1. Compare your test cases with the other people in your group. If you put them all together, which cases would be redundant? Make a new test strategy that you now consider to be thorough.

2. Suppose we were to add a this new method to class `AnotherTwoEndedList`:

   ```java
   // Removes all but the first occurrence of 'o' in the list.
   // Uses '.equals' to check for equality.
   public void removeDuplicates (Object o) {
   ```

Without knowing anything about the method body, design a testing strategy, and describe it using a table. Include the *significance* of each test case.

3. What do we call that kind of testing? __________________________________________
4. Recall that the instance variables in this class are `first`, `last`, and `numberOfNodes`. Now look at the method body:

```
// Removes all but the first occurrence of 'o' in the list.
// Uses '.equals' to check for equality.
public void removeDuplicates(Object o) {
    ListNode temp = first;
    while (temp != null) {
        if (temp.contents.equals(o)) {
            temp.previous.next = temp.next;
            temp.next.previous = temp.previous;
        }
        temp = temp.next;
    }
}
```

Design additional test cases that hit on potential weak spots that the code suggests to you.

5. What do we call that kind of testing? ________________________________

6. List any bugs that you can identify based on your test cases, and correct the code if necessary.