**COURSE: B.Tech SEMESTER: III**

**LAB: DATA STRUCTURES REGULATION: A2**

**WEEK 1(RECURSION):**

* Write a C Program using recursive function to compute the factorial of N for a given positive N value.
* Write a C Program using recursive function to generate Fibonacci Series
* Write a C Program using recursive function to compute GCD of two positive numbers
* Write a C Program using recursive function to implement Towers of Hanoi

**WEEK 2(SEARCH TECHNIQUES):**

* Write a C Program to search an element in the given list using Linear Search Technique. (using recursive and non-recursive functions)
* Write a C Program to search an element in the given sorted list using Binary search Technique. (using recursive and non-recursive functions)

**WEEK 3(SORTING TECHNIQUES):**

* Write a C Program using recursive function to sort a given list of integers in ascending order using Quick Sort Technique.
* Write a C Program using recursive function to sort a given list of integers in ascending order using Merge Sort Technique.

**WEEK 4(LINKED LIST):**

* Write a C Program to create a Single linked list and perform basic operations

(**Insertion** (at the beginning, at the end, before or after a given node),

**Deletion** (at the beginning, at the end, specified or a given node),

**Search, Traversal**)

**WEEK 5 (OTHER VARIANTS OF LINKED LIST):**

* Write a C Program to create a Circular linked list and perform basic operations

(**Insertion** (at the beginning, at the end, before or after a given node),

**Deletion** (at the beginning, at the end, specified or a given node),

**Search, Traversal**)

* Write a C Program to create a Double linked list and perform basic operations

(**Insertion** (at the beginning, at the end, before or after a given node),

**Deletion** (at the beginning, at the end, specified or a given node),

**Search, Traversal**)

**WEEK 6 (STACK):**

* Write a C Program to implement Stack operations (**push , pop , display**) using arrays
* Write a C Program to implement Stack operations (**push, pop, display**) using linked list.
* Write a C Program to implement Infix to postfix conversion using stacks.
* Write a C Program to evaluate the Postfix Expression using stacks.

**WEEK 7 (QUEUE):**

* Write a C Program to implement Queue operations (**enqueue** (insert), dequeue (delete), **display**) using arrays.
* Write a C Program to implement Queue operations ( **enqueue** (insert) ,**dequeue** (delete), **display**) using linked list
* Write a C Program to implement Circular Queue.

**WEEK 8(BINARY TREE):**

* Write a C Program to implement Binary Tree Creation.
* Write a C Program to implement Recursive Binary Tree Traversals (Depth first Traversals – In-Order, Pre-Order, Post-Order) for a given list of numbers or characters.

**WEEK 9(BINARY SEARCH TREE):**

* Write a C Program to implement Binary Search Tree Creation.
* Write a C program to implement Various Operations on Binary Search Tree( **Insertion, Deletion, Search**)

**WEEK 10(GRAPH):**

* Write a C Program to Creation a Graph ( using Adjacency Matrix or Adjacency List)
* Write a C Program to implement Graph Traversal (Breadth First and Depth First) Traversal.

**WEEK 11(GRAPH ALGORITHMS):**

* Write a C Program to implement Minimum Spanning Tree ( Prim’s Algorithm, Kruskal’s Algorithm)
* Write a C Program to implement Single Source Shortest Path(Dijkstra’s Algorithm)

**WEEK 12 (HEAP):**

* Write a C Program to implement Binary Heap (minimum heap or maximum heap)
* Write a C Program to implement Heap Sort
* Write a C Program to implement Collision Resolution Techniques using Linear probe (Open Addressing) Technique using Division method as hash function.