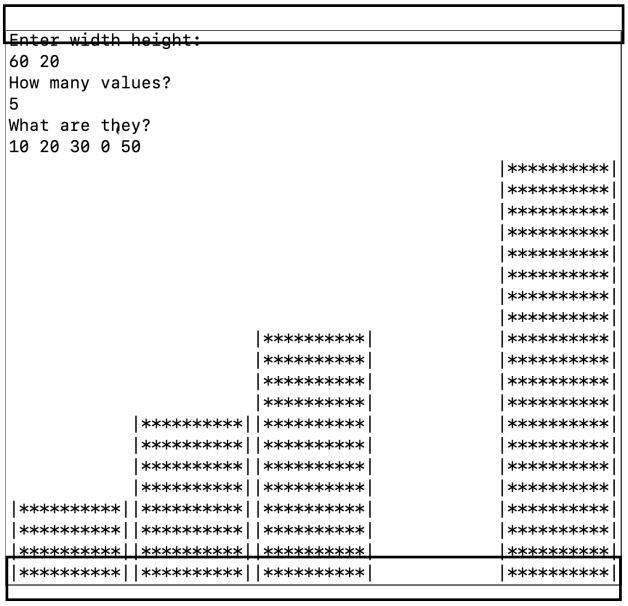
CSC171 — Lab 4

Ascii Bar Charts



The objective of this activity is to gain experience working with arrays. You will be reading a sequence of values from the user to produce a vertical ascii bar chart.

The key idea behind this project is to represent an "image" as a 2D array of characters. This representation is easily displayed using regular calls to `System.out.print`, by treating each "row" of the image to be printed as a line of text.

Submit your work as a Java file named `Histogram.java` (with public class `Histogram`) to blackboard by 1159pm Thu Oct 27th.

Requirements

1. **Setup the arrays.** The first stage is to read the data from the user. Prompt the user to enter the `height` and `width` (both integers) of the image array. Create a new array (call it the "image array") by constructing a new 2D array of dimension `height x width`.

   Next, prompt the user to enter a number (integer) of values to graph, then read each of the values into a new array of doubles. You can assume the values will always be nonnegative.

2. **Rescale the values.** One you have the values, rescale them such that the largest value is equal to the number of rows to print.

   (a) Identify the maximum value in the value array.

   (b) Divide each element in the value array by that maximum value.

3. **Mark the image buffer.** For each "bar" of the bar chart, you should "mark" the image, by setting the "pixels" to be the '*' character. The height of each bar is the rescaled value. To find the width of each bar, divide the total image width by the number of values to be graphed.

4. **Display the image.** Print the image to the screen row-by-row. You should print each row by printing each "pixel" character-by-character. At the end of each row, print a newline symbol to go to the next row.

5. **Testing.** You can test that your program works by choosing the largest values of width and height that fit in your output window, and then test very simple inputs (such as an incresing sequence 1 2 3 4, or a decreasing one 4 3 2 1).