Tuple Practice Problems

1. **Magic number:** If the summation of even indexed digits is equal to the summation of odd indexed digits, then the number is a magic number.

   Now, write a Python program that will take a string of numbers where each number is separated by a comma. The program should print a tuple containing two sub-tuples, where the first sub-tuple will hold the magic numbers and the second sub-tuple will hold the non-magic numbers.

   ==

   Sample Input1:

   "1232, 4455, 1234, 9876, 1111"

   Sample Output1:

   ((1232, 4455, 1111), (1234, 9876))

   Explanation1:

   For 1232, the sum of even indexed digits is = 4 & the sum of odd indexed digits is = 4. So, 1232 is a magic number.
   For 4455, the sum of even indexed digits is = 9 & the sum of odd indexed digits is = 9. So, 4455 is a magic number.
   For 1234, the sum of even indexed digits is = 4 & the sum of odd indexed digits is = 6. So, 1234 is a non-magic number.
   For 9876, the sum of even indexed digits is = 16 & the sum of odd indexed digits is = 14. So, 9876 is a non-magic number.
   For 1111, the sum of even indexed digits is = 2 & the sum of odd indexed digits is = 2. So, 1111 is a magic number.

   So, the final answer is ((1232, 4455, 1111), (1234, 9876))

   ==

   Sample Input2:

   "442310, 441310, 42251, 42241, 451, 452"

   Sample Output2:

   ((442310, 42251, 451), (441310, 42241, 452))

   Explanation2:

   For 442310, the sum of even indexed digits is = 7 & the sum of odd indexed digits is = 7. So, 442310 is a magic number.
   For 441310, the sum of even indexed digits is = 6 & the sum of odd indexed digits is = 7. So, 441310 is a non-magic number.

   For 42251, the sum of even indexed digits is = 7 & the sum of odd indexed digits is = 7. So, 42251 is a magic number.
   For 42241, the sum of even indexed digits is = 7 & the sum of odd indexed digits is = 6. So, 42241 is a non-magic number.
   For 451, the sum of even indexed digits is = 5 & the sum of odd indexed digits is = 5. So, 451 is a magic number.
   For 452, the sum of even indexed digits is = 6 & the sum of odd indexed digits is = 5. So, 452 is a non-magic number.
   So, the final answer is ((442310, 42251, 451), (441310, 42241, 452))

2.We all know that the additive primaries are red, green, and blue. Now, write a Python program that will take a color sequence as a string from the user where R represents Red, G represents Green and B represents Blue. The program should print the choice of colors that is actually a tuple containing the sub-tuples as (color_name, color_frequency) only if the color_frequency for that color is at least one in the given color sequence. [You are not allowed to use count() function here.]

==

Sample Input1:

"RGBRRGBBR"

Sample Output1:

(('Red', 4), ('Green', 2), ('Blue', 3))

Explanation2:

In the given color sequence, "R", "B", and "G" have color_frequency at least one (minimum1). So, "Red", "Green" and "Blue" are present in the output.

==

Sample Input2:

"RBRRBBR"

Sample Output2:

(('Red', 4), ('Blue', 3))

Explanation2:

In the given color sequence, "R" and "B" have color_frequency at least one (minimum1). So, "Red" and "Blue" are present in the output. On the other hand, "G" is not present in the given color sequence. So, "Green" is missing from the output.

==

Sample Input3:

"RRRRR"

Sample Output3:

(('Red', 5),)

Explanation3:

In the given color sequence, only "R" has color_frequency at least one (minimum1). So, only "Red" is present in the output. On the other hand, "B" or "G" are not present in the given color sequence. So, "Green" and "Blue" are missing from the output.

3.Write a Python program that takes a tuple of tuples as an input from the user. Then calculates the average value of the numbers for each tuple of tuples and find the tuple whose sum is the maximum. [You are not allowed to use max () function here. You are not allowed to use Regex split in this task.]

Hints: Since the input function converts everything to string by default. You might need to use strip() and split() to get the data/tuples.

==

Sample Input1:

((33, 22, 11), (30, 45, 56, 45,20), (81, 90, 39, 45), (1, 2, 3, 4,5,6))

Sample Output2:

Average of tuples: [22.0, 39.2, 63.75, 3.5]

Tuple with maximum sum is (81, 90, 39, 45)

Dictionary Practice Problems

1. Suppose two dictionaries are given. Write a Python program that combines two dictionaries and creates a list of values for each key. Then finally print the combined dictionary.

==

Given dictionaries1:

dict1 = {'w': 50, 'x': 100, 'y': 'Green', 'z': 400}

dict2 = {'x': 300, 'y': 'Red', 'z': 600}

Sample Output1:

{'w': [50], 'x': [100, 300], 'y': ['Green', 'Red'], 'z': [400, 600]}

==

Given dictionaries2:

dict1 = {'w': 50, 'y': 'Green', 'z': 400}

dict2 = { 'y': 'Red', 'a':200, 'b':4, 'z': 600}

Sample Output2:

{'w': [50], 'y': ['Green', 'Red'], 'z': [400, 600], 'a': [200], 'b': [4]}

==

2.   Assume, you have been given a dictionary named dict_1. The values of the dictionary will be a list or a tuple depending on the case of the dictionary key.

     If the key is in lower case, then the value will be in list format.
     If the key is in upper case, then the value will be in tuple format.

Now, write a Python program that creates a new dictionary named "dict_primes" that contains only the prime numbers in the value. Then, finally, print the dictionary dict_primes.

===

Given dictionary1:
{"a":[5,2,55,17],"P":(11,121,222),"B":(37,53,71),"c":[45,92,50]}

Sample Output 1:
dict_primes = {'a': [5, 2, 17], 'P': (11,), 'B': (37, 53, 71), 'c': []}

Explanation1:

Among the elements of [5, 2, 55, 17], the prime numbers are 5, 2, 17. So, they are added to the value.
The key a is in lowercase, so the value will be a list.
==
Among the elements of (11, 121, 222), the prime number is 11. So, it is added to the value.
The key P is in uppercase, so the value will be a tuple.
==
Among the elements of (37, 53, 71), the prime numbers are 37, 53, 71. So, they are added to the value.
The key B is in uppercase, so the value will be a tuple.
==
Among the elements of [45, 92, 50], none are prime numbers. So, none of the numbers are added to the value.
The key c is in lowercase, so the value will be a list.
===

Given dictionary2:
dict_1 = {"N":(4,9,3),"k":[95,37,197],"F":(32,5,97),"s":[31,94,55]}

Sample Output 2:
dict_primes = {'N': (3,), 'k': [37, 197], 'F': (5, 97), 's': [31]}

Explanation2:
Among the elements of (4, 9, 3), the prime number is 3. So, it is added to the value.
The key N is in uppercase, so the value will be a tuple.
==
Among the elements of [95, 37, 197], the prime numbers are 37, 197. So, they are added to the value.
The key k is in lowercase, so the value will be a list.
==
Among the elements of (32, 5, 97), the prime numbers are 5, 97. So, they are added to the value.
The key F is in uppercase, so the value will be a tuple.
==
Among the elements of [31, 94, 55], the prime number is 31. So, it is added to the value.
The key s is in lowercase, so the value will be a list.
==

3.Assume in a video game, there are three agents named Rage, Jett, and Sage. Each of these agents has three normal skills along with an ultimate skill. Each of these skills have a damage score. To calculate the additive of this score, you need to use the following formula.

additive_damage_score = sum of the Normal Skill Damage + Ultimate skill damage

Now, write a Python program that will detect the agent's name using "additive_damage_score" the from a given dictionary where the keys are "Normal Skills", "Ultimate Skill" and the values are the damages due to the use of those skills on the opponents.

- If the additive damage score is less than or equal to 70, then the agent's name is "Rage".

- If the additive damage score is greater than 70 and less than or equal to 100, then the agent's name is "Jett".

- Otherwise, the agent's name is "Sage".

==

Given Dictionary1:

given_dict = { "Normal Skills":[10,15,20], "Ultimate Skill":50 }

Sample Output1:

Additive Damage Score is 95

The agent's name is Jett

==

Given Dictionary2:

given_dict = { "Normal Skills":[10,10,10], "Ultimate Skill":30 }

Sample Output2:

Additive Damage Score is 60

The agent's name is Rage