

This project implements a Singly Linked List in Python with various functionalities such as adding nodes, removing nodes, inserting nodes at specific positions, and displaying the list. The main features and operations provided by this implementation are as follows:

## Implementation Details

### 1. Class Definitions:

- i. Node Class: Represents a single node in the linked list.
  - data: Stores the value of the node.
  - next: Points to the next node in the list (initially set to None).
- ii. LinkedList Class: Manages the linked list operations.
  - head: Points to the first node in the linked list (initially set to None).

### 2. LinkedList Methods:

- i. Insertion Methods:
  - insertAtBegin(val): Inserts a new node with the specified value at the beginning of the list.
  - insertAtEnd(val): Inserts a new node with the specified value at the end of the list.
  - insertAtSpecificPoint(val, index): Inserts a new node with the specified value at a given position in the list.
- ii. Deletion Methods:
  - removeNodeByVal(val): Removes a node with the specified value from the list.
  - removeNodeByIndex(index): Removes a node at the specified position from the list.
- iii. Sorting Method:
  - Implements a Bubble Sort algorithm.
  - Traverses the linked list and compares each node's data with the subsequent nodes.
- iv. Utility Methods:
  - displayList(): Returns the current state of the list as a string.
  - countEle(): Returns the number of elements in the list.
  - lengthOfList(): Returns a string indicating the length of the list.
  - exit(): Returns the final state of the list before exiting.

### 3. Interactive Menu:

- \* A user-friendly menu is provided for interacting with the linked list.
- \* Users can choose from various options to perform different operations such as adding, removing, and displaying nodes.
- \* The menu continues to prompt for user input until the user chooses to exit.

The interactive menu allows users to perform the following operations:

1. Create a new list by adding multiple elements.
2. Add a node at the beginning of the list.
3. Add a node at a specific position in the list.
4. Remove a node by its value.
5. Remove a node by its position.
6. Display the current state of the linked list.
7. Return the length of the list.
8. Exit the program.