

1. Implement operations (traverse, insert, delete, linear search, selection sort) on an array.
2. Implement insertion (at the beginning, at specified location, at the end) on single linked list and circular single linked list.
3. Implement deletion (at the beginning, at specified location, at the end) on single linked list and circular single linked list.
4. Implement insertion (at the beginning, at specified location, at the end) on double linked list and circular double linked list.
5. Implement deletion (at the beginning, at specified location, at the end) on double linked list and circular double linked list.
6. Write a program to count the number of nodes & reverse the single linked list.
7. Write a program to merge two sorted linked list and display the final sorted linked list.
8. Implement addition of two polynomial expressions using singly linked list.
9. Implement operations (push, pop) on a stack using arrays. Check the status of the stack whether there is underflow or overflow.
10. Implement the conversion of infix notation to postfix notation.
11. Implement the evaluation of postfix notation using stacks.
12. Implement binary search using recursion.
13. Implement operations (enqueue, dequeue) on a queue using arrays. Check the status of the queue whether it is empty or full.
14. Implement circular queue using arrays and linked list.
15. Implement stacks and queues using linked list.
16. Implement Sparse Array.
17. Implement operations on Binary Search Tree (Insertion, Deletion, Search, Traversals (using recursion)- Inorder, Preorder, Postorder).
18. Implement traversals on Binary Search Tree (using stacks) - Inorder, Preorder, Postorder).
19. Implement graph traversal (DFS & BFS)
20. Make a menu driven program to perform various sorting techniques (insertion, shell, merge, heap, bubble, quick).