

Mode-Histogram Lab

Part I: Implement these methods as given below. The **ModeHistogram.java** file has been started for you

Part II: Re-implement these methods using ArrayList instead of int[]. In the java file, you will see each method repeated twice (**overloading**)– once you'll use built-in java arrays and the other time you'll use ArrayList.

1. A *mode* is a value in an array that is larger than both the value immediately before it in the array and the value immediately after it. In other words, a mode occurs at index k in the array A if $A[k] > A[k-1]$ and $A[k] > A[k+1]$. The array is *unimodal* if the value increase until they reach a mode, then decrease, so that there is only one mode. For example, the array A shown below is unimodal with its mode occurring at index 4. Assume that the mode does not occur at the first or last entry in the array.

<u>Index k</u>	<u>A[k]</u>
0	3
1	5
2	9
3	10
4	12
5	11
6	9
7	4

- a) Write method **isMode**. `isMode` returns true if `data[k]` is larger than `data[k-1]` and larger than `data[k+1]`; otherwise, it returns false. In the example above, the call `isMode(A, 4)` returns true and the call `isMode(A, 5)` returns false.
- b) Write method **modeIndex**. `modeIndex` returns the index of the mode of data. You may assume that `data` is unimodal and the mode occurs at an index k , where $0 < k < \text{data.length}() - 1$. In the example above, the call `modeIndex(A)` returns 4.

- c) Write method **printHistogram**. `printHistogram` returns a `String` containing a character histogram of a unimodal array of nonnegative values, `data`, such that the longest bar of the histogram (the mode) has `longestBar` characters `barChar`, and all other bars have a number of `barChar` characters proportional to the corresponding value in the array `data` (rounding down).

For example, assume that vector `data` contains the values shown below.

The call `printHistogram(data, 20, 'x')` will return a `String` with the histogram shown in the square box below.

<u>Index k</u>	<u>data[k]</u>	<u>Length of bar</u>	<u>String returned as a result of the call printHistogram(data, 20, 'x')</u>
0	3	5	<pre>xxxxx xxxxxxxx xxxxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxxxxx xxxxxxxx</pre>
1	5	8	
2	9	15	
3	10	16	
4	12	20	
5	11	18	
6	9	15	
7	4	6	

note about the String returned here:

there are no spaces at the end of the lines

the last line does NOT have a '\n' character like the others

Unit Testing: We will give you (fairly soon) a Junit testing file for your class that will contain a basic set of tests for testing out your methods. You, of course, will be responsible for creating a more robust set of tests to ensure your methods are working properly.