

APCS Free Response A1 (ArrayList) Java Version

Assume that the standard library classes (e.g., `java.util.ArrayList`) are imported in any program that uses a program segment you write. If other classes are to be imported, that information will be specified in individual questions. Unless otherwise noted, assume that all methods are called only when their preconditions are satisfied. A Quick Reference to the AP Java subset classes is included in the case study insert.

A1

A researcher wishes to calculate some statistical properties for a collection of integer data values. The data values are represented by the array `tally`. The indexes of the array represent the possible values of the actual data values from zero to the maximal value (15 in the example below). Each array location contains the frequency (number of occurrences) of the value corresponding to its index. In the example below, `tally[4]` is 10, which means that the value 4 occurs ten times in the collection of data; whereas `tally[8]` is 0, which means that the value 8 does not occur in the data collection.

<u>tally</u>	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Frequency	0	0	10	5	10	0	7	1	0	6	0	10	3	0	0	1

Part A

You will write the static method `calculateModes` of the class `Stats` which is described as follows. Method `calculateModes` returns an `ArrayList` which contain the mode(s) found in parameter `tally`. The size of the returned array is equal to the number of modes. A **mode** is defined as a value that occurs with maximal frequency. If there is more than one such value, each is considered a mode of the data. In the example above, the modes are 2, 4, and 11, because they each occur 10 times and all other values occur fewer than 10 times.

In writing `calculateModes` you may call the `Stats` method `findMax` specified below which returns the maximum value in an `ArrayList`. Using the example array, `findMax(tally)` returns an `Integer` object, say `obj`, such that `obj.intValue()` returns the value 10. **Do NOT write `findMax`.**

```
/**
 * precondition: nums.size() > 0; nums contains Integer objects
 * postcondition: returns the maximal value in nums
 */

private static Integer findMax(ArrayList nums)
```

Do NOT write the body of findMax

In writing `calculateModes`, you may call method `findMax` specified above.

Complete method `calculateModes` below.

```
/**
 * precondition: tally.size() > 0; tally contains Integer objects
 * postcondition: returns an ArrayList that contains the modes(s);
 *                the ArrayList's size equals the number of modes.
 */

public static ArrayList calculateModes(ArrayList tally)
```

Part B

You will write the method `kthDataValue` of the `Stats` class which is described as follows. Method `kthDataValue` returns the `k`th data value when the data values are considered in sorted order. Recall that the indexes of the array represent possible data values and that each array location contains the frequency of the value corresponding to its index.

In the example reprinted below, the first ten data values are 2, the next five data values are 3, and the next ten data values are 4. For this example, `kthDataValue(tally, 1)` returns 2, `kthDataValue(tally, 14)` returns 3, `kthDataValue(tally, 15)` returns 3, and `kthDataValue(tally, 16)` returns 4.

Value	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Frequency	0	0	10	5	10	0	7	1	0	6	0	10	3	0	0	1

Complete method `kthDataValue` below.

```
/**
 * precondition: tally.size() > 0; tally contains Integer objects
 *                0 < k <= total number of values in data collection
 * postcondition: returns the kth value in the data collection
 *                represented by tally
 */

public static int kthDataValue(ArrayList tally, int k)
```

[*Owen L. Astrachan*](#)

Last modified: Wed Jul 10 12:05:08 EDT 2002