MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
(A constituent unit of MAHE, Manipal)

III SEMESTER B. TECH (COMPUTER SCIENCE & ENGINEERING)
IN SEMESTER EXAMINATION, DECEMBER 2021
SUBJECT: DATA STRUCTURES & APPLICATIONS (CSE 2152)
REVISED CREDIT SYSTEM

Date of Exam: 14/12/2021 **Time: 80 + 10 Minutes** **MAX. MARKS: 20**

Note: Answer ALL the questions.

SET-1

1	Given the following declarations:
	int num[10] = {23, 3, 5, 7, 4, -1, 6, 12, 10, -23};
	int i = 2;
	int j = 4;
	Write the values of the following expressions.
	a) *(num + 2)
	b) *(num + i + j)
	c) *(num + *(num + 1))
	d) *(num + j)
	e) *(num + i) + *(num + j)
	f) *(num + i) * *(num + j)
		3

2	Write a recursive function with the function declaration int rcntVowels(char *s); to find and return number of vowels (only lower case) in a string passed as parameter using static variable. Write main() to read a string from the keyboard, call the function, and display the count. To access element of the array use dereference operator(*) only.	3
3	Consider a **Circular Queue** implemented using a dynamic array of size 3 with array doubling. Show the status of the queue using the table below for each of the following operations. Show appropriate messages whenever required. [Table shows entries for the first 2 operations] i. Begin ii. Insert 5 iii. Insert 10 iv. Delete v. Insert 20 vi. Delete vii. Insert 30 viii. Delete ix. Delete x. Delete xi. Insert 5 xii. Insert 10 xiii. Insert 15	4

xiv. Insert 20

S.No	Operation	Element Inserted/Deleted/ Message	Front	Rear	Array
1	Begin	Create Array	0	0	0 1 2 [ ] [ ] [ ]
2	Insert	5	0	1	0 1 2 [ ] [5] [ ]

4	Write a function ***struct node** * **insert_order (struct node *first, int reg_no)*** which inserts a new node into a singly linked list without header node(list may be initially empty) in the ascending order of the registration number and returns the new list. The node structure of the linked list is as given below:
	
	struct node {
	int registration;
	struct node* next;
	};
		3

5	Given a singly linked list without header node consisting of nodes in the ascending order of registration number, write a function ***void Remove_Duplicates(struct node *first)*** which deletes the nodes with duplicate registration numbers, retaining the first occurrence in the list.
	The node structure of the linked list is as given below:
	
	struct node {
	int registration;
	struct node* next;
	};
		3

6	Given two **Circular Doubly Linked Lists** representing long binary numbers, write a function **Add (),** to add the two binary numbers and return a **Circular Doubly Linked List** representing the sum. The prototype of the Add function is as follows: **Nodeptr Add(Nodeptr A, Nodeptr B);**	4