Linked List Methods

```
// Removes the head of a linked list
public Object remove(){
    node tmp = null;
    if(isEmpty())
        return null;
    tmp = head;
    head = tmp.getNext();
    size--;
    return tmp.getInfo();
}
```

Linked List Methods

```
// Returns a pointer to the predecessor
// of the tail of a linked list
public node getPredTail(){
    node t = head;
    while(t.getNext().getNext() != null)
        t=t.getNext();
    return t;
}

public node getTail(){
    node t = head;
    while(t.getNext() != null)
        t=t.getNext();
    return t;
}
```

Node Declaration

```
// Inserts a node at the
// tail of a linked list
public void insertTail(Object obj){
    if(isEmpty())
        insert(obj);
    else{
        // find the tail of the list
        node t = getTail();
        // create a new node
        node tmp =
        new node(obj,null);
        // set a new node as the tail
        t.setNext(tmp);
        size++;
    }
}
```

Linked List Methods

```
// removes the tail of a linked list
public Object removeTail(){
    if(isEmpty())
        return null;
    if(head.getNext() == null)
        return remove();
    node t = getPredTail();
    Object obj = t.getNext().getInfo();
    t.setNext(t.getNext().getNext());
    size--;
    return obj;
}
```

Linked List Methods

```
public Object peek(int n){
    node t = head;
    for(int i = 0;i<n && t != null;i++)
        t = t.getNext();
    return t;
}

public linkedList copy(){
    if (head == null) return null;
    linkedList l = new linkedList();
    node t = head;
    while (t != null) {
        l.insertTail(t.getInfo());
        t = t.getNext();
    }
    return l;
}
```

Linked List Methods

```
public linkedList join(linkedList l){
    if (head == null) return l;
    node t = getTail();
    t.setNext(l.head);
    this.size += l.length();
    return this;
}

public void invert(){
    if (head == null) return;
    node curr = head;
    node pred = null;
    while (curr != null) {
        node succ = curr.getNext();
        curr.setNext(pred);
        pred = curr;
        curr = succ;
    }
    head = pred;
}
```