SE 226 – LAB#5 2023-2024 SPRING

Aim: Understanding Python data structures.

Please download the "1000words.txt" file in BlackBoard under Lab2 folder. It will give you the list declaration for 1000 words as "words" variable.

1. Take 2 numbers from the user between 1-999 (inclusive). The first number must be greater than the second one. Try your "words" variable by printing a slice from the list. Use the numbers for beginning and ending indexes of the slice.

2. Write a Python script that will generate random passwords from the given list variable *words that* you will get by utilizing the array declaration code given to you. Then, your script should assign the variable *user_number* to a value to be entered by the user. Your script should validate that the value of *user_number* must be between 3 and 7, inclusive. This value would refer to the number of words to be chosen randomly from the list of variable *words*. Depending on the value of *user_number*, your code should randomly choose a number of words from the words list. Lastly, your script should concatenate these words into one string and reverse it to create the password and print it on the screen.

3. Modify your script from the 2nd question. It should run for 8 times. This means you will generate 8 different passwords. Create 2 empty sets (i.e. x = set()). Name the first set as "shorter" and the second set as "longer". If *user_number* is smaller than 5, put the newly generated password to the "shorter" set, else put it to the "longer" set. Lastly use the union( ) method of the sets to combine these sets and print it on the screen.

4. Write a Python script that performs the following tasks:

a. Define and print a dictionary where the keys are integers from 1 to 30 (both included) and the values are the multiplication of the key and its predecessor. You can examine the sample.

SAMPLE OUTPUT:

{1: 0, 2: 3, 3: 8, 4: 15, 5: 24, 6: 35, 7: 48, 8: 63, 9: 80, 10: 99, 11: 120, 12: 143, 13: 168, 14: 195, 15: 224, 16: 255, 17: 288, 18: 323, 19: 360, 20: 399, 21: 440, 22: 483, 23: 528, 24: 575, 25: 624, 26: 675, 27: 728, 28: 783, 29: 840, 30: 899}

b. Perform the same task in part a, but print each key-value pair in a different line.

c. Considering the dictionary in part a, compute and print the sum of all the values in the dictionary without using *sum()* function.

d. Take a number from the user and remove the item with the same key as the entered number. Your script should handle and not crash, if the user enters a number that is not a key in the dictionary.

5. Consider the following two dictionaries:

divided = {'Tony': 41, 'Rhodey': 43, 'Nat': 35}

we_fall = {'Steve': 39, 'Clint': 35, 'Wanda': 28}

a. Merge divided and we_fall into a new dictionary called united_we_stand and print its all items to the screen in a table format as shown in sample output.

SAMPLE OUTPUT:

Name Age
Tony 41
Rhodey 43
Nat 35
Steve 39
Clint 35
Wanda 28

b. Remove the item with the key *'Nat'* from *united_we_stand* and print the updated result to the screen.

c. Print the items of *united_we_stand* in a sorted order by key.

d. Find and print the maximum and the minimum values of the items of *united_we_stand*.