CS 35101 Computer Organization

Project 2

25 points

Project Description

You will learn how fractions (floating point numbers) are stored in the computer and how to use MIPS assembly language instructions to work with such numbers.

In statistics, arithmetic mean μ (or simply mean, average, or expected value) of a series $a_0, a_1, a_2 \ldots a_{N-1}$ is defined as:

$$\mu = \frac{1}{N}\sum_{i=0}^{N-1} a_i$$

Write an assembly program named `average.asm` that reads a list of floating point numbers from the keyboard and displays the measure given above on the simulator's console.

The input specifications are given below:

- Prompt the user for the number of floating point values that the user will enter. You **MUST** save this number as in integer or 10 points are deducted from the assignment. Later you will need to convert it to a float.
- By using a loop, repeatedly prompt the user to enter the floating point numbers. You may read the floating point number the user entered into a floating point register. Assume the format for the number to enter is 3 decimal numbers, one for the integer part and two numbers after the decimal point; e.g. the user may enter 4.29, -5.90, 0.02, etc.

For example, given the following input:

A=[0.11 0.34 1.23 5.34 0.76 0.65 0.34 0.12 0.87 0.56]

The program's output should be similar to given in the following:

The Mean of A = 1.03

Prompt the user for each number separately. An example dialog is:

```
How many numbers would you like to average? 3
Please enter a 3 digit decimal d.dd:  4.12
Please enter a 3 digit decimal d.dd:  0.99
Please enter a 3 digit decimal d.dd:   1.87
The average is: 2.326666666666
```

Project Submission and Deliverables

Submit your code on Canvas. Please submit the following two files. Make certain that your assembly code is properly organized (indented, commented, contains your name at the top of the code) .

1. The **MIPS Assembly Code Language file** titled `average.asm`
2. The **Report file pdf** with screen prints of your working code titled `average.pdf`.

Your report must include:

1. Your name
2. A list of the assembly code file
3. A brief summary of project implementation
4. Results showing the working code via **screen prints.** Taking photographs of your working code is not acceptable. Use the print screen/print window capabilities for your particular operating system.
5. The conclusion listing the lessons learned and problems faced

If a portion of your code does not work, explain to potentially receive partial credit.
If your report does not load, you will lose points.

Grading

- Working Code (90%)
- Report including results (10%)

If your code does not assemble, a minimum of 50% will be deducted from your score.

If your code does not adhere to the specifications, a minimum of 80% will be deducted from your score.

You must use good programming standards; i.e. use loops when necessary, proper indentation, no hard coding of values, follow register conventions and comments in your code, etc.

Reference Materials

- Patterson and Hennessy: Chapters 2.1–2.4, 2.9–2.10, Chapter 3.1–3.6, Appendix A.10
- Slides on Mars Simulator and MIPS assembly language
- farh-to-cel.asm code file