

The primary methods for manipulating an `ArrayList<E>` are:

<code>add(E value)</code>	appends value at end of list
<code>add(int index, E value)</code>	inserts given value at given index, shifting subsequent values right
<code>clear()</code>	removes all elements of the list
<code>get(int index)</code>	returns the value at given index
<code>remove(int index)</code>	removes and returns value at given index, shifting subsequent values left
<code>set(int index, E value)</code>	replaces value at given index with given value
<code>size()</code>	returns the number of elements in list

1. Write a static method called `stutter` that takes an `ArrayList` of strings as a parameter and that replaces every string with two of that string. For example, if the list stores the values `["how", "are", "you?"]` before the method is called, it should store the values `["how", "how", "are", "are", "you?", "you?"]` after the method finishes executing.
2. Write a static method called `minToFront` that takes an `ArrayList` of integers as a parameter and that moves the minimum value in the list to the front, otherwise preserving the order of the elements. For example, if a variable called `list` stores the following values: `[3, 8, 92, 4, 2, 17, 9]` and you make the following call:

```
minToFront(list);
```

The value 2 is the minimum, so the list should store the following values after the call: `[2, 3, 8, 92, 4, 17, 9]`. You may assume that the list is not empty.

3. Write a static method called `maxLength` that takes an `ArrayList` of strings as a parameter and that returns the length of the longest string. It should return 0 if passed an empty `ArrayList`.
4. Write a static method called `removeEvenLength` that takes an `ArrayList` of strings as a parameter and that removes all of the strings of even length from the list.

Solution to Sample ArrayList Problems

1. Two possible solutions appear below.

```
public static void stutter(ArrayList<String> list) {
    for (int i = 0; i < list.size(); i += 2) {
        String s = list.get(i);
        list.add(i, s);
    }
}

public static void stutter(ArrayList<String> list) {
    for (int i = list.size() - 1; i >= 0; i--) {
        String s = list.get(i);
        list.add(i, s);
    }
}
```

2. One possible solution appears below.

```
public static void minToFront(ArrayList<Integer> list) {
    int minIndex = 0;
    for (int i = 1; i < list.size(); i++) {
        if (list.get(i) < list.get(minIndex)) {
            minIndex = i;
        }
    }
    int min = list.remove(minIndex);
    list.add(0, min);
}
```

3. One possible solution appears below.

```
public static int maxLength(ArrayList<String> list) {
    int max = 0;
    for (int i = 0; i < list.size(); i++) {
        String s = list.get(i);
        if (s.length() > max) {
            max = s.length();
        }
    }
    return max;
}
```

4. One possible solution appears below.

```
public static void removeEvenLength(ArrayList<String> list) {
    int i = 0;
    while (i < list.size()) {
        String s = list.get(i);
        if (s.length() % 2 == 0) {
            list.remove(i);
        } else {
            i++;
        }
    }
}
```