

CS35101 Project 3 - Report

Name: Matthew Miller

Included ASM files:

average.asm

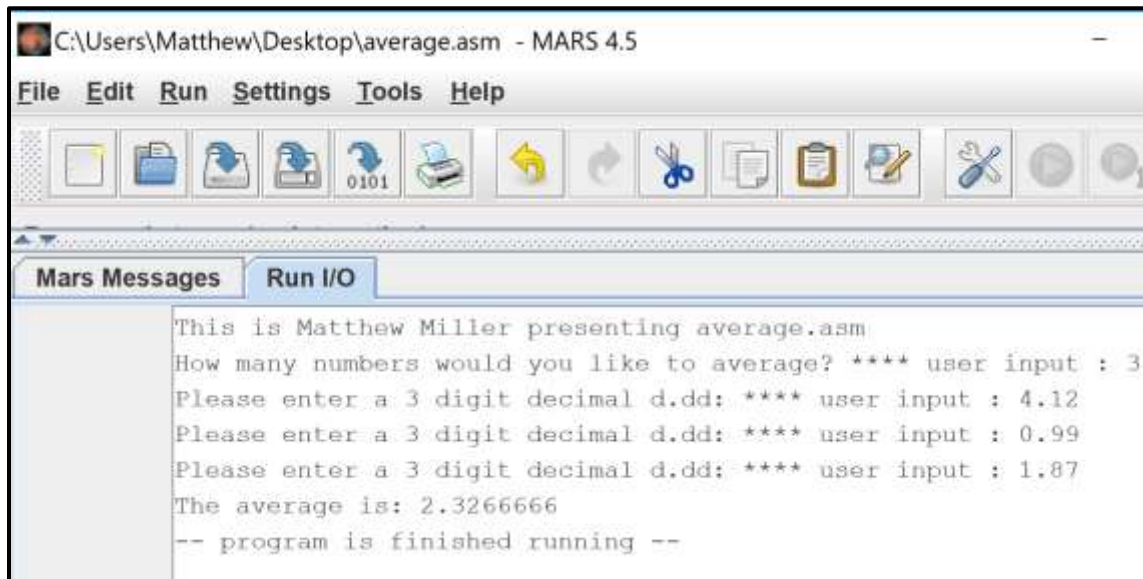
Implementation Summary:

I implemented the project's solution by prompting the user for an integer value that represents the number of floating-point-numbers to be averaged. First, I stored the user's input into a standard saved register (\$s0). Then, I placed a copy into a floating-point-register (\$f1). Next, I converted the integer-value in \$f1 to a floating-point-value with the cvt.s.w instruction.

After conversion, I called two procedures (collectUserFloat and averageFloat). The collectUserFloat was a simple procedure that collects all the user's input and stores a running total. After a running total has been stored, the averageFloat procedure conducts a division instruction and prints the resulting floating-point-value to the screen.

There are obviously string-based prompts which are used throughout the program for the user's instruction and clarity.

Results:



```
C:\Users\Matthew\Desktop\average.asm - MARS 4.5
File Edit Run Settings Tools Help
[Icons: New, Open, Save, Print, Run, etc.]
Mars Messages Run I/O
This is Matthew Miller presenting average.asm
How many numbers would you like to average? **** user input : 3
Please enter a 3 digit decimal d.dd: **** user input : 4.12
Please enter a 3 digit decimal d.dd: **** user input : 0.99
Please enter a 3 digit decimal d.dd: **** user input : 1.87
The average is: 2.3266666
-- program is finished running --
```

Conclusion:

I learned how to use and manipulate floating point numbers in the MIPS architecture. I initially had some problems figuring out how to conduct some of the floating-point specific instructions. However, upon review of the presentation slides, I was able to find the right instructions and complete the project without much issue.