1. **Write a Program to create a Binary Tree and perform following nonrecursive operations on it. a. Preorder Traversal, b. Postorder Traversal, c. Count total no. of nodes, d. Display height of a tree.**
2. **Write a Program to create a Binary Tree and perform following nonrecursive operations on it. a. inorder Traversal; b. Count no. of nodes on longest path; c. display tree levelwise; d. Display height of a tree.**
3. **Write a Program to create a Binary Search Tree holding numeric keys and perform following nonrecursive operations on it. a. Levelwise display, b. Mirror image, c. Display height of a tree, d. Find**
4. **Write a program to illustrate operations on a BST holding numeric keys. The menu must include: • Insert • Delete • Find • display in Inorder way**
5. **Write a program to illustrate operations on a BST holding numeric keys. The menu must include: • Insert • Mirror Image • Find • Post order (nonrecursive)**
6. **Write a Program to create a Binary Search Tree and perform following nonrecursive operations on it. a. Preorder Traversal b. Inorder Traversal c. Display Number of Leaf Nodes d. Mirror Image**
7. **Write a Program to create a Binary Search Tree and perform following nonrecursive operations on it. a. Preorder Traversal b. Postorder Traversal c. Display total Number of Nodes d. Display Leaf nodes.**
8. **Write a Program to create a Binary Search Tree and perform deletion of a node from it. Also display the tree in nonrecursive postorder way.**
9. **Write a Program to create a Binary Search Tree and display it levelwise. Also perform deletion of a node from it.**
10. **Write a Program to create a Binary Search Tree and display its mirror image with and without disturbing the original tree. Also display height of a tree using nonrecursion.**
11. **Write a program to efficiently search a particular employee record by using Tree data structure. Also sort the data on emp-id in ascending order.**
12. **Write a Program to create Inorder Threaded Binary Tree and Traverse it in Preorder way.**
13. **Write a Program to create Inorder Threaded Binary Tree and Traverse it in Inorder way.**
14. **Write a Program to implement AVL tree and perform different rotations on it and display it in sorted manner.**
15. **Write a Program to implement AVL tree and perform deletion on it and display it in sorted manner.**
16. **Write a Program to accept a graph from user and represent it with Adjacency Matrix and perform BFS and DFS traversals on it.**
17. **Write a Program to accept a graph from user and represent it with Adjacency List and perform BFS and DFS traversals on it.**
18. **Write a Program to implement Prim’s algorithm to find minimum spanning tree of a user defined graph. Use Adjacency Matrix to represent a graph.**
19. **Write a Program to implement Dijkstra’s algorithm to find shortest distance between two nodes of a user defined graph. Use Adjacency Matrix to represent a graph.**
20. **Write a Program to implement Kruskal’s algorithm to find minimum spanning tree of a user defined graph. Use Adjacency Matrix to represent a graph.**
21. **WAP to implement Heap sort on 1D array of Student structure (contains student\_name, student\_roll\_no, total\_marks), with key as student\_roll\_no. And count the number of swap performed.**
22. **WAP to convert a given Infix expression into its equivalent Postfix expression and evaluate it using stack.**
23. **WAP to implement stack using a singly linked list and perform following operations on it. A. PUSH, B. POP, C. StackeEmpty D. Display Stack.**
24. **WAP to implement following by using stack. A. Factorial of a given number B. Generation of Fibonacci series.**
25. **WAP to implement a linear queue using a singly linked list and perform following operations on it. A. enqueue, B. dequeue, C. QueueEmpty, D. Display queue, E. Display Front element, F. Display Rear element**
26. **      Write a Program to implement circular queue where user can add and remove the elements from rear and front end of the queue**
27. **      WAP to perform addition of two polynomials using singly linked list.**
28. **      Write a Reverse() function that reverses a Singly linked list and display the list**
29. **      WAP to create doubly linked list and perform following operations on it. A) Insert (all cases) 2. Delete (all cases).**
30. **  WAP to merge two sorted Doubly linked lists and display their result.**
31. **Write a Program to create Inorder Threaded Binary Tree and Traverse it in Postorder way.**