

## ArrayList – class WeatherData

---

This question involves analysis of weather data. The following WeatherData class has an instance variable, temperatures, which contains the daily high temperatures recorded on consecutive days at a particular location. The class also contains methods used to analyze that data. You will write two methods of the WeatherData class.

```
1 public class WeatherData {
2     /** Guaranteed not to be null and to contain only non-null entries */
3     public ArrayList<Double> temperatures;
4     /**
5      * Cleans the data by removing from temperatures all values that are
6      * less than lower and all values that are greater than upper, as
7      * described in part (a)
8      */
9     public void cleanData(double lower, double upper)
10    { /* to be implemented in part (a) */ }
11    /**
12     * Returns the length of the longest heat wave found in temperatures,
13     * as described in part (b)
14     * Precondition: There is at least one heat wave in temperatures
15     * based on threshold.
16     */
17    public int longestHeatWave(double threshold)
18    { /* to be implemented in part (b) */ }
19    /** There may be instance variable, constructors,
20     * and methods that are not shown */
21 }
```

The question continues on the following page.

## ArrayList – class WeatherData

---

Write the `cleanData` method, which modifies the `temperatures` instance variable by removing all values that are less than the `lower` parameter and all values that are greater than the `upper` parameter. The order of the remaining values in `temperatures` must be maintained.

For example, consider a `WeatherData` object for which `temperatures` contains the following.

99.1	142.0	85.0	85.1	84.6	94.3	124.9	98.1	101.0	102.5
------	-------	------	------	------	------	-------	------	-------	-------

The three shaded values shown would be removed by the method call `cleanData(85.0, 120)`.

99.1	142.0	85.0	85.1	84.6	94.3	124.9	98.1	101.0	102.5
------	-------	------	------	------	------	-------	------	-------	-------

The following shows the contents of `temperatures` after three three shaded values are removed as a result of the method call `cleanData(85.0, 120.0)`.

99.1	85.0	85.1	94.3	98.1	101.0	102.5
------	------	------	------	------	-------	-------

Complete method `cleanData`.

```
/**
 * Cleans the data by removing from temperatures all values that are
 * less than lower and all values that are greater than upper, as
 * described in part (a)
 */
public void cleanData(double lower, double upper)
```

## ArrayList – class WeatherData

---

Write the `longestHeatWave` method, which returns the length of the longest heat wave found in the `temperatures` instance variable. A heat wave is a sequence of two or more consecutive days with a daily high temperature greater than the parameter `threshold`. The `temperatures` instance variable is guaranteed to contain at least one heat wave based on the `threshold` parameter.

For example, consider the following contents of `temperatures`.

100.5	98.5	102.0	103.9	87.5	105.2	90.3	94.8	109.1	102.1	107.4	93.2
-------	------	-------	-------	------	-------	------	------	-------	-------	-------	------

In the following sample contents of `temperatures`, all heat waves based on `threshold` temperature of 100.5 are shaded. The method call `longestHeatWave(100.5)` would return 3, which is the length of the longest heat wave.

100.5	98.5	102.0	103.9	87.5	105.2	90.3	94.8	109.1	102.1	107.4	93.2
-------	------	-------	-------	------	-------	------	------	-------	-------	-------	------

In the following sample contents of `temperatures`, all heat waves based on `threshold` temperature of 92.5 are shaded. The method call `longestHeatWave(92.5)` would return 4, which is the length of the longest heat wave.

100.5	98.5	102.0	103.9	87.5	105.2	90.3	94.8	109.1	102.1	107.4	93.2
-------	------	-------	-------	------	-------	------	------	-------	-------	-------	------

Complete method `longestHeatWave`.

```
/**
 * Returns the length of the longest heat wave found in temperatures,
 * as described in part (b)
 * Precondition: There is at least one heat wave in temperatures
 * based on threshold.
 */
public int longestHeatWave(double threshold)
```