INSY 5336 001
Python Programming
Fall 2019
Homework 3 (50 points)
Due Date: November 10th 2019, 11:59 pm CST (no exceptions)

The following guidelines should be followed and will be used to grade your homework:

- All code to be implemented and submitted as a jupyter notebook (.ipynb) file.
- This is an individual homework assignment, no group submissions will be accepted. If you discuss in groups, please write your code individually and submit.
- Sample runs shown in the question should be used as a guide for implementation. However extensive testing needs to be done on your code to deal with all test cases that might possibly be executed.
- The instructions for running of each cell and the expected results should be documented in the cell preceding the code using markdown language.
- Every code segment in the jupyter notebook cells should be well documented with comments. Use # in the code to provide comments and they should explain the algorithm and what the code segment is doing.
- Error checking in your code is very important and differentiates a high quality programmer from a low quality one. Hence you should account for invalid user inputs, infinite loops, out of range results, etc. and resolve them by appropriate error messages. The homework will be graded for robustness of your code.
- Please read each assignment carefully. Note that you need to test your code with example input files. I will be using my own test input file to test your code. DO NOT hard coded file names in your program.

1. (25 points) A string is an anagram of another if the second string is simply a scrambled version of the first. Write a python program to implement the following game:
   a) Reads in a file that has words and their meanings in a text file. An example "words and their meanings" file is given in blackboard. Note that your program needs to ask the user for the "words and their meanings" file to use.
   b) The words and their meanings text file is of the csv (comma separated values) format. Use either notepad++ or notepad to create your file in the same format as the mywords.txt file given to you in blackboard
   c) Your program should then pick a word from the "words and their meanings" file, scramble the letters, and ask the user to unscramble it. Every run of your program should pick a word at random.
   d) The user may type in the unscrambled word or may ask for the definition/meaning of the word by entering a question mark. The game

continues until the user says "no" to the question: "Do you want to continue?". A sample run is shown below (user inputs are in red):

Give the name of the "words and their meanings" file: mywords.txt
Unscramble the following letters to form a word. Type "?" for the meaning of the unscrambled word: ulecenop

Enter the answer [or ? for the meaning]: ?
The word means: great wealth
Enter the answer [or ? for the meaning]: rich
Wrong, try again
Enter the answer [or ? for the meaning]: opulence
You got it! Do you want to continue [yes or no]: yes
Unscramble the following letters to form a word. Type "?" for the meaning of the unscrambled word: yrnuep
Enter the answer [or ? for the meaning]: injury
Wrong, try again
Enter the answer [or ? for the meaning]: ?
The word means: extremely poor
Enter the answer [or ? for the meaning]: penury
You got it! Do you want to continue [yes or no]: no
Goodbye!

Hint: a) The easiest way to store the "words and their meanings" file would be in a dictionary. You may have a dictionary that looks like the one shown below:

{ "opulence":"great wealth", "penury":"extremely poor", …..}

b) Please create your own "words and their meanings" file to test your code. I will be using my own test file to test your homework submission. Do not hard code filenames in your program.

2. (25 points) The file restaurant_reviews.txt is given to you.

   The file contains names of reviewers and their reviews of different restaurants in the comma separated format. The first string is the name of the reviewer followed by a name of a restaurant and its rating. You are required to write a Python program that computes a similarity score between any two reviewers using Euclidean distances. An example data from the restaurant_reviews.txt file is given below:

   Pete Wellsworth's reviews are:
   'Rooster and Owl': 3.0,
   'El Sapo Cuban Social Club': 4.0,
   'Punjab Grill': 3.0,
   'Shibumi': 5.0,

'Mama Chang': 3.5

Jay Samuel's reviews are:
'El Sapo Cuban Social Club':4.5,
'Mama Chang':1.0,
'Shibumi':4.0

The Euclidean distance between Jay Samuel and Pete Wellsworth is computed as follows:

1) Only consider the restaurants that both have reviewed.
2) Take the difference between corresponding reviews.
3) Sum the square of the differences
4) The square root of the sum of differences is the Euclidean score. The shorter the distance the closer the two reviewers.

For our example, this would be: $(4.0 - 4.5)^2 + (3.5 - 1.0)^2 + (5.0 - 4.0)^2 = 0.25 + 6.25 + 1 = 7.5$
Euclidean distance is Square Root of 7.5 = 2.7386

Your program should provide the following:

1) Ability to read in a user provided filename which contains restaurant reviews in the comma separated format similar to the given file restaurant_reviews.txt
2) Functionality to compute the similarity between two reviewers provided by the user
3) Functionality to compute the similarity between one user provided reviewer and all other reviewers in the database

A sample run for the two required functionalities is given below (user inputs are in red):

Give the name of the restaurant reviews file: restaurant_reviews.txt
What do you want to do? Input 1 for similarity between two reviewers, or Input 2 for similarity between one reviewer and all others in the database or 3 to quit: 1
Provide Reviewer1 name: Jay Samuel
Provide Reviewer2 name: Pete Wellsworth
The similarity score between Jay Samuel and Pete Wellsworth is: 2.7386

What do you want to do? Input 1 for similarity between two reviewers, or Input 2 for similarity between one reviewer and all others in the database or 3 to quit: 2
Provide Reviewer name: Jay Samuel
The Similarity Scores are:

Jay Samuel	**Tomm Sietsema**	**1.87**
Jay Samuel	**Corby Kumar**	**1.50**
Jay Samuel	**Jonathan Golder**	**2.87**

Jay Samuel Pete Wellsworth 2.74
Jay Samuel Brette Anderson 1.58
Jay Samuel Michael Baumer 1.80

What do you want to do? Input 1 for similarity between two reviewers, or Input 2 for similarity between one reviewer and all others in the database or 3 to quit: 3
Goodbye!

Hint:

a) A good way to convert the file contents to a Python data structure is to create a dictionary. For example the first two lines of the restaurant_reviews.txt file can be stored as a dictionary entry such as the following:

{'Tomm Sietsema': {'Rooster and Owl': 2.5, 'El Sapo Cuban Social Club': 3.5, 'The Godfather': 3.0, 'Shibumi': 3.5, 'Mama Chang': 2.5, 'Punjab Grill': 3.0},'Jonathan Golder': {'Rooster and Owl': 3.0, 'El Sapo Cuban Social Club': 3.5, 'The Godfather': 1.5, 'Shibumi': 5.0, 'Punjab Grill': 3.0, 'Mama Chang': 3.5}}