COMP 6481 ASSIGNMENT 1

Dept. of Computer Science & Software Eng., Concordia University

COMP 6481 --- Fall 2022

Programming and Problem Solving

Assignment 1

Student Name: Shalvi Saxena

Student ID: 40220846

Part I

Question 1

a)

```
Algorithm swap(A, m)
        for ( i from 0 to (n-1))
                if (A[i] equals to m)
                        A[i] = A[i] ^ A[i+1]
                        A[i+1] = A[i+1] ^ A[i]
                        A[i] = A[i] ^ A[i+1]
                         i++
        return A
```

b) O(n)

c) O(N)

Question 2

```
// Assuming capital alphabets can be included and sequence is consonants + repeating characters + numeric values + vowels
Algorithm reArrange(S)
        DECLARE vowels: ARRAY[10] of char { 'A', 'a', 'E', 'e', 'I', 'i', 'O', 'o', 'U', 'u'}
        String arrConsonants = "", arrVowels = "", arrRepeating = "", arrNumeric = "";
        for ( i from 0 to (S.length - 1) )
                if ( arrConsonants contains S.charAt(i) )
                        arrRepeating = arrRepeating append S.charAt(i)
                        arrConsonants remove S.charAt(i)
                else if ( arrVowels contains S.charAt(i) )
                        arrRepeating = arrRepeating append S.charAt(i)
                        arrVowels remove S.charAt(i)
                else if ( vowels contains S.charAt(i) )
                        arrVowels = arrVowels append S.charAt(i)
```

```
            else if ( S.charAt(i) is Alphabetic )
                    arrConsonants = arrConsonants append S.charAt(i)
            else
                    arrNumeric = arrNumeric append S.charAt(i)
    return arrConsonants + arrRepeating + arrNumeric + arrVowels
```

a) O(n)

b) O(N)

Question 3

i)

```
Algorithm tetradicNumbers(arr)
    for( i from 0 to n)
        If (i not equals to 1 && i not equals to 0 && i not equals to 8)
            Return false;
    If ( n is equals to n.reverse() )
        Return true;
    Else
        Return false;

main()
    DECLARE arr: ARRAY[10] of int
    int maxDiff = -1
    Int indDiff = -1
    int num1 = 0
    int num2 = 0
    conNum[] = [0,0];
    For ( i from 0 to n-1 )
        For ( j from n-1 to i+1)
            If ( | arr[i] - arr[j] | equals to 10 && isTetradic (arr[i]) )
                If ( indDiff smaller than i-j )
                    indDiff = i-j
                     Num1 = arr[i]
                    Num2 = arr[j]
                    conNum.add ( arr[i] )
                    conNum.add ( arr[j] )
        If ( | arr[i] - arr[i+1] | > maxDiff && isTetradic (arr[i]) )
            conNum.add ( arr[i] )
            conNum.add ( arr[i+1] )
END
```

ii) I have tried solving it in the optimized way possible using array and the given constraints.

iii) Time complexity = O(n^2) as we iterating through the array n^2 time using loops.

iv) Stack size will be O(1) as we have a single stack and have not used recursion. So, the stack size will remain constant.