

1 | Experiential Learning Component

1.1 | UNIT I : Experiential Learning - 2

[Level-1: 6Q, Level-2: 1Q]

1. **LEVEL - 1 : Construct a Basic Node:** Define a structure in C that represents a single node in a linked list containing an integer data and a pointer to the next node.
 - Hint: Use `struct` in C to define your node. The node typically contains data of type `int` and a pointer to the next `struct`.
2. **LEVEL - 1 : Initialization Function:** Write a function that initializes a linked list and returns a pointer to the head node.
 - Hint: This function should return a `NULL` pointer to signify that the list is empty at the beginning.
3. **LEVEL - 1 : Insert at Front:** Write a function that inserts a new integer node at the front of the linked list.
 - Hint: The new node should point to the current head of the list, and then the head pointer should be updated to the new node.
4. **LEVEL - 1 : Insert at Rear:** Write a function that appends a new integer node at the end of the linked list.
 - Hint: Traverse to the end of the list and then point the last node to the new node. Ensure that the new node points to `NULL`.
5. **LEVEL - 1 : Delete at Front:** Write a function to delete the front node of a linked list.
 - Hint: Adjust the head pointer to remove the first element.
6. **LEVEL - 1 : Delete at Rear:** Implement a function to delete the last node of a linked list.
 - Hint: Traverse the list to find and remove the last element.
7. **LEVEL - 2 : Insert at Position:** Write a function that inserts a new node at a given position in the linked list. If the position is out of range, print an error message.
 - Hint: Traverse to the node before the desired position and adjust the pointers accordingly. Validate the position first.