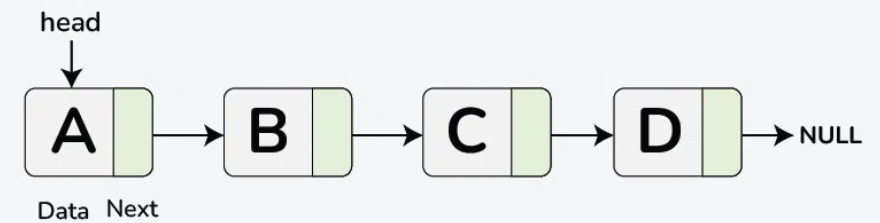
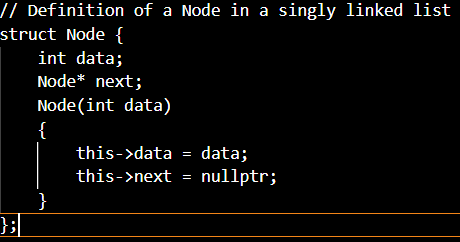
| **Feature** | **Linked List** | **Array** |
| --- | --- | --- |
| **Data Structure** | Non-contiguous | Contiguous |
| **Memory Allocation** | Typically allocated one by one to individual elements | Typically allocated to the whole array |
| **Insertion/Deletion** | Efficient | Inefficient |
| **Access** | Sequential | Random |

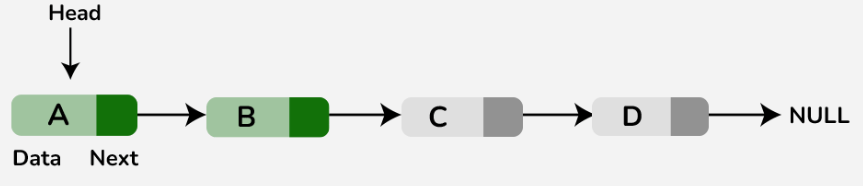
**Singly Linked List**

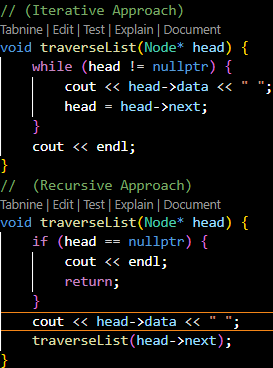
****



**operation on singly Linked list are-**

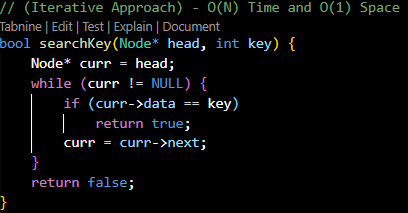
1. **Traversal -------**



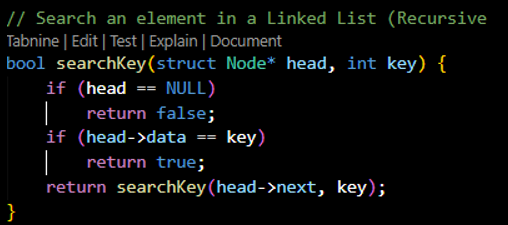


2. Searching------------

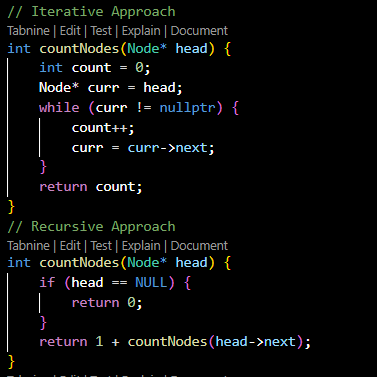
1-



2-

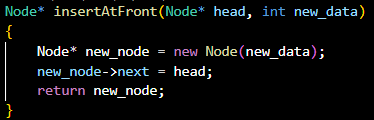


3. Length-----------

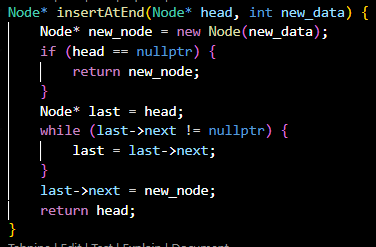


4. Insertion

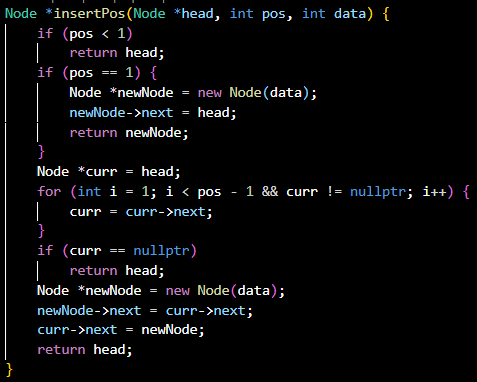
1-



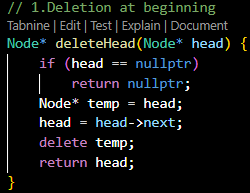
2-

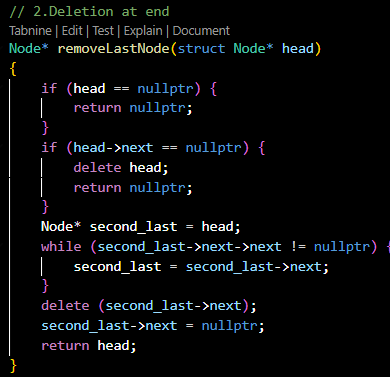


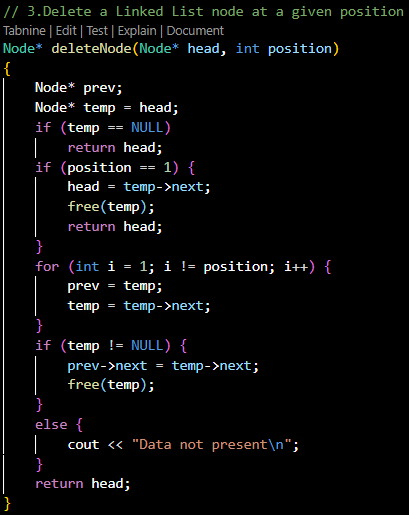
3-



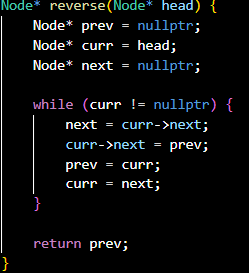
5. Deletion

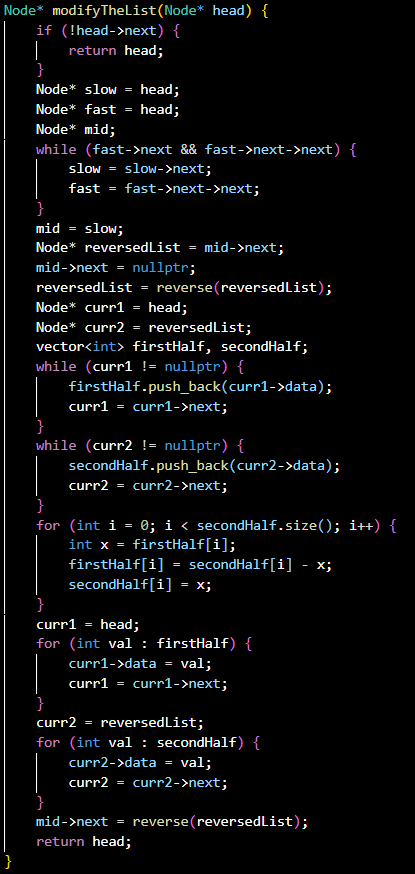




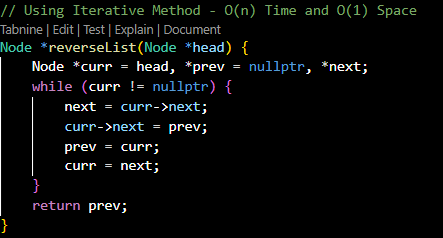


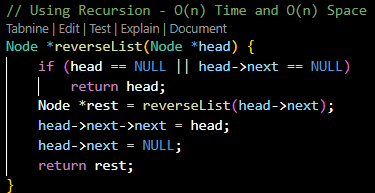
6. Modify

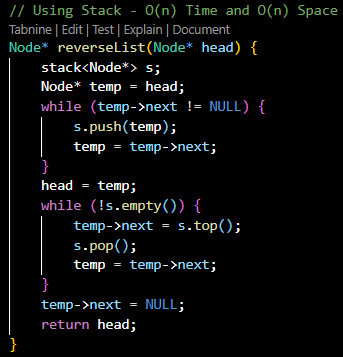




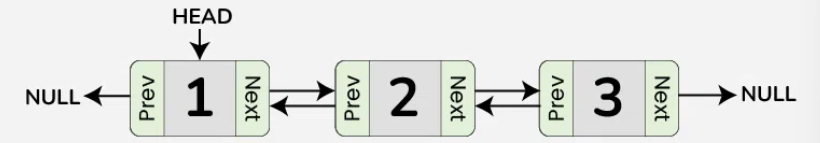
7. Reversing



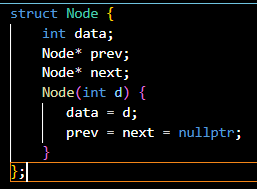




**Doubly Linked List**



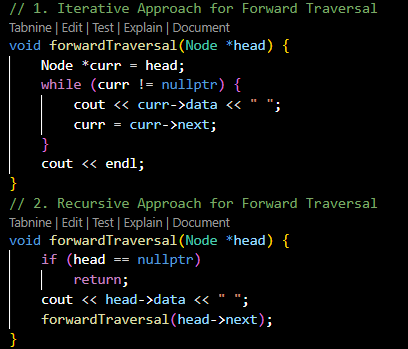
Node creation---



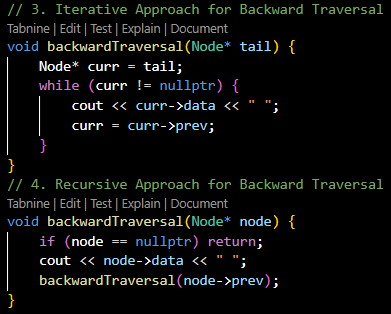
Operations---

1. Traversal----Types of Traversal in Doubly Linked List--Forward Traversal, Backward Traversal

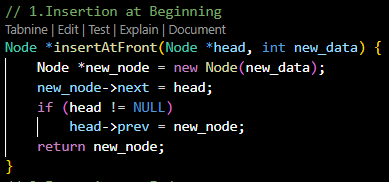
Forward traversal-

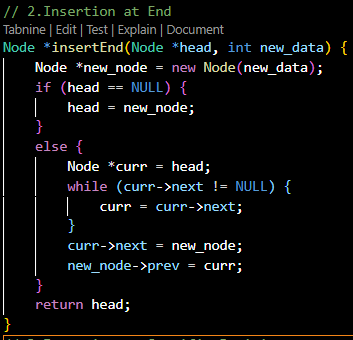


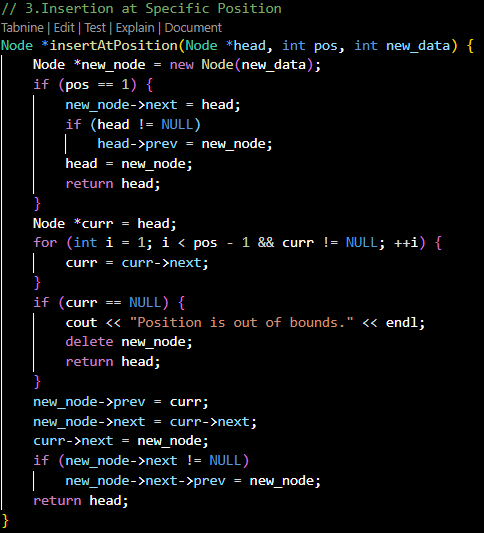
Backward Traversal



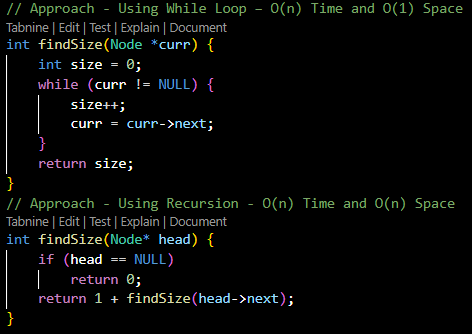
2. Insertion –



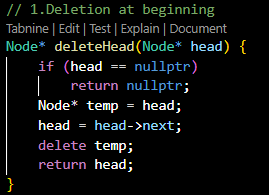


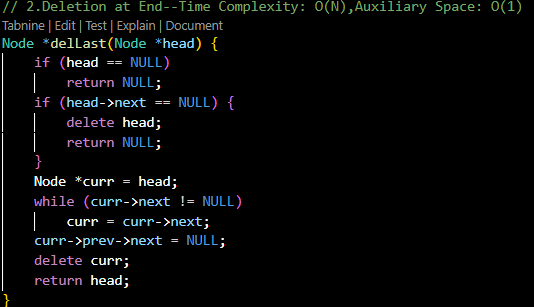


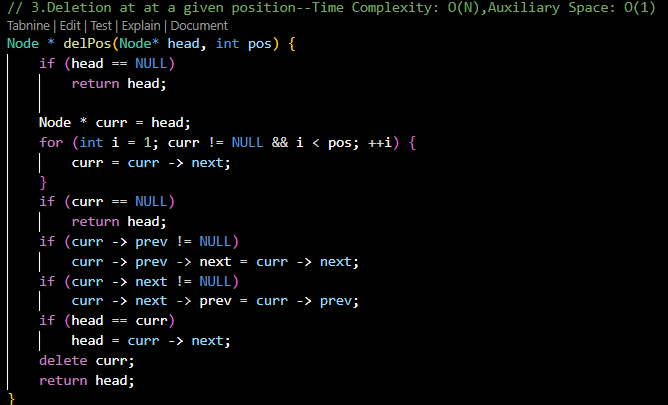
3. length----



4. deletion----







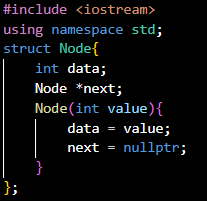
**Circular Linked List**

**Types of Circular Linked Lists**

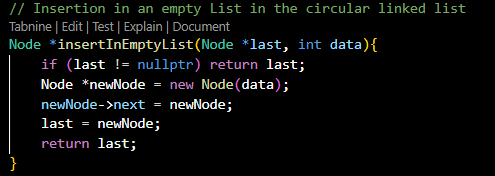
1. **Circular Singly Linked List**
2. **Circular Doubly Linked List**

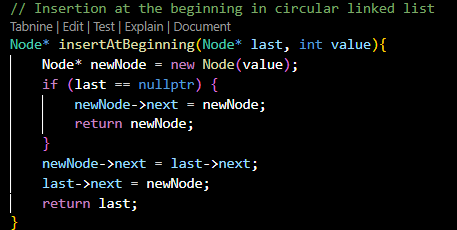
**Operations on the Circular Linked list**

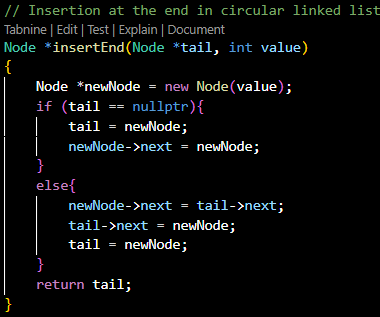
**Node creation-**

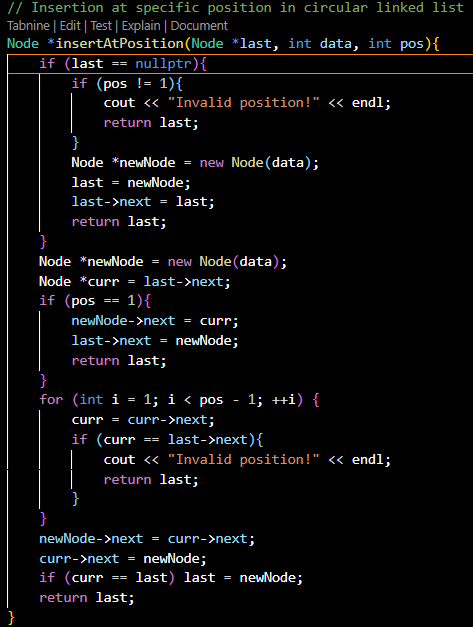
****

1. **Insertion-**

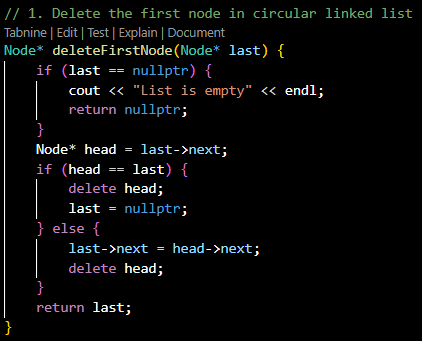
****

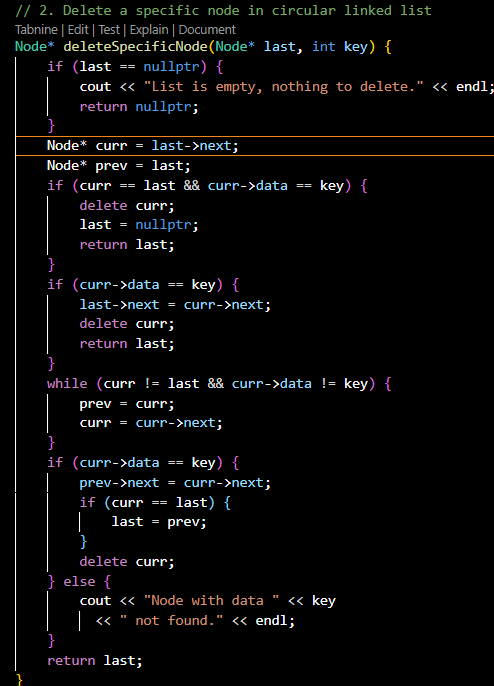
****

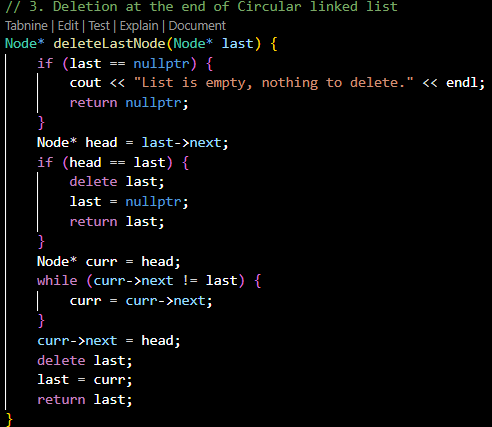
****

****

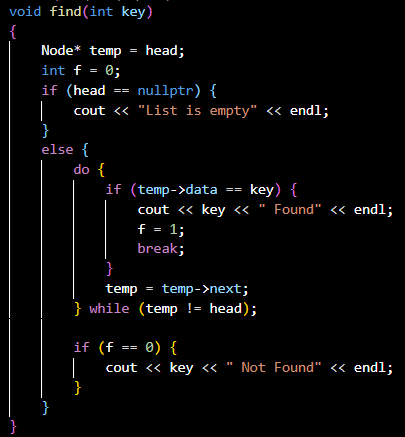
1. **Deletion**

****

****

****

1. **Searching**

****