIn, countIn);
        newNode.next = current.next;
        current.next = newNode;
    }
}


public void readList(String fileName) {
    try {
        File file = new File(fileName);
        Scanner scan = new Scanner(file);

        while (scan.hasNextLine()) {
            String line = scan.nextLine();
            String[] keyVal = line.split(" ");
            char character = keyVal[0].charAt(0);
            int key = Integer.parseInt(keyVal[1]);
            insertValue(character, key);
        }
    } catch (FileNotFoundException fnfe) {
        System.out.println(fnfe.getMessage());
    }
}

public void mergeLists(CharList other) {
    CharNode current = other.head;
    while (current != null) {
        insertValue(current.data, current.count);
        current = current.next;
    }
}

public void printRec1(CharNode front) {
    if (front == null) {
        return;
    }
    System.out.println(front.data + " : " + front.count);
    printRec1(front.next);
}

public void printRecBackwards1(CharNode front) {
    if (front == null) {
            return;
        }
        printRecBackwards1(front.next);
        System.out.println(front.data + " : " + front.count);
    }

    public int charCount() {
        CharNode current = head;
        int totalCount = 0;
        while(current != null) {
            totalCount += current.count; // Add the count of the
current node
            current = current.next;
        }
        return totalCount;
    }

    private class CharNode {
        public char data;
        public int count;
        public CharNode next;

        public CharNode(char dataIn, int count) {
            this.data = dataIn;
            this.next = null; // does not point to anything
initially
            this.count = count;
        }
    }
}
```

TestCharList.java:

```
public class TestCharList {
    public static void main (String[] args) {
        // TEST 1
        CharList charList1 = new CharList();
        CharList charList2 = new CharList();

        // System.out.println("List1:");
        charList1.readList("test1a.in");
        // charList1.printRec1(charList1.head);

        // System.out.println("List2:");
        charList2.readList("test1b.in");
        // charList2.printRec1(charList2.head);

        charList1.mergeLists(charList2);

        System.out.println("Total Character Count: " +
charList1.charCount());
        charList1.printRec1(charList1.head);

        System.out.println("Merged List (Backwards):");
        charList1.printRecBackwards1(charList1.head);

        // TEST 2
        System.out.println();
        System.out.println("TEST2");
        charList1.readList("test2a.in");
        charList2.readList("test2b.in");

        System.out.println("List 1: ");
        charList1.printRec1(charList1.head);
        System.out.println("List 2: ");
        charList1.printRec1(charList2.head);
        charList1.insertValue('z', 2); // inserting at the end
        charList1.insertValue('k', 2); // inserting at existing
node
        charList2.insertValue('a', 3); // inserting at the start
of the list
        System.out.println("List 1: (After)");
        charList1.printRec1(charList1.head);
        System.out.println("List 2: (After)");
        charList1.printRec1(charList2.head);
    }
}
```