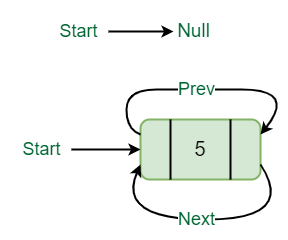
***Insert at End Doubly Linked List***

**Insertion at the end of the list or in an empty list:**

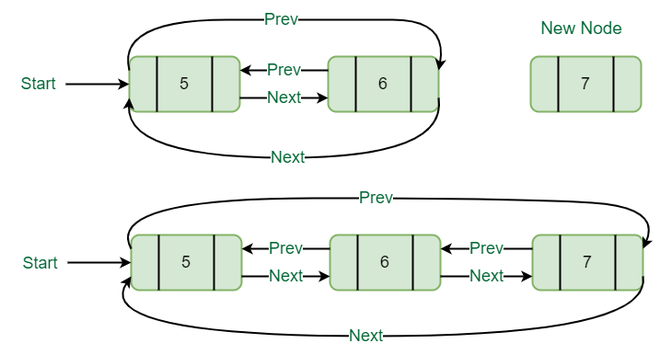
A node(Say **N**) is inserted with **data = 5**. So, the previous pointer of N points to N and the next pointer of N also points to N. But now start pointer points to the first node of the list.



*Insertion in an empty 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 at the end of list*

Below is the implementation of the above operations:

C++Java

// Function to insert at the end

static void insertEnd(int value)

{

// If the list is empty, create a single

// node circular and doubly list

if (start == null) {

Node new\_node = new Node();

new\_node.data = value;

new\_node.next = new\_node.prev = new\_node;

start = new\_node;

return;

}

// If list is not empty

// Find last node

Node last = (start).prev;

// Create Node dynamically

Node new\_node = new Node();

new\_node.data = value;

// Start is going to be

// next of new\_node

new\_node.next = start;

// Make new node previous of start

(start).prev = new\_node;

// Make last previous of new node

new\_node.prev = last;

// Make new node next of old last

last.next = new\_node;

}