1 | Experiential Learning Component

1.1 | UNIT I : Experiential Learning - 3

[Level-2: 4Q, Level-3: 3Q]

1. **LEVEL - 2 : Insert in Order:** Assuming the linked list stores integers, write a function that inserts a new node in such a way that the linked list remains sorted in increasing order.
   - Hint: Traverse the list to find the appropriate position where the new node's value fits, then insert the node at this position.
2. **LEVEL - 2 : Delete at Position:** Write a function to delete a node at a specific position.
   - Hint: Traverse to the specific position and adjust pointers.
3. **LEVEL - 2 : Delete by Content:** Implement a function to delete a node based on content.
   - Hint: Find the node with matching content and remove it by updating pointers.
4. **LEVEL - 2 : Reverse Linked List:** Implement a function that reverses the linked list.
   - Hint: You may need to keep track of three pointers as you reassign each node's next pointer to its previous node.
5. **LEVEL - 3 : Merge Two Sorted Lists:** Write a program that takes two sorted linked lists and merges them to produce a single sorted linked list.
   - Hint: Use a new list to store the merged result, comparing the current nodes of both lists and appending the smaller one.
6. **LEVEL - 3 : Sorting a Linked List**
   - **Sort Linked List Using New Header**: Implement a function that sorts a linked list of integers by creating a new header node and transferring nodes from the original list to the new list in sorted order.
     - **Hint**: Traverse the original list, remove the first node, and insert it into the new list at the correct position. Continue this process until the original list is empty.
7. **LEVEL - 3 : Polynomial Manipulation**
   - **Addition of Polynomials**: Implement a function that adds two polynomials represented as linked lists, where each node contains a coefficient and a degree.
     - **Hint**: Traverse both lists simultaneously, summing the coefficients of terms with the same degree. Add the resulting terms to a new polynomial linked list.