Assignment SPIM 1

Due Date: February 27

Purpose

You will use SPIM, our MIPS emulator, to write some code in assembly language. In this project, you will use arrays, loops, and conditionals. It's just like Comp 115!

Problem

Given a list of **up to** 20 positive integers, we wish to find some basic statistics and display the results in a nice way.

Input

The user will input a list of positive integers in the range of {1..15}, ending with a negative value. Note that the same number may be input more than once. Prompt the user only once.

Output

The program should display:

- the minimum value
- the maximum value
- the integer mean
- the variance, s^2. This is calculated as follows:

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

  where n = the number of values, x_i is a specific value, and $\bar{x}$ = mean of all the values.

- a bar graph that shows the counts of each value. For example, if the range were {1..4} and the 1 was seen twice, 2 was seen once, 3 was not seen, and 4 was seen three times, then a suitable bar graph, where a 0 is a placeholder symbol, would look like:

```
1 00
2 0
3
4 000
```

- display a message after the graph indicating that your program has completed

Be sure to label all output; display all of the output values as integers.

Specifics

- Since we have not yet covered floating point operations, all computations should be done using integer arithmetic.

- For various operations, you will need one or more arrays. To do this, create one or more 20-element arrays in the data portion of the program, using arbitrary integers. This will allocate storage for any data to come.
- You must have a good introductory comment including your name(s), a description of the program, a description of the input, and a description of the output. Comment registers as well as possible, so that a reader can figure out what each register holds. Of course, with a limited number of registers, some may be reused, so comment the best you can. Although we can't write true functions yet, you can group code together and use jumps to simulate functions. Comment each of these groups in a general way (e.g., "Find mean of the list."). Finally, line up the assembly code in some consistent way, so that it is as readable as possible.

Notes

For full credit, turn in your source code via email before 11:59:59 PM of the 27^{th}. Name your file *lastName*SPIM1.a; for example, my submission would be called gousieSPIM1.a. You may turn in more than one version, but I will grade only the latest submission.

Turn in a printed copy in class on February 28. You will have to figure out how to print this (Linux may not recognize the printer in csLab).

This is a fairly trivial problem in Python or C++, and I encourage you to first write the solution in one of those languages. You can then translate that program to SPIM. Nevertheless, it will probably take longer than you expect to get it working in assembly language. Do things in small stages, making sure everything works before moving on.

Computers in the future may weigh no more than 1.5 tons.
– *Popular Science* magazine, 1949