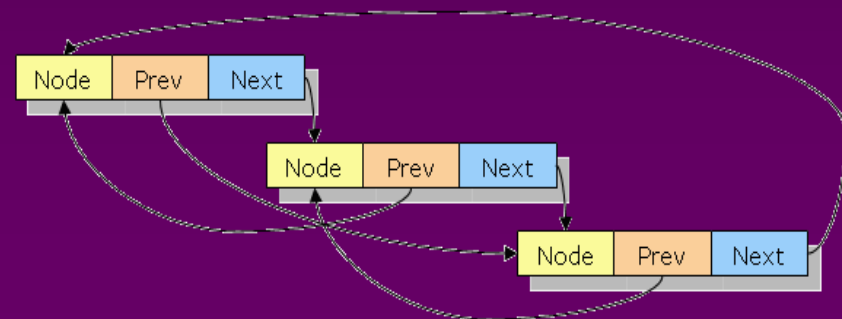
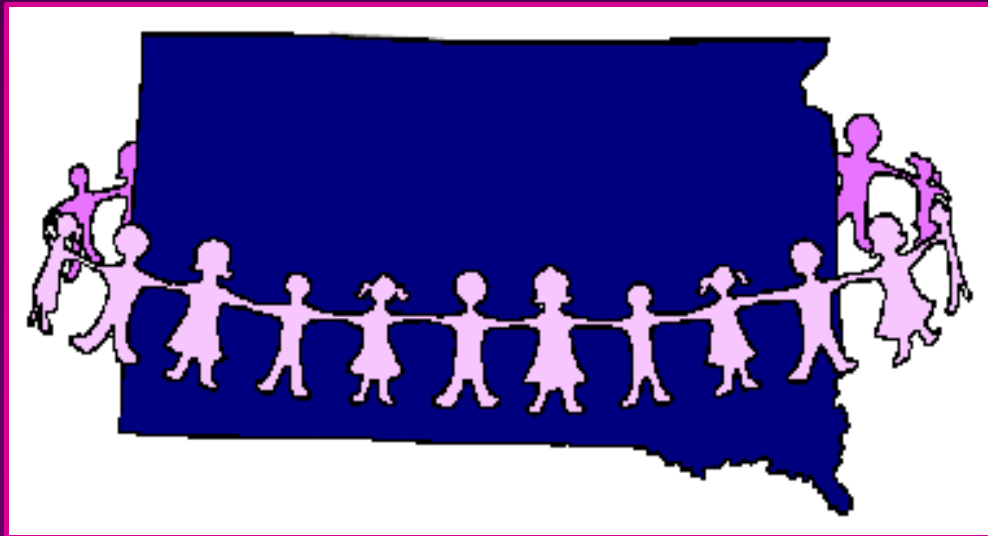
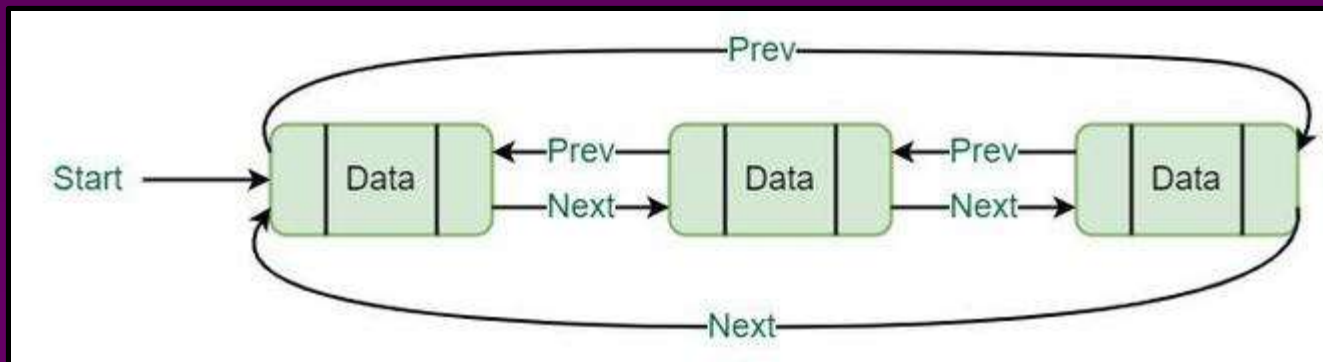


Circular Doubly Linked List - [CDL]



What is a Circular Doubly Linked Lists?

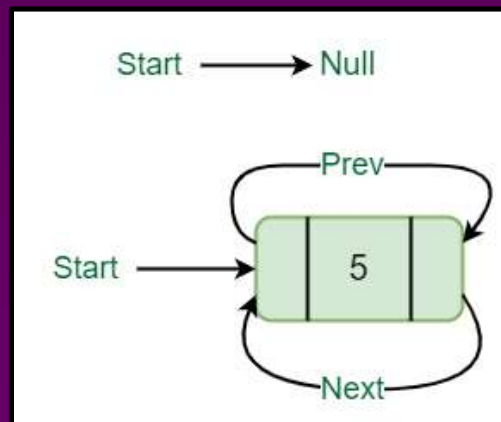
- ✉ Circular Doubly Linked List has properties of both doubly linked list and circular linked list in which two consecutive elements are linked or connected by the previous and next pointer and the last node points to the first node by the next pointer and also the first node points to the last node by the previous pointer .



- ✉ Circular doubly linked list doesn't contain **NULL** in any of the node.
- ✉ Circular doubly linked list is **a more complex type of data structure in which a node contains pointers to its previous node as well as the next node.**
- ✉ **Display forward** – Displays the complete list in a forward manner.
- ✉ **Display backward** – Displays the complete list in a backward manner.

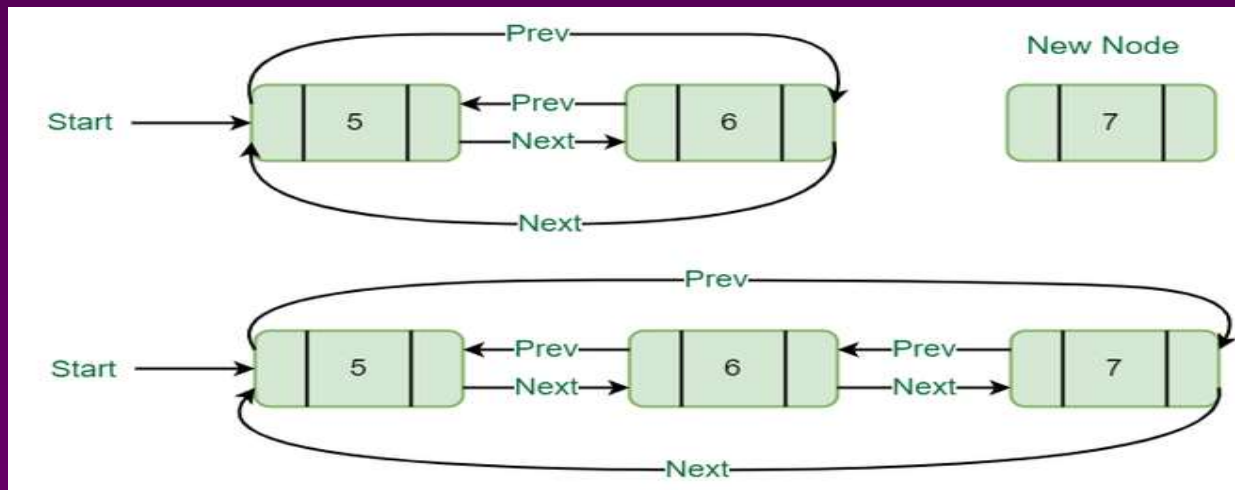
Insertion in Circular Doubly Linked List:

- ✉ **Insertion at the end of the list or in an empty list:**
- ✉ A node(Say **N**) is inserted with **data = 5**. So, the previous pointer of N points to N and the next pointer of N also points to N. But now start pointer points to the first node of the list.



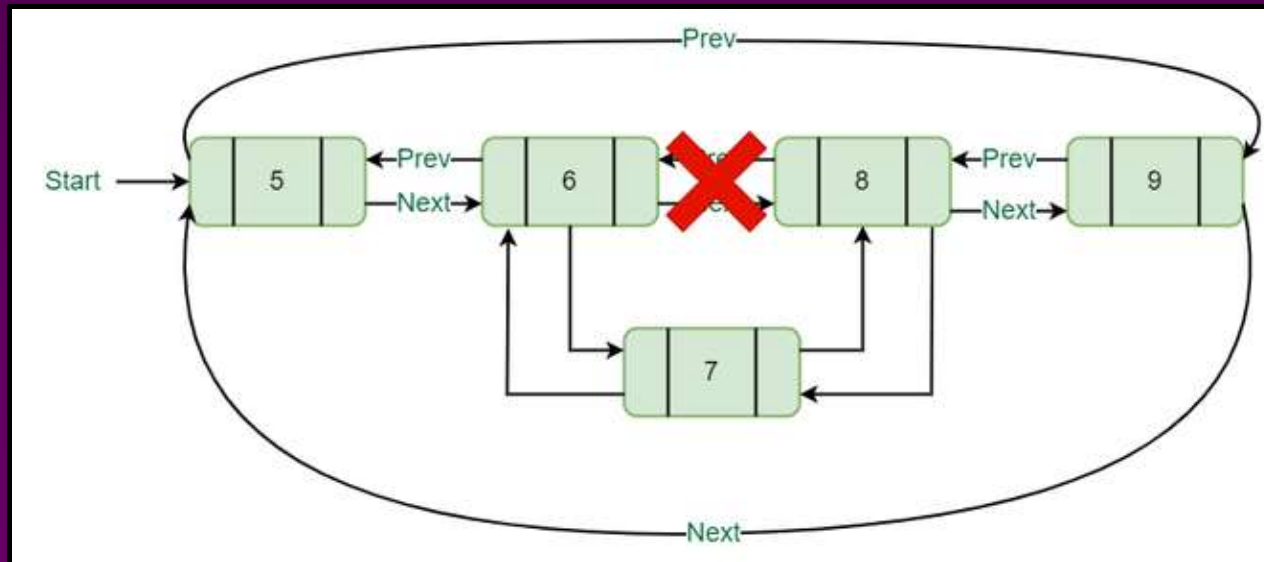
✉ **List initially contains some nodes, start points to the first node of the List:**

✉ A node(Say M) is inserted with data = 7, so the previous pointer of M points to the last node, the next pointer of M points to the first node and the last node's next pointer points to this M node, and first node's previous pointer points to this M node.



Insertion in between the nodes of the list:

- ✉ To insert a node in between the list, two data values are required one after which new node will be inserted and another is the data of the new node.



Difference between circular singly and circular doubly linked list

Circular Linked List (Intro.)

Circular **Singly** Linked List



Circular **Doubly** Linked List

