Student ID:
Name:
Signature:

# COMP 203 DATA STRUCTURES AND ALGORITHMS

## QUIZ 1 Date=09.11.2023 1:00 pm

## Total=100 points

**1.Array and recursion (30pt)**

a. Write a pseudocode or java code for **printTriangle(int[] A)** function that operates the following work: Given an array of integers, print a sum triangle from it such that the first level has all array elements. From then, at each level number of elements is one less than the previous level and elements at the level is the Sum of consecutive two elements in the previous level.

Example:

Input : A = {1, 2, 3, 4, 5}

```
Output :  [48]
          [20, 28]
          [8, 12, 16]
          [3, 5, 7, 9]
          [1, 2, 3, 4, 5]
```

Explanation :

```
[48]
[20, 28] -->(20 + 28 = 48)
[8, 12, 16] -->(8 + 12 = 20, 12 + 16 = 28)
[3, 5, 7, 9] -->(3 + 5 = 8, 5 + 7 = 12, 7 + 9 = 16)
[1, 2, 3, 4, 5] -->(1 + 2 = 3, 2 + 3 = 5, 3 + 4 = 7, 4 + 5 = 9)
```

**2.DLL (4*10=40 points)**

a. Write a psueudocode or java code for Node<E> class to implement node for a Doubly Linked list.

b. Write a psueudocode  or java code for DoublyLinkedList <E> class to implement node for a Doubly Linked list.

c. Write a psueudocode for a function with name InsertBetween(E first, E second, E addedValue) for Doubly Linked list that inserts a node having addedValue between the nodes that have value first and second.  Assume you have:

Student ID:
Name:
Signature:

**Example:** We have DLL head⇔ 2⇔3⇔1⇔4⇔5⇔8⇔tail

After InsertBetween(4,5,7)

head⇔ 2⇔3⇔1⇔4⇔7⇔5⇔8⇔tail

d. What is the average time complexity of InsertBetween(E first, E second, E addedValue) with Big-O?

**3. Recursion and Time Complexity (3*10=30 points)**

You have the following function:

```
public class ArraySum {
   public static int sumArrayElements(int[] array) {
      int sum = 0;
      for (int element : array) {
         sum += element;
      }
      return sum;
   }
   public static void main(String[] args) {
      int[] myArray = {1, 2, 3, 4, 5};
      int result = sumArrayElements(myArray);
      System.out.println("Sum of array elements: " + result);
   }
}
```

a. Now, you will write the binary recursion version of the function sumArrayElements. You may modify the parameters if it is necessary.

b. What is the average case complexity of the given function “sumArrayElements(int[] array)” with Big-O representation? Why.

c. What is the average case complexity of the function you implemented? Why?

d. Make a comparison between the given function and your function. Which one is more efficient in terms of Big-O complexity? Why?