B.Sc.(H) Computer Science
Semester III
Lab based on Data Structures(LOCF)
<u>List of Practicals</u>

1. Given a list of N elements, which follows no particular arrangement, you are required to search an element x in the list. The list is stored using array data structure. If the search is successful, the output should be the index at which the element occurs, otherwise returns -1 to indicate that the element is not present in the list. Assume that the elements of the list are all distinct. Write a program to perform the desired task.

2. Given a list of N elements, which is sorted in ascending order, you are required to search an element x in the list. The list is stored using array data structure. If the search is successful, the output should be the index at which the element occurs, otherwise returns -1 to indicate that the element is not present in the list. Assume that the elements of the list are all distinct. Write a program to perform the desired task.

3. Write a program to implement singly linked list which supports the following operations:
   (i) Insert an element x at the beginning of the singly linked list
   (ii) Insert an element x at i^{th} position in the singly linked list
   (iii) Remove an element from the beginning of the singly linked list
   (iv) Remove an element from i^{th} position in the singly linked list.
   (v) Search for an element x in the singly linked list and return its pointer
   (vi) Concatenate two singly linked lists

4. Write a program to implement doubly linked list which supports the following operations:
   (i) Insert an element x at the beginning of the doubly linked list
   (ii) Insert an element x at i^{th} position in the doubly linked list
   (iii) Insert an element x at the end of the doubly linked list
   (iv) Remove an element from the beginning of the doubly linked list
   (v) Remove an element from i^{th} position in the doubly linked list.
   (vi) Remove an element from the end of the doubly linked list
   (vii) Search for an element x in the doubly linked list and return its pointer
   (viii) Concatenate two doubly linked lists

5. Write a program to implement circularly linked list which supports the following operations:
   (i) Insert an element x at the front of the circularly linked list
   (ii) Insert an element x after an element y in the circularly linked list
   (iii) Insert an element x at the back of the circularly linked list
   (iv) Remove an element from the back of the circularly linked list
   (v) Remove an element from the front of the circularly linked list
   (vi) remove the element x from the circularly linked list
   (vii)Search for an element x in the circularly linked list and return its pointer
   (viii) Concatenate two circularly linked lists

6. Implement a stack using Array representation

7. Implement a stack using Linked representation

8. Implement Queue using Circular Array representation

9. Implement Queue using Circular linked list representation

10. Implement Double-ended Queues using Linked list representation

11. Write a program to implement Binary Search Tree which supports the following operations:

    (i) Insert an element x

    (ii) Delete an element x

    (iii) Search for an element x in the BST and change its value to y and then place the node with value y at its appropriate position in the BST

    (iv) Display the elements of the BST in preorder, inorder, and postorder traversal

    (v) Display the elements of the BST in level-by-level traversal

    (vi) Display the height of the BST

12. Write a program, using templates, to sort a list of *n* elements. Give user the option to perform sorting using Insertion sort, Bubble sort or Selection sort.

13. Write a program to implement:

    (i) Diagonal Matrix using one-dimensional array.

    (ii) Lower Triangular Matrix using one-dimensional array

    (iii) Upper Triangular Matrix using one-dimensional array

    (iv) Symmetric Matrix using one-dimensional array.

14. Write a program to implement AVL Tree.

15. Write a Program to implement a priority queue using heap data structure.

16. Write a program to evaluate a prefix/postfix expression using stacks.