B.M.S. College of Engineering, Bengaluru-560019

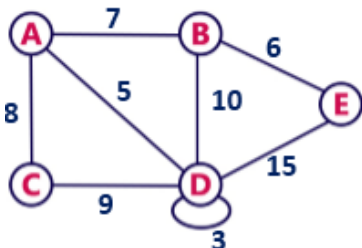**Autonomous Institute Affiliated to VTU**

January / February 2025 Semester End Main Examinations

Programme: B.E. **Semester: III**

Branch: Computer Science and Business Systems **Duration: 3 hrs.**

Course Code: 23BS3PCDSA **Max Marks: 100**

Course: Data Structures and Applications

Instructions: 1. Answer any FIVE full questions, choosing one full question from each unit.
2. Missing data, if any, may be suitably assumed.

			UNIT - I	CO	PO	Marks
1	a)		Analyze the advantages and disadvantages of using a linked list over an array for data storage.	CO1	PO1	**6**
	b)		You are asked to store **employee details** in a **structure**, where each employee has a name, age, and salary. You need to: • Dynamically allocate memory for an array of 5 employee structures. • Accept input for each employee's details. • Display the details using pointers.	CO1	PO3	**8**
	c)		You are developing a library management system where books are stored in a linked list. Each node in the linked list represents a book and contains the following information: Book ID (integer), Book Title (string), Author (string). Design a function to search for a book in the linked list using Book Title. The function should take two parameters i.e. pointer to the linked list and key book and it should return 1 on successful search and 0 on unsuccessful search.	CO2	PO3	**6**
			OR			

2	a)		Examine the following code snippet carefully. Analyze its functionality, explain its purpose, and predict the output of the program
			#include <stdio.h>
			#include <stdlib.h>
			int main() {
			  int *ptr, i;
			  ptr = (int *)malloc(5 * sizeof(int));
			  for (i = 0; i < 5; i++) {
			    *(ptr + i) = i + 1; }
			  ptr = (int *)realloc(ptr, 3 * sizeof(int));
			  for (i = 0; i < 3; i++) {
			    printf("%d ", *(ptr + i)); }
			  free(ptr);
			 return 0;}
				CO3	PO2	**6**

		b)	Implement the following functions in C for a singly linked list of integers to perform the following Operations.
			
			   a) Insert a new node at the given position of the linked list.
			   b) Deleting a node with a specified key value.
			   c) Traverse through each node of the linked list
				CO2	PO3	**8**

		c)	Discuss the advantages of using structures in programming and identify situations where using a structure is preferable over individual variables to store related data.	CO1	PO1	**6**
			UNIT - II			

3	a)		Write a C routine to implement the following operations on doubly linked list.
			
			  i. Define a structure to store node of Doubly linked list
			  ii. Create a node and insert a node at the middle of the list
			  iii. delete a node at given position
				CO2	PO3	**8**

	b)		You have an empty stack. Perform the following operations:
			  • Push 5, 10, and 15 onto the stack.
			  • Pop the top element from the stack.
			  • Push 20 onto the stack.
			  • Peek the top element of the stack.
			  • Pop all elements from the stack and print them in the order they are popped.
			What will be the sequence of stack states? What are the elements popped from the stack?
				CO3	PO2	**7**

	c)		Write a C function to reverse the linked list.	CO2	PO3	**5**
			OR			
4	a)		Write a C program to implement a stack data structure using an array. The program should support the following operations: push, pop and peek.	CO2	PO3	**8**
	b)		Given two polynomials: Polynomial A: $3x^3 + 2x^2 + 5x + 7$ Polynomial B: $4x^3 + 6x^2 + 1x + 3$ Represent these polynomials using linked lists and provide C code to add two polynomials which takes pointer to each polynomial as parameter and returns the resultant polynomial.	CO3	PO3	**7**
	c)		Given a scenario where data needs to be processed repeatedly (e.g., buffering in music players), justify the use of a circular linked list over a doubly linked list or a queue.	CO1	PO2	**5**
			UNIT - III			
5	a)		Consider the following Postfix expression 3 4 + 5 2 * - 6 2 / + Show the step-by-step process of how the expression is evaluated, including the stack's state after each operation.	CO3	PO2	**5**
	b)		Write a pseudo code for conversion of infix expression into postfix expression. Consider the following infix expression	CO3	PO2	**8**

			(A + B) * (C - D) / E apply above pseudo code to convert the given infix expression into postfix expression.			
		c)	Trace the function call stack when calculating the Factorial for n = 3 using recursion. Show the stack's state during the execution of factorial (3)	*CO4*	*PO2*	**7**
			OR			
	6	a)	Implement a priority queue using a linked list in C. The queue should support the following operations: • Insert an element based on its priority (higher priority elements come first). • Remove the element with the highest priority. • Display the elements of the priority queue in order of priority.	*CO2*	*PO3*	**10**
		b)	Discuss the applications of queues in real-life scenarios and Computing.	*CO1*	*PO1*	**5**
		c)	A linear queue is being used to manage customer orders in a restaurant system. As customer orders are processed and dequeued, the system becomes less efficient. Analyze the issue that arises with a linear queue in this scenario. Based on your analysis, suggest an efficient method or alternative data structure that could improve the performance of the queue when elements are dequeued.	*CO2*	*PO2*	**5**
			UNIT - IV			
	7	a)	Consider the following binary tree.	*CO2*	*PO2*	**5**
		b)	Provide recursive C functions to implement Inorder, Preorder, and Postorder traversals for the given binary tree. Traverse the binary tree given in Fig. 7(a) using Inorder, Preorder, Postorder traversals and provide the output for each traversal.	*CO3*	*PO3*	**8**
		c)	Write a C function to implement the insertion operation in a binary search tree (BST). Demonstrate how the tree is updated by inserting the following values into the BST: 50, 30, 70, 20, 40, 60, 80	*CO3*	*PO3*	**7**
			OR			
	8	a)	Define binary tree. Analyze the following binary tree and determine its properties:	*CO2*	*PO2*	**5**

For row 7 a), the cell content includes:

```
    10
   / \
  15  25
 /     \
2      30
/ \    / \
34 20 70 80
   Fig. 7(a)
```

Show the array and linked list representation of this binary tree.

			• Number of levels
			• Height of the tree
			• Number of nodes at each level
			• Is this tree a full binary tree?
			
			    1
			   / \\
			  2 3
			 / \\ \\
			4 5 6
			Fig. 8(a)			

		b)	Write a C function to delete a node with a given value from the BST. Perform the deletion of node 2 from the binary tree provided in Fig. 8(a) and update the tree. Show the updated tree structure after deletion.	*CO3*	*PO3*	**8**
		c)	Discuss at least two real-world applications where trees are used. And Given the following Inorder and Postorder traversals of a binary tree: Inorder Traversal: D, B, E, A, F, C, G Postorder Traversal: D, E, B, F, G, C, A Construct the binary tree from the provided Inorder and Postorder traversals.	*CO3*	*PO2*	**7**
			UNIT - V			
	9	a)	Discuss need of Balance tree and Construct an AVL tree by inserting the following sequence of numbers: 1,2,3,4,5,6,7,8. For each insertion, illustrate the tree and apply the necessary rotations to maintain balance.	*CO2*	*PO2*	**8**
		b)	Discuss Red-Black Trees. Mention the rules that every Red-Black tree should follow.	*CO2*	*PO1*	**7**
		c)	Mention the advantages and disadvantages of Splay Trees.	*CO2*	*PO1*	**5**
			OR			
	10	a)	Assume a splay tree of six nodes and demonstrate Splaying on any chosen leaf node.	*CO2*	*PO2*	**8**
		b)	Write adjacency matrix and adjacency list representation of graph given below: Fig. 10(b)	*CO2*	*PO2*	**6**
		c)	Explain the properties of a Red-Black Tree and discuss the advantages and disadvantages of using Red-Black Trees over other types of self-balancing binary search trees	*CO2*	*PO2*	**6**

******