VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

Faculty Name: C. Oswald

CSE2003 Digital Assignment 1: On ADT and Data Structures

Due: **11:59 PM 09 August 2020** (Sunday)

This assignment needs be done **individually**. It contributes **10%** to your final mark.

As always, please feel free to approach me by emails/calls, if you have any questions. My mobile number is 99413-13457. I'd be happy to help out. If I do not answer your call, please send me a SMS. I will call you back. Read the assignment instruction carefully. Good luck!

What to Submit

This is to be completed and submitted to Moodle. By the due date, submit one pdf file in the given naming format:

Individual Assignment: YourReg.No_DA1.pdf

Marking Criteria

Your submission will be marked using the following criteria.

- Showing good efforts through completed tasks in terms of designing logic and coding skills.
- Demonstrating creativity in providing unique individual solutions.
- Showing attention to details through a good quality assignment report.

Copying others contents will be noted, which will lead to penalty for both the donor and the recipient:). Late submissions will also fetch penalty. So, I encourage you to submit at the earliest and not to rush at the last moment.

Steps to proceed

Step 1. Read the question carefully and design your own logic for each of the questions given below. Each question carries 10 points.

Step 2. Code them in either C/C++/Java. If you use OOP, do not use standard template libraries. Please note: Clearly giving short comments in the code wherever necessary and proper indentation will fetch extra points.

Step 3. Test your code with sample test cases.

Step 4. In the Assignment report, for every question number, record your code and a snapshot of one test case.

Sl. No.	Question	Rubrics in points
1.	Using recursion solve, a) Given two strings, returns whether the first string is a subsequence of the second. For example, given **`tar`** and **`metaphor`**, you should return true, given **`col`** and **`lockdown`**, you should return false. **(5 points)** b) Given a list of n distinct elements and a number k, write a function that lists all k-element permutations of the list. **(5 points)**	Recursive condition – 3 Base case – 2
2.	For a Matrix *M*, write a function **int duplicatevalue(matrix* *M*, int value)** that returns 1 if a node with the value exists in the matrix. Return *0* if not. **(4 points)** Also write a function **bool magic(matrix* M)** to find out whether this matrix is a magic square or not. A magic square of order *n* is an arrangement of *n^2* numbers, usually distinct integers, in a square, such that the *n* numbers in all rows, all columns, and both diagonals sum to the same constant. **(6 points)**	Logic – 5 Use of pointers in coding - 5
3.	Your friend has given you a sorted doubly linked list of distinct nodes(no two nodes have the same data) and a value *x*. He has challenged you to count the triplets in the list that product up to a given value *x*. Can you help your friend.	Logic – 6 Use of pointers in coding - 4
4.	In computer science and mathematics, the Josephus Problem (or Josephus permutation) is a theoretical problem. Following is the problem statement: There are *n* people standing in a circle waiting to be executed. The counting out begins at some point in the circle and proceeds around the circle in a fixed direction. In each step, a certain number of people are skipped and the next person is executed. The elimination proceeds around the circle (which is becoming smaller and smaller as the executed people are removed), until only the last person remains, who is given freedom. Given the total number of persons *n* and a number *k* which indicates that *k-1* persons are skipped and *k*th person is killed in circle. The task is to choose the place in the initial circle so that you are the last one remaining and so survive. For example, if *n* = 5 and *k* = 2, then the safe position is 3. Firstly, the person position 2 is killed, then person at position 4 is killed, then person at position 1 killed. Finally, the person at position 5 is killed. So the person at position survives. If *n* = 7 and *k* = 3, then the safe position is 4. The persons at positions 6, 2, 7, 5, 1 are killed in order, and person at position 4 survives. *Use Linked list representation to implement you logic.*	Use of a suitable ADT – 2 Logic – 4 Linked List Representation, use of pointers in coding – 4

END OF ASSIGNMENT ONE