**V.E.S. Institute of Technology, Collector Colony,   
Chembur, Mumbai**

**Department of M.C.A**

**INDEX**

|  |  |  |
| --- | --- | --- |
| **Sr.**  **No** | **Contents** | **Files** |
| **1** | **Implementation of different sorting techniques.** | **Sorting.cpp, BubbleSort.cpp, SelectionSort.cpp, InsertionSort.cpp, QuickSort.cpp, ShellSort.cpp, RadixSort.cpp** |
| **2** | **Implementation of searching algorithms.** | **Searching.cpp, LinearSearch.cpp, BinarySearch.cpp** |
| **3** | **Implementation of stacks(Using arrays and Linked List)** | **StackArray1.cpp, StackArray2.cpp,**  **StackLinkedList1.cpp, StackLinkedList2.cpp** |
| **4** | **Implementation of Stack Applications like:**  **a. Postfix evaluation**  **b. Balancing of Parenthesis** | **Postfix1.cpp, Postfox2.cpp,**  **Balancing1.cpp, Balancing2.cpp** |
| **5** | **Implement all different types of queues.** | **SimpleQueue1.cpp, SimpleQueue2.cpp,**  **CircularQueue2.cpp,**  **DoubleEndedQueue1.cpp, DoubleEndedQueue2.cpp** |
| **6** | **Demonstrate application of queue (eg. Priority Queue, Breadth First Search)** | **PriorityQueue1.cpp, PriorityQueue2.cpp,**  **BreadthFirstSearch.cpp** |
| **7** | **Implementation of all types of linked lists.**  **Insert, Display, Delete,**  **Search, Count Reverse operation on:**  **Singly Linked Lists**  **Circular Linked List**  **Doubly Linked Lists** | **SLL.cpp, CLL.cpp, DLL.cpp** |
| **8** | **Demonstrate application of linked list (Polynomial addition, Sparse matrix)** | **PolynomialAdd.cpp, SparseMatrix.cpp\*,SparseMatrix2.cpp** |
| **9** | **Create and perform various operations on BST.** | **BST1.cpp, BST2.cpp** |
| **10** | **Implementing Heap with different operations performed.** | **HeapOP.cpp, HeapSort.cpp** |
| **11** | **Create a Graph storage structure (eg. Adjacency matrix)** | **AdjacencyMatrix1.cpp, AdjacencyMatrix2.cpp** |
| **12** | **Perform various hashing techniques with Linear Probe as collision resolution scheme.** | **ModuloDivision.cpp, MidSqaure.cpp\*** |
| **13** | **Create a minimum spanning tree using any method Kruskal’s Algorithm for Prim’s Algorithm** | **Kruskal.cpp,**  **Prim1.cpp, Prim2.cpp** |
| **14** | **Implementation of Graph traversal. (DFS and BFS)** | **DFS.cpp, BFS.cpp** |
| **15** | **Group Project** |  |