

1. Write a program to
  - (a.) create an array of integers and initialize it at compile-time
  - (b.) create another array of floating values and initialize it at run-time
  - (c.) display the elements of both the arrays with proper headings
2. Write a Program to implement Linear Search for
  - (a.) First occurrence of search item
  - (b.) All occurrences of search item
3. Write a program to
  - (a.) create an array of integers and initialize it
  - (b.) Find minimum and maximum elements in the array
  - (c.) Find sum and average of array elements
4. Write a program to Merge unsorted arrays
5. Write a program to Merge sorted arrays
6. Write a program to insert a number at a given location in an array.
7. Write a program to delete a number from a given location in an array.
8. Write a program to search a number in an array and delete it, if found.
9. Write a Program to implement Binary Search
10. Write a Program to perform elimination of duplicate elements from an existing list of elements.
11. Create a Matrix. Perform addition, subtraction, Transpose and Multiplication using Switch-Case statement.
12. Read and display Upper, Lower and Tri-diagonal matrices
13. Implement sparse matrices using 3-tuple notation.
14. Write a Program to implement Selection Sort.
15. Write a Program to implement Insertion Sort.
16. Write a Program to implement Bubble Sort.
17. Write a Program to implement Merge Sort.
18. WAP to implement Singly Linked List that stores data as integer and perform following operations:
19. Traverse the list to display each element
20. Search for a specific element in the list
21. WAP to implement Singly Linked List that stores data as integer and perform following operations:
  - (a.) Count the number of nodes in the list
  - (b.) Find minimum and maximum value in the linked list.
22. WAP to implement Singly Linked List that stores data as integer and perform following operations:
  - (a.) Insert a new node in the beginning and end of the list
  - (b.) Insert a new node after a given node in the list.
  - (c.) Insert a new node before a given node in the list.

23. WAP to implement two Singly Linked List that stores data as integer and perform following operation:
- (a.) Merge the two lists to create a new sorted list
24. WAP to implement a Singly Linked List that stores data as integer and perform following operation:
- (a.) Create a new list that is reverse of the first linked list
25. WAP to implement Singly Linked List that stores data as integer and perform following operations:
- (a.) Delete a node in the beginning and end of the list.
  - (b.) Delete the node that comes after a given node in the linked list.
  - (c.) Search an element in the linked list. If found, delete it.
26. WAP to implement Doubly Linked List that stores data as integer and perform following operations:
- (a.) Traverse the list to display each element
  - (b.) Search for a specific element in the list
27. WAP to implement Doubly Linked List that stores data as integer and perform following operations:
- (a.) Insert a new node in the beginning, end and middle of the list
  - (b.) Delete a node in the beginning, end and middle of the list
  - (c.) WAP to implement Header Linked List with operations:
  - (d.) Insertion(Beginning, Between, End)
  - (e.) Deletion(Beginning, Between, End)
  - (f.) Traverse
28. WAP to implement Circular Linked List with operations:
- (a.) Insertion(Beginning, Between, End)
  - (b.) Deletion(Beginning, Between, End)
  - (c.) Traverse
  - (d.) Linear Search
29. WAP to perform polynomial addition using linked lists
30. Write a Program to Perform Pop, Push, Traverse operations on the stack using array (Static Stack).
31. (Optional) WAP to Convert Infix Expression to Postfix form using Stack.
32. (Optional) WAP to Convert Infix Expression to Prefix form using Stack.
33. WAP to evaluate Postfix expression using Stack.
34. WAP to reverse the String using Stack
35. WAP to perform different operations with Queue such as Insert, Delete, Display of elements using array. (Linear Queue or Static Queue)
36. WAP to perform different operations with Queue such as Insert, Delete, Display of elements using Circular Queue.
37. Write a Program to Perform Pop, Push, Traverse operations on the stack using Pointer (Dynamic Stack).

38. WAP to perform different operations such as Insert, Delete, Display elements using Dynamic Queue.
39. WAP to implement priority queue with three priority values (1: Lowest, 2, 3: Highest)
40. WAP to find factorial of a number using recursion
41. WAP to generate Fibonacci Series of n-terms
42. Write a program to calculate the GCD of two numbers using recursive functions
43. WAP to create a Binary tree and traverse the tree in Inorder, Preorder and Postorder manner using recursive functions
44. WAP to create a Binary Search Tree (BST) and traverse the tree in Inorder, Preorder and Postorder manner using recursive functions
45. WAP to implement following recursive operations on a Binary Search Tree (BST)
  - a. Find an element
  - b. Insert an element
  - c. Delete an element
  - d. Count the number
  - e. Find maximum element
  - f. Find minimum element
  - g. Find height of the tree