Adding Some Numbers

Brian C. Ladd

Spring 2024

Learning Outcomes

After completing this program, students will be able to

- Write a MIPS program with a single `main` and global data.
- Use `git` to turn in a program.
- Use MIPS system calls to read and write data and strings.

Note: For *this* assignment you may make use of the **`mult`** instruction it MIPS.

Overview

Students will write a program that prompts the user for three integer values, a, b, and c. The program will then use three constants defined in the program, x, y, and z, to calculate $value = a \times z + b \times y + c \times x$. The program will store the value in its own variable and print the results of the calculation.

Procedure

1. Read this entire assignment. Spend a few minutes thinking about what you are going to do.
2. This program (and all programs in this course) are expected to follow the current Departmental Coding Standards (see the `notes` folder for this class for a copy). In particular:
   - Every file must identify the programmer, course, and assignment in a comment.
   - Every *function* should have a header comment identifying the purpose, parameters (and, in this class, where they are passed in), and return value. Document what it *means*.
   - Every program must include a `README` (Coding Standards discuss formatting) that discusses how you *tested* your program (how do you know it works) and notes as required by the assignment.
   - Your code **must** assemble without error as submitted. I will load it into Mars and press the **Assemble** button. The code should be ready to run.

   This is an upper-division programming course. No program will be graded if it fails to meet these requirements.

3. You will write a program in MIPS assembler using the MARS simulator found at `http://www.cs.missouristate.edu/MARS/`. The MIPS program itself is just a text file; there is no requirement that you type it into the simulator.

4. The program will:

```
Define global data: a,b,c, value, x,y,z
  (x, y, z have initial values that matter)
main:
  Prompt user for a
  Read integer
  Store integer in a
  Prompt user for b
  Read integer
  Store integer in b
  Prompt user for c
  Read integer
  Store integer in c

  Evaluate a*z+b*y+c*x
  Store result in value

  Print The value is ${value}
```

5. The vaules of x, y, z come from your three initials (if you have fewer than three, pick letters to get to three; if you have more than three, pick three of them). The point is to have three letters, each of which you turn into a number:

   1. 7 if the initial is in the first third of the alphabet
   2. 31 if the initial is in the middle third of the alphabet
   3. 57 if the initial is in the last third of the alphabet

   Then pick one of them and subtract 100 from it (so it is a negative number).**Note:** I am not very concerned about which values you use, just having different values and one that is negative.

6. Consider how you would write this in Java. All variables are `static` integers (equivalent to global). It is very important to do this **before** trying to write the assembly.

   Write your Java solution and include it in your `README`.

7. Define the seven variables in the **`.data`** section. You can initialize all but x, y, z to 0. Initialize those three to the expected constant values.

   You will also have to define strings in the data for *prompts* and for the *results*. **`.asciiz`** (ASCII string ending in zero) is what you use define a string. Put all strings *after* your integers (we will talk about why).

8. Write `main` in the **`.text`** segment to do what your Java program says to do.

Submit through Gitea

Check your working code into `git`. When you have one working "feature" in your program, checking it in to `git` is a *good thing*. Protect yourself from making things *worse*.

Use a `.gitignore` file to exclude any garbage files your IDE produces from the repository. They will cost you points.

You have an account on the departmental **Gitea** server. Submit your work in your shared organization in a repo named `pAddingSomeNumbers`