Subject : 2305CS201 – Data Structure

Faculty: Prof. Nidhi K. Chitroda

LAB	PRACTICAL
1.	**Hands on practice to get familiar with basic C programming concepts** 1. Write a program to find factorial of a number using loop and recursion. [A] 2. Write a program to find factors of a given number. [A] 3. Write a program to check whether a number is prime or not. [A] 4. Write a program to find GCD using loop and recursion. [B] 5. Write a program to calculate power using loop and recursion. [B] 6. Write a program to display prime numbers between two intervals. [C]
2.	**Regular operations on 1-D Array Data Structure** 1. Read n numbers in an array and print their sum and average. [A] 2. Write a program to find the largest element in an array. [A] 3. Read n numbers in an array then read two different numbers, replace 1st number with 2nd number in an array and print the final array. [A] 4. Write a program to copy all the elements of one array to another array. [B] 5. Read n numbers in an array and print it in reverse order. [B] 6. Write a program to find common elements between two arrays. [C] 7. Write a program to remove duplicates from sorted array. [C]
3.	**Regular operations on 2-D Array Data Structure** 1. Read two 2x2 matrices and perform addition of matrices into third matrix and print it. [A] 2. Read two matrices, first 3x2 and second 2x3, perform multiplication operation and store result in third matrix and print it. [B] 3. Write a program to find transpose of a square matrix. [C]
4.	**Implementation of the Pointer concept** 1. Read n numbers in an array and print it using pointer. [A] 2. Write a C program to swap two numbers, calling an UDF by value. [A] 3. Write a C program to swap two numbers, calling an UDF by reference. [A] 4. Write a program to find largest element in the array using Pointer. [B] 5. Write a program to check if the string is a palindrome or not using Pointer. [C]
5.	**Implementation of the Structure concept** 1. Create a structure Employee_Detail (Employee_id, Name, Designation, Salary). Write a program to read the detail from user and print it. [A] 2. Create an array of structure Student_Detail (Enrollment_no, Name, Sem, CPI) for 5 students, scan their information and print it. [B] 3. Create a structure Employee_Detail (Employee_id, Name, Designation, Salary). Write a program to read the detail from user and print it using Structure Pointer. [B] 4. Write a program to add two complex numbers by passing structure to a Function. [C]
6.	**Implementation of Data Structure Stack** 1. Write algorithms to perform following operations on Stack: PUSH, POP, PEEP, CHANGE and DISPLAY. [A] 2. Write a menu driven program to perform following operations on Stack: PUSH, POP, PEEP, CHANGE and DISPLAY. [B] 3. Write a program to reverse a string using Stack. [C]
7.	**Implementation of Stack application: Converting Infix Expression to Postfix** 1. Convert the following infix expressions to postfix expressions: [A]

	◦ (A + B * C / D - E + F / G / (H + I)) ◦ (A + B) * C + D / (B + A * C) + D 2. Write an algorithm to convert infix to postfix notation. [A] 3. Write a program to convert the given infix notation to postfix notation. [B] 4. Write a program to convert postfix notation to infix notation. [C]
8.	**Implementation of Stack application: Converting Infix Expression to Prefix** 1. Convert the following infix expressions to prefix expressions: [A] ◦ (A + B * C / D - E + F / G / (H + I)) ◦ (A + B) * C + D / (B + A * C) + D 2. Write an algorithm to convert infix to prefix notation. [B] 3. Write a program to convert the given infix notation to prefix notation. [C]
9.	**Implementation of Stack application: Evaluating Postfix expression** 1. Evaluate the following expressions: [A] ◦ 5, 4, 6, +, *, 4, 9, 3, /, +, * ◦ 7, 5, 2, +, *, 4, 1, 1, +, /, - 2. Write an algorithm to evaluate post-fix expression using Stack. [A] 3. Write a program for evaluation of post-fix Expression having single digit operand using Stack. [B] 4. Write a program for evaluation of post-fix Expression having more than on digit operand using Stack. [C]
10.	**Implementation of Stack application: Evaluating Prefix expression** 1. Evaluate the following expressions: [A] ◦ *, +, 6, 9, -, 3, 1 ◦ +, -, *, 2, 2, 1, 16, 8, 5 2. Write an algorithm to evaluate pre-fix expression using Stack. [B] 3. Write a program for evaluation of pre-fix Expression using Stack. [C]
11.	**Implementation of Data Structure Simple Queue** 1. Write algorithms to perform following operations on Queue: ENQUEUE and DEQUEUE. [A] 2. Write a menu driven program to perform following operations on Simple Queue: ENQUEUE, DEQUEUE and DISPLAY. [B] 3. Write a program to reverse the elements of Queue. [C]
12.	**Implementation of Data Structure Circular Queue** 1. Write algorithms to perform following operations on Circular Queue: ENQUEUE and DEQUEUE. [A] 2. Implement Circular Queue using array that performs following operations: [B] ◦ ENQUEUE ◦ DISPLAY ◦ DEQUEUE 3. Consider the following circular queue having 6 memory cells. Front=2, Rear=4 Queue: _, A, C, D, _, _. Describe queue as following operation take place: [C] ◦ F is added to the queue ◦ Two letters are deleted ◦ R is added to the queue ◦ S is added to the queue ◦ One letter is deleted
13.	**Implementation of Data Structure Double Ended Queue** 1. Write algorithms to perform following operations on Double Ended Queue: ENQUEUE and DEQUEUE. [A] 2. Implement Double Ended Queue that performs following operations: [B] ◦ ENQUEUE Front

	◦ ENQUEUE Rear ◦ DISPLAY ◦ DEQUEUE Front ◦ DEQUEUE Rear 3. Consider a Deque with size 6, initially empty. Perform the following sequence of operations: ◦ ENQUEUERear(10) ◦ ENQUEUEFront(20) ◦ ENQUEUERear(30) ◦ DEQUEUEFront() ◦ ENQUEUEFront(40) What will the Deque look like after all operations? [C]
14.	**Implementation of Dynamic Memory Allocation concept** 1. Explain following functions for dynamic memory allocation: [A] ◦ Malloc () ◦ Calloc() ◦ Realloc() ◦ Free () 2. Write a program to get n elements of an array from user and print those elements using pointer. [B] 3. Write a program to display n elements and sum of those elements using dynamic memory allocation. Also release the memory occupied after displaying [C]
15.	**Implementation of Data Structure Singly Linked List – Insertion** 1. Write algorithms to perform following operations on Singly Linked List: [A] ◦ Insert a node at the beginning of the linked list ◦ Insert a node at the end of the linked list 2. Write a menu driven program to implement following operations on the singly linked list: ◦ Insert a node at the beginning of the linked list [A] ◦ Insert a node at the end of the linked list [A] ◦ Display the list [B] ◦ Count number of nodes [C]
16.	**Implementation of Data Structure Singly Linked List – Deletion** 1. Write algorithms to perform following operations on Singly Linked List: [A] ◦ Delete a first node ◦ Delete a last node ◦ Delete a specific node 2. Write a menu driven program to implement following operations on the singly linked list: ◦ Delete the first node of the linked list [A] ◦ Delete the last node of the linked list [A] ◦ Display the list [B] ◦ Delete a specific node [C]
17.	**Implementation of Data Structure Ordered Linked List - Insertion** 1. Write an algorithm to do ordered insertion in the Singly Linked List. [A] 2. Write a menu driven program to implement following operations on the Ordered Linked List. [B] ◦ Insert a node such that linked list is in ascending order ◦ Display the list 3. Write a program which allows user to insert unique node values only in ordered linked list. [C]

18.	**Implementation of Data Structure Ordered Linked List - Deletion** 1. Write an algorithm to do ordered deletion in the Singly Linked List. [A] 2. Write a menu driven program to implement following operations on the Ordered Linked List. [B] ◦ Delete a specific node such that linked list is in ascending order ◦ Display the list 3. Write a program which removes duplicate nodes from ordered linked list if any. [C]
19.	**Implementation of Stack and Queue using Linked List** 1. Write a program to implement stack using linked list. [A] 2. Write a program to implement queue using linked list. [B] 3. Write a program to implement queue using linked list. [C]
20.	**Implementation of Data Structure Circular Linked List - Insertion** 1. Write algorithms to perform following operations on Circular Linked List: [A] ◦ Insert a node at the beginning of the circular linked list ◦ Insert a node at the end of the circular linked list 2. Write a menu driven program to implement following operations on the Circular Linked List. ◦ Insert a node at the beginning of the circular linked list [B] ◦ Insert a node at the end of the circular linked list [B] ◦ Display the list [B] ◦ Count the nodes [C]
21.	**Implementation of Data Structure Circular Linked List - Deletion** 1. Write algorithms to perform following operations on Circular Linked List: [A] ◦ Delete a node at the beginning of the circular linked list ◦ Delete a node at the end of the circular linked list 2. Write a menu driven program to implement following operations on the Circular Linked List. ◦ Delete a node at the beginning of the circular linked list [B] ◦ Delete a node at the end of the circular linked list [B] ◦ Display the list [B] ◦ Delete a specific node [C]
22.	**Implementation of Data Structure Doubly Linked List** 1. Write algorithms to perform following operations on Doubly Linked List: [A] ◦ Insert a node ◦ Delete a node 2. Write a menu driven program to implement following operations on the Doubly Linked List. [B] ◦ Insert a node in middle of doubly linked list ◦ Delete a specific node in doubly linked list 3. Write a program to swap first node with last, second to second last and so on in doubly linked list. [C]
23.	**Implementation of Linear Search** 1. Find index of searched element x for given list: [A] ◦ arr[] = {1, 2, 3, 4}, x = 3 ◦ arr[] = {10, 8, 30, 4, 5}, x = 5 ◦ arr[] = {10, 8, 30}, x = 6 2. Write algorithm for linear search. [A] 3. Write a program to implement Linear/Sequential Search. [B] 4. Write a program to implement Linear Search using recursion. [C]
24.	**Implementation of Binary Search** 1. Write algorithm for binary search. [A]

	2. Write a program to implement Binary Search using loop. [B] 3. Write a program to implement Binary Search using recursion. [C]
25.	**Implementation of Bubble sort and Selection sort** 1. Show step by step sorting using bubble sort and selection sort algorithm for given array: [A] ◦ arr[] = {38, 27, 43, 10} ◦ arr[] = {10, 80, 40, 30, 90, 40} 2. Write algorithm for bubble sort and selection sort. [A] 3. Read n numbers in an array from user and sort them in ascending order using Bubble Sort algorithm and print sorted array. [B] 4. Read n numbers in an array from user and sort them in ascending order using Selection Sort algorithm and print sorted array. [C]
26.	**Implementation of Insertion, Quick and Merge sort** 1. Show step by step sorting using insertion sort, quick sort and merge sort algorithm for given array: [A] ◦ arr[] = {38, 27, 43, 10} ◦ arr[] = {10, 80, 40, 30, 90, 40} 2. Write algorithm for insertion sort, quick sort and merge sort. [A] 3. Read n numbers in an array from user and sort them in ascending order using Insertion Sort algorithm and print sorted array. [B] 4. Read n numbers in an array from user and sort them in ascending order using Quick Sort algorithm and print sorted array. [C] 5. Read n numbers in an array from user and sort them in ascending order using Merge Sort algorithm and print sorted array. [C]