**DATA STRUCTURES USING C -LAB CYCLE**

1. Write programs to demonstrate the use of storage classes (local variable, global variable, static variable, register variable) in C.

2. Use a menu-driven program to insert, search, delete and sort elements in an array using functions (use global variables)

3. Use a menu-driven program to insert, search, delete and sort elements in an array using functions (use only local variables)

4. Search for all the occurrences of an element in an integer array (positions)

5. Sort the array elements in ascending order (minimum three functions: read, disp and sort)

6. Display the array elements in the same order using a recursive function

7. Display array elements in reverse order using a recursive function.

8. Write a program to Perform the addition of two matrix and Subtraction of one matrix from another

9. Write a program to perform multiplication of two matrix

10. Write a program to find the transpose of a matrix

11. Write a program to find the Determinant of a matrix (2x2 and 3x3)

**Section 2: Stack**

12. Implement stack operations using arrays

**Session 3 - String**

13. Read a String and Just print it in the reverse order

14. Read a String and Reverse the string in the same array itself

15. Read n Strings and display them in the ascending order.

**Session 4 - Stack .. Continued**

16. Reverse a string using Stack

17. Convert an expression from infix expression to postfix using stack

18. Convert an expression from infix expression to prefix using stack

19. Evaluate an infix expression using stack

20. Evaluate an expression using stack by converting it into postfix before evaluating

21. A letter means push and an asterisk means pop in the following sequence. Give the sequence of values returned by the pop operations when this sequence of operations is performed on an initially empty LIFO stack.

E A S \* Y \* Q U E \* \* \* S T \* \* \* I O \* N \* \* \*

**Session 5 - Sparse Matrix**

**22.** Read and display a sparse matrix

23. Write a program to add two sparse matrix

24. Write a program to multiply two sparse matrix

**Session - 6 Polynomial using array**

25. Read a polynomial and display

26. Add two polynomials

27. Subtract two polynomials

28. Multiply two polynomials

**Session-7**

29. Demonstrate queue using array

30. Demonstrate circular queue using array.

**Session 8**

31. Implement heapsort

32. Demonstrate a priority queue using an array.

**Session 9: Structure**

33. Define a structure for dates with dd/mm/yyyy. Provide functions for reading, displaying and comparing two dates are equal or not

34. Define a structure for employees with eno,ename, esal and dno. Read n employees information and provide functions for the following:

a. Searching an employee by no

b. Sorting the employees by

i. Name

ii. Salary

c. Deleting an employee

**Session 10: Polynomials using Structure**

35. Read a polynomial and display it; use structure array

36. Add two polynomials

37. Subtract two polynomials

38. Multiply two polynomials

**Session 11: Dynamic Memory Allocation**

39. Implement a) malloc , b) calloc and c) free functions

40. Use malloc to read n integers and find the mean.

41. Use calloc to read n numbers and find the mode.

42. Declare a structure for Books having author\_name and book\_name. Create an array of books using a pointer variable. Provide functions for reading n books and displaying the same using pointers.

43. Use realloc to implement varchar for any length.

44. Demonstrate a linked list creation and display

45. Write a program with functions to insert a new node

a. at the beginning of a Singly Linked List.

b. At the end of the linked list

c. after a specified element in a linked list.

46. Write a program with functions to delete a node

a. From the beginning of the linked list

b. From the end of the linked list

c. The node with specified data element

47. Write a program to create a singly linked list of n nodes and display it in reverse order.

48. Sort the elements in a linked list using

a. changing the values (swapping the values)

b. Changing the address (Swapping the address)

**Section 13: Polynomial using Linked List**

49. Polynomial using linked list - addition and multiplication

50. Linked list using names - insert, delete, display, sort, reverse, count

**Section 14- linked list**

51. Perform the respective operations on Linked Stack

52. Perform the respective operations on Linked Queue

53. Perform the respective operations on Circular Linked List

54. Perform the respective operations on Circular Linked Queue

55. Perform the respective operations on Doubly Linked List

56. Perform the respective operations on Circular doubly linked list - store string values as data part

**Section 15: Linear and Non Linear Search**

57. Implement linear search for finding an element in an array.

58. Apply binary search for searching an element in an array of integers

59. Apply binary search for searching an element in an array of Strings

**Section 16: Binary Search Tree**

60. Binary search tree insertion and display Traversal using inorder, preorder and postorder using recursion

61. Binary search tree insertion and display in-order without using recursion

62. .Binary search tree insertion and display pre-order without using recursion

63. Binary search tree insertion and display post-order without using recursion

64. Binary search tree insertion using names and display the names in ascending order using inorder traversal.

**Section -17: Graphs**

65. Demonstrate the data structure of adjacent matrix using arrays

66. Demonstrate the data structure of adjacent matrix using linked lists