Week 2 Assignment: 

Strings and Collection Data Types

Strings are extensively utilized in numerous Python applications. Additionally, data collection types like lists, tuples, dictionaries, and sets are frequently employed in many Python scripts. In this assignment, you will complete several programming activities/exercises to become familiar with using these data types in Python.

# Activity 1 – Palindrome

A palindrome is a word that looks the same when read forward and backward. For example, "kayak" is a palindrome, while "loyal" is not.

Your task is to write a Python program which:

* asks the user for a text.
* checks whether the entered text is a palindrome, and prints the result.

Save the program as week2-1.py.

# Activity 2 – Count vowels

Write a program to print the occurrences of vowel letters in a string.

The vowels are, A, E, I, O, U, a, e, i, o, u.

Example:

Input:

“Hello, world!”

Output:

The number of occurrences of vowels is: 3

The number of occurrences of e/E is: 1

The number of occurrences of o/O is: 2

Save the program as week2-2.py.

# Activity 3 – Receipt

Write a Python program to implement the following steps:

Step 1: Create a grocery and prices dictionary having the following elements (key:value pairs) in it.

'Apple': 0.89

'Banana': 0.39

'Bread': 2.50

'Chocolate': 1.50

Step 2: Take the user(cashier)’s inputs for the numbers of each product the customer purchased, and print the receipt.

Save the program as week2-3.py.

# Activity 4 – Find Common and Unique Elements

Given two lists (or tuples), find the common elements and the unique elements in each list (or tuple) that are not in the other.

Example:

Input:

list1 = [1, 2, 2, 3, 4, 5]

list2 = [3, 4, 4, 5, 6, 7]

Output:

Common elements: {3, 4, 5}

Unique elements in list1: {1, 2}

Unique elements in list2: {6, 7}

Save the program as week2-4.py.

Hint: An easy solution is to use sets.

# Optional Challenge: Merge k Sorted Lists

You are given a list of *k* lists, each list is sorted in ascending order.

Merge all the lists into one sorted list and return it.

Consider time complexity (*O(Nlogk)* ) and not use the existing sort() method.

Example:

Input:

lists = [[1,4,5],[1,3,4],[2,6]]

Output:

[1,1,2,3,4,4,5,6]

# Completion / Submission:

Zip the Python script files into one file, week2\_*yourname*.zip. Before the due date, submit the zipped file to: BB -> CS5160 -> …-> Week 2 Assignment