CSL201–DATA STRUCTURES LAB.

CYCLE 1:

1.	Write a program to perform Linear search on n numbers.Write Linear Search as a function and call it from the function main () by passing the appropriate parameters.
2.	Implement Polynomial in one variable using Structure.Write function to perform addition of two polynomials in one variable.
3.	Represent Sparse Matrices using arrays and read two sparse matrices. Write functions to perform addition of two sparse matrices and Transpose of a Sparse Matrix.
4.	Implement a Menu driven program to perform the following operations on a Stack Data Structure using Arrays. a. Pushing elements to the stack. b. Popping Elements from the stack. c. Display the contents of the Stack.
5.	Implement a Menu driven program to perform the following operations on Queue Data Structure using Arrays. (i) Inserting an element to the Queue. (ii) Deleting an element from the Queue. (iii) Display the contents of the Queue
6.	Implement a Menu driven program to perform the following operations on Circular Queue Data Structure using Arrays. (i) Inserting an element to the Circular Queue. (ii) Deleting an element from the Circular Queue. (iii) Display the contents of the Circular Queue
7.	Write functions to implement the following. a. Convert infix expression to postfix expression. b. Evaluate a Postfix Expression.

CYCLE 2:

8.	Write a menu driven program to perform the following operations on a Singly Linked List. a. Insert at the beginning. b. Insert at the end. c. Insert in sorted order d. Count the number of nodes. e. Reverse the linked list. f. Delete from beginning g. Delete from end h. Display the linked list.
9.	Write a menu driven program to perform the following operations on a Doubly Linked List. a. Insert at the beginning. b. Insert at the end. c. Insert in sorted order d. Count the number of nodes. e. Reverse the linked list. f. Delete from beginning g. Delete from end h. Display the linked list.
10.	Write a menu driven program to perform the following operations on a Circular Singly Linked list. a. Insert at the beginning. b. Insert at the end. c. Delete from Beginning. d. Delete from end e. Display the linked list.
11.	Implement a Stack Using Singly Linked List.
12.	Implement a Queue using a Circular Singly Linked List.
13.	Discuss the implementation of polynomials using Linked lists. Write a program to read two polynomials in one variable and store it using linked lists, and perform addition and multiplication of the polynomials, display the resultant polynomials.

CYCLE 3

14.	Write a menu driven program to perform the following operations on a Binary Search Tree. a. Insert b. Delete c. Search d. Display
15.	Write a program to create a Binary Tree and perform the following traversals on it. a. Inorder b. Preorder c. Postorder
16.	Write a program to read a graph and store it using adjacency lists, and perform the following traversals on the Graph. a. BFS b. DFS
17.	Write functions to perform the following Sorting on a Set of numbers. a. Selection Sort b. Bubble Sort c. Insertion Sort
18.	Write program to perform Merge Sort on a set of Numbers.
19.	Write a program to perform Quicksort on a set of numbers.
20.	Implement Binary search as a. Iterative function b. Recursive function

Additional Programs:

1. Write a program to convert an infix expression to a prefix expression using stacks.
2. Write a program to reverse the content of queue using stack.
3. Implement a Priority Queue using arrays with the operations:
    3.1. Insert elements to the Priority Queue.
    3.2. Delete elements from the Priorit Queue.
    3.3. Display the contents of the Priority Queue after each operation.
4. Implement a Double-Ended Queue(DEQUEUE) with the operations:
        4.1. Insert elements to the front of the queue.
        4.2. Insert elements to the rear of the queue.
        4.3. Delete elements from the front of the queue.
        4.4. Delete elements from the rear of the queue.
        4.5. Display the queue after each operation.
5. Implement two Stacks –Stack1 and Stack2 using a single Array.
6. Implement the following operation on a Hash table using the hash function **h(k)=k%size** and double hashing for collision resolution, with mid-square method as the second hash function.
7. Write a program to perform Heap sort on a set of numbers.