Modes

**Part A**

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 indices 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.

tally

| 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 |

You will write the static method calculateModes of the class Stats which is described as follows. Method calculateModes returns an int array which contains the mode(s) found in parameter tally. The length of the returned array is equal to the number of modes. A **mode** is defined as a value that occurs most often (greatest frequency) in the set, and more often than at least one other value. If several values occur with the same frequency, each value is a mode of the set. In the example above, the modes are 2, 4, and 11, because they each occur 10 times and 10 times is more other than the other values. As another example, the set {2,2,5,5,8,8} does not have a mode (why?). If there is no mode, return an empty array. How should your code handle that?

In writing calculateModes you may call the Stats method findMax specified below which returns the maximum value in array. Using the example array, findMax(tally) returns 10.

/\*\*

\* precondition: nums.length > 0

\* postcondition: returns the maximal value in nums

\*/

private static int findMax(int[] nums)

**Do NOT write the body of findMax**

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

Complete method calculateModes below.

/\*\*

\* precondition: tally.length > 0

\* postcondition: returns an int array that contains the modes(s);

\* the array's length equals the number of modes.

\*/

public static int[] calculateModes(int[] tally)

**Part B**

You will write the method kthDataValue of the Stats class which is described as follows. Method kthDataValue returns the kth data value when the data values are considered in sorted order. Recall that the indices 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.length > 0  
 \* 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(int[] tally, int k)

**Testing**: as always, you will test your code with the input that is given in the problem and see if you get the expected output. But that’s only the start of testing. What situation can you dream up that causes the program to crash?