Assignment List SY-Data Structures

Note: Students should complete all the assignments in the list. Instructor can select any one or two problem statements from the given assignment to implement during the lab hours, and the remaining problem statements in that assignment must be discussed and gottendone by the students. The practical exam will be based on all the assignments given in this list.

Assignment #	Concept / Unit Mapping	Problem #	Problem Statement	Evaluation Head Lab / HA
1	Unit - I Arrays	1	Implement **basic string operations** such as length calculation, copy, reverse, and concatenation using character single dimensional arrays without using built-in string library functions.	
		2	Write a program to construct and verify a **magic square** of order 'n' (for both even & odd) such that all rows, columns, and diagonals sum to the same value.	
		3	Implement **matrix multiplication and analyse** its performance using row-major vs column-major order access patterns to understand how memory layout affects cache performance.	
		4	Develop a program to identify and efficiently store a **sparse matrix** using compact representation and perform basic operations like display and simple transpose.	
		5	Develop a program to compute the **fast transpose** of a sparse matrix using its compact (triplet) representation efficiently.	
2	Unit – I Searching and Sorting	1	In Computer Engg. Dept. of VIT there are S.Y., T.Y., and B.Tech. students. Assume that all these students are on ground for a function. We need to identify a student of S.Y. div. (X) whose name is "XYZ" and roll no. is "17". Apply **appropriate Searching method** to identify the required student.	
		2	WAP to implement **Bubble sort and Quick Sort** on a 1D array of Student structure (contains student_name, student_roll_no, total_marks), with key as student_roll_no. And count the number of swap performed by each method.	
		3	Write a program to input marks of n students Sort the marks in ascending order using the **Quick Sort** algorithm without using built-in library functions and analyse the sorting algorithm pass by pass. Find the minimum and maximum marks using **Divide and Conquer** (recursively).	
		4	Write a program using **Bubble sort algorithm**, assign the roll nos. to the students of your class as per their previous years result. i.e. topper will be roll no. 1 and analyse the sorting algorithm pass by pass.	
		5	Write a program to arrange the list of employees as per the average of their height and weight by using **Merge and Selection** sorting method. Analyse their time complexities and conclude which algorithm will take less time to sort the list.	

3	Unit- II	1	Implementation of Singly Linked List to Manage ‘Vertex Club’ Membership Records. The **Department of Computer Engineering** has a student club named **‘Vertex Club’** for second, third, and final year students. The **first member is the President** and the **last member is the Secretary**. Write a C++ program to: ● Add/delete members (including President/Secretary) ● Count members ● Display members ● Concatenate two division lists Also implement: reverse, search by PRN, and sort by PRN operations.	
		2	The **ticket reservation system for Galaxy Multiplex** is to be implemented using a **C++ program**. The multiplex has **8 rows**, with **8 seats in each row**. A **doubly circular linked list** will be used to track the availability of seats in each row.Initially, assume that some seats are randomly booked. An **array will store head pointers for each row’s linked list**.The system should support the following operations: **a)** Display the current list of available seats. **b)** Book one or more seats as per customer request. **c)** Cancel an existing booking when requested.	
		3	Develop a **C++ program** to store and manage an **appointment schedule for a single day**. Appointments should be scheduled randomly using a **linked list**. The system must define the **start time, end time**, and specify the **minimum and maximum duration allowed for each appointment slot**. The program should include the following operations: **a)** Display the list of currently available time slots. **b)** Book a new appointment within the defined time limits. **c)** Cancel an existing appointment after validating its time, availability, and correctness. **d)** Sort the appointment list in order of appointment times. **e)** Sort the list based on appointment times using **pointer manipulation** (without swapping data values).	
		4	In the **Second Year Computer Engineering class**, there are two groups of students based on their favorite sports:	

			● **Set A** includes students who like **Cricket**. ● **Set B** includes students who like **Football**. Write a **C++ program** to represent these two sets using **linked lists** and perform the following operations: **a)** Find and display the set of students who like **both Cricket and Football**. **b)** Find and display the set of students who like **either Cricket or Football, but not both**. **c)** Display the **number of students who like neither Cricket nor Football**.	
		5	Write a **C++ program** to store a **binary number** using a **doubly linked list**. Implement the following functions: **a)** Calculate and display the **1's complement** and **2's complement** of the stored binary number. **b)** Perform addition of **two binary numbers** represented using doubly linked lists and display the result.	
4	Unit- II	1	Write a **C++ program** to implement a **Set** using a **Generalized Linked List (GLL)**. For example: Let **S = { p, q, {r, s, t, {}, {u, v}, w, x, {y, z}, a1, b1} }** Store this structure using a **Generalized Linked List** and display the elements in correct **set notation format**.	
		2	WAP to perform addition o f two polynomials using singly linked list.	
		3	Implement Bubble sort using Doubly Linked List	
		4	WAP to create a doubly linked list and perform following operations on it.A) Insert (all cases) 2. Delete (all cases).	
		5	Given a list, split it into two sublists — one for the front half, and one for the back half. If the number of elements is odd, the extra element should go in the front list. So FrontBackSplit() on the list {2, 3, 5, 7, 11} should yield the two lists {2, 3, 5} and {7, 11}. Getting this right for all the cases is harder than it looks. You should check your solution against a few cases (length = 2, length = 3, length=4) to make sure that the list gets split correctly near the shortlist boundary conditions. If it works right for length=4, it probably works right for length=1000. You will probably need special case code to deal with the (length <2) cases.	

5	Unit-III Stack	1	WAP to build a simple stock price tracker that keeps a history of daily stock prices entered by the user. To allow users to go back and view or remove the most recent price, implement a stack using a linked list to store these integer prices. Implement the following operations: 1. record(price) – Add a new stock price (an integer) to the stack. 2. remove() – Remove and return the most recent price (top of the stack). 3. latest() – Return the most recent stock price without removing it. 4. isEmpty() – Check if there are no prices recorded.	
		2	Convert given infix expression Eg. a-b*c-d/e+f into postfix form using stack and show the operations step by step.	
		3	Write a program to implement multiple stack i.e more than two stack using array and perform following operations on it. A. Push B. Pop C. Stack Overflow D. Stack Underflow E. Display	
		4	You are given a string containing only parentheses characters: '(', ')', '{', '}', '[', and ']'. Your task is to check whether the parentheses are balanced or not. A string is considered balanced if: 1. Every opening bracket has a corresponding closing bracket of the same type 2. Brackets are closed in the correct order.	
		5	You are given a postfix expression (also known as Reverse Polish Notation) consisting of single-digit operands and binary operators (+, -, *, /). Your task is to evaluate the expression using stack and return its result.	
6	Unit-III Queues	1	Write a program to keep track of patients as they checked into a medical clinic, assigning patients to doctors on a first-come, first-served basis.	
		2	Pizza parlour accepting maximum n orders. Orders are served on an FCFS basis. Order once placed can’t be cancelled. Write C++ program to simulate the system using circular QUEUE.	
		3	Write a program that maintains a queue of passengers waiting to see a ticket agent. The program user should be able to insert a new passenger at the rear of the queue, Display the passenger at the front of the Queue, or remove the passenger at the front of the queue. The program will display the number of passengers left in the queue just before it terminates.	

		4	Write a program to implement multiple queues i.e. two queues using array and perform following operations on it. A. Add Queue, B. Delete from Queue, C. Display Queue	
		5	In a call center, customer calls are handled on a first-come, first-served basis. Implement a queue system using Linked list where: ● Each customer call is enqueued as it arrives. ● Customer service agents dequeue calls to assist customers. ● If there are no calls, the system waits.	
7	Unit-IV	1	Write a program to perform Binary Search Tree (BST) operations (Create, Insert, Delete,Levelwise display)	
		2	Write a program to perform Binary Search Tree (BST) operations (Count the total number of nodes, Compute the height of the BST, Mirror Image).	
		3	Write a Program to create a Binary Tree Search and Find Minimum/Maximum in BST	
		4	Write a Program to create a Binary Tree and perform following Nonrecursive operations on it. a. Inorder Traversal b. Preorder Traversal c. Display Number of Leaf Nodes d. Mirror Image	
		5	Write a Program to create a Binary Tree and perform the following Recursive operations on it. a. Inorder Traversal b. Preorder Traversal c. Display Number of Leaf Nodes d. Mirror Image	
8	Unit-IV	1	Write a program, using trees, to assign the roll nos. to the students of your class as per their previous years result. i.e topper will be roll no. 1	
		2	Write a program to illustrate operations on a BST holding numeric keys. The menu must include: • Insert • Delete • Find • Show	
		3	Write a program to efficiently search a particular employee record by using Tree data structure. Also sort the data on emp-id in ascending order.	
		4	**Write a program to implement** a product inventory management system for a shop using a search tree data structure. Each product must store the following information: ● Unique Product Code ● Product Name ● Price ● Quantity in Stock	

			● Date Received ● Expiration Date **Implement the following operations:** 1. **Insert** a product into the tree (organized by product name). 2. **Display all items** in the inventory using inorder traversal. 3. **List expired items** in prefix (preorder) order of their names.	
		5	**Write a program to implement** deletion operations in the product inventory system using a search tree. Each product must store the following information: ● Unique Product Code ● Product Name ● Price ● Quantity in Stock ● Date Received ● Expiration Date **Implement the following operations:** 1. **Delete a product** using its unique product code. 2. **Delete all expired products** based on the current date.	
9	Unit-V	1	Write a Program to accept a graph from a user and represent it with Adjacency Matrix and perform BFS and DFS traversals on it.	
		2	Write a Program to implement Prim's algorithm to find minimum spanning tree of a user defined graph. Use Adjacency List to represent a graph.	
		3	Write a Program to implement Kruskal's algorithm to find the minimum spanning tree of a user defined graph. Use Adjacency List to represent a graph.	
		4	Write a Program to implement Dijkstra's algorithm to find shortest distance between two nodes of a user defined graph. Use Adjacency List to represent a graph.	

		5	Write a Program to accept a graph from a user and represent it with Adjacency List and perform BFS and DFS traversals on it.	
10	Unit-V	1	Write a Program to implement Kruskal's algorithm to find the minimum spanning tree of a user defined graph. Use Adjacency Matrix to represent a graph.	
		2	Write a Program to implement Dijkstra's algorithm to find shortest distance between two nodes of a user defined graph. Use Adjacency Matrix to represent a graph.	
		3	Write a Program to implement Prim's algorithm to find minimum spanning tree of a user defined graph. Use Adjacency List to represent a graph.	
		4	Write a Program to implement Kruskal's algorithm to find the minimum spanning tree of a user defined graph. Use Adjacency List to represent a graph.	
		5	Write a Program to implement Dijkstra's algorithm to find shortest distance between two nodes of a user defined graph. Use Adjacency List to represent a graph.	
11	Unit-VI	1	Implement a hash table with collision resolution using linear probing.	
		2	Implement collision handling using separate chaining.	
		3	Implement collision resolution using **linked lists**.	
		4	Store and retrieve student records using roll numbers.	
		5	WAP to simulate a faculty database as a hash table. Search a particular faculty by using MOD as a hash function for linear probing method of collision handling technique. Assume suitable data for faculty record.	
12	Unit-VI	1	WAP to simulate a faculty database as a hash table. Search a particular faculty by using 'divide' as a hash function for linear probing with chaining without replacement method of collision handling technique. Assume suitable data for faculty record.	
		2	WAP to simulate a faculty database as a hash table. Search a particular faculty by using MOD as a hash function for linear probing with chaining with replacement method of collision handling technique. Assume suitable data for faculty record.	
		3	WAP to simulate employee databases as a hash table. Search a particular faculty by using Mid square method as a hash function for linear probing method of collision handling technique. Assume suitable data for faculty record.	
		4	WAP to simulate student databases as a hash table. a student database management system using **hashing techniques** to allow **efficient insertion, search, and deletion** of student records.	
		5	Design and implement a **smart college placement portal** that uses **advanced hashing techniques** to efficiently manage student placement records with high performance and low collision probability, even under dynamic data growth.	