Lab Assignment 9, TCSS 142 Autumn 2014

Due: Thursday, Dec. 4, 2014, at the end of lab

OBJECTIVE

The objective of this assignment is to give you further practice with lists, functions, and file reading.

ASSIGNMENT SUBMISSION

While the lab instructor walks around checking lecture exercises (pre-lab) on individual basis, start working with the partner of your choosing on exercise sets 2 and 3. Once you are done with a set, check with the one of the other pairs sitting next to you regarding their progress – help each other. Then, as a pair, present your solutions to the lab instructor. Each student is to have his version of the programs/answers and be capable of presenting them for the pair. The presenter will be chosen at random by the lab instructor. There is NO set 4 since this is the last lab for the quarter and all the exercises need to be shown to the lab instructor before leaving the lab for credit. Use IDLE unless indicated otherwise.

1. **Lecture Exercises (30%)**

Show the following exercises you were to create during the lecture:

* charcounter.py
* webcounter.py
* stockCorrelation.py
* point.py and pointDriver.py

**2.** **File Reading (35%)**

1. Write a program that asks the user for the name of the txt file. The program should display the first five lines of the file's contents. If the file contains less than five lines, it should display the file's entire contents. You are to implement file reading using a while loop. Test files are provided (test2a1.txt and test2a2.txt).
2. Write a program that processes the file houseData.csv (sample input file provided). The input file contains information related to the house sales and is organized as follows: each line is a record consisting of a listing number, asking price, and a list of room sizes (in sq feet) – the number of rooms may differ between the records. You are to write a report to another csv file that summarizes this information in the following fashion: listing number, asking price, size of the house, number of rooms.

For example for the input data

KZ35979004,$100000,200,300,400

AZ35979002,$200000,200,200,300,500,300

The output should look as follows:

KZ35979004,$100000,900,3

AZ35979002,$200000,1500,5

**3. List of lists (35%)**

A list that contains lists as its elements is referred to as a two-dimensional list. Typically when we process all elements in such a list, we need to construct one loop to iterate over all elements in an outer list and a nested loop that iterates over the inner list components. You may picture such a list as consisting of rows and columns, e.g.

[ [2, 3, 4],

[5, 6, 7],

[8, 9, 0] ]

Where the iteration will be constructed as

for each row

for each column

process an element

When processing a single row or a single column, we just process a single list, so a simple loop iterating over that list suffices.

Download the program matrix.py that contains main and 5 functions that try to manipulate the 2d list (called matrix) in some fashion. The functions are not complete, however, and your job is to add correct statements to complete them – each place where the code is to be added is marked by a line (think of this exercise as fill-in-the-blanks). Do not change any existing code. Do not change function main – it contains a test case matrix you can use to test your functions. Your function implementations are to be scalable, so do not hardcode values related to the matrix size. Based on the way the main is written, your code should produce the following output:

the sum of matrix is: 45

average for row 0 is 2.0

average for row 1 is 5.0

average for row 2 is 8.0

[[1, 2, 3], [0, 0, 0], [7, 8, 9]]

[[2, 3, 4], [1, 1, 1], [8, 9, 10]]

[[0, 1, 2, 3, 4], [5, 6, 7, 8, 9], [10, 11, 12, 13, 14], [15, 16, 17, 18, 19]]

----- THE END -----