0. Implementing `frontToBack`

Write a method `frontToBack` that could be added to the `LinkedIntList` class that moves the first element of the list to the back end of the list. Suppose a `LinkedIntList` variable named `list` stores the following elements from front (left) to back (right):

```
list: 18 -> 4 -> 27 -> 9 -> 54 -> 5 -> 63
```

If you made the call of `list.frontToBack();`, the list would then store the elements in this order:

```
list: 4 -> 27 -> 9 -> 54 -> 5 -> 63 -> 18
```

If the list is empty or has just one element, its contents should not be modified. Do not call any other methods on the `LinkedIntList` object such as `add`, `remove`, or `size`. Do not create new `ListNode` objects (though you may have as many `ListNode` variables as you like. Do not use other data structures or mutate the data of any existing nodes.

Solution:

```java
public void frontToBack() {
    if (front != null) {
        ListNode current = front;
        while (current.next != null) {
            current = current.next;
        }
        current.next = front;
        front = front.next;
        current.next.next = null;
    }
}
```