```
class Node {
    int value
    Node next;

    public Node(int value){
        this.value=value;
        this.next=null;
    }
}

class LinkedListUtils{
    public static Node insert(Node root, int value) {
        if(root == null){
            return new Node(value);
        } else{
            //adding it on front of linkedlist
            Node newNode = new Node(value);
            root.next=newNode;
            return newNode;
        }
    }

    public static void printLinkedList(Node head){
        while(head != null){
            System.out.print(head.value+",");
        }
        System.out.println();
    }

    public static Node reverse(Node head, int size){
        if(head==null || head.next == null)) {
            return head;
        }

        Node previous = null;
        Node tempHead = head;
        Node next = head.next;

        for(int i=0;i<size-1;i++){
            if(next==null) {
                break;
            }
            previous = previous.next;
            tempHead = tempHead.next;
            next = next.next;
        }
        if(next == null) {
            return reverse(head);
        }
        tempHead.next = null;
        Node newHead = reverse(head);
        Node tail = getTail(newHead);
        tail.next=reverse(next,size);
        return newHead;
    }

    private static Node reverse(Node head){
        if(head == null || head.next == null){
            return head;
        }

        Node prev=null;
        Node curr = head;
        Node next = head.next;

        while(next != null){
            curr.next=prev;
            prev=curr;
            curr = next;
            next = next.next;
        }
        return curr;
    }

    private static Node getTail(Node head){
        while(head.next != null){
            head = head.next;
        }

    }
}


class LinkedList{
    public static void main(String [] str){
        Node head=null;
        head = LinkedListUtils.insert(head,5);
        head = LinkedListUtils.insert(head,10);
        head = LinkedListUtils.insert(head,15);
        head = LinkedListUtils.insert(head,25);
        head = LinkedListUtils.insert(head,35);
        head = LinkedListUtils.insert(head,45);

        head = LinkedListUtils.reverse(head,3);
    }

}
```