Unit-1

|  |  |  |  |
| --- | --- | --- | --- |
| Unit 1 | Marks | CO | BTL Level |
| What is a data structure? | 2 | CO-1 | BTL-1 |
| Define an array in C. | 2 | CO-1 | BTL-2 |
| What is the difference between static and dynamic arrays? | 2 | CO-1 | BTL-2 |
| How do you declare and initialize an array in C? | 2 | CO-1 | BTL-1 |
| What are the limitations of arrays? | 2 | CO-1 | BTL-2 |
| What is the time complexity of accessing an element in an array? | 2 | CO-1 | BTL-1 |
| What is array indexing? | 2 | CO-1 | BTL-1 |
| Explain the difference between a one-dimensional and two-dimensional array. | 2 | CO-1 | BTL-2 |
| How do you traverse an array in C? | 2 | CO-1 | BTL-2 |
| How is an array stored in memory in C? | 2 | CO-1 | BTL-1 |
| Discuss the concept of arrays and how they are implemented in C. | 6 | CO-1 | BTL-4 |
| Write a C program to perform insertion and deletion in an array. | 6 | CO-1 | BTL-5 |
| Explain the advantages and disadvantages of arrays. | 6 | CO-1 | BTL-6 |
| Write a C program to reverse an array. | 6 | CO-1 | BTL-4 |
| Explain the concept of a 2D array in C. Provide an example. | 6 | CO-1 | BTL-5 |
| Write a C program to find the second largest element in an array. | 6 | CO-1 | BTL-6 |
| Explain the memory allocation of arrays in C. | 6 | CO-1 | BTL-4 |
| Write a C program to merge two arrays. | 6 | CO-1 | BTL-5 |
| Discuss the role of arrays in sorting and searching algorithms. | 6 | CO-1 | BTL-6 |
| Write a C program to find the sum of elements in an array. | 6 | CO-1 | BTL-4 |

Unit-2

|  |  |  |  |
| --- | --- | --- | --- |
| Unit 2 | Marks | CO | BTL Level |
| What is a linked list in C? | 2 | CO-2 | BTL-1 |
| Differentiate between singly linked list and doubly linked list. | 2 | CO-2 | BTL-2 |
| How do you define a node in a linked list in C? | 2 | CO-2 | BTL-2 |
| What is a circular linked list? | 2 | CO-2 | BTL-1 |
| Explain how to insert a node at the beginning of a linked list in C. | 2 | CO-2 | BTL-2 |
| What are the advantages of using linked lists over arrays? | 2 | CO-2 | BTL-1 |
| What is a head pointer in a linked list? | 2 | CO-2 | BTL-1 |
| How do you delete a node from a linked list in C? | 2 | CO-2 | BTL-2 |
| What is the time complexity of searching in a linked list? | 2 | CO-2 | BTL-1 |
| How do you reverse a linked list in C? | 2 | CO-2 | BTL-1 |
| Write a C program to implement a singly linked list and perform insertion and deletion operations. | 6 | CO-2 | BTL-4 |
| Explain the concept of dynamic memory allocation in linked lists in C. | 6 | CO-2 | BTL-5 |
| Write a C program to reverse a linked list. | 6 | CO-2 | BTL-2 |
| Discuss the operations of a circular linked list. Write a program to implement it. | 6 | CO-2 | BTL-4 |
| Write a C program to merge two sorted linked lists. | 6 | CO-2 | BTL-5 |
| Explain the concept of doubly linked lists and how they differ from singly linked lists. | 6 | CO-2 | BTL-6 |
| Write a C program to delete a node from a singly linked list. | 6 | CO-2 | BTL-2 |
| Explain the advantages of linked lists over arrays with examples. | 6 | CO-2 | BTL-5 |
| Write a C program to find the length of a linked list. | 6 | CO-2 | BTL-6 |
| Discuss how linked lists are used in real-world applications. | 6 | CO-2 | BTL-4 |

Unit-3

|  |  |  |  |
| --- | --- | --- | --- |
| Unit 3 | Marks | CO | BTL Level |
| What is a stack in data structures? | 2 | CO-3 | BTL-1 |
| What is the Last-In-First-Out (LIFO) principle in stacks? | 2 | CO-3 | BTL-2 |
| How do you implement a stack using arrays in C? | 2 | CO-3 | BTL-1 |
| How do you implement a stack using linked lists in C? | 2 | CO-3 | BTL-1 |
| What is a queue in data structures? | 2 | CO-3 | BTL-2 |
| Explain the First-In-First-Out (FIFO) principle in queues. | 2 | CO-3 | BTL-2 |
| What is the difference between a stack and a queue? | 2 | CO-3 | BTL-1 |
| What is a circular queue? | 2 | CO-3 | BTL-2 |
| How do you implement a queue using arrays in C? | 2 | CO-3 | BTL-1 |
| How do you implement a queue using linked lists in C? | 2 | CO-3 | BTL-1 |
| Write a C program to implement a stack using an array. | 6 | CO-3 | BTL-2 |
| Explain the applications of stacks in C programming. | 6 | CO-3 | BTL-5 |
| Write a C program to implement a queue using linked lists. | 6 | CO-3 | BTL-6 |
| How do you implement a circular queue using an array? Write a C program. | 6 | CO-3 | BTL-4 |
| Discuss the differences between stacks and queues. | 6 | CO-3 | BTL-5 |
| Write a C program to implement a priority queue. | 6 | CO-3 | BTL-2 |
| Explain how stacks are used in recursion. | 6 | CO-3 | BTL-4 |
| Write a C program to reverse a stack using recursion. | 6 | CO-3 | BTL-5 |
| How do you implement a deque in C? Write a program. | 6 | CO-3 | BTL-6 |
| Write a C program to check if parentheses are balanced using a stack. | 6 | CO-3 | BTL-4 |

Unit-4

|  |  |  |  |
| --- | --- | --- | --- |
| Unit 4 | Marks | CO | BTL Level |
| What is a binary tree? | 2 | CO-4 | BTL-1 |
| What is the difference between a binary tree and a binary search tree? | 2 | CO-4 | BTL-2 |
| What is a complete binary tree? | 2 | CO-4 | BTL-1 |
| What is a leaf node in a tree? | 2 | CO-4 | BTL-1 |
| What is inorder traversal of a binary tree? | 2 | CO-4 | BTL-2 |
| What is a full binary tree? | 2 | CO-4 | BTL-1 |
| What is the height of a tree? | 2 | CO-4 | BTL-1 |
| What is preorder traversal of a binary tree? | 2 | CO-4 | BTL-2 |
| What is a balanced binary tree? | 2 | CO-4 | BTL-2 |
| What is the root node of a tree? | 2 | CO-4 | BTL-1 |
| Write a C program to implement a binary search tree. | 6 | CO-4 | BTL-4 |
| Explain different tree traversal methods (inorder, preorder, postorder) in C. | 6 | CO-4 | BTL-5 |
| Write a C program to find the height of a binary tree. | 6 | CO-4 | BTL-6 |
| How do you insert and delete nodes in a binary search tree? Provide a C program. | 6 | CO-4 | BTL-4 |
| Write a C program to perform level-order traversal of a binary tree. | 6 | CO-4 | BTL-2 |
| Discuss the different types of binary trees and their properties. | 6 | CO-4 | BTL-6 |
| Write a C program to implement a balanced binary search tree. | 6 | CO-4 | BTL-4 |
| Explain the concept of AVL trees. Provide an algorithm and code for insertion. | 6 | CO-4 | BTL-2 |
| Write a C program to find the minimum and maximum elements in a binary search tree. | 6 | CO-4 | BTL-6 |
| Discuss the applications of binary trees in computer science. | 6 | CO-4 | BTL-4 |

Unit-5

|  |  |  |  |
| --- | --- | --- | --- |
| Unit 5 | Marks | CO | BTL Level |
| What is a graph in data structures? | 2 | CO-5 | BTL-1 |
| What are the different types of graphs? | 2 | CO-5 | BTL-2 |
| What is an adjacency matrix? | 2 | CO-5 | BTL-2 |
| What is an adjacency list? | 2 | CO-5 | BTL-1 |
| What is the difference between directed and undirected graphs? | 2 | CO-5 | BTL-2 |
| What is a weighted graph? | 2 | CO-5 | BTL-1 |
| Explain depth-first search (DFS). | 2 | CO-5 | BTL-1 |
| Explain breadth-first search (BFS). | 2 | CO-5 | BTL-2 |
| What is a cycle in a graph? | 2 | CO-5 | BTL-2 |
| What is a connected graph? | 2 | CO-5 | BTL-1 |
| Write a C program to implement a graph using an adjacency matrix. | 6 | CO-5 | BTL-4 |
| Discuss the differences between BFS and DFS. | 6 | CO-5 | BTL-5 |
| Write a C program to implement BFS for graph traversal. | 6 | CO-5 | BTL-2 |
| Write a C program to implement DFS for graph traversal. | 6 | CO-5 | BTL-4 |
| Explain Dijkstra’s algorithm with an example. Provide a C program. | 6 | CO-5 | BTL-5 |
| Write a C program to implement Prim’s algorithm for finding a minimum spanning tree. | 6 | CO-5 | BTL-6 |
| Explain Kruskal’s algorithm with an example. Provide a C program. | 6 | CO-5 | BTL-4 |
| Discuss the applications of graphs in computer networks. | 6 | CO-5 | BTL-2 |
| Write a C program to detect a cycle in a graph. | 6 | CO-5 | BTL-6 |
| Explain how graphs are used in social networks. | 6 | CO-5 | BTL-4 |