1. Write a program to implement stack using array. Implement functions for below operations.
2. Push element
3. Pop element
4. Peep element
5. Check if stack is full
6. Check if stack is empty
7. Write a program to reverse a string using stack
8. Write a program to convert a decimal number into its binary form using stack.
9. Write a program to implement double ended stack.

class Dstack

{

int \*arr;

int top1, top2, size;

};

In double ended stack addition and deletion of elements take place from both ends. Maintain two top indicators to perform operations.

1. Write a program to convert the infix expression into its postfix form using stack. Accept infix string from user.
2. Write a program to convert infix expression into its prefix form using stack. Accept infix string from user.
3. Write a program to evaluate a postfix expression.
4. Write a program to implement queue using array. Implement functions for below operations.
5. Insert element in queue
6. Remove element from queue.
7. Print elements of queue.
8. Check if queue is full
9. Check if queue is empty.
10. Reverse elements of stack using queue.
11. Implement circular queue using array with all operations mentioned in question 8.
12. Implement priority queue using array with all operations mentioned in question 8.
13. Write a program to implement double ended queue with array i.e. de-queue. Implement operations
14. Insert from front end
15. Insert from rear end
16. Remove from front end
17. Remove from rear end
18. Check if queue is full
19. Check if queue is empty
20. Write a menu driven program to create a singly linked list to perform following operations
21. Insert node at end, begin, middle of list
22. Remove node from end, begin, middle of list
23. Display list elements
24. Modify assignment 13 to print singly linked list in reverse way using recursion.
25. Implement doubly linked list with all operations mentioned in assignment 13
26. Implement singly circular linked list with all operations mentioned in assignment 13
27. Write a program to implement stack using linked list. Implement functions for below operations.
28. Push element
29. Pop element
30. Peep element
31. Check if stack is empty
32. Write a program to implement Binary Search Tree dynamically. Implement below operations
33. Insert a node with user entered value
34. Print inorder, preorder and postorder traversal of a tree
35. Delete a node with user entered value
36. Print height of a tree
37. Write a program to implement threaded binary tree with all operations mentioned in question 18
38. Write a program to implement AVL tree with all operations mentioned in question 18
39. Write a program to sort the elements of array using bubble sort
40. Write a program to sort the elements of array using selection sort
41. Write a program to sort the elements of array using insertion sort
42. Write a program to sort the elements of array using quick sort
43. Write a program to sort the elements of array using merge sort
44. Write a program to search an element inside array using linear search
45. Write a program to search an element inside array using binary search