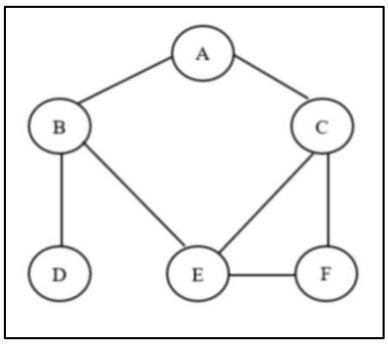
(Affiliated to University of Mumbai)	***End Semester Examination (R-24/R-25) SH 2026 Answer Key with marking scheme***
Branch: ECS	Course: Data Structures
Year/ Semester: II / III	Course code: ECC302
Time: 03 hours	Marks: 80

Note: 1. All questions are compulsory.
2. Figures to right indicate full marks.
3. Assume suitable data wherever necessary.

		Marks
Q. 1	All questions are compulsory.	
A.	Write a C program to implement a linear queue using an array and perform ENQUEUE and DEQUEUE operations. Solution: Queue array, front and rear initialization – 1m Correct ENQUEUE operation - 1.5m Correct DEQUEUE operation - 1.5m Overflow/underflow handling and demonstration/output – 1m	05
B.	Write a C program to create a singly linked list and insert a new node at a specified position. Display the list before and after insertion. Solution: Node structure and memory allocation – 1m Creation and linking of the initial list – 1m Correct insertion logic, including position 1 – 2m Display before and after insertion – 1m	05
C.	Explain the properties of a binary tree and differentiate between full and complete binary trees with suitable examples. Solution: Definition of binary tree – 1m Any two correct properties – 2m Difference between full and complete binary trees – 1m Suitable examples/diagrams – 1m	05
D.	Analyze the insertion of the keys 30, 20, 10 into an AVL tree. Draw the tree after insertion and demonstrate the rotation required to balance it. Solution: Correct tree after inserting 30 and 20 – 1m Tree after inserting 10 showing LL imbalance – 1m Identify LL imbalance and required right rotation – 1m Correct final balanced tree – 1m Explain balance factor or why rotation is needed – 1m	05
E.	Describe any five applications of data structures in real-world computing systems with suitable examples. Five valid applications with suitable examples (1 mark each) – 5m	05
F.	Represent the polynomial $P(x) = 5x^3 + 4x^2 + 2x+1$ using a linked list. Draw the linked-list representation showing the coefficient, exponent, and link fields. Explain node fields: coefficient, exponent, link – 1m Correct coefficient/exponent pairs – 2m Correct linked-list diagram in descending exponent order - 1.5m NULL link at the final node - 0.5m	05

Q.2A.	A college campus has several buildings connected through a computer network, represented by the given undirected graph. The main server is located at vertex A. The network administrator wants to visit all reachable buildings using Depth First Search (DFS) and Breadth First Search (BFS). Using the given graph, starting from vertex A: a. Apply DFS and BFS to determine the traversal sequence for each technique. b. Write the algorithms for DFS and BFS. c. Compare the working principles of DFS and BFS.  Solution: Correct DFS traversal sequence for the supplied graph – 2m Correct BFS traversal sequence for the supplied graph – 2m DFS algorithm – 2m BFS algorithm – 2m Comparison of working principles, data structures, or applications – 2m	10
	OR	
Q.2B.	A software project contains tasks with prerequisite dependencies represented by a directed acyclic graph (DAG). Apply topological sorting to determine a valid task execution order. Demonstrate the algorithm and state its applications in project scheduling. Solution: Define topological sorting and DAG – 2m Explain algorithm/steps, including in-degree or equivalent method – 3m Demonstrate a valid ordering for the given dependencies – 3m Applications in project scheduling – 2m	10
Q.3A	Explain the algorithm for evaluating a postfix expression using a stack, with a suitable example. Explain the stack-based evaluation algorithm – 3m Correct handling of operands and operators, including operand order – 2m Suitable worked example with stack states - 3m Correct final result and complexity or conclusion - 2m	10
	OR	
Q.3B	Discuss the working principles of circular queues, priority queues, and deques with suitable examples. Differentiate between the operations of a deque and a simple queue. Circular queue: principle, operation, and example – 3m Priority queue: principle and example – 2m Deque: principle and example – 2m Difference between deque and simple queue - 3m	10

Q.4A	Apply linear search and binary search to locate the key 45 in the following sorted array. Illustrate the search steps and compare the number of comparisons required. Array: 10, 20, 30, 40, 45, 50, 60, 70, 80. Solution: Linear search steps – 2m Correct linear search result: 5 comparisons – 1m Binary search steps – 3m Correct binary search result: 1 comparison – 1m Comparison of number of comparisons and efficiency – 3m	10
	OR	
Q.4B	Explain the division method of hashing. Insert the following keys into a hash table of size 10 using linear probing and chaining for collision resolution. Show the final hash table for each technique. Keys: 23, 43, 13, 27, 37, 33, 53, 18. Division method and correct hash calculations -2m Linear probing: collision handling and final table – 4m Chaining: correct chains and final table - 3m Brief comparison of the two techniques – 1m	10
Q.5A	Develop a C program to perform the following operations on a doubly linked list: a. Insert a node at the beginning. b. Delete a node from the end. c. Display the elements of the list in forward and reverse order. Solution: Correct node structure with prev and next pointers – 1m Insert a node at the beginning – 2m Delete a node from the end, including empty/single-node cases – 3m Display in forward order - 1.5m Display in reverse order - 1.5m Correct main()/demonstration and overall program logic – 1m	10
	OR	
Q.5B	Develop a C program to implement a stack using a singly linked list. Perform PUSH and POP operations for the sequence PUSH(10), PUSH(20), PUSH(30), POP(), PUSH(40), and POP(), and display the stack after each operation. Solution: Node structure and top pointer – 1m Correct PUSH operation – 2m Correct POP operation with underflow handling – 2m Execute the specified sequence correctly – 2m Display stack after each operation – 2m Overall program structure and output – 1m	10
Q.6A	Analyze the expression tree for the postfix expression AB+C*DE/-. Draw the expression tree, determine its in-order, preorder, and post-order traversals, and analyze how the tree structure represents the order of operations in the expression. Solution: Correct expression tree – 3m In-order traversal – 2m Pre-order traversal - 1.5m Post-order traversal - 1.5m Explain how the tree represents order of operations – 2m	10

	OR	
Q.6B	Analyze the construction of a Huffman tree for the characters with frequencies A: 5, B: 9, C: 12, D: 13, E: 16, and F: 45. Show the steps involved in combining the nodes, generate the Huffman codes for all characters, calculate the total number of bits required to encode the given frequencies, and analyze how character frequencies influence code lengths. Solution: Correct sequence of combining the two smallest frequencies – 3m Correct Huffman tree – 2m Valid Huffman codes for all characters – 2m Correct total weighted bit count – 2m Explain the relationship between frequency and code length – 1m	10
	*********** All the Best***********	