Assignment SPIM 1

Due Date: February 23

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 30 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 up to 30 positive integers in the range of \{1..20\}, ending with a negative value. The same number may appear multiple times. Prompt the user only once.

Output
The program should display:

- the number of values input
- the minimum value
- the maximum value
- the integer mean
- the mode (the value seen most often)
- a bar graph that shows the counts of each value. For example, if the range were \{1..3\} and 1 was seen twice, 2 was seen once, and 3 was seen three times, then a suitable bar graph, where a 0 is a placeholder symbol, would look like:

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

Be sure to label all output; display all of the output values as integers. You may display any suitable placeholder symbol in your graph.

Specifics
- Since we have not yet covered floating point operations, all computations should be done using integer arithmetic.
- To find the mode, you will need one or more arrays. To do this, create an \(n\)-element array in the data portion of the program, using arbitrary integers. This will allocate storage for any data to come.
At the end of the program, display some kind of finishing statement, such as “Program completed.” This will show me that your program finished properly and is not in an infinite loop.

You must include a good introductory comment including your name, 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
This is a fairly trivial problem in C or C++, but it will take longer than you expect to get it working in assembly. Do things in small stages, making sure everything works before moving on.

Turn in your source code via email as an attachment. Name your file with your first initial followed by your last name and finally “SPIM1.a,” as in mgousieSPIM1.a.

Turn in a printed copy in class on February 24th. You will have to figure out how and where to print. Write/print and sign the Wheaton Honor Code Pledge on what you turn in: “I have abided by the Wheaton College Honor Code in this work.”

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