Assignment P2

Due Date: March 1

Purpose
Hi-ho! In this new and funtastic assignment, you’ll have the pleasure of trying out some graphics as well as working with strings and lists. Along the way, you will also practice your basic Python statements.

Problem
Professor Gousilini is lazy. He wants to compare one student’s grades with the class averages, but he doesn’t want to do this in a spreadsheet. Instead, he hires you to write a Python program that will graph the data in a nice way.

Input
The program should prompt for two items: the student’s data and the class data. The user will type in four grades and the student’s last name all on one line, with no spaces between each portion of the data. Similarly, the user will type in four grades for the class averages for the second prompt. For example:

Enter data for student: 70908565Smith
Enter data for class: 95829971

Thus, the grades for Smith are 70, 90, 85, and 65. You may assume that each grade will be entered as two digits. The lowest grade is 00 and the highest is 99; note that 100 is not a valid score. You may also assume that the data is correct (no need to make sure the user is typing in the correct values).

Output
For the input above, the program should produce a nice graphic similar to the following:

The graphic does not need to look exactly like the one shown above, and in fact, should be spaced better. The output must include:

- A line graph representing the student’s scores.
- A line graph representing the average scores of the class.
- A bar graph representing the student’s average.
• A bar graph representing the class average.

• Appropriate labels for everything, including a legend for the colors used for the student and class.

Lastly, the numeric averages should be displayed as floating point values with one decimal place in the Python shell.

Specifics

• Because the data is all on one line and has numbers and text mixed together, you will have to read it as a string.

• Store the scores in two lists. You should not have eight separate variables!

• The graphic should be a reasonable size (not tiny, not huge), well aligned, and be as neat as possible.

• You may use the `round()` function to find the integer value of the averages for graphing purposes.

• Follow good programming practices, such as the introductory comment, as in the first project. Write a one-line comment at the start of each section of your code.

Notes

• There are quite possibly some unforeseen problems in this project! Start early and develop an algorithm (not to hand in) to solve the problem.

• Work on the solution in parts. Be sure one part works (for example, getting the input correctly) before moving on to the next.

• Your project must work in Wing IDE 101 using Python 3 and `graphics.py`.

• As before, the name of your source code file should be your first initial, last name, and project number; e.g., mgousieP2.py.

• As before, send your program as an attachment via email to mgousie@wheatoncollege.edu. Submit by 11:59:59 on the due date for the project to be on time.

• A printed version of your source code is due in the envelope on my office door (SC 1325) on March 2nd. Write/print and sign the Wheaton Honor Code Pledge on what you turn in: “I have abided by the Wheaton College Honor Code in this work.”

• Remember to save all of your work until your project is returned.

*I think there is a world market for maybe five computers.*

– Thomas Watson, IBM chairman, 1943