MCA DS Lab Cycle

Section 1: Basics

1. Write programs to demonstrate the use of storage classes 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. Two-dimensional matrix: using functions
   a. Addition
   b. Subtraction
   c. Multiplication
   d. Transpose   [Need to discuss before implementing]
   e. Determinant
7. Display the array elements in the same order using a recursive function
8. Display array elements in reverse order using a recursive function

Section 2: Stack

9. Implement stack operations using arrays.
10. Reverse a string using Stack

Session 3: Stack

11. Convert an expression from infix to postfix using stack
12. Evaluate an expression using stack

Session 4: Struct

13. Define a structure for dates with dd/mm/yyyy. Provide functions for reading, displaying and comparing two dates are equal or not
14. 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 5: Polynomials using Array

15. Read a polynomial and display it; use array
16. Add two polynomials using the array itself

Session 6: Polynomials using Structure

17. Read a polynomial and display it; use structure array
18. Add two polynomials
19. Subtract two polynomials
20. Multiply two polynomials

****

Session 7: Dynamic Memory Allocation

21. Implement a) malloc , b) calloc and c) free functions
22. Use malloc to read n integers and find the mean.
23. Use calloc to read n numbers and find the mode.
24. 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.
25. Use realloc to implement varchar for any length.

Session 8- Queue

26. Implement Queue using array
27. Implement priority queue

Section 9- Linked List- Basics

1. Demonstrate a linked list creation and display
2. 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.
3. 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
4. Write a program to create a singly linked list of n nodes and display it in reverse order.
5. Sort the elements in a linked list using
    a. changing the values (swapping the values)
    b. Changing the address (Swapping the address)

Section 10: Polynomial using Linked List

1. Polynomial using linked list - addition and multiplication
2. Linked list using names - insert, delete, display, sort, reverse, count

Section 11- linked list

1. Perform the respective operations on the following [Separate Question]
    a. Linked Stack
    b. Linked Queue

Section 12: Binary Search Tree

1. Binary search tree insertion and display Traversal using inorder, preorder and postorder using recursion
2. Binary search tree insertion and display in-order without using recursion
3. .Binary search tree insertion and display pre-order without using recursion
4. Binary search tree insertion and display post-order without using recursion
5. Binary search tree insertion using names and display the names in ascending order using inorder traversal.

Section -13: Graphs

1. Demonstrate the data structure of adjacent matrix using arrays
2. Demonstrate the data structure of adjacent matrix using linked lists