National University of Computer and Emerging Sciences

Laboratory Manual

for

Data Structures Lab

Course Instructor	Mr. Razi-ul-din
Lab Instructor(s)	Mamoona Akbar Mateen
Section	BSE-3B
Semester	Fall 2023

Department of Computer Science

FAST-NU, Lahore, Pakistan

Objectives:

In this lab, students will practice:

1. Linked List

Question 1: Link List

1. Implement a template class 'List' with nested 'Node' struct that contains data members: A template variable 'data' and a Node pointer 'next'. You may define any member functions, if required, for the template class.
2. Now using the above class, implement a linked list that supports the following operations:
   a. Insert at start `void insertAtStart(T const element);`
   b. Insert at end `void insertAtEnd(T const element);`
   c. Delete from start `void DeleteAtStart();`
   d. Delete from end `void DeleteAtEnd();`
   e. Print element head to tail void printForward() () const;
   f. Print element tail to head void PrintReverse()const;
   g. Return size of link list int size()const;
   h. Return middle element of link list int ReturnMiddle()const;.
   i. Return true if FRONT/TAIL is pointing to NULL otherwise false. bool IsEmpty();
   j. Return maximum element of link list int FindMax() const;
   k. It should enter the new Node with the value key, after the first occurrence of value val. If not found insert at Tail. void InsertAfter(val, key);
   l. It should enter the new Node with the value key, before the first occurrence of value val. If not found insert at Tail void InsertBefore(val, key);
   m. Make a function **insertSorted** that takes an element as argument and inserts in linked list in sorted order
   n. Make a function **DeleteDuplicates** that deletes the duplicate elements from this sorted list (Traverse only once)
   o. Destructor

Create a suitable main function to test the above functions.

Question 2:

Given a single link list of size n which is initialized by numbers from 1 to n in random order. User

deletes four elements at random from list. Write a function that finds which elements are

missing from the list.

E.g. singleLinkList

1->7>8->10->9->4->2->6->3->5

After user removed four elements from single link list.

singleLinkList

1->7->10->9->2->5

Missing elements

3->4->6->8