**TESTING REPORT**

***UNROLLED LINKED LIST***

**=>Test Case 1: Insertion**

1. **Insert Single Element:**
   * Input: **1**
   * Expected Output: List displays **1**.
2. **Insert Multiple Elements:** 
   * Input: **2, 4, 6, 8, 10**
   * Expected Output: List displays:

2 4 6 8 10

1. **Insert into a Full Node (Triggering Node Split):** 
   * Input: **12** (to fill up a node and trigger split)
   * Expected Output: New node created and list displays appropriately split nodes.

**=>Test Case 2: Deletion**

1. **Delete Element from Head Node:** 
   * Input: **2**
   * Expected Output: **2** is removed from the list.
2. **Delete Element from Non-Head Node:** 
   * Input: **8**
   * Expected Output: **8** is removed from the list.
3. **Delete Non-existent Element:** 
   * Input: **100**
   * Expected Output: No change in the list and appropriate message indicating element not found.
4. **Delete All Elements from a Node:** 
   * Input: Delete all elements from a node until empty.
   * Expected Output: Node should be removed from the list if empty after deletion.

**=>Test Case 3: Search**

1. **Search for Existing Element:** 
   * Input: **6**
   * Expected Output: Message indicating **6** is found in the list.
2. **Search for Non-existent Element:** 
   * Input: **99**
   * Expected Output: Message indicating **99** is not found in the list.
3. **Search in an Empty List:** 
   * Input: Search for any element.
   * Expected Output: Message indicating list is empty.

**=>Test Case 4: Display**

1. **Display an Empty List:** 
   * Input: Display command.
   * Expected Output: Message indicating list is empty.
2. **Display List with Single Node:** 
   * Input: List with a few elements.
   * Expected Output: Elements displayed in a single line.
3. **Display List with Multiple Nodes:** 
   * Input: List with more elements than can fit in a single node.
   * Expected Output: Elements displayed across multiple nodes, demonstrating proper structure.

**=>Boundary Conditions**

1. **Insert at Maximum Node Capacity:** 
   * Input: Insert enough elements to fill a node.
   * Expected Output: Proper splitting of nodes when the threshold is reached.
2. **Delete to Empty Nodes:** 
   * Input: Delete elements until a node is empty.
   * Expected Output: Node is removed from the list when empty.
3. **Edge Cases with Insert/Delete/Search Operations:** 
   * Input: Test with very small and very large numbers.
   * Expected Output: Correct handling of edge case values.
4. **Test Empty List Operations:** 
   * Input: Perform operations on an empty list (e.g., search, delete).
   * Expected Output: Proper error handling for operations on an empty list.

**=>Functional Testing**

* **Stress Testing:** 
  + Perform operations with a large number of elements.
* **Error Handling:** 
  + Input invalid data (e.g., non-integer values).
  + Exceed input boundaries (e.g., large values).
* **Performance Testing:** 
  + Test the time complexity for large operations.
* **Memory Usage:** 
  + Monitor memory usage with varying sizes of data.