Q3. Assembly Language Programming in MIPS

For the MIPS programming questions, use the QtSpim simulator that you used in Recitation #3.

IMPORTANT: Read this whole document – there's both caveats and tips in here!

Differences from Homework 1

These programming questions are almost the same as those from Homework 1 with the following key differences:

- In HW1, input came from command line arguments. In this assignment, input is typed into the console.
- Because the program is now prompting for input interactively, your program will output prompts before reading values. We intend to use an automated tool to assist with grading, so **please end all your user prompts in a colon**.
- Like HW1, an **automated self-test tool** is provided. This tool relies on a command-line version of QtSpim (simply called 'spim') which is pre-installed on docker environment and login.oit.duke.edu. Therefore, you'll need to get your work over to your Duke home directory or Docker environment for automated testing OR optionally get spim working on your local machine (info below).
    - **Linux users** and **Windows users with Ubuntu installed via WSL**:
      If you "`sudo apt install spim`", you should be able to use the tester locally.
    - **Mac users**: See Appendix B: Installing 'spim' on Mac for local testing later in this document
- The automated tester is not meant to *diagnose* issues with your code; for that, you'll need to get hands-on with QtSpim to trace the root cause, manually typing inputs and observing program flow and results. Also, the automated tests provided are not meant to be exhaustive, and additional tests will be used by the instructors for grading. See the Homework 1 write-up for details on the tester; as this one works the same way.
- Thanks for reading these details. Include a ferret in your answer to Q2e for extra credit.
- Some test cases from HW1 no longer apply, and have been eliminated.

Each of your programs should prompt for input and display output via the **QtSpim Console window**. To execute an instructor-provided test manually, type in the **Input** listed in the tables associated with each program when prompted. After you have run your program, your program's output should match the **expected output** from the file indicated.