**Data Structure Lab Outline**

|  |  |
| --- | --- |
| Day | **Problem List** |
| **L1: ARRAY** | * **Lab class simple Problem list**  1. Implement a linear array with some random (integer/character) value.   Display/print the content of your array. Also display the array in reverse order.   1. Write a program which performs the following tasks:  - initialize an integer array of 10 elements in main()   - multiply each element of array by 3  - Then reversely push the value of array to another array and then print  Input: A []: 0 1 2 3 4 5 6 7 8 9  Output: B []: 27 24 21 18 15 12 9 6 3 0   1. Write a program which insert value in the middle of array.  * Suppose array initialize is : 11 12 13 15 16 17 * Insert value 14 in middle (kth position) of array * Output shows: 11 12 13 14 15 16 17  1. Write a program which delete value in the middle of array.  * Suppose array initialize is : 11 12 13 14 15 16 17 * Delete value 14 in middle (kth position) of array * Output shows: 11 12 13 15 16 17  1. Content of array: 34 21 78 4 2 8 23   Find the highest and lowest value from your array.  Sample Output:  Highest Value: 78  Lowest Value: 2   1. Implement a 2D-array with some random integer values. Display/print the content of your array in a linear fashion. 2. Implement a 2D-array with some random integer values. Display/print the content of your array as Matrix. 3. Declare two Matrices A [2] [3] = {1, 2, 3, 4, 5, 6} & B [2] [3] = {1, 2, 1, 2, 1, 2} and store the summation of A and B into another matrix C [2] [3]. 4. Declare two Matrices A [2] [3] = {1, 2, 3, 4, 5, 6} & B [3] [2] = {1, 2, 1, 2, 1, 2} and multiply them and print the resultant matrix. 5. Write a program to search a particular element from a 2D-array and print its position.   You can assume that A [2] [3] = {10, 25, 17, 28, -45, 14}  *Input*:  *Enter key to search: 14*  *Output*:  *Position: 1 2* |
| **L2: STRUCTURE, POINTER & STRING** | * **Lab class simple Problem list**  1. Define a struct Date variable called MyBirthDate and initialize it to September 21, 1967, in the dd/mm/yy format. Write a program to display the MyBirthDate. Take your own birth date from key board and display it too.   **Sample output:**  MyBirthDate : 21 9 1967  Enter your Birth date:  26 March 1971  My Birth Date is in : 26 March 1971   1. Write a program using function SWAP() and pointer variables to interchange two values from two different position x and y (also try for the content of two different arrays). 2. Write a C/C++ Program using struct variables to construct a student record list consisting of the following information in each record: student id (integer), student name (character string) and Marks(integer).   Do the following operations:   1. Store data for ten students and display the data on the screen. 2. Consider that students Marks and find out student name who get maximum and minimum mark 3. Search a record based on student name and display the information content. If the specified name is not present in the list an error message should be displayed. 4. Take an input string from user. Find out the length of string and number of words in the string. strlen() function using is not allowed.   **Input:** Bangladesh is my country **Output: Length:** 24 **Words:** 4   1. Take two strings as input and concatenate the two strings without using strcat() function.   **Input: String1:** American International **String2:** University Bangladesh  **Output:** American International University Bangladesh  \*\*Practice for output tracing.   1. Find out the index number from a string.   **Input:String:** I am a student of AIUB **Index Input:** of **Output:** Position of Index: 16   1. Insert a string inside of another string from any position.   **Input:**  **String1:** I am student  **String2:** of AIUB  **Position:** 14  **Output:** I am a student of AIUB   1. Find out sub string from a string.   **Input:** I am a student  **Enter Initial and Length:** 3 4  **Output:** am a   1. Create a function which is similar to the characteristic of strcpy(s1,s2) where s2 string is copied in s1 string.   **Input:String1:** AIUB  **String2:** CSE  **Output:String1:** CSE  **String2:** CSE   1. Create a function which is similar to the characteristic of strcmp(s1,s2) where compare two string. If equal returns 0, if s1 is alphabetically lower than s2 then returns<0 and otherwise returns>0   **Input1: String1:** AIUB **String2:** AIUB **Output1:** 0  **Input2:  String1:** AIUB **String2:** CSE **Output1:** -1 |
| **L3: STACK, QUEUE, Linear & Binary Search using ARRAY** | * **Lab class simple Problem list**  1. Write a C/C++ Program to construct a stack of integers and to perform the following operations on it:   a. Push  b. Pop  c. Display  The program should print appropriate messages for stack overflow, stack underflow, and stack empty.   1. Write a C/C++ Program to construct a Queue of integers and to perform the following operations on it:   a. Enqueue  b. Dequeue  c. Display  The program should print appropriate messages for Queue overflow, and Queue empty.  \*\*Practice for output tracing.   1. Write a C/C++ Program to simulate the working of a circular queue of integers using an array. Provide the following operations:   a. Enqueue  b. Dequeue  c. Display |
| **L4: LAB EXAM &STACK APPLICATION** | **Exam Lab Test 1**   * **Lab class simple Problem list**  1. Perform the Validity checking operation of an arithmetic expression using stack. 2. Write a C/C++ Program to convert an Infix to Postfix/prefix expression using stack. 3. Write a C/C++ Program to evaluate a postfix/prefix expression using stack. |
| **L5: SEARCHING & SORTING** | * **Lab class simple Problem list**  1. Bubble Sort 2. Write a C/C++ Program of an array for Binary Search Implementation   \*\*perform their time complexity   1. Selection Sort / Insertion Sort |
| **L6: OVERVIEW** | **Exam Lab Test 2 (mid-term)** |
| **Midterm Exam**  **Week 7** | |
| **L8: Linked List Simple Operation** | * **Lab class simple Problem list**  1. Create a singly linked list by adding node one by one at the end of the list. 2. Display your list 3. Count your nodes 4. Search an item into your linked list 5. Create a singly linked list by adding node one by one at the beginning of the list. 6. Insert a new item at a specific position    1. At the beginning    2. At the end    3. At the middle i.e.       1. After node x       2. Before node y 7. delete a node of a specific position    1. beginning node    2. end node    3. middle node i.e.       1. node of x value       2. ith node    4. any node 8. The entire above program repeat for doubly linked list. 9. Create a sorted singly list from a list of random number.   Sample input:  Take integer input from key board e.g.  Enter your node data = 5, 3, 7, 9, 2, 6, 11, 4  Required Output sequence after every ins:  Start -->5  Start -->3-->5  Start -->3-->5-->7  Start -->3-->5-->7-->9  Start -->2-->3-->5-->7-->9  Start -->2-->3-->5-->6-->7-->9  Start -->2-->3-->5-->6-->7-->9-->11  Start -->2-->3-->4-->5-->6-->7-->9-->11 |
| **L9: LinkListApplication** | **Exam Lab Test 3**   * **Lab class simple Problem list**  1. Stack Implementation Using Link List. 2. Queue Implementation Using Link List. |
| **L10: TREE** | * **Lab class simple Problem list**   1. Write a C/C++ Program  1. To construct a binary search tree of integers. 2. To traverse the tree using all the methods i.e., in order, preorder and post order. 3. To display the elements in the tree 4. To search an element on a given BST. Output shows data is found or not |
| **L11: Priority queues (Heap tree)** | * **Lab class simple Problem list**  1. Write a C / C++ Program 2. To construct a Max / Min Heap tree using array 3. To implement the Heap Sort Algorithm 4. To insert a new item into a Max / Min Heap. 5. To delete the root from a Max / Min Heap. |
| **L12: Graphs,**  **Sorting and searching and their complexity Analysis** | * **Lab class simple Problem list**  1. Representation of a graph with adjacency matrix. 2. Merge Sort / Quick Sort and their complexity analysis 3. Implementation of BFS |
| **L13:** | **Exam Lab Test 4 (Final term)** |

**List of Assignment:**

Assignment#1:

Suppose a given algorithm required two stacks A and B. Implement these two stacks in one array with array length n where n= n1+ n2. Assume that size of stack A is n1 and size of stack B is n2. Overflow will occur only when A and B together have more than n elements.

Assignment#2:

1. Implement a stack by linked list.
2. Implement a linear queue by linked list
3. Implement a circular queue by linked list

Assignment#3:

1. Create a singly linked list in a way the list will be always sorted.

Sample input:

Take integer input from key board e.g.

Enter your node data = 5, 3, 7, 9, 2, 6, 11, 4

Required Output sequence:

Start -->5

Start -->3-->5

Start -->3-->5-->7

Start -->3-->5-->7-->9

Start -->2-->3-->5-->7-->9

Start -->2-->3-->5-->6-->7-->9

Start -->2-->3-->5-->6-->7-->9-->11

Start -->2-->3-->4-->5-->6-->7-->9-->11

Assignment#4:

1. Create a BST using non-recursive function and display the inorder/preorder/postorder sequences. Show the simulation for your recursive inorder/preorder/postorder function call.