

```
// Function to reverse a linked list, return new head after reversing the list.
// Node{int data, Node next}
Node reverseList(Node head) {
    // list is empty or list has only one node -> no need to reverse.
    if (head == null || head.next == null) {
        return head;
    }
    // working with every three consecutive node name as first, second and third.
    Node first = head;
    Node second = first.next;
    Node third = second.next;
    first.next = null;           // marks first node(head) pointer to nothing(null).
    while (true) {
        third = second.next;    // Third node pointer.
        second.next = first;    // Assign the next node of the second node to the first node
        first = second;         // Then assign second node as first(head) node.
        second = third;         // Then assign third node as second node.
        if (third == null) {
            break;
        }
    }
    return first;
}
```