Programming Assignment #2

Bicycle Race Results

CS 2308.253 Spring 2020
Instructor: Jill Seaman

Due: Monday, 2/17/2020: upload electronic copy by 11:59pm!

Problem:

Write a C++ program that will allow a user to access the results from a time-trial bike race. The race consists of laps around a 6.1 mile course, and the racers try to complete as many laps as they can within 6 hours. Only complete laps count. Your program will read in the race results from a file and display the data in the format requested.

Your program will be provided with a dataset in a file containing the following information for each racer:
1) Bib number - a positive integer
2) Racer's name - a string (may include spaces)
3) Distance covered by the racer (equal to the number of laps completed times 6.1)
3) Recorded time - a string in "05:23:32" format
The number of laps is not provided and does not need to be computed.

The program should first read the results from a text file named "dataset.txt". It will contain up to 100 results. See the sample file on the class website.

Then, it should offer the user a menu with the following options:

1. Display Results sorted by bib number
2. Display Results sorted by distance, then time
3. Lookup a bib number given a name
4. Lookup a result by bib number
5. Quit the Program

The program should perform the selected operation and then re-display the menu. **Do not change the menu numbers** associated with the operations. Display an error message if the user enters an inappropriate number.

For options 1 and 2, display the results for each racer on a single, separate line. The results should include the Bib number, the name, the distance, the time, and the **average speed** in miles per hour (distance/time in hours). The values should line up in columns (use setw). Headers for the table are optional. For options 3, output the bib number with a label, and for option 4, output the results for the racer on a single line. For options 3 and 4, if the racer is not found display an appropriate message.

Additional Requirements:

- This program must be completed in a **Linux or Unix** environment, using a command line compiler like g++.

- Your program **must compile** and run, otherwise you will receive a score of 0.

- The program must be **modular** (use top-down design), with significant work done by **functions**. Each function should perform a single, well-defined task.

- Use a **partially filled array** of structures to store the results**:**
  Use a counter variable to count the number of results that are read in from the file, and use this value as the size of the array for the search and sort functions.

- Your program should work for an input file with any number of results up to 100.

- For option 2 Display Results sorted by distance, then time, if two racers have the same distance (because they completed the same number of laps) the one with the faster time value (the lesser value) should come first.

- You MUST use **binary search** for option 4, Lookup result by Bib number.

- You may use (and modify) the code from the book. See the Files tool in Canvas for these files. Please look at this code before you start implementing your program!

- I will put a sample input file on the class website (dataset.txt) and the console output from running my solution on that file (output2.txt).

- This is a long assignment. Please start early and implement one piece at a time. Make sure you can input the dataset file and display it before you do anything else.

- The tricky part is calculating the average speed. Do this last. You need to convert the time value like "5:40:52" to a precise time in total hours: 5.6811. See this calculator. Hint: I used stringstream to get the hours, minutes, and seconds from the string.

Logistics:

Name your file **assign2_xxxxx.cpp** where xxxxx is your TX State NetID (your txstate.edu email id).

Submit an **electronic copy** using the Assignments tool on the Canvas website for this class (canvas.txstate.edu).

See the assignment policy on the course website (cs.txstate.edu/~js236/cs2308) for more details, including late deadlines and penalties.