**CSE 143: Computer Programming II**

**QuickCheck: ArrayIntList Solutions (due Tuesday, January 13)**

**0. runningTotal**

Add a method `runningTotal` to the `ArrayIntList` class, that returns a new `ArrayIntList` that contains a running total of the original list. In other words, the \( i^{th} \) value in the new list should store the sum of elements 0 through \( i \) of the original list. For example, suppose a variable `list` stores the following sequence of values:

\[ [2, 3, 5, 4, 7, 15, 20, 7] \]

If the following call is made:

```
ArrayIntList list2 = list.runningTotal();
```

The variable `list2` should store the following sequence of values:

\[ [2, 5, 10, 14, 21, 36, 56, 63] \]

The original list should not be changed by the call. If the original list is empty, the result should be empty.

**Solution:** Two solutions are shown below:

```java
public ArrayIntList runningTotal() {
    ArrayIntList result = new ArrayIntList(this.elementData.length);
    if (size > 0) {
        result.add(this.elementData[0]);
        for (int i = 1; i < this.size; i++) {
            result.add(result.get(i - 1) + this.elementData[i]);
        }
    }
    return result;
}
```

```java
public ArrayIntList runningTotal() {
    ArrayIntList result = new ArrayIntList(this.elementData.length);
    if (size > 0) {
        result.elementData[0] = this.elementData[0];
        for (int i = 1; i < this.size; i++) {
            result.elementData[i] = result.elementData[i - 1] + this.elementData[i];
        }
        result.size = size;
    }
    return result;
}
```