LUCIANO GABBANELLI

EXTRA EXTRA

1. Exercices

All exercises must be performed by means of the functions or methods requested (more complex structures are optional):

1. Write a Python program to triple all numbers of a given list of integers. Use Python map

```
nums = (1, 2, 3, 4, 5, 6, 7)

print("Original list: ", nums)

result = map(lambda x: x + x + x, nums)

print("\nTriple of said list numbers:")

print(list(result))
```

2. Create a function that takes a list of digits and returns the number they correspond to. For example [1,2,3] corresponds to the number one hundred twenty-three (123). Use the reduce function.

```
from functools import reduce

def digits_to_numbers(digits):

    return reduce(lambda x,y:x*10 + y,digits)


digits_to_numbers([4,3,9,2])
```

3. Write a Python program to listify the list of given strings individually using Python map. Listify means to create a list with each character: Input: `['Red', 'Blue']`. Output: [[‘R’, ‘e’, ‘d’] , [‘B’, ‘l’, ‘u’, ‘e’]]

```
color = ['Red', 'Blue', 'Black', 'White', 'Pink']

print("Original list: ")

print(color)

print("\nAfter listify the list of strings are:")

result = list(map(list, color))

print(result)
```

4. Create a function that returns the words from a list of words that start with a specific letter. Use the filter function.

```
def filter_words(list_words, letter):

    return list(filter(lambda word:word[0]==letter, list_words))

filter_words([Dog,'Cat','Ball','Apple','Book','Python'], 'B')
```

5. Write a Python program to add (element to element) three given lists using Python map and lambda functions.

```
nums1 = [1, 2, 3]

nums2 = [4, 5, 6]

nums3 = [7, 8, 9]

result = map(lambda x, y, z: x + y + z, nums1, nums2, nums3)

print("\nNew list after adding above three lists:", list(result))
```

6. Make a function that takes a list and returns a dictionary containing the values of the list as the key and the index as the value. Use the enumerate function.

```
def d_list(L):

    return {key:value for value,key in enumerate(L)}

d_list(['a','b','c','d','e'])
```

7. Make a function that returns the count of the number of elements in the list whose value is equal to its index. Use the enumerate function.

```
def count_match_index(L):
     return len([num for count,num in enumerate(L) if num==count])


count_match_index([0,2,2,1,5,5,6,10])
```

8. Make a function that takes a list comprehension to return a list of equal length, where each value is the two strings of L1 and L2 concatenated with a connector between them. Example: Lists: ['A', 'a'] ['B','b'] Connector: '-' Output: ['A-B'] ['a-b']. Use the zip function.

```
def concatenation(L1, L2, connector):
    return [word1+connector+word2 for (word1,word2) in zip(L1,L2)]


concatenation(['A', 'a'],['B','b'],'-')
```

9. Write a Python program to create a list containing the power of said number in bases raised to the corresponding number in the index using Python map.

   pow() is given to map two list objects, one for each base and index parameter.

```
bases_num = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]

index = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

print("Base numbers abd index: ")

print(bases_num)

print(index)

result = list(map(pow, bases_num, index))

print("\nPower of said number in bases raised to the corresponding number in the index:")

print(result)
```