

	<ul style="list-style-type: none"> • Undo • Redo 		
Q4.	<p>A software company designs a task scheduler that manages jobs with varying priorities and requirements:</p> <ul style="list-style-type: none"> • Regular tasks are added to the rear of the task queue and processed in the order they arrive (FIFO). • High-priority tasks must be added to the front of the queue to be handled as soon as possible. • Occasionally, some low-priority tasks must be removed from the rear of the queue to free up resources. <p>a) Explain how a deque (double-ended queue) data structure is suitable for this task scheduling system. Mention the operations that will be used to add or remove tasks from both ends.</p> <p>b) Given an initially empty deque, show the queue state after performing each of the following operations in sequence:</p> <ul style="list-style-type: none"> • Add regular Task A to the rear • Add regular Task B to the rear • Add high-priority Task C to the front • Remove one task from the rear • Add high-priority Task D to the front • Remove one task from the front 	4	CO2
Q5.	<p>The professors at MANIT have chosen a special grading system for their 10 top students in the Computer Science and Engineering (CSE) branch. Each student receives a unique score reflecting their performance throughout the semester. To ensure fairness and creativity, the professors decide to use a computational method to determine each student's academic rank. Here's how their innovative ranking system works:</p> <ul style="list-style-type: none"> • All student scores are sequentially stored as nodes in a Binary Search Tree (BST), mirroring the order in which results are processed: 14, 35, 7, 22, 3, 9, 50, 8, 40, 60. • For each student, their computed "rank" is defined as the sum of their own score plus all the scores that are higher (greater) among the other students in the BST. This method uniquely accounts for how close a student's score is to the top. • Professors want to analyze the distribution of ranks using three types of Depth First Traversal (Preorder, Inorder, Postorder). This helps them understand student performance from different perspectives. <p>a) Build the rank tree 'R'.</p> <p>Hint: For each student, replace their original score with their computed rank following the described grading normalization.</p> <p>b) Print the ranks of the students in Preorder, Inorder, and Postorder traversals, simulating different report formats the professors can use for their analysis.</p>	4	CO3

*****END*****

Name of the Student.....

Scholar Number. 2011201129

MAULANA AZAD NATIONAL INSTITUTE OF TECHNOLOGY BHOPAL
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

EXAMINATION: MID TERM

MONTH and YEAR: October 2025

Course: B.Tech

Semester: III

Subject Code: CSE 24211

Subject Name: Data Structures and Algorithm

Maximum Marks: 20

Duration: 60 mins

Date: 03/10/2025

Time: 09:30 AM to 10:30 AM

Note: Attempt all questions. Assume missing data (if any) and mention the assumptions.

Q. No.	Questions	Marks	COs
Q1.	<p>a) Given a sorted array of 0's and 1's where all 0's appear before 1's, write an algorithm to find the index of the first occurrence of 1. Solve the problem using both linear search and binary search, and compare their time and space complexities.</p> <p>b) Evaluate the value of func(4,7) for the given recursive algorithm, and illustrate its execution using the function call stack.</p> <pre> algorithm func(x, y) if (x > y) then return -1 elseif (x equal y) then return 1 else return (x * func(x + 1, y)) end if end func </pre>	4	CO2 & CO3
Q2.	<p>a) Write a function to concatenate two circular doubly linked lists.</p> <p>b) Write a function to split a Linear Linked List into two Linear Linked Lists on the basis of data values, such that the even numbers are stored in the first list and the odd numbers are stored in the second list.</p>	4	CO1 & CO2
Q3.	<p>An office worker is using a word processor that supports both 'Undo' and 'Redo' functionalities for editing documents. Each editing action (typing, deleting, formatting) is recorded to allow reverting or reapplying changes efficiently.</p> <p>a) Identify which data structure(s) can be used to implement the Undo and Redo features in such a word processor. Explain briefly why these data structure(s) are suitable for this purpose.</p> <p style="margin-left: 40px;">Two empty stacks</p> <p>b) Suppose the employee performs these actions in sequence, show the status of both Undo data structure and Redo data structure:</p> <ul style="list-style-type: none"> • Types "Section 1" • Types "Section 2" • Types "Section 1" • Undo 	4	CO2