# Lab # 7: Linked List

**Subject:** **Data Structures and Algorithms**

**Course:** **BESE15 B**

**Date:** **4th November, 10.**

---

**TASK:**

Implement singly linked list. Each node of linked list will store a character. Define following operations for the singly linked list:

- **InsertAtEnd:** Inserts new node at the end of linked list.
- **Delete:** Deletes a node from the linked list. The node to be deleted can be first, last or any node. This function will delete only the first occurrence of the character to be deleted.
- **IsEmpty**: Returns true if linked list is empty.
- **ListSize**: Returns total number of nodes present in the linked list.
- **DisplayList**: Displays all nodes present in the linked list.
- **Find:** Search linked list for the node specified by user.
- **ClearList**: Empties linked list (deletes all nodes).

Modify the program to adjust the following functions also.

- **Merge**: Takes a string as an input from user, reads one character at a time and stores that character at the end of the existing linked list.
- **Insert:** Inserts a node at the location specified by user. For example, if user gives location 0, the node will be inserted as a first node, if location doesn't exist, the node will be inserted as a last node. Consider following case:

| Location | 0 | 1 | 2 |
|----------|---|---|---|
| Node     | a | c | d |

If user wants to enter b at location 1, the linked list will be modified as follows:

| Location | 0 | 1 | 2 | 3 |
|----------|---|---|---|---|
| Node     | a | b | c | d |

- **DeleteAllOccurrences:** Deletes all occurrences of a character to be deleted.
- **DeleteFirstNode:** Deletes first node from the linked list.
- **DeleteLastNode:** Deletes last node from the linked list.

**Submission Deadline**: 22nd November, 10.