Programming Project 4: List of Dictionaries

I am told the average time for a full marathon is 280 minutes, more or less. For this program you will need a txt data file with 10 records. Each record has a runner's name, a time in minutes the person ran in the Boston marathon, a time in minutes the person ran in the Chicago marathon, and a time in minutes the person ran in the New York marathon. The items in each row are *comma-separated*. The assignment will use a menu-driven model we have often used where each option is selected by entering an integer.

The data file, *runners.txt*, is given in a link in the same folder as this assignment. Your program can add new runner data, but the new runner's name must not be the same as a runner's name that is already on the list. That it, each name must be unique. As you add runners the original runner data must survive. When you submit your program, you *must* include the data file, runners.txt, zipped in the folder with your program.

There are example programs similar to this assignment given in the lecture notes discussing dictionaries.

Use the python CSV module's csv.DictReader() approach to read and write the data to and from the file. Read the comma-separated file into a list of dictionaries when opening the file for reading, and use that list of dictionaries as the program runs. When storing the data back to the file, use the cvs.DictWriter() to store the data back as comma-separated data just like the file you read.

An image of the options you need to offer your program's user is given below. Your menu need not look identical, but should be very similar to the following.

```
Menu options.  Choose 1, 2, 3, or 4
   1. Display all data for all racers
   2. Display a runners individual race average in minutes
   3. Add a new runner and race data
   4. Save and exit


Enter your choice, 1, 2, 3, or 4:
```

For full credit, all displayed data must line up in neat columns, and you should have a main function and an additional function for each of the following tasks:

- a function to display the list of all the data (option 1): all names and their 3 race scores. You do not have to display the average in this option

- a function allowing the user to enter an existing runner's name, then calculate and display the 3-race average of that individual runner (option 2). Round the average to a whole number.
- a function to add a new name and 3 times to the collection of runners. This info should be saved when the program exits. (option 3). If the program is restarted the new data should appear along with the original data.
- a function to save and store all data back to the csv file (option 4) in the comma-separated format. use csv.DictWriter()
- optionally, a function to read the data from the file. do use the csv.DictReader() library. up to you on this one.
- also, you *must* submit the data file you received along with your program code.

For option 2 you should be able to enter a name, like Elaine, and have Elaine's three race average displayed in whole minutes. You don't need fractions of a minute..

For option 3 you should be able to enter a new name and 3 race times, and have that new record be properly saved to the data file on option 4.

A user of your program should be able to interact with the program as much as they want before deciding to save an exit.

I *strongly* recommend making a backup copy of your data file before running any code in case the data file gets corrupted. Please start this program early.

As always, let me know if you have questions.

How To Submit Your Program

Please do put *both* the program file and your data file in a folder, zip it, and attach it to an (student) email when ready or you need help. For this program do not send me a link to code in the cloud. Thanks.