CircularLinkedList Overview

This structure connects nodes in both directions (Next and Prev), and forms a circle — the last node links back to the first, and vice versa.

insertFirst(int Val)

- Inserts a node at the **beginning**.
- If the list is empty:
    - New node points to itself (Next and Prev).
- Otherwise:
    - Links new node before the current head.
    - Updates Head and adjusts surrounding pointers.

insertLast(int Val)

- Inserts a node at the **end**.
- If empty, defers to insertFirst().
- Otherwise:
    - Links the new node just before Head, preserving circularity.

insertNth(int Val, int Pos)

- Inserts at a **specific position** (0-based).
- If Pos == 0, calls insertFirst().
- Otherwise:
    - Traverses to node at position Pos - 1.
    - Inserts new node after it.
    - Checks for out-of-bounds (i.e., wrap-around).

insertCenter(int Val)

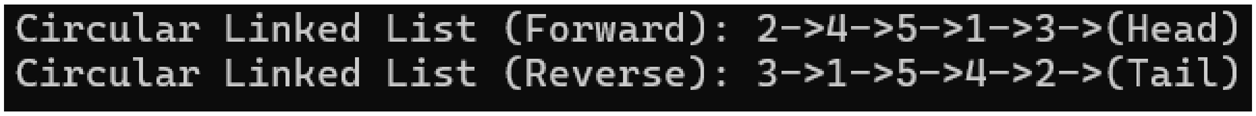
- Inserts in the **middle** using slow/fast pointer technique.
- Slow moves 1 step, Fast moves 2.
- Inserts node after Slow.

displayForward()

- Starts at Head, moves via Next, and stops when it loops back.

displayReverse()

- Starts at Tail (Head->Prev), moves via Prev, and loops back.



```
Circular Linked List (Forward): 2->4->5->1->3->(Head)
Circular Linked List (Reverse): 3->1->5->4->2->(Tail)
```