SRM VALLIAMMAI ENGINEERING COLLEGE

(An Autonomous Institution)

SRM Nagar, Kattankulathur – 603 203

DEPARTMENT OF INFORMATION TECHNOLOGY

(Common to Computer Science and Engineering, Artificial Intelligence and Data Science & Cyber Security)

QUESTION BANK

III SEMESTER

IT3361 - PROGRAMMING AND DATA STRUCTURES

Regulation – 2023

Academic Year 2025-2026 (ODD Semester)

Prepared by

Dr. S. Jeyalakshmi, Associate Professor / AI-DS

Dr. A. Vidhya, Assistant Professor / CSE

Ms. V. Prema, Assistant Professor / CSE

Ms. T. Sathya, Assistant Professor / CYS

Ms. M. Abinaya, Assistant Professor / AI&DS

Ms. M. Mohanapriya, Assistant Professor / CSE

Ms. N. J. Nithya Nandhini, Assistant Professor / IT

Ms. S. Priya, Assistant Professor / IT

SRM VALLIAMMAI ENGINEERING COLLEGE

(An Autonomous Institution)

SRM Nagar, Kattankulathur – 603 203.

DEPARTMENT OF INFORMATION TECHNOLOGY

Common to Computer Science and Engineering, Cyber Security and Artificial Intelligence & Data Science

QUESTION BANK

SUBJECT   : **IT3361 PROGRAMMING AND DATA STRUCTURES**
SEM / YEAR : **III Sem / II Year**

	UNIT I - INTRODUCTION TO C PROGRAMMING		
Data Types – Variables – Operators – Expressions and Statements – Conditional Statements – Arrays – Single and Multi-Dimensional Arrays.		
	PART-A		
Q.No.	**Question**	**Level**	**Competence**
1	Define data type in C.	BTL2	Remember
2	What is a variable?	BTL1	Remember
3	List any four data types in C.	BTL1	Remember
4	Define an operator with example.	BTL1	Understand
5	What is the purpose of an assignment operator?	BTL1	Remember
6	Differentiate between an expression and a statement.	BTL2	Understand
7	What is an arithmetic expression in C? Give an example.	BTL2	Remember
8	Name two conditional statements in C.	BTL2	Remember
9	Explain the use of the if-else statement.	BTL1	Remember
10	What is the difference between if and switch statements?	BTL2	Understand
11	Define an array.	BTL2	Remember
12	What is the syntax to declare an array in C?	BTL1	Remember
13	List two advantages of using arrays.	BTL1	Understand
14	How is a 1D array initialized? Give an example.	BTL2	Remember
15	Define a multi-dimensional array.	BTL1	Remember
16	Provide an example of a 2D array declaration.	BTL1	Understand
17	Describe how to access elements in a 2D array.	BTL2	Remember
18	Explain the difference between a single and multi-dimensional array.	BTL2	Understand
19	What is the default value of uninitialized local variables in C?	BTL1	Understand

20	Define constant and how it differs from variable.	BTL1	Remember
21	What is the role of the sizeof operator in C?	BTL2	Remember
22	Write the syntax and use of the ternary operator.	BTL2	Remember
23	Explain the concept of precedence in operators.	BTL2	Remember
24	What are logical operators? Give examples.	BTL1	Understand

PART-B

Q.No.	Question	Marks	Level	Competence
1	Write a C program to read two numbers and perform all arithmetic operations (addition, subtraction, multiplication, division). Display the results using formatted output.	16	BTL3	Analyze
2	Develop a program that uses relational and logical operators to compare two integer values and prints the appropriate message.	16	BTL3	Analyze
3	Implement a program to demonstrate the use of if, if-else, and else-if ladder with a real-life example like grading a student.	16	BTL3	Apply
4	Write a C program to input and store 10 numbers in a single-dimensional array, then print the maximum and minimum numbers.	16	BTL3	Evaluate
5	Write a program that initializes and displays a 3x3 matrix using a two-dimensional array.	16	BTL3	Create
6	Analyze the difference between various types of operators in C (arithmetic, logical, relational, assignment) with suitable examples.	16	BTL4	Evaluate
7	Differentiate between single-dimensional and multi-dimensional arrays. Provide a scenario where each is better suited.	16	BTL4	Apply
8	Examine the behavior of switch vs. if-else conditional constructs with a program that uses both for the same logic.	16	BTL4	Evaluate
9	Identify errors in a given code snippet involving variables, data types, and conditional statements. Suggest corrections and explain the logic.	16	BTL4	Create
10	Analyze the memory layout of a multi-dimensional array and explain how the elements are stored and accessed in C.	16	BTL4	Analyze
11	Evaluate the use of different conditional statements (if, switch, ternary operator) for designing a menu-driven program. Which one is most efficient and why?	16	BTL5	Apply
12	Critically assess the advantages and disadvantages of using arrays over individual variables in C programs.	16	BTL5	Evaluate
13	Justify the importance of operator precedence in evaluating expressions. Use complex expressions as examples to support your argument.	16	BTL5	Analyze
14	Evaluate a program's correctness that calculates the average of numbers using arrays and conditional statements. Suggest improvements.	16	BTL5	Create
15	Design and implement a C program that takes an array of integers and categorizes them into even and odd numbers using conditional statements.	16	BTL6	Apply

16	Create a program that calculates the result of a mathematical expression input by the user (e.g., "a + b * c") and respects operator precedence and associativity.	16	BTL6	Evaluate
17	Develop a C program to simulate a simple calculator using switch statement with options for all arithmetic and logical operations.	16	BTL6	Create

UNIT II - ADVANCED CONCEPTS IN C
Functions – Recursive Functions, Parameter Passing Techniques, Pointer - Pointer to Structure, Structures
PART-A

Q.No.	Question	Level	Competence
1	What is a function in C?	BTL1	Remember
2	Define a recursive function.	BTL1	Remember
3	What is a pointer?	BTL1	Remember
4	Define structure in C.	BTL1	Remember
5	What is the difference between call by value and call by reference?	BTL1	Remember
6	Give the syntax for declaring a pointer to a structure.	BTL1	Remember
7	List any two advantages of using functions in C.	BTL1	Remember
8	What is meant by parameter passing?	BTL1	Remember
9	Give the base condition of a recursive function.□	BTL1	Remember
10	What is a formal parameter?	BTL1	Remember
11	Explain the need for recursive functions.	BTL2	Understand
12	Describe the role of a return statement in a function.	BTL2	Understand
13	Differentiate between actual and formal parameters.	BTL2	Understand
14	Explain how pointers can be used to pass parameters to functions.	BTL2	Understand
15	Describe how memory is accessed using pointers.	BTL2	Understand
16	Illustrate the use of structure with an example.	BTL2	Understand
17	Explain the concept of nested structures.	BTL2	Understand
18	Why are pointers important in C programming?	BTL2	Understand
19	Describe how to access structure members using pointers.	BTL2	Understand
20	Explain the difference between structure and array.	BTL2	Understand
21	Explain why recursion requires a base condition.	BTL2	Understand

22	What is the difference between local and global variables?	BTL2	Understand
23	Describe function prototype with an example.	BTL2	Understand
24	Explain how recursion differs from iteration.	BTL2	Understand

PART-B

Q.No.	Question	Marks	Level	Competence
1	Write a C program to calculate factorial of a number using recursion.	16	BTL3	Apply
2	Implement a function in C to swap two numbers using pointers.	16	BTL3	Apply
3	Write a C program to read and display student information using structures.	16	BTL3	Apply
4	Implement a recursive function to generate Fibonacci series.	16	BTL3	Apply
5	Write a function to pass structure as a parameter and display student details.	16	BTL3	Apply
6	Analyze the difference between passing by value and passing by reference with code examples.	16	BTL4	Analyze
7	Compare recursive and iterative approaches for computing factorial.	16	BTL4	Analyze
8	Analyze how pointers can be used to dynamically allocate memory for structures.	16	BTL4	Analyze
9	Compare the use of arrays and structures in data storage.	16	BTL4	Analyze
10	Examine the behavior of pointer arithmetic in C with suitable examples.	16	BTL4	Analyze
11	Evaluate the benefits of using recursion in real-time applications.	16	BTL5	Evaluate
12	Justify the use of structures over multiple individual variables in complex data models.	16	BTL5	Evaluate
13	Critically assess the readability and maintainability of recursive vs. iterative code.	16	BTL5	Evaluate
14	Evaluate how pointer misuse can lead to security issues or memory leaks.	16	BTL5	Evaluate
15	Design a C program that manages employee records using structures and pointers.	16	BTL6	Create
16	Develop a recursive C program to solve the Tower of Hanoi problem.	16	BTL6	Create
17	Construct a mini-application in C that uses structures and pointers to manage book records in a library.	16	BTL6	Create

	UNIT III - LINEAR DATA STRUCTURES		
Introduction to Data Structures- Abstract Data Types (ADTs) – List ADT – Array-Based Implementation – Linked List – Doubly- Linked Lists – Circular Linked List – Stack ADT – Implementation of Stack–Applications – Queue ADT – Priority Queues – Queue Implementation – Applications.
	PART-A		
Q.No.	Question	Level	Competence
1	Define Abstract Data Type (ADT).	BTL1	Remember
2	What is a linked list?	BTL1	Remember
3	Define stack.	BTL1	Remember
4	What is a queue?	BTL1	Remember
5	Define circular linked list.	BTL1	Remember
6	What is the difference between static and dynamic memory allocation?	BTL1	Remember
7	List two applications of stack.	BTL1	Remember
8	Define priority queue.	BTL1	Remember
9	What is the use of rear and front in a queue?	BTL1	Remember
10	What is a node in a linked list?	BTL1	Remember
11	What is the difference between singly and doubly linked lists?	BTL1	Remember
12	Write the syntax for declaring a structure in C.	BTL1	Remember
13	Explain the concept of list ADT.	BTL2	Understand
14	Describe array-based implementation of a list.	BTL2	Understand
15	Explain how insertion is performed in a singly linked list.	BTL2	Understand
16	Compare stack and queue.	BTL2	Understand
17	Differentiate between linear and circular queue.	BTL2	Understand
18	Explain the term LIFO with respect to stacks.	BTL2	Understand
19	Describe the enqueue operation in a queue.	BTL2	Understand
20	How does a circular linked list differ from a regular linked list?	BTL2	Understand
21	Why do we use doubly linked lists instead of singly linked lists?	BTL2	Understand
22	Explain how a stack can be implemented using arrays.	BTL2	Understand
23	Describe a real-time application of queue.	BTL2	Understand
24	How is memory managed in linked list implementation?	BTL2	Understand

PART-B

Q.No.	Question	Marks	Level	Competence
1	Write a program to implement a singly linked list with insertion and deletion operations.	16	BTL3	Apply
2	Implement stack operations using arrays in C.	16	BTL3	Apply
3	Write a program to insert and delete elements in a queue using linked list.	16	BTL3	Apply
4	Implement a circular linked list and perform insert at beginning and end.	16	BTL3	Apply
5	Develop a C program to reverse a string using stack.	16	BTL3	Apply
6	Analyze the time and space complexity of array-based vs linked list-based list implementations.	16	BTL4	Analyze
7	Compare stack implementation using array and linked list.	16	BTL4	Analyze
8	Analyze the differences between queue and priority queue with examples.	16	BTL4	Analyze
9	Explain the differences between linear, circular, and doubly linked lists using diagrams and code snippets.	16	BTL4	Analyze
10	Evaluate the performance of stack in recursive function calls.	16	BTL5	Evaluate
11	Justify the use of circular queue over linear queue in memory-constrained environments.	16	BTL5	Evaluate
12	Evaluate the role of pointers in efficient memory management in linked list implementations.	16	BTL5	Evaluate
13	Discuss the pros and cons of using array-based stacks in embedded systems.	16	BTL5	Evaluate
14	Design a menu-driven program to manage a student record system using structures and linked lists.	16	BTL6	Create
15	Create a priority queue system for a hospital using structures and linked lists.	16	BTL6	Create
16	Develop a stack-based expression evaluator for postfix notation.	16	BTL6	Create
17	Construct a circular queue for real-time task scheduling with enqueue and dequeue operations.	16	BTL6	Create

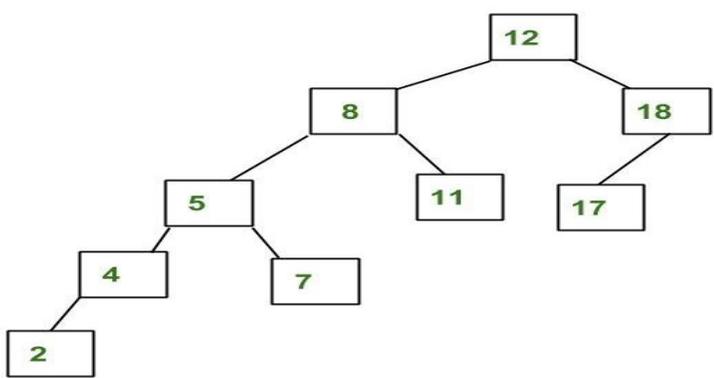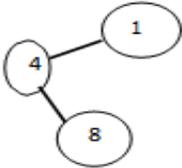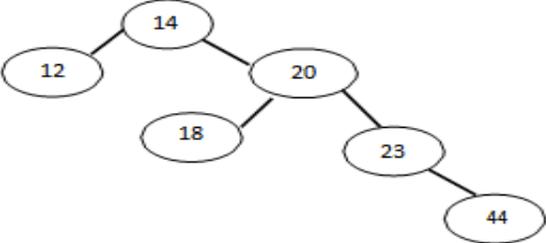| UNIT IV - NON-LINEAR DATA STRUCTURES – TREES |
|---|
| Tree ADT – tree traversals - Binary Tree ADT – expression trees – binary search tree ADT – applications of trees. Implementation of Binary search tree and its operations, tree traversal methods, finding height of the tree using C, AVL Trees. |

PART-A

Q.No.	Question	Level	Competence

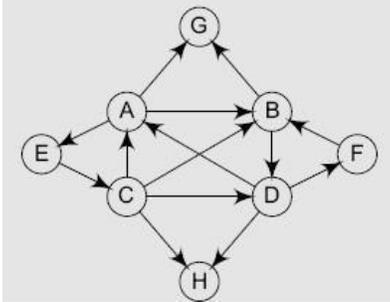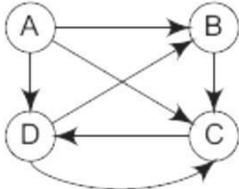| 1	What is a Tree Abstract Data Type (ADT)?	BTL1	Remember
2	Differentiate AVL tree and Binary search tree.	BTL2	Understand
3	Define preorder traversal of a binary tree.	BTL1	Remember
4	Show the difference between inorder and postorder traversal?	BTL2	Remember
5	Write the inorder traversal of a tree with root A and children B (left), C (right).	BTL1	Remember
6	Compare the time complexity of tree traversal algorithms?	BTL2	Understand
7	Which traversal of a binary tree gives sorted order in a binary search tree?	BTL1	Remember
8	Define a leaf node in a binary tree.	BTL2	Understand
9	What is a full binary tree?	BTL1	Remember
10	Define a complete binary tree.	BTL2	Understand
11	What is the maximum number of nodes in a binary tree of height h?	BTL2	Understand
12	State the relationship between the number of leaf nodes and internal nodes in a full binary tree.	BTL2	Understand
13	Show the degree of a node in a binary tree?	BTL1	Remember
14	What is an expression tree?	BTL2	Understand
15	Convert the expression a + b * c to a binary expression tree.	BTL1	Remember
16	What traversal of an expression tree gives the postfix expression?	BTL1	Remember
17	Write the infix expression for a tree with root + and children a (left), * (right) with b and c as children of *.	BTL1	Remember
18	What is a Binary Search Tree?	BTL2	Understand
19	State the time complexity for searching an element in a balanced BST?	BTL2	Understand
20	Give the condition must be satisfied for a tree node to be inserted in a BST?	BTL1	Remember
21	List the properties of a BST.	BTL1	Remember
22	How is deletion handled in a BST node with two children?	BTL2	Understand
23	Write a function prototype for inserting a node in a BST in C.	BTL1	Remember
24	Calculate the height of a binary tree in C using function.	BTL2	Understand

PART-B

Q.No.	Questions	Marks	Level	Competence
1	Apply the tree traversal techniques (inorder, preorder, postorder) on the given binary tree and demonstrate the traversal results.	16	BTL3	Apply
2	Explain the following operations on a binary search tree with suitable algorithm i. Find a node ii. Find minimum and maximum elements of BST.	6 10	BTL3	Apply
3	Write an algorithm for inserting nodes in a binary search tree and explain with an example.	16	BTL3	Apply
4	Describe the various operations that can be performed on a Binary Search Tree with example for each operation.	16	BTL3	Apply
5	i. Discuss in detail the various methods in which a binary tree can be represented. ii. Discuss the advantage and disadvantage of each method.	10 6	BTL4	Analyze
6	i. Create a binary search tree using the following data elements **45, 39, 56, 12, 34, 78, 32, 10, 89, 54, 67, 81** ii. Explain the steps to convert general tree to binary tree?	10 6	BTL5	Evaluate
7	Describe the three cases for deleting a node from a binary search tree with an example for each case.	16	BTL5	Evaluate
8	i. Discuss whether the following tree is AVL tree or not. If yes explain, how is it balanced and if not balance the tree ii. Explain how deletion can take place in AVL trees with suitable algorithm.	10 6	BTL6	Create
9	i. What are AVL trees? Describe the different rotations defined for AVL tree. ii. Insert the following elements step by step in sequence into an empty AVL tree **63, 9, 19, 27, 18, 108, 99, 81**.	8 8	BTL5	Evaluate
10	Describe the rotations for height balancing the tree with the algorithm for all rotations.	16	BTL4	Analyze
11	Write an algorithm for deleting a node in a binary search tree considering the different cases of a node to be deleted.	16	BTL3	Apply

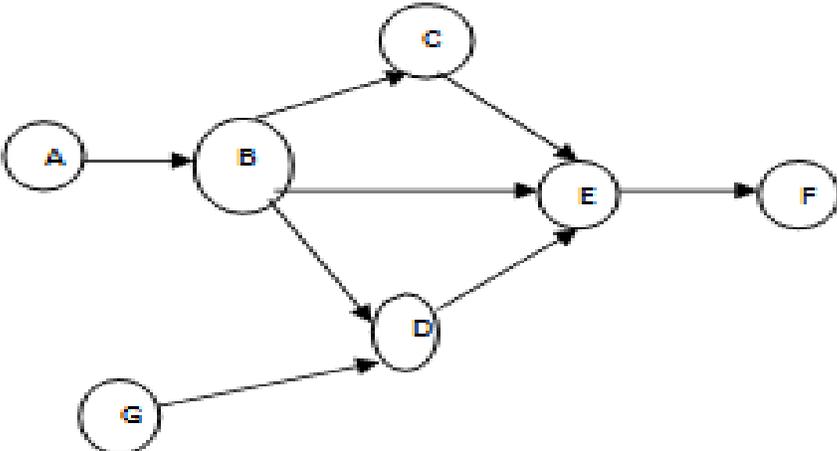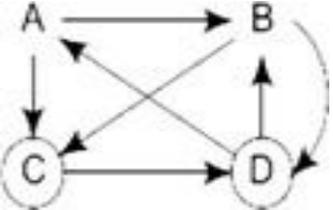|				
12	Explain the construction of expression tree with example. Give the applications of trees.	16	BTL4	Analyze
13	Explain Deletion in Binary tree by merging and copying.	16	BTL3	Apply
14	i. Explain how to delete an element from the binary search tree. ii. Write recursive algorithm for pre order traversal.	8 8	BTL3	Apply
15	Construct an expression tree for the expression (a+b*c) + ((d*e+f)*g). Give the outputs when you apply inorder, preorder and postorder traversals.	16	BTL3	Apply
16	i. Develop a routine for post order traversal. Is it possible to find minimum and maximum value in the binary search tree using traversals? Discuss. ii. Display the given tree using Inorder,Preorder,Postorer traversals 	8 8	BTL3	Apply
17	**Construct** AVL tree for the followings after rotation. i. ii. iii. 	5 5 6	BTL6	Create

UNIT V - NON-LINEAR DATA STRUCTURES –GRAPHS

Definition – Representation of Graph – Breadth-first traversal - Depth-first traversal – Topologial Sort – Dynamic programming Technique – Warshall's and Floyd's algorithm – Greedy method – Dijkstra's algorithm – applications of graphs. Implementation of graph, graph traversal methods, finding shortest path using Dijkstra's algorithm in C.
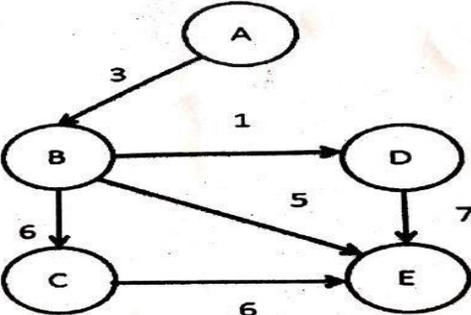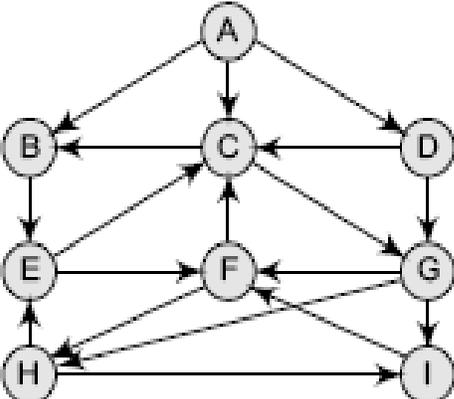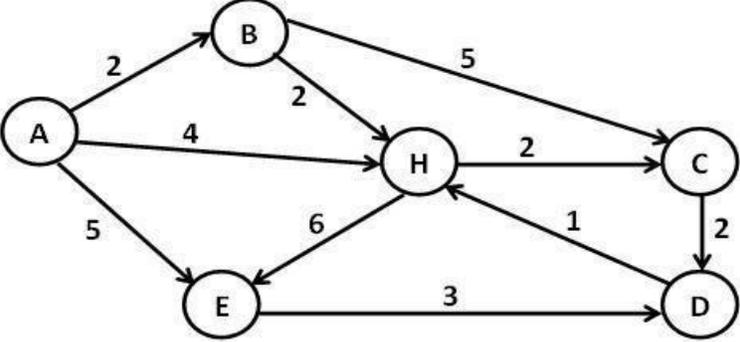
PART-A

Q.No.	Question	Level	Competence
1	Define graph.	BTL 1	Remember
2	Consider the graph given below. Create the adjacency matrix of it	BTL 2	Understand
3	Find out the in-degree and out-degree of each node in the given graph	BTL 2	Understand
4	Create an undirected graph and its adjacency matrix for the following specification of a graph G. V(G)=1,2,3,4 E(G) ={ (1,2),(1,3),(3,3),3,4),(4,1)}	BTL 2	Understand
5	Differentiate BFS and DFS.	BTL 2	Understand
6	What is meant by bi-connected graph?	BTL 1	Remember
7	Give the purpose of Dijikstra's algorithm.	BTL 2	Understand
8	Differentiate cyclic and acyclic graph	BTL 1	Remember
9	Classify strongly connected and weakly connected graph.	BTL 1	Remember
10	Illustrate an articulation point with example.	BTL 1	Remember
11	What are the representations of the graph?	BTL 1	Remember
12	Define minimum spanning tree. Give an example	BTL 1	Remember
13	State the principle of Topological sorting.	BTL 1	Remember
14	Explain procedure for Depth first search algorithm.	BTL 2	Understand

15	What is Dynamic programming technique? Explain it with an example.	BTL 2	Understand
16	Prove that the number of edges in a complete graph of n vertices inn(n-1)/2	BTL 2	Understand
17	Assess the minimum number spanning tree possible for a completegraph with n vertices.	BTL 2	Understand
18	Give two applications of graphs.	BTL 2	Understand
19	What is visiting and traversing in a graph?	BTL 1	Remember
20	When is a graph said to be weakly connected?	BTL 2	Understand
21	What is Greedy method? Explain it with an example.	BTL 1	Remember
22	what is in-degree of a graph. Give suitable example.	BTL 2	Understand
23	Assess out-degree of a graph.	BTL 1	Remember
24	What is the use of Warshall's and Floyd's algorithm?	BTL2	Understand

PART-B

Q.No.	Question	Marks	Level	Competence

1	Describe in detail about the following representations of a graph.
	  i. Adjacency Matrix
	  ii. Adjacency List
		8
		8
			BTL5	Evaluate

2	i) Consider the given directed acyclic graph D. Sort the nodes D by applying topological sort on 'D'. ii.Consider the graph given below and show its adjacency list in the memory. 	8 8	BTL6	Create

3	Differentiate depth-first search and breadth-first search traversal of a graph with suitable examples.	16	BTL3	Apply
4	i. Explain with algorithm, How DFS be performed on an undirected graph. ii. Show the algorithm for finding connected components of an undirected graph using DFS, and derive the time complexity of the algorithm.	10 6	BTL4	Analyze
5	Describe an appropriate algorithm to find the shortest path from 'A' to every other node of A for the given graph. 	16	BTL4	Analyze
6	i. Examine topological sorting of a graph G with suitable example. ii. Explain Dynamic programming with suitable examples.	8 8	BTL3	Apply
7	i. Discuss an algorithm for Breadth first Search on a graph. ii. Give an example based on the algorithm.	10 6	BTL4	Analyze
8	Compare any two applications of Graph with your own example.	16	BTL4	Analyze
9	Develop an algorithm to compute the shortest path using Dijkstra's algorithm. Validate the algorithm with suitable example.	16	BTL6	Create
10	Explain the depth first approach of finding articulation points in a connected graph with necessary algorithm.	16	BTL6	Create
11	i. Write short notes on Bi-connectivity. ii. Express different types of graphs with example.	8 8	BTL3	Apply
12	Explain the various applications of graphs.	16	BTL3	Apply
13	Consider the graph G given below. The adjacency list of G is also given. Assume that G represents the daily flights between different cities and we want to fly from city A to I with minimum stops. One alternative is to use a breadth-first search of G starting at node A. 	16	BTL4	Analyze
14	Explain Warshall's and Floyd's algorithm with suitable example.	16	BTL5	Evaluate

15	i) Explain weakly connected graph and strongly connected graph with example. ii) State the various graph traversal algorithms. Explain each in detail.	8 8	BTL 6	Create
16	Write a C program to implement finding shortest path using Dijkstra's algorithm with necessary algorithm and output.	16	BTL 6	Create
17	Using Dijkstra's algorithm to find the shortest path from the source node A. 	16	BTL3	Apply