CS 1510: Intro to Computing - Fall 2017
Assignment 8: Tracking the Greats of the NBA

Code Due: Tuesday, November 7, 2017, by 11:59 p.m.

The Assignment

The purpose of this assignment is to give you more practice with functions, files, and lists while using a real data set populated with career data from NBA players!

Remember, before you begin this project, review the **function commenting that you shoud use**.

Getting Started

Download **hw8_data.csv** and save it into the same directory where you will be writing your program for this assignment. The format of this file is easy to understand. Open the file by right clicking on the file and selecting "Open With" and selecting a text editor like Notepad++ or Wordpad. The first line tells you the names of all the columns.* After that, each line's data corresponds to one player's career statistics. Each field is separated by a comma.

*To understand the meanings of each of the abbreviations, look at the page: http://www.databasebasketball.com/basketball/rankings.htm?ltype=RotoPts ; you'll see the headings in the "3 Enter Points For Each Scoring Category" panel on the rihgt. Move the mouse over a cell heading and a tool tip will appear with the full title.

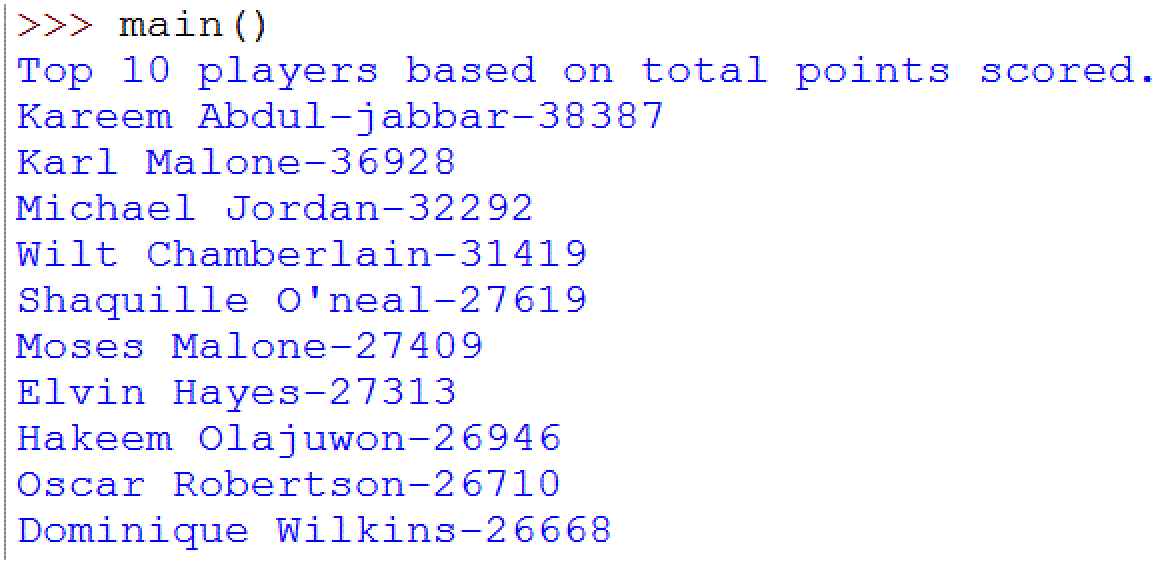
Your Assignment, Part 1

1. Begin by creating a file called **hw8.py**.

2. In this file create a function called **readData()**. This function should:
   - ***take no parameters***. It will be specialized for the file that you are using.
   - open the file "hw8_data.csv"
   - each line in this data file should be read as a list, and then each line list should be then appended to a master data list. (you should ignore any garbage/empty rows that may be at the bottom of the file.)
   - close the file "hw8_data.csv"
   - ***return*** the master data list, which is a list of line lists. (In other words, you should return a list of length 4051. Each element in the list is itself a list representing the career stats of a single player)

3. Write a function called **points()**. This function should:
   - ***take in one parameter*** - a list of player data (this will be the data you "loaded" and "cleaned" using readData())
   - For each player in the data list you should create a list consisting of 2 items: (1) the points the player earned and (2) the player's name.
   - Each of these lists should be appended to a separate list.
   - When you are done processing the input data you should ***return*** this new list of lists.

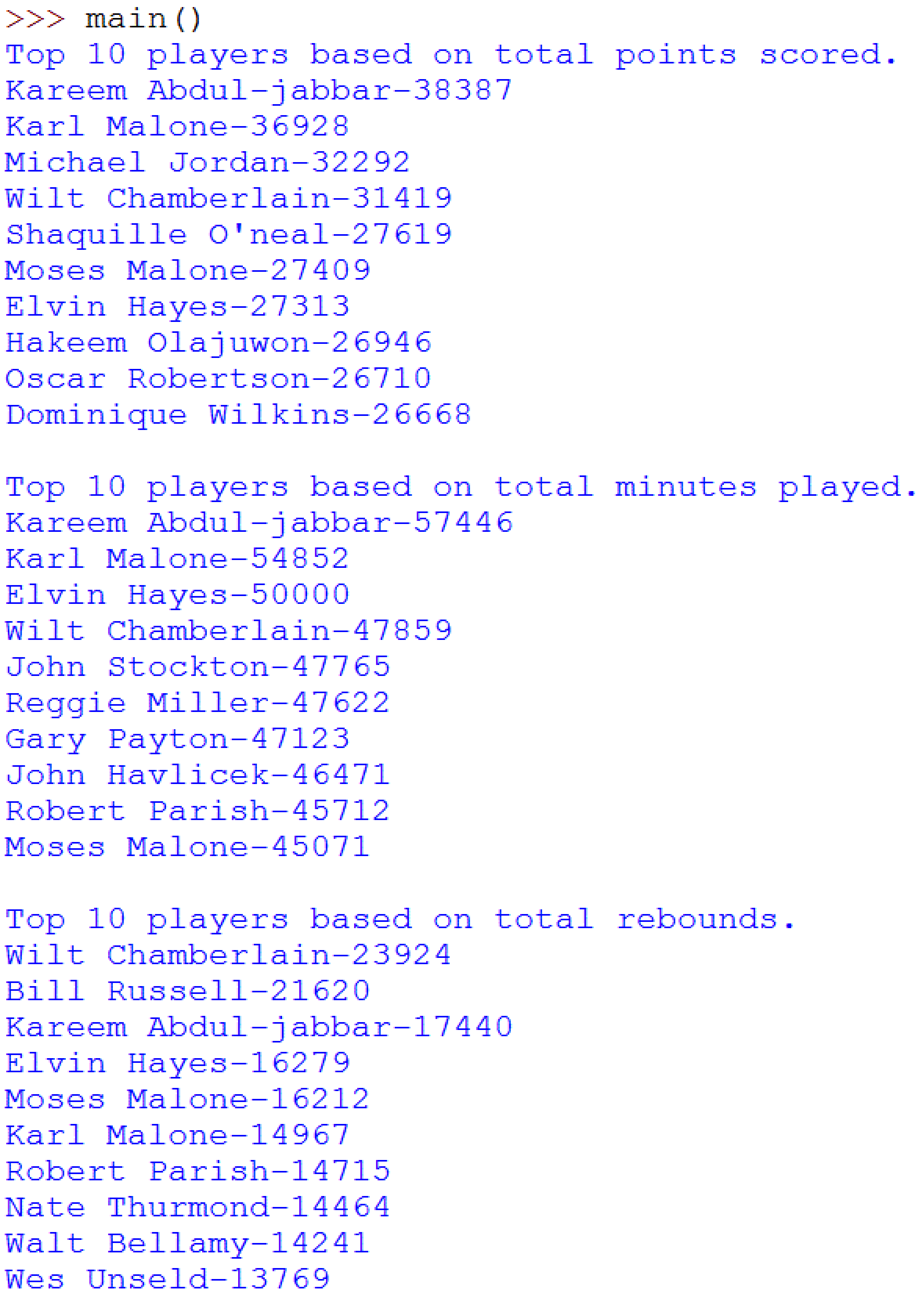
4. Write a function called **main()**. This function should:
   - ***take no parameters***
   - invoke readData() to get the data about all of the players from our stat file
   - send the list returned by readData() to points()
   - sort the list returned by points() to identify and print information about the ten players from this data file who scored the most points during their careers.



```
>>> main()
Top 10 players based on total points scored.
Kareem Abdul-jabbar-38387
Karl Malone-36928
Michael Jordan-32292
Wilt Chamberlain-31419
Shaquille O'neal-27619
Moses Malone-27409
Elvin Hayes-27313
Hakeem Olajuwon-26946
Oscar Robertson-26710
Dominique Wilkins-26668
```

Your Assignment, Part 2

1. Write additional functions called minutes() and rebounds(). These functions should work just like the points() function does. That is, they should take in the main data list, process each player in the list to make appropriate lists, and return a list of these lists.
2. You should also modify your main() method so that it prints the ten players with the largest number of **each** of these items. When I run this function I should see a long screen dump of the top 10 players in each of these three statistic categories:



```
>>> main()
Top 10 players based on total points scored.
Kareem Abdul-jabbar-38387
Karl Malone-36928
Michael Jordan-32292
Wilt Chamberlain-31419
Shaquille O'neal-27619
Moses Malone-27409
Elvin Hayes-27313
Hakeem Olajuwon-26946
Oscar Robertson-26710
Dominique Wilkins-26668

Top 10 players based on total minutes played.
Kareem Abdul-jabbar-57446
Karl Malone-54852
Elvin Hayes-50000
Wilt Chamberlain-47859
John Stockton-47765
Reggie Miller-47622
Gary Payton-47123
John Havlicek-46471
Robert Parish-45712
Moses Malone-45071

Top 10 players based on total rebounds.
Wilt Chamberlain-23924
Bill Russell-21620
Kareem Abdul-jabbar-17440
Elvin Hayes-16279
Moses Malone-16212
Karl Malone-14967
Robert Parish-14715
Nate Thurmond-14464
Walt Bellamy-14241
Wes Unseld-13769
```

Part 1 and Part 2 are enough for full credit. Part 3 will earn you an extra 10% for this assignment.

Extra Credit: Your Assignment, Part 3 - The Efficiency Statistic (up to +10%)

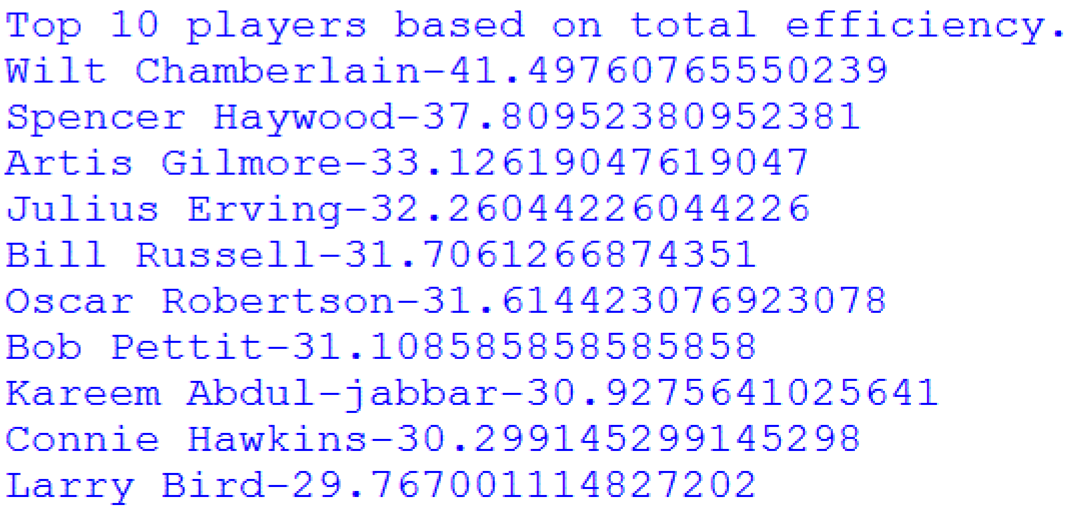
Each of the above statistics is interesting, but it only tells us how good a player is at one specific statistic. How do many NBA coaches quickly evaluate a player's overall game performance? They check his efficiency. This statistic is something like the QB passer rating we calculated earlier in the course. It is a calculation that tries to assign a number to how "well" a player played the game. Higher numbers mean a better performance from that player.

NBA.com evaluates all players based on the efficiency formula indicated below (and shown on the aboutstats.htm page). In this project, we will follow this efficiency formula. Since we are not evaluating a player based on one game, we need to divide the total efficiency by the number of games the player played. So the formula is:

$$Efficiency = \frac{(pts + reb + asts + stl + blk) - ((fga - fgm) + (fta - ftm) + turnover)}{gp}$$

The abbreviations on the right hand side of the equation correspond to the fields in the statistics file. Again, you can check out the meanings of each of the abbreviations at: http://www.databasebasketball.com/basketball/rankings.htm?ltype=RotoPts

1. Create a function called **efficiency().** This function should behave very similarly to the functions you wrote in parts 1 and 2 in that it should:
    - ***take in one parameter*** - a list of player data (this will be the data you "loaded" and "cleaned" using readData())
    - For each player in the data list you should create a list consisting of the player's career long efficiency and the player's name)
    - Each of these lists should be appended to a separate list.
    - When you are done processing the input data you should return this new list of lists.
    - Now modify main() so that it uses this as it has the prior functions.



```
Top 10 players based on total efficiency.
Wilt Chamberlain-41.4976076555O239
Spencer Haywood-37.8095238095238l
Artis Gilmore-33.126190476l9047
Julius Erving-32.2604422604422б
Bill Russell-31.706126687435l
Oscar Robertson-31.6l44230769230 78
Bob Pettit-31.1085858585858
Kareem Abdul-jabbar-30.9275641025641
Connie Hawkins-30.299145299145298
Larry Bird-29.767001114827202
```

Helpful Hints

1. Remember the split() function, which takes as an argument the character to split on, and returns a LIST of STRINGS
2. Pay attention to the type of data you are working with. Don't forget to convert strings to numbers or vice versa as needed:
3. A 2-item list would be: myListA = [x,y]
4. To append this list to a list you can just say myListB.append(myListA). Then to access the different items in the list you index into the list twice, so for example if you appended the above list as the first item in a list:
5. myListB[0][0] would return x
6. myListB[0][1] would return y

7. Since there are so many fields, do some testing (E.g. output some parsed data) to make sure that you get the correct data.
8. List’s sort function and reverse function should be useful.
   - myList = [[3,2], [1,2], [2,5]]
     myList.sort() # myList will be [[1,2], [2,5], [3,2]]
     myList.reverse() # myList will be [[3,2], [2,5], [1,2]]

Notes and requirements:

1. Use detailed commenting (as in the **rps.py** file)
2. Make sure that you save your program in the correctly named files
3. Use meaningful variable names with the proper style (use_snake_casing)
4. Every file containing python code that you submit should contain a header commen block containing three pieces of information as shown below:

```
"""
File: filename
Author: your-name
Description: one-line description of the file
"""
```

Final Submission

Submit the modified file **hw8.py** to eLearning.