

USN 

--	--	--	--	--	--	--	--	--	--

Course Code 

<b>22CS/AI/CA/CD/34</b>
-------------------------

Third Semester B.E. Degree Examinations, September 2024

**DATA STRUCTURES AND APPLICATIONS**

(Common to CSE, AIML, CSE- AI, CSE- DS)

Duration: 3 hrs

Max. Marks: 100

Note: 1. Answer any FIVE full questions choosing ONE full Question from each Module.  
2. Missing data, if any, may be suitably assumed

<u>Q.No</u>	<u>Question</u>	<u>Marks</u>	<u>(RBTL:CO:PI)</u>
<b>Module-1</b>			
1.	a. List and explain different types of data structures.	05	(2 :1: 1.7.1)
	b. Define pointer. Write a C program to read and find sum of two integer numbers using pointer.	05	(3:1: 1.7.1)
	c. List any two disadvantages of arrays. Write a C function to perform insertion and deletion of an element at the specified position in a given array.	10	(3:1: 1.7.1)
<b>OR</b>			
2.	a. Write an algorithm to add two given polynomials using array of structures.	06	(2:1: 1.7.1)
	b. Define sparse matrix. Write the following matrix in triple form and find its transpose of the matrix	08	(3:1: 1.7.1)
	$\begin{bmatrix} 15 & 0 & 0 & 22 & 0 & -15 \\ 0 & 11 & 3 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 & -6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 91 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 28 & 0 & 0 & 0 \end{bmatrix}$		
	c. Write a C program to find the average of three internal marks using structures.	06	(3:1: 1.7.1)
<b>Module-2</b>			
3.	a. Write an algorithm to perform the following operations (i) To convert infix to postfix expression (ii) To Evaluate the given postfix expression	10	(3:2: 2.8.1)
	b. List different types of recursions. Write a recursive function (i) To find sum of N natural numbers. (ii) To find factorial of a given number.	10	(3:2: 2.8.1)
<b>OR</b>			
4.	a. Write a C program to implement stack data structure using arrays.	10	(3:2: 2.8.1)
	b. Write a C program to implement circular queue data structure using arrays.	10	(3:2: 2.8.1)
<b>Module-3</b>			
5.	a. Write an algorithm to perform the following operations on singly linked list. (i) Inserting an element at the beginning of the linked list. (ii) Deleting an element at the ending of the linked list.	08	(3:3: 1.7.1)

Note: (RBTL - Revised Bloom's Taxonomy Level: CO - Course Outcome: PI- Performance Indicator)

- b. Write a diagrammatic linked representation of given sparse matrix. **04** (3:3: 1.7.1)

$$\begin{bmatrix} 2 & 0 & 0 & 0 \\ 4 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 \\ 8 & 0 & 0 & 1 \\ 0 & 0 & 6 & 0 \end{bmatrix}$$

- c. Write a C program to implement stack data structure using singly linked list. **08** (3:3: 1.7.1)

**OR**

6. a. List and explain different types of linked lists. **10** (3:3: 1.7.1)

- b. Write an algorithm to add two polynomials using linked list. **06** (3:3: 1.7.1)

- c. Write a C function to perform the following operations on doubly linked list. **04** (2:3: 1.7.1)

- (i) Counting total number of nodes in a linked list.  
(ii) Merging two given linked list.

**Module-4**

7. a. Define binary tree. Write the C routine for the following operations **10** (3:4: 1.7.1)

- (i) Inorder Traversal (ii) Preorder Traversal (iii) Postorder Traversal

- b. List different properties of binary search tree, Construct binary search tree for the given data **1,2,3,4,5,6,7,8,9,10,11,12,12,13,14,15** and show the array and linked representation of the same. **10** (3:4: 1.7.1)

**OR**

8. a. List the disadvantages of binary tree. Demonstrate, how we can use threaded binary trees. **10** (3:4: 1.7.1)

- b. Write a C function to perform the following operations on binary search tree. **10** (3:4: 1.7.1)

- (i) Searching an Element using Recursion  
(ii) Inserting a node in a Binary search tree

**Module-5**

9. a. Write an algorithm to perform the following graph traversals **10** (3:5: 2.8.1)

- (i) Depth First Search (DFS)  
(ii) Breadth First Search (BFS).

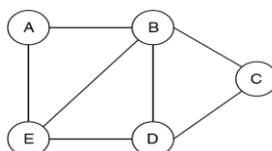
- b. Define collision. Explain how to resolve the collision in hashing technique. **05** (2:5: 2.8.1)

- c. List and explain the different hashing functions with examples. **05** (2:5: 2.8.1)

**OR**

- 10 a. Write a C program to implement Breadth First Search (BFS) graph traversal. **10** (3:5: 2.8.1)

- b. Define graph. Give the adjacency matrix and adjacency list for the graph given below. **06** (3:5: 2.8.1)



- c. Define (i) Collision (ii) Hash Table (iii) Hash Function (iv) Linear Probing **04** (2:5: 2.8.1)

\*\* \*\* \*