Task5

public class SinglyLinkedList {  
 static class Node  
 {  
 int data;  
 Node next;  
 Node(int d)  
 {  
 this.data = d;  
 this.next = null;  
 }  
 }  
 static class LinkedList{  
 Node start;  
 LinkedList()  
 {  
  
 start = null;  
 }  
  
 public void push(int data)  
  
 {  
 if(this.start == null)  
 {  
 Node temp = new Node(data);  
 this.start = temp;  
 }  
 else  
 {  
 Node temp = new Node(data);  
 temp.next = this.start;  
 this.start = temp;  
 }  
 }  
   
 public int findSecondLastNode(Node ptr)  
  
 {  
  
 Node temp = ptr;  
   
 if(temp == null || temp.next == null)  
  
 return -1;  
   
 while(temp.next.next != null)  
  
 {  
  
 temp = temp.next;  
  
 }  
  
 return temp.data;  
  
 }  
   
 public static void main(String[] args)  
  
 {  
  
 LinkedList ll = new LinkedList();  
 the below list 8 -> 23 -> 11 -> 29 -> 12 \*/  
  
 ll.push(12);  
  
 ll.push(30);  
  
 ll.push(11);  
  
 ll.push(23);  
  
 ll.push(8); System.*out*.println(ll.findSecondLastNode(ll.start));  
 } }  
  
}

..................................................................

public class Revers {  
 static class Node  
  
 {  
  
  
  
 int data;  
  
 Node next;  
  
 };  
  
 static Node *head* = null;  
 static void reverseLL()  
  
 {  
 Stack<Node> s = new Stack<>();  
 Node temp = *head*;  
 while (temp.next != null)  
 {  
 s.add(temp);  
 temp = temp.next;  
 }  
 *head* = temp;  
   
 while (!s.isEmpty())  
 {  
 temp.next = s.peek();  
 s.pop();  
 temp = temp.next;  
 }  
 temp.next = null;  
 }  
 static void printlist(Node temp)  
 {  
 while (temp != null)  
 {  
 System.*out*.print(temp.data+ " ");  
 temp = temp.next;  
 }  
 }  
 static void insert\_back( int value)  
  
 {  
 Node temp = new Node();  
 temp.data = value;  
 temp.next = null;  
   
 if (*head* == null)  
 {  
 *head* = temp;  
 return;  
 }  
 else  
 {  
 Node last\_node = *head*;  
 while (last\_node.next != null)  
 {  
 last\_node = last\_node.next;  
 }  
 last\_node.next = temp;  
  
 return;  
  
 }  
 }  
 public static void main(String[] args)  
  
 {  
  
 *insert\_back*( 1);  
  
 *insert\_back*( 2);  
  
 *insert\_back*(3);  
  
 *insert\_back*( 4);  
  
 System.*out*.print("Given linked list\n");  
  
 *printlist*(*head*);  
  
 *reverseLL*();  
  
 System.*out*.print("\nReversed linked list\n");  
  
 *printlist*(*head*);  
  
 }  
  
}

...................................................................

public class Rotate {  
  
 static class LinkedList {  
  
 Node head;   
 class Node {  
  
 int data;  
  
 Node next;  
  
 Node(int d)  
  
 {  
  
 data = d;  
  
 next = null;  
  
 }  
  
 }  
  
 void rotate(int k)  
  
 {  
  
 if (k == 0)  
  
 return;  
  
 Node c = head;  
  
 int count = 1;  
 while (count < k && c != null) {  
 c = c.next;  
 count++;  
 }  
 if (c == null)  
  
 return;  
 Node kthNode = c;  
 while (c.next != null;  
 c = c.next;  
 c.next = head;  
 head = kthNode.next;  
 kthNode.next = null;  
  
 }  
  
 void push(int new\_data)  
  
 {  
 Node new\_node = new Node(new\_data);  
 new\_node.next = head;  
 head = new\_node;  
 }  
 void printList()  
 {  
 Node temp = head;  
 while (temp != null) {  
 System.*out*.print(temp.data + " ");  
 temp = temp.next;  
 }  
 System.*out*.println();  
 }  
 public static void main(String args[])  
 {  
 LinkedList llist = new LinkedList();  
 for (int i = 60; i >= 10; i -= 10)  
 llist.push(i);  
 System.*out*.println("Given list");  
 llist.printList();  
 llist.rotate(4);  
 System.*out*.println("Rotated Linked List");  
 llist.printList();  
 }  
 }  
}