Suppose we have this class for nodes in a doubly-linked list of integers:

```java
public class IntNode {
    public int data;
    // The forwards and back pointers in the node.
    public IntNode forwards;
    public IntNode backwards;

    public IntNode (int i) {
        data = i;
        forwards = null;
        backwards = null;
    }
}
```

You will answer questions about this method:

```java
// Appends the doubly-linked list beginning at node front2 to the end of
// the doubly-linked list beginning at node front1.
// Preconditions: front1 and front2 each refer to the first node
// in a doubly-linked list.

public static void append (IntNode front1, IntNode front2) {
    // Go to the end of the first list.
    IntNode temp = front1;
    while (temp != null)
        temp = temp.forwards;

    // Now attach the second list to the end of this one.
    temp.forwards = front2;
    // Go over to the front of the second list and link it back
    // to the end of the first.
    temp = temp.forwards;
    temp.back = temp;
}
```
1. Complete the table below with a good set of test cases for this method:

<table>
<thead>
<tr>
<th>Values in the nodes in the list 1 (in order)</th>
<th>Values in the nodes in the list 2 (in order)</th>
<th>Significance of this test case</th>
</tr>
</thead>
</table>

2. Trace the method on each of your test cases. Write a one-sentence description of each bug you find.

3. Fix as many of the bugs as you can. You can write corrections directly on the code.