Reg. No.										

MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL

A Constituent unit of MAHE, Manipal

III SEMESTER B.TECH. (COMPUTER SCIENCE & ENGINEERING)

END SEMESTER EXAMINATIONS, JAN 2021

SUBJECT: DATA STRUCTURES & APPLICATIONS [CSE 2152]

REVISED CREDIT SYSTEM
(05/03/2021)

Time: 3 Hours MAX. MARKS: 50

Instructions to Candidates:

- Answer **ALL FIVE** questions.
- Missing data may be suitably assumed.

1A.	Create a structure as shown: struct DISTANCE {int feet; float inch;}; In main(), create pointers to the above struct and allocate memory using dynamic memory allocation. Read in and store values of distances d1and d2. Find the sum of the two distance values using function: void addDist (struct DISTANCE *d1, struct DISTANCE *d2, struct DISTANCE *result); Display the value of result in main().	**5**
1B.	Write a complete program to check whether the given string is a palindrome using recursion.	**3**

1C.	Write the output of the following program:
	#include<stdio.h>
	void main() {
	int num[5] = {3, 4, 6, 2, 1};
	int *p = num;
	int* q = num+2;
	int* r = &num[1];
	printf("\n%d %d", num[2], *(num+2));
	printf("\n%d %d", *p, *(p+1));
	printf("\n%d %d", *q, *(q+1));
	printf("\n%d %d", *r, *(r+1));
	}
		2

2A.	Write a C program(Menu driven) to implement 'n' stacks using a single 1-D array containing 'm' locations with following prototypes,
	i) push(int i, int item, STACK *S); //pushing an item on ith stack
	ii) pop(int i, STACK *S);//poping an item from ith stack
	iii) display(S) // displaying all 'n' stack contents
	For the STACK structure, the members boundary[i] and top[i] represents boundary and top respectively for the ith stack along with element represented by other member. While pushing if the particular stack is full, and if there is space available elsewhere in the array of 'm' locations, it should shift the stacks so that space is allocated to the full stack.
		5

2B.	Write a program to convert the given prefix expression to postfix equivalent	**3**

<table>
<tr><td>2C.</td><td>Write C functions for the following operations on ordinary queue using array with proper validation: a) insert b) Delete</td><td>2</td></tr>
<tr><td>3A.</td><td>Write a function “poly_add (poly_pointer, poly_pointer)” to add two polynomials represented by two singly linked lists, A and B and return the new polynomial, C.</td><td>5</td></tr>
<tr><td>3B.</td><td>Give any six differences between Array and Linked List.</td><td>3</td></tr>
<tr><td>3C.</td><td>Write code to implement stack operations (push and pop) using singly linked list.</td><td>2</td></tr>
<tr><td>4A.</td><td>Consider all five possible binary search trees for the key set (a1, a2, a3) = (Creta, City, Punto). Find the optimal binary search tree
(i) with equal probabilities, pi = qi = 1/7 for all i and j
(ii) with p1=0.05, p2=0.5, p3=0.1, q0=0.05, q1=0.1, q2=0.15, q3=0.05</td><td>5</td></tr>
<tr><td>4B.</td><td>Given a prefix expression, write a function in C to create an expression tree.</td><td>3</td></tr>
<tr><td>4C.</td><td>Discuss with an example for each, the storage representations of binary trees using arrays and linked representations.</td><td>2</td></tr>
<tr><td>5A.</td><td>Write an iterative function in C to traverse a binary tree in post-order traversal showing the node structure of the stack.</td><td>4</td></tr>
<tr><td>5B.</td><td>Write function “struct node* findunion (struct node *LLOne, struct node *LLTwo)” which returns the union of two lists represented using singly linked list.</td><td>4</td></tr>
<tr><td>5C.</td><td>Convert the given infix expression, A + B - C * D / (E – F + G) *H
to prefix, by showing the Scanned Symbol, Action Taken, Stack Contents and Current Prefix in the form of a table, as shown below:
<table>
<tr><td>Scanned symbol</td><td>Action Taken</td><td>Stack Contents</td><td>Current Prefix</td></tr>
</table></td><td>2</td></tr>
</table>