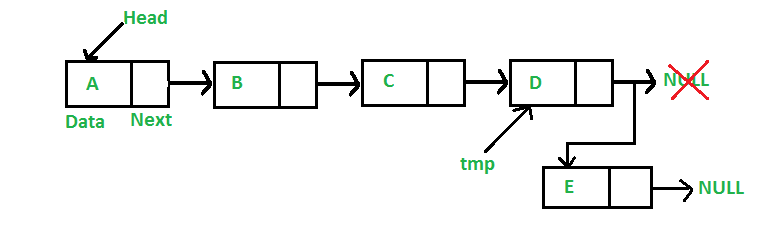
***Insert at the end of Singly Linked List***

**Add a node at the end: (6 steps process)**

* The new node is always added after the last node of the given Linked List. For example if the given Linked List is 5->10->15->20->25 and we add an item 30 at the end, then the Linked List becomes 5->10->15->20->25->30.
* Since a Linked List is typically represented by the head of it, we have to traverse the list till the end and then change the next to last node to a new node.

[](https://media.geeksforgeeks.org/wp-content/cdn-uploads/gq/2013/03/Linkedlist_insert_last.png)

Following are the 6 steps to add a node at the end.

C++Java

/\* Appends a new node at the end. This method is

defined inside LinkedList class shown above \*/

public void append(int new\_data)

{

/\* 1. Allocate the Node &

2. Put in the data

3. Set next as null \*/

Node new\_node = new Node(new\_data);

/\* 4. If the Linked List is empty, then make the

new node as head \*/

if (head == null)

{

head = new Node(new\_data);

return;

}

/\* 4. This new node is going to be the last node, so

make next of it as null \*/

new\_node.next = null;

/\* 5. Else traverse till the last node \*/

Node last = head;

while (last.next != null)

last = last.next;

/\* 6. Change the next of last node \*/

last.next = new\_node;

return;

}

**Complexity Analysis:**

* **Time complexity:**O(N), where N is the number of nodes in the linked list. Since there is a loop from head to end, the function does O(n) work.
  + This method can also be optimized to work in O(1) by keeping an extra pointer to the tail of the linked list/
* **Auxiliary Space:** O(1)