COM161 Problem Solving for Computing

Practical Lab 8

This week, we explore Python’s support of Dictionaries and Sets for data manipulation, which is a common programming requirement. As with previous Lab exercises, take time to digest and attempt each exercise yourself before seeking guidance and support from the teaching team. Refer to the week 8 slides for useful insight to Dictionaries and Sets.

Task 1

Two words are anagrams if they contain all of the same letters, but in a different order. For example, ‘Python and ‘Typhon, and ‘Study’ and ‘Dusty’ are anagrams. One way to determine if something is an anagram is to tally the individual letters that make up each word, and then to compare those tallies. Write and program that reads two words in from the keyboard, determines whether or not these are anagrams and produces a report detailing the character make-up of each word.

Task 2

Morse Code is an encoding scheme that uses dashes and dots to represent numbers and letters. In this exercise, you will write a program that uses a dictionary to store the mapping from letters and numbers to Morse code. Use a full stop to represent a dot, and a hyphen to represent a dash. You will find a copy of the international Morse code on Wikipedia.

Your program should read a message from the user. Then it should translate each letter and number in the message to Morse code, leaving a space between each sequence of dashes and dots. Your program should ignore any characters that are not letters or numbers. The Morse code for ‘Hello World’ Is shown here, as an example: …. . .-.. .-.. --- .-- --- .-. .-. .-..

Optional Task: Extend the program so that the encoded Morse Code is outputted to a file. Write a new second program that is able to read in the file and decode the message into English.

Task 3

In the game of Scrabble, each letter has points associated with it. The total score of a word is the sum of the score of its letters. More common letters are worth fewer points while less common letter are worth more points. The points associated with each letter is as follows:

- (1 point)-A, E, I, O, U, L, N, S, T, R
- (2 points)-D, G
- (3 points)-B, C, M, P
- (4 points)-F, H, V, W, Y
- (5 points)-K
- (8 points)- J, X
- (10 points)-Q, Z

Write a program that computes and displays the score for any word entered. Note: in order to simplify the logic, you should use the UPPER() function to convert all words entered into uppercase.

Task 4

Write a program that opens a specified text file and then displays a list of all the unique words found in the file. You will need to create a sample list of words into a file yourself. *Hint: store each word as an element of a set.*

Task 5

Create a text file that lists your expenses for the last month in the following categories: rent, transport, food, clothing, socializing, misc. Write a program that reads the data from a file and uses matplotlib to plot a pie chart showing how you spend your money. The file format may look something like:

```
#Expenses
rent 1000
transport 300
food 400
clothing 150
socializing 200
misc 100
```

Task 6

Review the Week 8 lecture materials and implement the programs presented in Examples 2 (tally_staff()) and 3 (at_the_movies).

Apply this grouping approach to the coffee example outlined in the Week 5/6 practical lab, towards tallying occurrences of each coffee type.