

WAP to convert a given Infix expression into its equivalent Postfix expression and evaluate it using stack.

✓ **WAP to convert a given Infix expression into its equivalent Prefix expression and evaluate it using stack.**

✓ **WAP to implement two stack using array and perform following operations on it. A. PUSH, B. POP, C. StackFull D. StackEmpty E. Display Stack.**

✓ **WAP to implement following by using stack.**
A. Factorial of a given number B. Generation of Fibonacci series

✓ **Write a Program to implement circular double ended queue where user can add and remove the elements from both front and rear of the queue**

✓ **Write a Program to implement multiple two queues using array and perform following operations on it. A. Addq, B. Delq, C. Display Queue.**

Fifth Assignment



WAP to perform addition of two polynomials using singly linked list.



Write an iterative Reverse() function that reverses a list by rearranging all the next pointers and the head pointer. Ideally, Reverse() should only need to make one pass of the list.



WAP to create doubly linked list and perform following operations on it. A) Insert (all cases) 2. Delete (all cases).



WAP to merge two sorted Doubly linked lists and display their result.



Implement Push and POP operations of STACK on Doubly linked lists



Implement ADD and DELETE operations of QUEUE on Doubly linked lists

Seventh Assignment

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.

✓ Write a Program to create a Binary Tree and perform following nonrecursive operations on it. a. Levelwise display b. Mirror image c. Display height of a tree.

✓ Write a program to illustrate operations on a BST holding numeric keys. The menu must include: • Insert • Delete • Find • Show

✓ Write a program to illustrate operations on a BST holding numeric keys. The menu must include: • Insert • Mirror Image • Find • Post order (nonrecursive)

✓ Write a Program to create a Binary Tree and perform following Nonrecursive operations on it. a. Inorder Traversal b. Preorder Traversal c. Display Number of Leaf Nodes d. Mirror Image

✓ 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

✓ 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.

✓ 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.

✓ Write a Program to create a Binary Search Tree and display it levelwise. Also perform deletion of a node from it.

✓ 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.

✓ 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.

Eighth Assignment

Write a Program to create Inorder Threaded Binary Tree and Traverse it in Preorder way.

✓ Write a Program to create Inorder Threaded Binary Tree and Traverse it in Inorder way.

- ✓ Write a Program to implement AVL tree and perform different rotations on it and display it Levelwise.

Ninth Assignment

Write a Program to accept a graph from user and represent it with Adjacency Matrix and perform BFS and DFS traversals on it.

- ✓ Write a Program to implement Prim's algorithm to find minimum spanning tree of a user defined graph. Use Adjacency List to represent a graph.

- ✓ Write a Program to implement Kruskal's algorithm to find minimum spanning tree of a user defined graph. Use Adjacency List to represent a graph.

- ✓ Write a Program to implement Dijkstra's algorithm to find shortest distance between two nodes of a user defined graph. Use Adjacency List to represent a graph.

- ✓ Write a Program to accept a graph from user and represent it with Adjacency List and perform BFS and DFS traversals on it.

- ✓ 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.

✓ 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.

✓ Write a Program to implement Prim's algorithm to find minimum spanning tree of a user defined graph. Use Adjacency List to represent a graph.

✓ Write a Program to implement Kruskal's algorithm to find minimum spanning tree of a user defined graph. Use Adjacency List to represent a graph.

✓ Write a Program to implement Dijkstra's algorithm to find shortest distance between two nodes of a user defined graph. Use Adjacency List to represent a graph.

✓ 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.

WAP to implement Heap sort and Quick 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.

WAP to implement Heap sort and Merge 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.

WAP to implement Quick sort and Merge Sort on 1D array of Employee structure (contains employee_name, emp_no, emp_salary), with key as emp_no. And count the number of swap performed.

Assume that an array A with n elements was sorted in an ascending order, but two of its elements swapped their positions by a mistake while maintaining the array. Write a code to identify the swapped pair of elements and their positions in the asymptotically best possible time. [Assume that all given elements are distinct integers.]

Implement following hashing Techniques by assuming suitable input and Table Size.

a. Linear Probing

b. Quadratic Probing

c. Double Hashing.

Also mention number of collisions occurred while inserting a Data in hash table.