20.4.2022 **Exercise -3**

Python Compound Datatypes

1. Given a list of integers, multiply the even integers by 3 and the odd integers with 5, return the sum of all the items after the multiplication.

2. Find the ***"centered"*** average of elements in a list which is mean average of elements except the smallest and the largest element in the list. If there are multiple copies of smallest (or) largest element, ignore only one copy of the element during computation.

   Example: [2,1,3,4,1,2,4]
   "centered" average = (1 + 2 + 2 + 3+ 4) / 5 = 2.4

3. State the inbuilt function for reversing a list.

   Write a script to reverse a list of given elements without using built-in function. Ex: For a list with three elements [18,34,27], the reversed list is [27,34,18]

   Extend the same script by accepting two integers *start* and *stop* whose values are between 0 and n-1 where n is the length of the list and reverse only the portion of numbers between the index locations *start* and *stop.*

   Ex: For a list with five elements [18,34,27,45,16], start=1 and stop=3, then the script should print [18,45,27,34,16].

4. Given a list which contains values as 0 and 1, process and find the lengths of the group of consecutive 1's in the input list, in the order from left to right.
   Example: If the input list contains [1,1,0,1,0,0,1,1,1,0,0], then the script should print [2,1,3]

5. Create a list of n elements, process the elements based on the following condition:
   if element(x) is a prime number, then print x*x
   else print the sum of digits in the number

6. Implement a lookup table concept (LUT) using python lists. Lookup table concept is used to retrieve values from the table in faster manner.

   Create a list which stores the computations of a function "3x+2" for a range of integer values from 0 to 7

0	1	2	3	4	5	6	7
2	6	8	11	14	17	20	23

Now, use the LUT to access values associated with elements 0 to 7 Create another list which holds only elements from 0 to 7, refer the lookup table and find the output of computations

Example: If the list is [0,7,6,3,1] then the output is a list [2,23,20,11,6]

Additional marks will be provided if the student is able to prove that LUT decreases the amount of time required for computation when compared with sequential processing of elements.

7. There are 12 musical notes in chromatic scale namely

C	C#	D	D#	E	F	F#	G	G#	A	A#	B

Interval between each pair of notes is called as semitone. D# is 3 semitones above C, E is 2 semitones above D. Every note has a major scale which comprises of 7 out of 12 notes which are 0,2,4,5,7,9 and 11 semitones above the current note.

Example: For note D, major scale comprises of (D, E, F#, G, A, B, C#), For note C, major scale comprises of (C, D, E, F, G, A, B)

Write a script which accepts a note and prints its major scale.

8. Given a tuple of n integers, iteratively form tuples which records the absolute difference between the neighbouring integers. Find the exact number of steps the tuple takes to reach either tuple of all zeros (or) original tuple sequence.

Example: Input tuple = (1,2,1,2,1,0)

In the first iteration we get a new tuple of form (1,1,1,1,1,1). In the next iteration we get (0,0,0,0,0,0). It **takes two steps** for the tuple to reach all zeros.

#Note: Not all tuples will descend to zeros. So, limit the number of iterations to 100. If it does not end up with input sequence or all zeros with the specified number of iterations, print 'Invalid Tuple'

9. Accept a list of donors and their information (name, blood type, age) and maintain it as a dict with blood type as the key value

There will be more than one donor. Hence, the details of the donors are maintained as list of tuples. Each tuple corresponds to one donor

which stores the information in a particular order (name, blood type, age)

Accept a patient detail in particular the blood type and the age.

Print the most compatible donor who should have the same blood type and small age difference

Also print other compatible donors which is a list of donors with same blood type sorted by age difference.

10. Accept two strings
Now, process them independently where you find the count of each character in a string and maintain it in a dictionary. As a result of the above process, two dicts will be created

Example: "hello" will be processed as h -> 1 e->1 l->2 o->1
"balloon" will be processed as b->1 a->1 l->2 o->2 n->1

Extend the program and merge the dictionaries and find the total count of each character. Also print the counts sorted by key in alphabetical order.