Programming Assignment 3

Problem Statement:

- You are to refine Program 3's Player List to be implemented with a **doubly-linked list** instead of an array in the class to store the player stats. The input and output file formatting requirements are unchanged from Program 2.
- Also, you must maintain the list in sorted order (alphabetical by last name, then first name). This means that every time a player is added to the list, you must locate, and place that item in the correct spot. This will require you to be able to compare two players by name. For example, these names are in sort order:
    - Russell Andre
    - Warner David
    - Maxwell Glenn
    - Buttler Jos
    - Kohli Virat
- You must provide the user with the ability to delete a player from the list by providing their firstname and lastname. Only delete the player node with an exact match. In order to test this feature, you will need to add a loop to the main program that prompts the user for the name to remove and then prompt them to see if there are more to remove.
- Write the list of players contents to the report file both before and after testing the remove operation.

Other Requirements:

- Make sure your name and the compiler version are noted in the comments at the top of your main program.
- You **may not** use any built-in or C++ library sort functions.

Helpful Information on how to compare two strings in C++:
http://www.cplusplus.com/reference/string/string/compare/

Turn In:

- Submit the electronic version of your project as a zip file to canvas for this assignment. DO NOT INCLUDE any large database file (extension .sdf) or the DEBUG folder in your zip file. The later versions of Visual create a hidden folder in your project directory named .vs, please do not allow that to be included in your zip. It is very large. Submit only the single zip file to canvas.
- The zip file name format should be firstname_lastname_Program3.
- For example, if your name is Manaswini Maddu your zip file should be named as **Manaswini_Maddu_Program3.zip**

Grading Requirements:

- Your program must be well-commented. Comment all variables, functions and remember to have a section of comments at the top of your program that includes your name, date, course section and a description of what your program does. (Internal documentation on programs in my courses counts for up to 20% of credit.)
- Always use good variable names. If I don't understand the program I cannot grade it.
- Use proper code indentation to make sure your program is easy to read and understand.
- You will receive no more than 50% credit if your program does not compile.
- If your program compiles but does not execute correctly, you will receive no more than 70% credit.
- Do not work with other students on the solution to this program. Copying code and unauthorized collaboration falls under plagiarism.

Sample Execution:

```
Welcome to the Cricket Player Stats Program.
This program reads the data from an input file. This program stores the data in a doubly
linked list, sorts the data based on first name and then writes the result which stores the
players data and stats in the output file. The input file name and output file name taken
from the user by prompting.

Enter input file name: playerinput.txt
Enter output file name: report.txt

Reading players data from: playerinput.txt
The sorted list has been written to output file: report.txt

Would you like to remove any player from your list? (yes/no): yes
Please enter the FIRST and LAST name of player: Maxwell Glenn
Maxwell Glenn removed successfully.

Would you like to remove any player from your list? (yes/no): yes
Please enter the FIRST and LAST name of player: Hardik Pandya
Hardik Pandya was not found in your list.

Would you like to remove any player from your list? (yes/no): no
The new version of the list is added to the report file.
Program 3 is completed
```

Sample Input File and Corresponding Output File:

Sample input text file:

```
Russell Andre 56 1034 794 10
Maxwell Glenn 121 3390 2689 40
Kohli Virat 256 12471 13408 54
Buttler Jos 152 4150 3429 36
Warner David 141 6007 2048 28
```

Note – There should be NO blank lines after the last line of data in an input file.

Sample output text file:

```
Player Report - - - 5 Players found in the input file

Sorted List:
PLAYER NAME                 :       Dismissals    Batting_Strike_Rate   Batting_Average
----------------------------------------------------------------------------------------
       Andre, Russell       :            46             130.23                 22.48
       David, Warner        :           113             293.31                 53.16
       Glenn, Maxwell       :            81             126.07                 41.85
       Jos, Buttler         :           116             121.03                 35.77
       Virat, Kohli         :           202              93.01                 61.74

New list after removing player:
PLAYER NAME                 :       Dismissals    Batting_Strike_Rate   Batting_Average
----------------------------------------------------------------------------------------
       Andre, Russell       :            46             130.23                 22.48
       David, Warner        :           113             293.31                 53.16
       Jos, Buttler         :           116             121.03                 35.77
       Virat, Kohli         :           202              93.01                 61.74
```

TIP:

I only tested this with one test case, removing a player from somewhere in the middle of the list. You should test other scenarios such as removing players from the ends of the list.