Data Structures Lab Manual

Lab 2: Singly link list

Prepared for: Students of Data Structures

Department of Computer Science  
Fast School of Computing

# Objectives

* Create and manipulate singly linked lists.
* Implement common linked list operations (insertion, deletion, traversal, searching).

# Lab Outcomes

After completing this lab, students will be able to:  
1. Apply linked lists to solve practical problems.

## 1. Create Link List

***Note: Link List will be generic these are only examples*** just for your reference.

# Lab Task

## Display the list elements after creation.

*Example:* Input: 5 → 10 → 15  
Output: 5 10 15

1. Insert At beginning

Implement a function insertAtBeginning() to insert a new node at the start of the linked list.

Example: Initial list: 10 → 20 → 30  
Insert 5 → 5 → 10 → 20 → 30

1. Insert At End

Implement a function insertAtEnd() to insert a new node at the end of the list.

Example: Initial list: 10 → 20 → 30  
Insert 40 → 10 → 20 → 30 → 40

1. Insert at Specific Position

Implement a function insertAtPosition(int pos, int val) to insert a node at the given position.

Handle invalid positions gracefully.

Example: Initial list: 10 → 20 → 40  
Insert 30 at position 3 → 10 → 20 → 30 → 40

1. Delete from Beginning

Write a function deleteFromBeginning() to delete the first node of the linked list.

Example: Initial list: 10 → 20 → 30

After deletion → 20 → 30

1. Delete from End

Write a function deleteFromEnd() to delete the last node.

Example: Initial list: 10 → 20 → 30

After deletion → 10 → 20

1. Delete by Value

Write a function deleteFromEnd() to delete the last node.

Example: Initial list: 10 → 20 → 30

After deletion → 10 → 20

1. Search an Element

Implement a function search(int val) that returns the position of the element in the list, or -1 if not found.

Example: List: 5 → 10 → 15

Search 10 → Found at position 2

1. Reverse the Link List

Write a function to reverse the linked list.

Example: 10 → 20 → 30 → 30 → 20 → 10

# Student Tasks

1. Use a singly linked list to represent a polynomial of the form:  
   5x³ + 2x² + 3x + 4
2. Implement functions to:
3. Create polynomial using linked list.
4. Display polynomial in standard format.
5. Add two polynomials represented using linked lists.

# Submission Guidelines

- Submit your .cpp file with proper comments.  
- Make sure your program compiles and runs successfully.