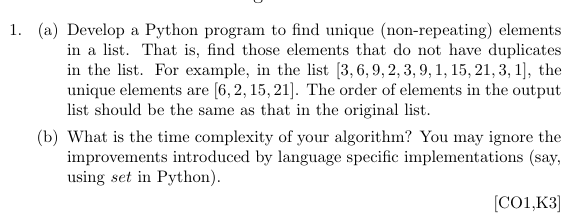
**ASSIGNMENT – 2**

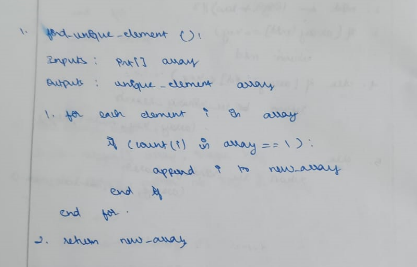
**AIM:**

To solve the given problems using Python and analyze the time complexities of the problems.

**Qn1:**

****

**Psuedo Code:**

****

**Source Code:**

def getUnique(l):

    unique = []

    for i in l:

        if l.count(i) == 1:

            unique.append(i)

    return unique

l = [3,6,9,2,3,9,1,15,21,3,1]

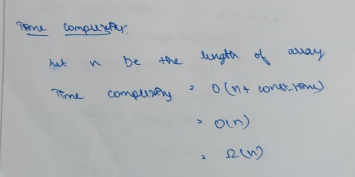
unique = getUnique(l)

print("Unique elements: ", unique)

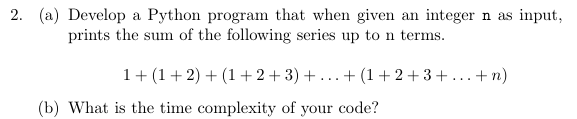
**Output:**

****

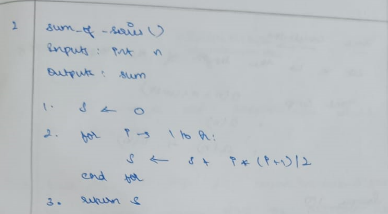
**Time Complexity:**

****

**Qn2:**

****

**Psuedo Code:**

****

**Source Code:**

def sumSeries(n):

    sum = 0

    for i in range(n):

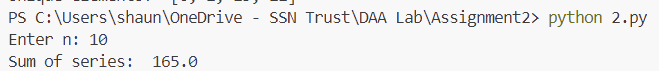
        sum += (i\*(i+1))/2

    return sum

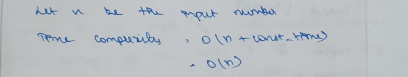
n = int(input("Enter n: "))

print("Sum of series: ", sumSeries(n))

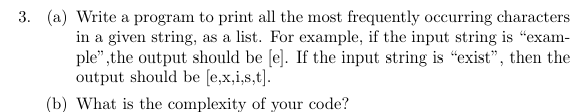
**Output:**

****

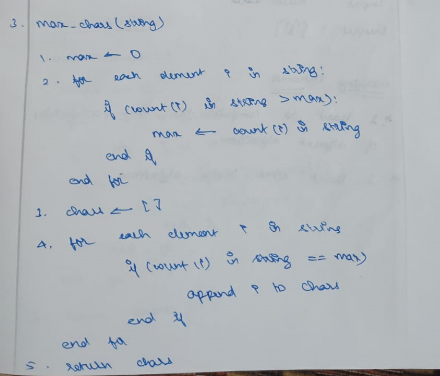
**Time Complexity:**

****

**Qn3:**

****

**Psuedo Code:**

****

**Source Code:**

def getMostFreq(str):

    max = 1

    d = {}

    for i in str:

        if i in d:

            count = d[i] + 1

            if count > max:

                max = count

            d[i] = count

        else:

            d[i] = 1

    max\_ = []

    for k,v in d.items():

        if v == max:

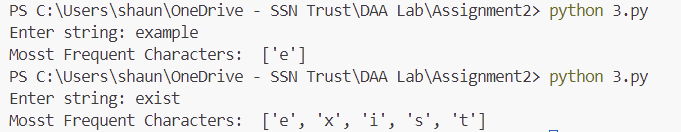
            max\_.append(k)

    return max\_

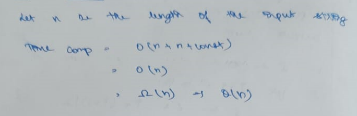
str = input("Enter string: ")

print("Mosst Frequent Characters: ", getMostFreq(str))

**Output**

****

**Time Complexity:**

****

**Learning Outcomes:**

* I learnt to analyse the time complexities of various algorithms
* I learnt how to implement various sorting and searching algorithms in Python