**LAB TASK #6**

**Submitted By :** Wajahat Safeer

**Roll Number :** 12407

**Class :** BSSE

**Section :** C

**Semester :** 3rd

**Submitted To :** Sir Jamal Abdul Ahad

**QUESTION #01**

**PROGRAM:**

def are\_anagrams(str1, str2):

Str1 = str1.replace(“ “, “”).lower()

Str2 = str2.replace(“ “, “”).lower()

If len(str1) != len(str2):

Return False

Freq\_table1 = {}

Freq\_table2 = {}

For char in str1:

Freq\_table1[char] = freq\_table1.get(char, 0) + 1

For char in str2:

Freq\_table2[char] = freq\_table2.get(char, 0) + 1

Return freq\_table1 == freq\_table2

String1 = “Listen”

String2 = “Silent”

Result = are\_anagrams(string1, string2)

If result:

Print(f”{string1} and {string2} are anagrams.”)

Else:

Print(f”{string1} and {string2} are not anagrams.”)

**QUESTION #02**

**PROGRAM:**

def count\_frequency(numbers):

Frequency\_dict = {}

For num in numbers:

Frequency\_dict[num] = frequency\_dict.get(num, 0) + 1

Return frequency\_dict

Integer\_list = [1, 2, 3, 4, 2, 3, 1, 4, 5, 6, 1, 2, 3, 4, 5]

Result\_dict = count\_frequency(integer\_list)

Print(“Input List:”, integer\_list)

Print(“Frequency Dictionary:”, result\_dict)

**QUESTION #03**

**PROGRAM:**

def longest\_subarray\_with\_sum(nums, K):

Sum\_index\_map = {} # Dictionary to store cumulative sums and their indices

Max\_length = 0

Current\_sum = 0

For I, num in enumerate(nums):

Current\_sum += num

If current\_sum == K

Max\_length = I + 1

Elif current\_sum – K in sum\_index\_map:

Max\_length = max(max\_length, I – sum\_index\_map[current\_sum – K])

If current\_sum not in sum\_index\_map:

Sum\_index\_map[current\_sum] = i

Return max\_length

Arr = [10, 5, 2, 7, 1, 9]

K = 15

Result = longest\_subarray\_with\_sum(arr, K)

Print(f”The length of the longest subarray with sum {K} is: {result}”)