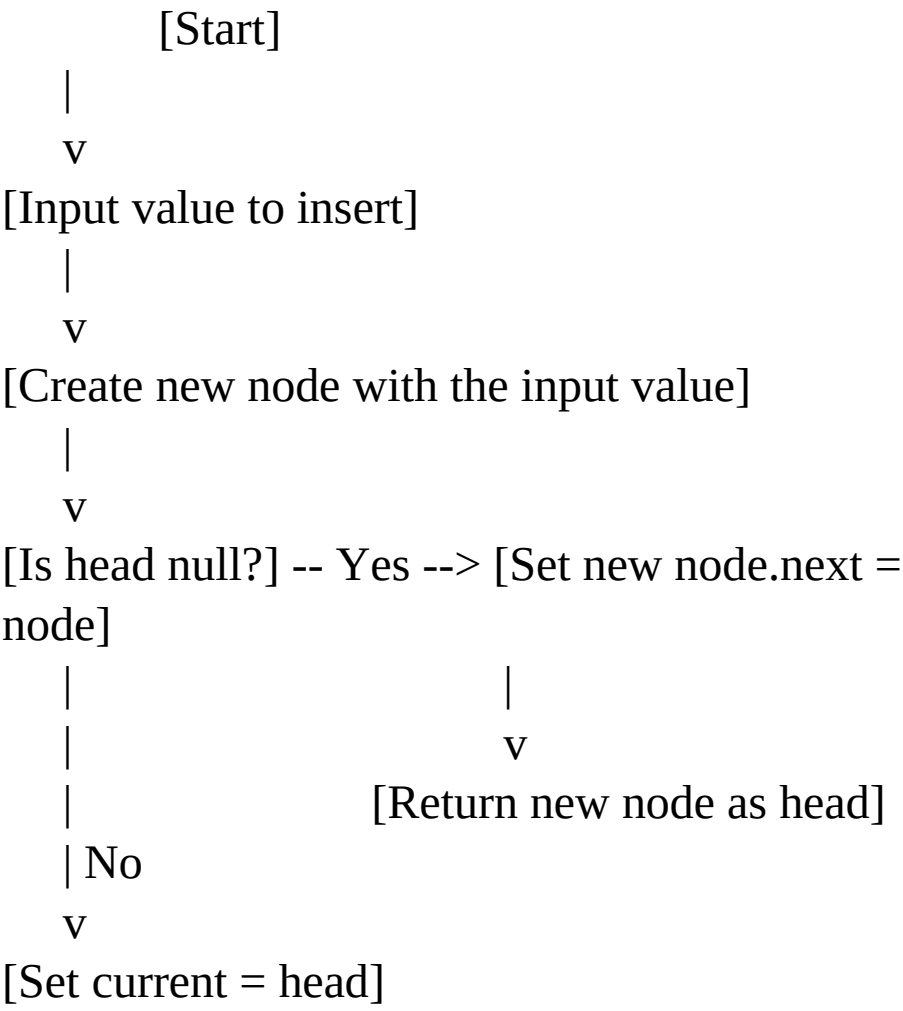
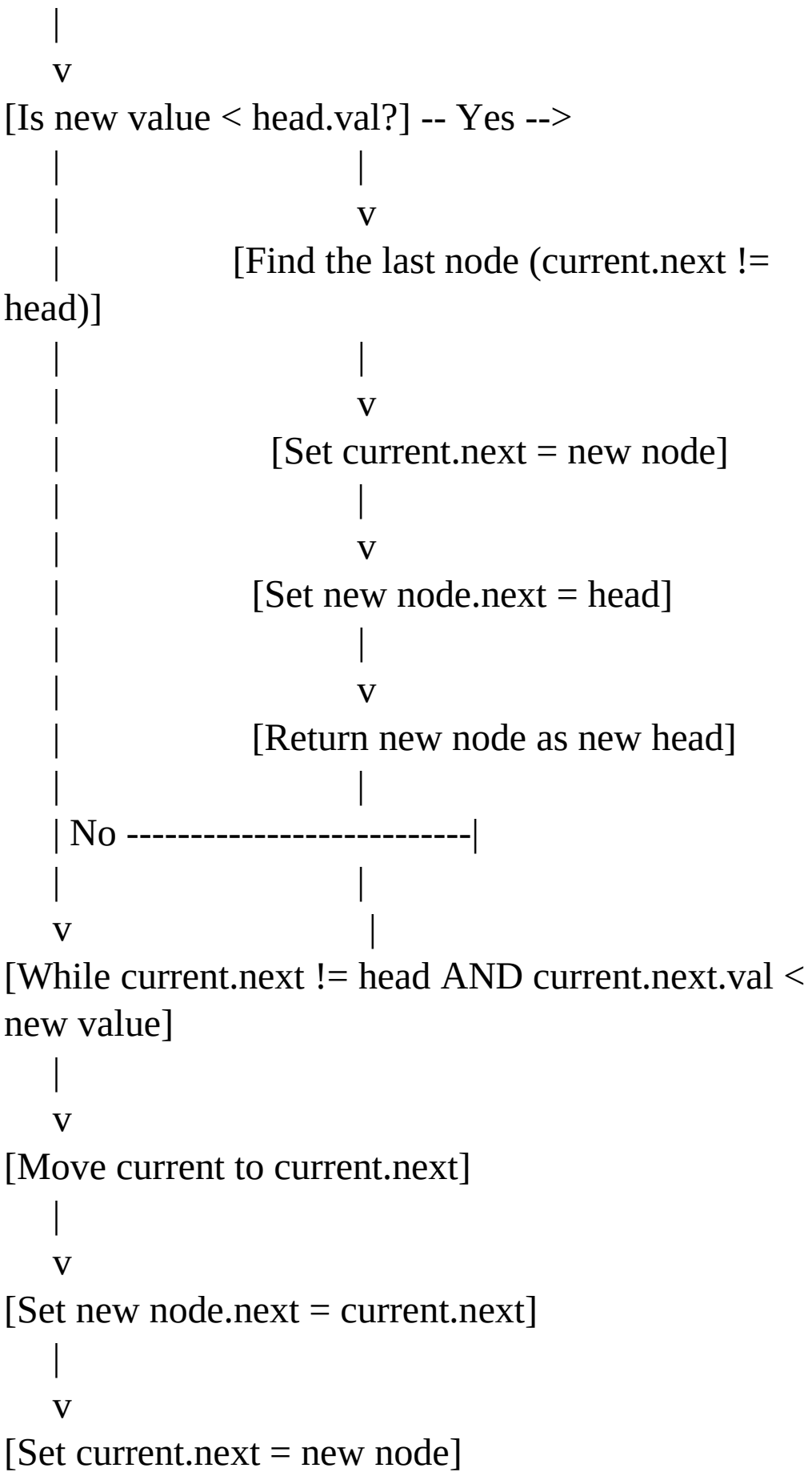
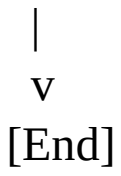
❖ Insert in Circular List:

- Explanation:
  The program inserts a new node into a sorted circular linked list by first creating the node with the given value and checking if the list is empty. If not, it traverses the list to find the appropriate insertion point while maintaining the sorted order, adjusting the next pointers accordingly. Finally, it updates the list structure to include the new node and returns the head of the list.
- Time Complexity and Space Complexity:
  - O(n) O(1)
- Flowchart:




[Is new value < head.val?] -- Yes -->
[Find the last node (current.next != head)]
[Set current.next = new node]
[Set new node.next = head]
[Return new node as new head]
No
[While current.next != head AND current.next.val < new value]
[Move current to current.next]
[Set new node.next = current.next]
[Set current.next = new node]


[End]