* First need to check if head is null
  insert at head. (if position < 0 should be handled).

* Second need to check if head -> next == null
  insert at rear

* (3) And that is why we check for :
  curr -> next != null ,
  if that is true, we insert at position (some where in the middle of the Linked List).

* Another way that we do not need to check for
  curr -> next != null is to have a counter
keep track/count number of nodes.
If position > 0 and position < number of nodes ,
then insert at middle.

* The theory above is very similar for delete (int position).