National University of Computer and Emerging Sciences

Lab Manual

for

Data Structures

Course Instructor	Ms. Arooj Khalil
Lab Instructor(s)	Ms. Mamoona Akbar Ms. Sonia Anum
Section	OOP BSE-3B
Semester	FALL 2022

Department of Computer Science
FAST-NU, Lahore, Pakistan

Lab Manual 03

Objectives:

After performing this lab, students shall be able to revise:

- ✓ Double Link list
- ✓ Stack using link list

Problem 1
Double link list

1. Implement a Struct ‘Node’ that contains three data members: A template variable ‘data’, Node pointer ‘next’ and Node pointer ‘prev’.
2. Now implement a double linked list class having two private data members Node pointer ‘head’ and Node pointer ‘tail’.
3. Now make an iterator class having one private data member Node pointer current. Please note that iterator class is a nested class of double linked list class. **(Note that iterator class is defined inside the List class)**
4. Now implement the following operations for iterator class:
   a. default constructor
   b. dereference operator
   c. post increment operator
   d. pre increment operator
   e. not equal operator
5. Now implement the following operations for linked list class:

a. begin iterator begin() const;
b. end iterator end() const;
c. Insert at start void insertAtHead(T const element);
d. Insert at end void insertAtTail (T const element);
e. Delete at Start void eraseAtHead ();
f. Delete at End void eraseAtTail();
g. Print element head to tail void printForward() ()const;
h. Print element tail to head void PrintReverse()const;
i. Return size of circular double link list. int size()const;
j. Return middle element of circular double link list int ReturnMiddle()const;.
k. Return true if FRONT/TAIL is pointing to NULL otherwise false. bool IsEmpty();
l. Return maximum element of circular double link list. int FindMax()const;
m. 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);
n. 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);
o. Make a function **insertSorted** that takes an element as argument and inserts in doubly linked list in sorted order
p. Make a function **DeleteDuplicates** that deletes the duplicate elements from this sorted list (Traverse only once)
q. Copy Constructor

r. Destructor

Create a suitable main function to test the above functions.

Problem 2
Infix to postfix using stack

Implement a Struct ‘Node’ that contains two data members: A template variable ‘data’ and Node pointer ‘next’.
Create a Stack that contain two data members, A Node variable ‘top’ and template variable ‘size’.
Now implement the following operations for stack class:

- It delete top element from stack. void pop();
- Insert in stack void push(T const element);
- Return true if top is pointing to NULL otherwise false. bool IsEmpty();
- Get top element from stack T gettopelement();

Make appropriate function according to your need.
Solve below these problem Infix to postfix and show its result.

- (AX * (BX * (((CY + AY) + BY) * CX)))
- ((H * ((((A + ((B + C) * D)) * F) * G) * E)) + J)