Programming and Problem Solving, Assignment 1

Omer Sayem, 40226505

Department of Computer Science & Software Eng., Concordia University

COMP 6481 --- Fall 2023

Answer to the question 1

a)

```
Start
Declare and initialize for the list x, num1 and num2
Input x
Input num1
Input num2
N= length(x)
For I from 0 to N-1
        if x[I] == num1
        x[I] = num2
        if x[I] == num2
        x[I] = num1
End For
Print x
 End
```

b)

The time complexity of the above algorithm is O(n)

c)

The space complexity of this algorithm is O(n)

Answer to the question 2

a)

```
Start
Declare and initialize for the random string list x
Sort the elements of the list
Declare and initialize three string array list abc, num, cap
N= length(x)
For I from 0 to N-1
        If x[I] matches pattern [a-z]
                Add x[I] in abc
        Else if    x[I] matches pattern [0-9]
                Add x[I] in num
        Else if x[I] matches pattern [A-Z]
                Add x[I] in cap
End For
Print ( Concat abc , num , cap)
End
```

b)

Since the time complexity of the sorting function is O(nlog(n) and the loop time complexity is 0(n), the significant complexity for this algorithm is O(nlog(n)).

c)

The space complexity of this algorithm is O(n).

Answer to the Question 3

i)

```
Function isPrime(val)
    If val is less than or equal to 1
        Return false
    End If

    For i from 2 to val - 1
        If val is divisible by i
            Return false
        End If
    End For

    Return true
End Function

Start
    Declare an array 'arr' with given values
    Declare a target value 'target'
    Create an empty hashmap 'hm'

    For i from 0 to the length of 'arr' - 1
        If 'hm' contains 'arr[i]'
            If isPrime(arr[i]) is true and isPrime(target - arr[i]) is true
                Print "Two consecutive prime numbers with a sum of " + target +
                    " are " + arr[i] + " and " + (target - arr[i]) +
                " found at indices " + i + " and " + hm[arr[i]][1] + " respectively"
              Exit the loop
            End If
          Else
            Insert (target - arr[i]) as the key and [arr[i], i] as the value into 'hm'
          End If
        End For
      End
```

ii) Justification of my design:

The brute force solution to this problem could be for each of the items in the list, I could search for the subtracted number from the target in the rest of the element of the list. However, this design would take much time because the time complexity for the worst case of this algorithm is O (n^2 * val). Instead, I choose to hash mapping difference from the target of each element from the start, so that if I get the difference as a key existing in my HashMap, I can say that I found the two sum numbers after I have to check whether those two numbers are prime or not by a simple isPrime function which checks whether the numbers are divisible by any other number except 1 and itself.

iii) Time complexity of this algorithm is O (n * val), in for worst-case scenario the hash mapping would grow as numbers in the list, so O(n), and for the checkprime inside the loop would grow as a number gets bigger. Thus, combining both the time complexity would be O(n * val)

iv) Maximum stack growth of the hashmap of this algorithm is o(n)