Practice Problems for List

Question 1.

Suppose you are given two lists of the same length.

Given:

keys = ['key1', 'key2', 'key3', 'key1', 'key2']

values = [10, [20], 30, [40, 50], 10]

You have to make a dictionary and a tuple combining both lists. [Your program should work for any given lists]

Sample Output:

my_dict : {'key1': [10, [40, 50]], 'key2': [[20], 10], 'key3': [30]}

my_tuple : (('key1', [10, [40, 50]]), ('key2', [[20], 10]), ('key3', [30]))

Question 2.

Write a python program that asks the user to give multiple words separated by space. Your program should count the length of each word. Create a list that contains no repeating lengths . Please consider the input as a fixed one.

Sample Input 1:

The purpose of our lives is to be happy

Sample Output 1:

[3. 7. 2, 51]

Explanation:

Here the lengths of the words are as 3, 7, 2, 3, 5, 2, 2, 2, 5. Therefore, you must create a list without any repeating length.

Question 3.

Suppose you are given a tuple of tuples. You have to make a dictionary where the keys will be the tuples of the given tuple and the values will be the average of each tuple. Also, print the key whose value is the maximum. [Your program should work for any given tuple]

Given Tuple:

((22, 22, 13), (30, 78, 45,4), (81, 959, 39, 45), (1, 4, 4,5,6))

Output:

{(22, 22, 13): 19.0, (30, 78, 45, 4): 39.25, (81, 959, 39, 45): 281.0, (1, 4, 4, 5, 6): 4.0}

Max = (81, 959, 39, 45)

Question 4.

Suppose you are given a tuple of tuples with odd and even numbers. You have to make a dictionary where the keys will be the average of the even numbers of each tuple and its corresponding value will be the sum of the odd values of each tuple. Print the new dictionary. [Your program should work for any given tuple]

Given Tuple:

((20, 22, 13), (30, 33, 45,4), (81, 76, 30, 45), (1, 7, 4,2,6))

Output:

{21: 13, 17: 78, 53: 126, 4: 8}

Explanation:

The even numbers and odd numbers of the first element of the tuple are 20, 22 and 13 respectively. So the average of even numbers is (20+22)/2 = 21 and the summation of the odd numbers is 13. Therefore, the first (key: value) pair is (21:13).

Question 5.

Suppose you are given a dictionary where the keys are characters and values are numbers. You need to create a new dictionary with only one key: value pair. The key will be the concatenation of all the uppercase keys and the value will be the summation of the lowercase values. You are not allowed to use upper(), lower(), isdigit() functions. [Your program should work for any given dictionary]

Given Dictionary:

my_dict = {'A': 7, 'B': '8', 'c': 14, 'D': 10, 'e': '70', 'F': 9, 'G': 2, 'h' : '32'}

Output:

{'ABDFG': 116}

Explanation:

The uppercase keys are 'A', 'B', 'D', 'F' and 'G'. So the key is 'ABDFG'. The summation of the values of lowercase keys is (14+70+32) = 116. Therefore the new value is 116.

Question 6.

Suppose you are given a list of lists. You have to make a dictionary where the key will be the last element of each list and the value will be a tuple of the remaining elements of the list. [Your program should work for any given dictionary] **You are not allowed to use slicing**.

Given List 1:

[['Red', 'Blue', 'Violet'], ['Blue', 'Yellow','Green'], ['Red', 'Yellow','Orange']]

Output:

{'Violet': ('Red', 'Blue'), 'Green': ('Blue', 'Yellow'), 'Orange': ('Red', 'Yellow')}

Given List 2:

[['Red', 'Yellow','Blue', 'Black'], ['Red', 'White','Pink']]

Output:

{'Black': ('Red', 'Yellow', 'Blue'), 'Pink': ('Red', 'White')}

Question 6.

Suppose you are given a list of strings. You have to create a dictionary where the key will be the elements of the list in reverse order and its corresponding value will be a list containing each character of the elements. The output must be exactly in the same form as the sample output. **You are not allowed to use reverse functions or slicing to reverse the list.**

Given Input1:

['Emma', 'Olivia', 'Amelia']

Sample Output1:

{'Amelia': ['A', 'm', 'e', 'l', 'i', 'a'], 'Olivia': ['O', 'l', 'i', 'v', 'i', 'a'], 'Emma': ['E', 'm', 'm', 'a']}

Given Input2:

['1', '6543', '976','23']

Sample Output2:

{'23': ['2', '3'], '976': ['9', '7', '6'], '6543': ['6', '5', '4', '3'], '1': ['1']}

Question 7.

Assume, you have been given two lists. Print both lists simultaneously such that list1 should display items in original order and list2 in reverse order.

==========

Given lists: List1 = [10 ,20, 30, 40]

List2 = [100,200,300,400]

Sample Output:

10 400

20 30

30 200

40 100

==========

Given lists:

List1 = [90, 30,-30, 20 ,12]

List2 = [-8,-5,-2, 100 , 6]

Sample Output:

90 6

30 100

-30 -2

20 -5

12 -8

Question 8.

Write a Python program to reverse strings in a given list of string values without using reverse() or reversed().

Sample Input: ['Red','Green','Blue','White','Black']

Sample Output: ['deR', 'neerG', 'eulB', 'etihW', 'kcalB']

Question 9.

Write a Python program that takes a list from the user. Then find the first even and first odd number in the list.

Sample Input1:

[1, 3, 5, 7, 4, 1, 6, 8]

Sample Output1:

First even : 4 and First odd:1

Sample Input2:

[55, 5, 7, 9, 11]

Sample Output2:

First even: not found and First odd: 55

Sample Input3:

[50, 52, 6, 2, 10]

Sample Output3:

First even : 50 and First odd: not found

Question 9.

Assume, you have been given a list T and a number, N. Now, create a list by concatenating the input list

which range goes from 1 to N.

==========

Sample input1:

T = ['p', 'q']

N = 5

Sample Output1:

['p1', 'q1', 'p2', 'q2', 'p3', 'q3', 'p4', 'q4', 'p5', 'q5']

==========

Sample input2:

T = ['p', 'q', 'r']

N = 3

Sample Output2:

['p1', 'q1', 'r1', 'p2', 'q2', 'r2', 'p3', 'q3', 'r3']

==========

Sample input3:

T = ['p', 'q', 'r','s']

N = 2

Sample Output3:

['p1', 'q1', 'r1', 's1', 'p2', 'q2', 'r2', 's2']

Question 10.

Assume that you have given the following dictionary:

d = {

"Agent Name": ["Sova", "Jett", "Chamber", "Brimstone", "Reyna"],

"Agent Type": ["Initiator", "Duelist", "Sentinel", "Controller", "Duelist"],

"Sequence" : [1, 3, 5, 2, 4]

}

However, you have to write a Python program that will generate the following dictionary:

r_d = {

1: ("Sova", "Initiator"),

2: ("Brimstone", "Controller"),

3: ("Jett", "Duelist"),

4: ("Reyna", "Duelist"),

5: ("Chamber", "Sentinel")

}

Here, the "Agent Name" and "Agent Type" have the same number of indices along with a correlation. For example, at index 0 for "Agent Name" and "Agent Type" we get "Sova" and "Initiator". So, in the new dictionary there should be a key which will hold a tuple value of ("Sova", "Initiator").

On the other hand, here the "Sequence" holds the sequence of those tuples in that new dictionary. For example,

At position 1, we have to place the value ("Sova", "Initiator"),

At position 3, we have to place the value ("Jett", "Duelist")

At position 5, we have to place the value ("Chamber", "Sentinel")

At position 2, we have to place the value ("Brimstone", "Controller")

At position 4, we have to place the value ("Reyna", "Duelist")

Question 11.

Let's assume that a player plays a multi-agent game where every gun has a type of "Side Arms" or "Rifles" or "Sniper Rifles". Here, in every dictionary inside the tuple, the key is the gun name, the first index of the value-list is the gun type and the second index of the value-list is the kills done by that gun in a certain match.

Write a Python program that will generate a simple dictionary, where the unique gun types will be the keys and the value will be a list holding the specific gun names at the front and the summation of the total kills by those guns as the last index.

[YOU DO NOT NEED TO TAKE ANY INPUT FROM THE USER. YOU CAN SIMPLY STORE THE SAMPLE INPUT DATA INTO A VARIABLE AND PROCEED.]

Given Sample Input 1:

({"Ghost" : ["Side Arms", 3]}, {"Vandal" : ["Rifles", 15]}, {"Sheriff" : ["Side Arms", 5]}, {"Operator" : ["Sniper Rifles", 7]}, {"Phantom" : ["Rifles", 10]})

Expected Sample Output 1:

{"Sidearms" : ["Ghost","Sheriff", 8], "Rifles" : ["Vandal", "Phantom", 25], "Sniper Rifles" : ["Operator", 7]}

Explanation 1:

Here, the unique gun types are: "Sidearms", "Rifles" and "Sniper Rifles", which will be the keys for the output dictionary.

For "Side Arms" we get two gun names: "Ghost", "Sheriff" and the total kills by them is = 3+5 = 8. So, the value for the key "Side Arms" will be ["Ghost","Sheriff", 8]. And so on.

Given Sample Input 2:

({"Classic" : ["Side Arms", 2]}, {"Guardian" : ["Rifles", 10]}, {"Shorty" : ["Side Arms", 4]}, {"Vandal" : ["Rifles", 20]})

Expected Sample Output 2:

: {"Sidearms" : ["Classic","Shorty", 6], "Rifles" : ["Guardian", "Vandal", 30]}

Explanation 2:

Here, the unique gun types are: "Sidearms" and "Rifles", which will be the keys for the output dictionary. For "Side Arms" we get two gun names: "Classic", "Shorty" and the total kills by them is = 2+4 = 6. So, the value for the key "Side Arms" will be ["Classic","Shorty", 6]. And so on.

Question 12.

Write a function find difference which takes 2 user inputted strings as parameters. The 1st parameter is the original string and the 2nd parameter is a string made by random shuffling of the 1st parameter and then one more letter is added at a random position. Your task is to find out the random letter and its position and print them inside the function. Finally call the function.

Sample Input 1:
User input 1 = ‘abcd’
User input2 = ‘dcbea’
find_difference(‘abcd’,’dcbea’)

Sample Output 1:
e was added at position 3

Explanation 1:
The original string was “abcd” and after shuffling it became ‘dcba’ and a random letter,
In this case ‘e’ was added at position 3.

Question 13.

x=['h','a','h','p','g','a','a','v','p','p','a','a','b','a','p']
Your task is to find and print the first unique character in it. You must use a dictionary to solve this problem. [Marks: 101

Sample Input:
Give List : x=['h','a','h','p','g','a','a','v','p','p','a','a','b','a','p']

Sample Output:
'g' was the 1st letter which was entered only once

Explanation: From our given list it can be seen that, 'h' occurs 2 times, 'a' occurs 6 times, 'p' occurs 4 times, 'g' occurs 1 time, 'v' occurs 1 time and 'b' occurs 1 time. So among these, 'g', 'v' and 'b' are unique as they occurred only once but 'g' came before 'v' and 'b' and so 'g' is the first unique character.