B. P. PODDAR INSTITUTE OF MANAGEMENT AND TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

AY: 2022-2023
COURSE: DATA STRUCTURE & ALGORITHM LAB
PAPER CODE: PCC-CS391

ASSIGNMENT LIST

Lab 1: 1D Array

1. Write a C program to insert and delete an element from a 1-D static array.
2. Write a C program to reverse the elements present in a 1-D static array.
3. Write a C program to delete duplicates elements in a 1-D static array.
4. Write a C program to find the largest and smallest element present in a 1-D static array.
5. Write a C program to find the second largest and second smallest element present in a 1-D static array.
6. Write a C program for linear search and binary search.

Lab 2: 2D Array

7. Write a C program to store the elements in a 2D array and display it & represent it in row major order & display it.
8. Write a C program to test a given matrix is sparse or not. If it is sparse then represent it as 3-tuple format.
9. Write a C program to find the transpose of a sparse matrix.
10. Write a C program to find the matrix multiplication of two matrices.

Lab 3: Singly and Circular Linked list

11. Write a C program for representation of singly linked lists (create and display it) and then implementation of relevant operations – add, delete from beginning, end at and after specified locations (i.e., before and after a given node).
12. Write a C program for representation of circular linked lists (create and display it) and then implementation of relevant operations – add, delete from beginning, end and at and after specified locations (i.e., before and after a given node).

Lab 4: Doubly Linked list

13. Write a C program for representation of doubly linked lists (create and display it) and then implementation of relevant operations – add, delete from beginning, end and at and after specified locations (i.e., before and after a given node).

Lab 5: Implementations of Stack

14. Write a C program to implement a stack using array, implementation including the function to check whether the stack is empty. Push an element into the stack; pop an element from a stack. Return the top element from the stack, display the stack elements.
15. Write a C program to implement a stack using linked list, implementation including the function to check whether the stack is empty. Push an element into the stack; pop an element from a stack. Return the top element from the stack, display the stack elements.

Lab 6: Applications of Stack

16. Write a C program to implement "Tower of Hanoi" problem.
17. Write a C program to reverse a string using stack.
18. Write a C program to convert infix expression to postfix expression.
19. Write a C program to evaluate of postfix expression.

Lab 7: Applications of Linked list

20. Write a C program for reversing the singly linked list.
21. Write a C program for merging two linked lists (SLL).
22. Write a C program for linked representation of polynomials and also perform addition, subtraction of two polynomials.

Lab 8: Implementations of Queue

23. Write a C program for array representation of queue and implement of basic queue operations.
24. Write a C program for linked list representation of queue and implement of basic queue operations.
25. Write a C program for array representation of queue using circular queue concept.
26. Write a C program for implementation of Deque.

Lab 9: Nonlinear data structure: Tree and Graph

27. Write a C program to create a binary search tree (insert and delete) and implement recursive traversal of the tree – i.e., post-order, pre-order, and in-order traversal.
28. Write a C program to create a binary search tree and implement non-recursive traversal of a tree – i.e., post-order, pre-order, and in-order traversal.
29. Write a C program to implement the two-way threading (threaded binary tree).
30. Write a C program for implementation of AVL tree and its operations.
31. Write a C program to create and represent a graph using adjacency matrix.

Lab 10: Sorting & Hashing

32. Write a C program for implementing bubble sort.
33. Write a C program for implementing insertion sort.
34. Write a C program for implementing selection sort.
35. Write a C program for implementing quick sort.
36. Write a C program for implementing merge sort.
37. Write a C program to implement hashing technique.

Lab 11: Beyond the syllabus

38. Write a C program for implementing breadth first search (BFS) in a graph.
39. Write a C program for implementing depth first search (DFS) in a graph.
40. Write a C program to convert infix to equivalent postfix.
41. Write a C program for implementation of Red-Black tree and its operations.