**EXTRA TASK:**

**DATA STRUCTURES**

1. Create a list of given structure and run

**x=[100,200,300,400,500,[1,2,3,4,5,[10,20,30,40,50],6,7,8,9],600,700,800]**

* Access list [1, 2, 3, 4]
* x=[100,200,300,400,500,[1,2,3,4,5,[10,20,30,40,50],6,7,8,9],600,700,800]
* print(x[5][0:4])
* Access list [600, 700]
* x=[100,200,300,400,500,[1,2,3,4,5,[10,20,30,40,50],6,7,8,9],600,700,800]
* print(x[6:][:-1])
* Access list [100, 300, 500, 600, 800]

x = [100,200,300,400,500,[1,2,3,4,5,[10,20,30,40,50],6,7,8,9],600,700,800]

del x[1::2]

print(x)

* Access list [[800, 700, 600, [1, 2, 3, 4, 5, [10, 20, 30, 40, 50], 6, 7, 8, 9], 500, 400, 300, 200, 100]
* def Reverse(lst):
* lst.reverse()
* return lst
* lst = [100,200,300,400,500,[1,2,3,4,5,[10,20,30,40,50],6,7,8,9],600,700,800]
* print(Reverse(lst))
* Access list [10]

x=[100,200,300,400,500,[1,2,3,4,5,[10,20,30,40,50],6,7,8,9],600,700,800]

print(x[5][5][0])

* Access list [ ]

x=[100,200,300,400,500,[1,2,3,4,5,[10,20,30,40,50],6,7,8,9],600,700,800]

print(x[5:-5])

2. Create a list of thousand numbers using range and xrange and see the difference between each other.

list=[i for i in range(1000)]

print(list)

list=[i for i in xrange(1000)]

print(list)

In Python 2.x:

* range creates a list, so if you do range(1, 10000000) it creates a list in memory with 9999999 elements.
* xrange is a sequence object that doesn’t generate a static list at run time.

In Python 3:

range does the equivalent of python's xrange.

3. How Tuple is beneficial as compared to the list?

A. In Python tuples are immutable but list are mutable.

Tuples are identified by python interpreter as one immutable constant literal, and hence is built as 1 single entity and stored in hashtable and are fetched when some execution is done on them.

Whereas list are mutable object, so each time some execution is done new objects are created hence arent interpreted just once and hence lists are slower than tuple.

4. Write a program in Python to iterate through the list of numbers in the range of 1,100 and print the number which is divisible by 3 and a multiple of 2.

lower = int(input("Enter lower range limit:"))

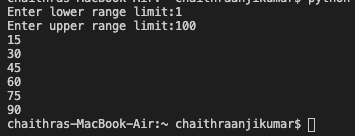
upper = int(input("Enter upper range limit:"))

for i in range(lower, upper+1):

if((i%3==0) & (i%5==0)):

print(i)

Output:



5. Write a program in Python to reverse a string and print only the vowel alphabet if it exists in the string with their index.

def reverse\_vowels(str1):

vowels = ""

for char in str1:

if char in "aeiouAEIOU":

vowels += char

result\_string = ""

for char in str1:

if char in "aeiouAEIOU":

result\_string += vowels[-1]

vowels = vowels[:-1]

else:

result\_string += char

return result\_string

print(reverse\_vowels("Panthom"))

print(reverse\_vowels("palindrome"))

6. Write a program in Python to iterate through the string “hello my name is Abcde” and print the string which has even length of the word.

def printWords(s):

# split the string

s = s.split(' ')

# iterate in words of string

for word in s:

# if length is even

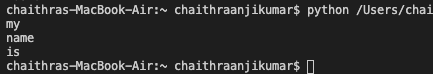
if len(word)%2==0:

print(word)

s = "hello my name is Abcde"

printWords(s)

Output:



7. Write a program in python to print the pair of numbers whose sum is equal to result number that is let's say 8.

**x=[1,2,3,4,5,6,7,8,9,-1]**

def pairs\_count(arr, n, sum):

ans = 0

arr = sorted(arr)

i, j = 0, n - 1

while (i < j):

if (arr[i] + arr[j] < sum):

i += 1

elif (arr[i] + arr[j] > sum):

j -= 1

else:

x = arr[i]

xx = i

while (i < j and arr[i] == x):

i += 1

y = arr[j]

yy = j

while (j >= i and arr[j] == y):

j -= 1

if (x == y):

temp = i - xx + yy - j - 1

ans += (temp \* (temp + 1)) // 2

else:

ans += (i - xx) \* (yy - j)

return ans

arr = [1, 2, 3, 4, 5, 6, 7, 8, 9, -1]

n = len(arr)

sum = 8

print(pairs\_count(arr, n, sum))

**Output:**

**4**

8. Write a program in Python to complete the following task:

* Create two different lists as in even\_list and odd\_list

ls = [8, 11, 20, 30, 33, 41, 54, 69]

lsOdd = []

lsEven = []

for num in ls:

if num%2 == 0:

lsEven.append(num)

else:

lsOdd.append(num)

# print lists

print ("list: ", ls)

print ("list of Even: ", lsEven)

print ("list of Odd: ", lsOdd)

Output:



* Ask the user to enter the number in the range of 1,50 and make sure if the entered number is even appended it to the even\_list and if the entered number is odd append it to the odd list.

ls = range(1,50)

lsOdd = []

lsEven = []

for num in ls:

if num%2 == 0:

lsEven.append(num)

else:

lsOdd.append(num)

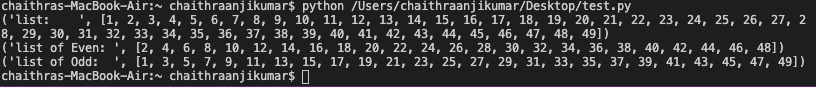
# print lists

print ("list: ", ls)

print ("list of Even: ", lsEven)

print ("list of Odd: ", lsOdd)

Output:



* Keep that in mind you can only add 5 items in each list

def listsum(numList):

if len(numList) == 1:

return numList[0]

else:

return numList[0] + listsum(numList[1:])

print(listsum([1,3,5,7,9]))

* Make sure once you entered the total 5 elements calculate the sum of the list and return the maximum out of the list.

9. Write a program to find out the occurrence of a specific word from an alphanumeric statement. **Input:** 12abcbacbaba344ab

**Output:** a=5 b=5 c=2

Note: Make sure to avoid counting the occurrence numeric values in the string.

def char\_frequency(str1):

dict = {}

for n in str1:

keys = dict.keys()

if n in keys:

dict[n] += 1

else:

dict[n] = 1

return dict

print(char\_frequency('12abcbacbaba344ab'))

10. Generate and print another tuple whose values are even numbers in the given tuple (1,2,3,4,5,6,7,8,9,10).

tup=(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)

ls= list()

for i in tup:

if tup[i-1]%2==0:

ls.append(tup[i-1])

tup\_even=tuple(ls)

print(tup\_even)