Dictionaries

1. Напишите программу на Python, которая принимает два списка L1 и L2, С тем же числом элементов, n. Элементы списка L1 могут быть целыми числами (int) или (str), а элементы L2 могут быть (list) или (tuple).

Программа создаст словарь D, так что каждый элемент L1 будет ключом в словаре, и каждый L2 будет иметь значение, связанное с этим соответствующим ключом. Можно предположить, что количество элементов одинаково в обоих списках.

L1=[1,2,'a','b']

L2=[(22,'aa'),['x'],('3',),('s',3)]

D= {1:(22,'aa'), 2:['x'],'a':(3,),'b':('s',3)}

1. Напишите программу, которая получает один список с тем же количеством значений, которые hashable, и значений, которые не являются hashable. Значения могут быть смешаны, а не отсортированы по этим двум типам. Программа создаст словарь D, в котором ключи - это хешируемые значения, найденные во входных данных, а все остальные значения - это значения, связанные по одному с ключами.

Подсказка: создайте два отдельных списка: один со значениями, которые можно хэшировать, а другой со списками, которые не являются хэшируемыми, и создайте из них словарь.

L=[1,2,[22],'a','b',[33],[44],[55]]

d = {'a': [44], 1: [22], 2: [33], 'b': [55]}

1. Напишите программу на Python, которая создает список из 100 целых чисел, которые выбираются случайным образом в диапазоне 1-100. Программа рассчитает частоту каждого из чисел от 1 до 100 в приведенном выше списке. Программа напечатает следующий отчет:

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

Вывод будет выглядеть следующим образом (должны быть напечатаны только первые 10 строк):

1,2 times

2,5 times

3,4 times

4,2 times

5,1 times

6,1 times

7,0 times

8,0 times

....

Примечания: Для написания всех вышеперечисленных программ можно предположить, что входные данные верны, то есть нет необходимости проверять целостность ввода.