SINGLY LINKED LIST

public static class LLi {  
 private static int *size*;  
 LLi(){  
 this.*size*=0;  
 }  
 static class Node{  
  
 String data;  
 Node next;  
 Node(String data){  
 this.data=data;  
 this.next=null;  
 *size*++;  
 }  
  
 }  
 Node head;  
 public void addFirst(String data) {  
 Node newn = new Node(data);  
 if (head == null) {  
 head = newn;  
 return;  
 }  
  
 newn.next = head;  
 head = newn;  
  
 }  
 public void addLast(String data) {  
 Node newn = new Node(data);  
 Node curr=head;  
  
 while(curr.next!=null){  
 curr=curr.next;  
 }  
  
 curr.next=newn;  
  
 }  
  
 public void rev(Node prev,Node curr)  
 {  
 if(head==null||head.next==null){  
 return;  
 }  
 if(curr==null){  
 head.next=null;  
 head=prev;  
 return;  
 }  
 Node nextNode= curr.next;  
 curr.next=prev;  
 rev(curr,nextNode);  
  
  
 }  
 public void deleFirst() {  
 if (head == null) {  
 System.*out*.println("empty list");  
 return;  
 }  
 *size*--;  
 head=head.next;  
 }  
 public void delLast() {  
 if (head == null) {  
 System.*out*.println("empty list");  
 return;  
 }  
 *size*--;  
 if(head.next==null){  
 head=null;  
  
 }  
  
 Node curr=head;  
 while(curr.next.next!=null){  
 curr=curr.next;  
 }  
 curr.next=null;  
 }  
  
  
 public void print() {  
  
 if (head == null) {  
 System.*out*.println("empty list");  
 return;  
 }  
 Node curr=head;  
 while(curr!=null){  
 System.*out*.print(curr.data+"-->");  
 curr=curr.next;  
 }  
 System.*out*.print("NULL");  
 }  
 public void insert(int index,String val){  
  
 Node newn=new Node(val);  
 if(head==null){  
 addFirst(val);  
 }  
 Node curr=head;  
 for(int i=1;i<index-1;i++){  
 curr=curr.next;  
 }  
 newn.next=curr.next;  
 curr.next=newn;  
  
 }  
 public void delete(int index){  
  
  
 if(head==null){  
 return;  
 }  
 if(head.next==null){  
 deleFirst();  
 }  
 Node curr=head;  
 for(int i=1;i<index-1;i++){  
 curr=curr.next;  
 }  
 curr.next=curr.next.next;  
  
 }  
 public boolean find(String val){  
 Node curr=head;  
 while(curr!=null){  
 if(curr.data==val){  
 return true;  
 }  
 curr=curr.next;  
 }  
 return false;  
 }  
  
public int getSize(){  
 return *size*;  
}  
  
 }  
 public static void main(String[] args) {  
  
 LLi newn=new LLi();  
 newn.addFirst("a");  
 newn.addFirst("is");  
// newn.print();  
 newn.addLast("list");  
 newn.addFirst("this");  
// newn.rev(newn.head,newn.head.next);  
// newn.print();  
 newn.insert(4,"good");  
 newn.insert(2,"verygood");  
 newn.delete(2);  
 newn.print();  
// System.out.println();  
// System.out.println(newn.find("is"));  
//// newn.print();  
//  
// newn.deleFirst();  
//// newn.print();  
// newn.delLast();  
// newn.print();  
// System.out.println();  
// System.out.println(newn.getSize());  
 }