

National University of Computer and Emerging Sciences



Lab Manual
for
Data Structure

Course Instructor	Dr. Zareen Alamgir
Lab Instructor(s)	Ms. Mamoona Akbar Ms. Ammara
Section	DS BCS-3C
Semester	FALL 2022

Department of Computer Science
FAST-NU, Lahore, Pakistan

Lab Manual 04

Objectives:

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

- ✓ iterator
- ✓ Circular link list

Problem 1

1. Implement a Struct 'Node' that contains two data members: A template variable 'data', Node pointer 'next'.
2. Now implement a circular linked list class having one private data members 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 circular 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
 - f. 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 tail `void insertAtTail (T const element);`
 - d. remove all nodes from the CLL which contain element whose digit sum is even and find sum and product of remaining element

```
input: 9-> 11-> 34-> 6 -> 13 -> 21
output: 9->34->21
sum : 64
product : 6426
```

- e. Reverse CLL `void Reverselist();`

```
Input: 1 -> 2 -> 3 -> 4 -> 5 -> 6 -> 7 -> 8 -> 9 -> 10
Output: 10 -> 9 -> 8 -> 7 -> 6 -> 5 -> 4 -> 3 -> 2 -> 1
```

- f. Move all occurrence of an element to end of CLL
`void MoveOccurence(T keyvalue);`

Input: 1 -> 2 -> 2 -> 3

KEY VALUE = 2

Output: 1 -> 3 -> 2 -> 2

- g. Return size of circular link list. `int size()const;`
- h. Return true if TAIL is pointing to NULL otherwise false. `bool IsEmpty();`
- i. 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);`
- j. 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);`
- k. Make a function **DeleteDuplicates** that deletes the duplicate elements from sorted list (Traverse only once)
- l. Copy Constructor
- m. Destructor

Create a suitable main function to test the above functions.