Due date: Wednesday, Jan 30 (11:59 PM)

CS 271, Winter 2013 — Programming Assignment 1

Submission instructions

- Submit a single MIPS assembly file via TEACH: `http://engr.oregonstate.edu/teach`
- Your file must be named `PA1`[your-username]`.asm`
- For example, my submission would be named `PA1walkiner.asm`

Learning objectives

The goal of this assignment is to get some initial experience programming in MIPS assembly language. Specifically, by the end of this assignment, you should feel comfortable:

1. manipulating registers,
2. doing basic integer arithmetic,
3. defining constants and variables in data memory,
4. reading from and writing to data memory, and
5. using the system call interface to perform I/O.

Description and Requirements

Write and test a MIPS assembly program that performs the following tasks, in order:

1. Print an introduction that includes: your name, a title, and a brief description of the program.
2. Prompt the user for two numbers and save these in two registers.
3. Perform several calculations using these two numbers, and *save each result in data memory*.
   Results to compute: sum, difference, product, (integer) quotient, and remainder of the two numbers.
4. For each computed result, *retrieve it from memory* and print it in a nice way.
5. Print a concluding message and exit.

Important: In Task 3, each of the results must be stored to a different location in data memory. In Task 4, these results must each be retrieved from data memory.

You should begin by declaring all of the constants and variables you will need in the data segment.

Your text segment should be divided into five sections, corresponding to the five tasks. Use comments to indicate the sections. This means that you will perform all of the calculations and store all of the results into memory (Section 3) before you retrieve and print each of them (Section 4).

At the end of your program you should use the "exit" system call (10) to exit cleanly from your program.

Finally, your program must be fully documented. You should not comment every line. Instead, break it into small, logical chunks that can be succinctly described (e.g. "calculate the difference and save to memory"). Use of pseudo-code to help you reason about your program is encouraged. This will become more important as your assembly programs become more complex.

An example execution is provided on the next page.

Example execution

Below is an example execution that illustrates what your program should do. Your output need not match this output exactly. However, the structure (title, name, intro, results, goodbye) and the order of the results (sum, difference, product, quotient, remainder) should be the same. Most importantly, it must be clear and easy to tell that your program is computing and printing the correct things.

Program output is in `typewriter font` while user input is in ***red, bold, italic font***.

```
Math in MIPS! by, Eric Walkingshaw

Enter two numbers and I'll show you the sum,
difference, product, quotient, and remainder.

First number: 9
Second number: 2

9 + 2 = 11
9 - 2 = 7
9 x 2 = 18
9 / 2 = 4 R 1

Goodbye.
```