|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **B. TECH. SECOND YEAR (3rd Semester)- CSE/IT/CS/M.Tech. Integrated/ Data Science/AI/AI-ML** | | | | |
| **Course code** | | **ACSE0351** | **L T P** | **Credit** |
| **Course title** | | **DATA STRUCTURES LAB** | **0 0 2** | **1** |
| **List of Experiments:** | | | | |
| **Sr. No.** | **Name of Experiment** | | | **CO** |
| 1 | Program to create and display Linear Array | | | CO1 |
| 2 | Program to insert a data item at any location in a linear Array | | | CO1 |
| 3 | Program to delete a data item from a Linear Array | | | CO1 |
| 4 | Program to implement multiplication of two matrices. | | | CO1 |
| 5 | Program to create sparse matrix. | | | CO1 |
| 6 | Program to implement linear search in an Array | | | CO4 |
| 7 | Program to implement binary search in an Array | | | CO4 |
| 8 | Program to implement bubble sort in a non-recursive way | | | CO4 |
| 9 | Program to implement selection sort in a non-recursive way | | | CO4 |
| 10 | Program to implement insertion sort in a non-recursive way | | | CO4 |
| 11 | Program to implement Merge sort in a non-recursive way | | | CO4 |
| 12 | Program to implement Merge sort in a recursive way | | | CO4 |
| 13 | Program to implement Quick sort in a recursive way | | | CO4 |
| 14 | Program to implement Queue Using array | | | CO3 |
| 15 | Program to implement Circular Queue Using array | | | CO3 |
| 16 | Program to implement Stack Operation using array | | | CO3 |
| 17 | Program to implement the Single Linked List   |  |  |  |  | | --- | --- | --- | --- | | 1. Insertion | 1. Deletion | 1. Traversal | 1. Reversal | | 1. Searching | 1. Updation | 1. Sorting | 1. Merging | | | | CO2 |
| 18 | Program to implement the doubly Linked List   |  |  |  |  | | --- | --- | --- | --- | | 1. Insertion | 1. Deletion | 1. Traversal | 1. Reversal | | 1. Searching | 1. Updation | 1. Merging |  | | | | CO2 |
| 19 | Program to implement the circularly Single Linked List   |  |  |  |  | | --- | --- | --- | --- | | 1. Insertion | 1. Deletion | 1. Traversal | 1. Reversal | | 1. Searching | 1. Updation |  |  | | | | CO2 |
| 20 | Program to implement Queue Using linked list | | | CO3 |
| 21 | Program to implement Circular Queue Using linked list | | | CO3 |
| 22 | Program to implement Priority Queue Using linked list | | | CO3 |
| 23 | Program to implement Stack Operation using Linked list | | | CO3 |
| 24 | Program to convert infix to postfix expression. | | | CO3 |
| 25 | Program to evaluate postfix expression | | | CO3 |
| 26 | Program to compute factorial using tail recursion | | | CO3 |
| 27 | Program to implement Tower of Hanoi | | | CO3 |
| 28 | Program implementing Addition of two polynomials via Linked Lists | | | CO2 |
| 29 | Program to implement binary tree using linked list   |  |  |  |  | | --- | --- | --- | --- | | 1. Insertion | 1. Deletion | 1. Traversal | 1. Searching | | | | CO5 |
| 30 | Program to implement binary search tree using linked list   |  |  |  |  | | --- | --- | --- | --- | | 1. Insertion | 1. Deletion | 1. Traversal | 1. Searching | | | | CO5 |
| 31 | Program to implement Heap sort in a non-recursive way | | | CO5 |
| 32 | Program to implement Radix sort. | | | CO4 |
| 33 | Program to implement BFS algorithm | | | CO5 |
| 34 | Program to implement DFS algorithm | | | CO5 |
| 35 | Program to implement the minimum cost spanning tree | | | CO5 |
| 36 | Program to implement the shortest path algorithm | | | CO5 |
| **Lab Course Outcome:** After completion of this course students will be able to | | | | |
| CO 1 | Implement operations on single and multi-dimensional array. | | | K3 |
| CO 2 | Implement various linear data structures like single Linked-list, doubly Linked-list, Circular linked-list. | | | K3, K6 |
| CO 3 | Implement Stack and Queue using array and linked list. | | | K3 |
| CO 4 | Analyze and Implement sorting and searching algorithms. | | | K4, K6 |
| CO5 | Solve complex problems using non-linear data structures like tree and graph. | | | K6 |