Question 1: Sound

<table>
<thead>
<tr>
<th>Part (a)</th>
<th>limitAmplitude</th>
<th>4½ points</th>
</tr>
</thead>
</table>

**Intent:** Change elements of samples that exceed ±limit; return number of changes made

+3 Identify elements of samples to be modified and modify as required
  +1 Consider elements of samples
  +½ Accesses more than one element of samples
  +½ Accesses every element of samples (no bounds errors)

+2 Identify and change elements of samples
  +½ Compares an element of samples with limit
  +½ Changes at least one element to limit or −limit
  +1 Changes all and only elements that exceed ±limit to limit or −limit appropriately

+1½ Calculate and return number of changed elements of samples
  +1 Initializes and updates a counter to achieve correct number of changed samples
  +½ Returns value of an updated counter (requires array access)

<table>
<thead>
<tr>
<th>Part (b)</th>
<th>trimSilenceFromBeginning</th>
<th>4½ points</th>
</tr>
</thead>
</table>

**Intent:** Remove leading elements of samples that have value of 0, potentially resulting in array of different length

+1½ Identify leading-zero-valued elements of samples
  +½ Accesses every leading-zero element of samples
  +½ Compares 0 and an element of samples
  +½ Compares 0 and multiple elements of samples

+1 Create array of proper length
  +½ Determines correct number of elements to be in resulting array
  +½ Creates new array of determined length

+2 Remove silence values from samples
  +½ Copies some values other than leading-zero values
  +1 Copies all and only values other than leading-zero values, preserving original order
  +½ Modifies instance variable samples to reference newly created array

**Question-Specific Penalties**

−1 Array identifier confusion (e.g., value instead of samples)
−½ Array/collection modifier confusion (e.g., using set)
Question 1: Sound

Part (a):

```java
public int limitAmplitude(int limit) {
    int numChanged = 0;
    for (int i = 0; i < this.samples.length; i++) {
        if (this.samples[i] < -limit) {
            this.samples[i] = -limit;
            numChanged++;
        }
        if (this.samples[i] > limit) {
            this.samples[i] = limit;
            numChanged++;
        }
    }
    return numChanged;
}
```

Part (b):

```java
public void trimSilenceFromBeginning() {
    int i = 0;
    while (this.samples[i] == 0) {
        i++;
    }
    int[] newSamples = new int[this.samples.length - i];
    for (int j = 0; j < newSamples.length; j++) {
        newSamples[j] = this.samples[j+i];
    }
    this.samples = newSamples;
}
```