Functional Programming in python

1. Write a Python script for calculating the sum of the positive and negative numbers of a given list using lambda and filter.

   Output structure:

   Given list: [4, 5, 10, -4, -2, 8, -10]

   Sum of positive numbers: 27

   Sum of negative numbers: -16

2. Write a Python script using python's "sorted" function (and lambda) to sort a list of dictionaries based on the age key.

   Output structure:
   Given dictionary: [{'name':'Zahra', 'age':26, 'salary':100},{'name':'Fatemeh', 'age':21, 'salary':50}, {'name':'Mobina', 'age':32, 'salary':250},{'name':'Hanieh', 'age':17, 'salary':10}]
   Sorted list: [{'name':'Hanieh', 'age':17, 'salary':10},{'name':'Fatemeh', 'age':21, 'salary':50},{'name':'Zahra', 'age':26, 'salary':100}, {'name':'Mobina', 'age':32, 'salary':250}]

3. Using lambda and reduce, create Fibonacci series up to n.

   Output structure:

   fib(5) = [0, 1, 1, 2, 3]

   hint: reduce takes a third argument as the initial value, use [0,1] as the initial value for the third argument. Reduce will be applied on range(n-2) (for given n by user) as the iterable object. Define the first argument for reduce using lambda with two arguments, but you may not use the second argument in the body of the lambda function. The first argument can be considered a list.

4. Find the intersection of two given lists using lambda and filter.

   Output structure:

   Original arrays: [5, 7, 8, 9, 10], [4, 8, 9]

   Intersection of arrays: [8, 9]

5. Use lambda to find palindromes in a given list of strings. Palindrome is a string that reads the same backwards as forwards, e.g. madam.

   Output structure:

Original list of strings: [‘Python’, 'php', 'Java', 'abcba']

List of palindromes: ['php', 'abcba']

Hint: use *reversed* to reverse the string in a list and then use *“”.join* to join the list in one string.

---

6. Using lambda, write a python script to count the number of times that items in a given list appear in the list. The output should be declared in a dictionary.

Output structure:

Original list: [0,0,1,0,2,2,2,3,5,4]

Count dictionary: {0: 3, 2: 3, 3: 1, 4: 1, 5: 1}

Hint: use .count() method. You may map each item in the original list to a tuple. Convert the whole obtained pairs of tuples to a dictionary using dict. It will take care of removing duplicate items as well.

---

7. Consider a list of full names in a list where each element is formatted as "Firstname Lastname", like ["Ali abbasi", "Mohammad Mohammadi", "Mojtaba Shokri"]. Write a list comprehension that produces a list with the full names in the format "Lastname, Firstname". The resulting list should look like: [“Abbasi, Ali”, “Mohammadi, Mohammad”, “Shokri, Mojtaba”]. Use only one line of code.

Hint: You can use .split()[0] and .split()[-1] to separate the string.

---

8. Using list comprehension, write a Python script to remove the 0th, 3rd and 6th element from the list in [10, 9 , 8 ,7, 6, 5, 4, 3, 2, 1]. Use only one line of code.

Hint: You may use “enumerate” in your inner for loop to access index of the elements.

---

9. Use list comprehension to calculate the list of prime numbers up to given number n.

Hint: You may use nested list comprehension. This can be written in one line of code.

---

10. A) Use nested list comprehension to generate the following matrix:

$$list1 = \begin{bmatrix}0 & 1 & 2\\ 3 & 4 & 5\\ 6 & 7 & 8\end{bmatrix}$$

B) Use another nested list comprehension to generate the following list from list1:

$$list2 = [[0,4], [16], [36,64]]$$