Homework Turnin

Email:	rgalanos@fcps.edu
Section:	6G
Course:	TJHSST APCS 2016-17
Assignment:	04-03
Receipt ID:	14d5b7352f97b380a1b98ebc80057a70

Execution failed with return code 1 (general error). (Expected for JUnit when any test fails.)

Warning: Your program failed to compile :

```
Josephus.java:39: error: cannot find symbol
      ListLab1.pointerToLast(list).setNext(list);
      ^
  symbol:   variable ListLab1
  location: class Josephus
Josephus.java:96: error: cannot find symbol
         p = ListLab1.insertLast(p, obj);
             ^
  symbol:   variable ListLab1
  location: class Josephus
2 errors
```

Please correct your file(s), go back, and try to submit again. If you do not correct this problem, you are likely to lose a large number of points on the assignment. Please contact your TA if you are not sure why your code is not compiling successfully.

Turnin Failed! (See above)

There were some problems with your turnin. Please look at the messages above, fix the problems, then Go Back and try your turnin again.

GradeIt has a copy of your submission, but we believe that you will want to fix the problems with your submission by resubmitting a fixed version of your code by the due date.

We have received the following file(s):

Josephus.java (3598 bytes)

```java
// name:     date:

import java.util.*;
import java.io.*;
import javax.swing.JOptionPane;

public class Josephus
{
   private static String WINNER = "Josephus";
   public static void main(String[] args) throws FileNotFoundException
   {
      /* run it first with J_numbers.txt  */
      ListNode p = null;
      int n = Integer.parseInt(JOptionPane.showInputDialog("How many names (2-20)?"));
      File f = new File("J_numbers.txt");
      p = readNLinesOfFile(n, f);
      int countOff = Integer.parseInt(JOptionPane.showInputDialog("How many names to count off each time?"));
      countingOff(p, countOff, n);

      /* run it next with J_names.txt  */
      System.out.println("\n ****  Now start all over.  Enter the winning position in the JOptionPane.  *** \n"
      p = readNLinesOfFile(n, new File("J_names.txt"));
      int winPos = Integer.parseInt(JOptionPane.showInputDialog("Enter Josephus's preferred position."));
      replaceAt(p, WINNER, winPos);
      countingOff(p, countOff, n);
      System.out.println(WINNER + " wins!");
   }
   /* reads the names, builds the linked list.
     */
   public static ListNode readNLinesOfFile(int n, File f) throws FileNotFoundException
   {
      ListNode list = null;
      Scanner infile = new Scanner(f);
      for(int i=0;i<n;i++)
      {
         list = insert(list, infile.next());
      }

      ListLab1.pointerToLast(list).setNext(list);

      return list;

   }
  /* Runs a Josephus game, counting off and removing each name. Prints after each removal.
     Ends with one remaining name, who is the winner.
     */
   public static void countingOff(ListNode p, int count, int n)
   {
      print(p);
      for(int i = 0; i<n-1;i++)
      {
         p = remove(p, count);
         print(p);
      }
   }
   /* removes the node after counting off count-1 nodes.
     */
   private static ListNode remove(ListNode p, int count)
   {
      if(count==1)
      {
         Object end = p.getValue();
         while(p.getNext().getValue()!=end)
            p = p.getNext();
      }
      else
         for(int i = 0; i<count-2;i++)
            p = p.getNext();
      p.setNext(p.getNext().getNext());
      return p.getNext();
   }
   /* prints the circular linked list.
     */
   public static void print(ListNode p)
   {
      ListNode p2 = p;
      Object end = p2.getValue();
```

```
         System.out.print(p2.getValue()+" ");
         p2=p2.getNext();
         while(p2.getValue()!=end)
         {
            System.out.print(p2.getValue() + " ");
            p2 = p2.getNext();
         }
         System.out.print("\n");
      }
    /* helper method to build the list.  Creates the node, then
       inserts it in the circular linked list.
       */
      private static ListNode insert(ListNode p, Object obj)
      {
         if(p==null)
            p= new ListNode(obj, null);
         else
         {
            p = ListLab1.insertLast(p, obj);
         }
         return p;
      }

      /* replaces the value (the string) at the winning node.
         */
      private static void replaceAt(ListNode p, Object obj, int pos)
      {
         for(int i=0;i<pos-1;i++)
            p = p.getNext();
         p.setValue(obj);
      }
   }

     //the College Board's standard ListNode class
   class ListNode
   {
      private Object value;
      private ListNode next;
      public ListNode(Object v, ListNode n)
      {
         value=v;
         next=n;
      }
      public Object getValue()
      {
         return value;
      }
      public ListNode getNext()
      {
         return next;
      }
      public void setValue(Object newv)
      {
         value=newv;
      }
      public void setNext(ListNode newn)
      {
         next=newn;
      }
   }
```