Systems I
Lab 6
Simple assembly programming

1 The assembler/simulator

Assembly programming in RISC-V will show us how a general-purpose programmable circuit—a *processor*—may be used, and what it would need to be capable of doing. We will write assembly programs and run them through an *assembler/simulator* for a RISC-V processor and memory. This simulator will allow us to move step-by-step through our programs, watching the registers and memory change as we do so.

We will use the *Venus Simulator*, which you can access in your web browser at:

`https://venus.cs61c.org`

You may also find useful, as we do this work, the following RISC-V assembly reference, which includes a number of details and examples about how to write the instructions and annotate your assembly code so that it can be correctly parsed:

`https://marks.page/riscv/`

2 A pre-written program

Get some code: The best way to understand how this simulator works is to try using it. Begin by opening a tab with some sample source code that we will load into the simulator and then step through:

`https://bit.ly/COSC-175-2526f-add-two`

Copy it into the simulator: Open the simulator in another browser tab. On top, if it is not already selected, click on the `Venus` tab. Below, you will see a basic terminal and shell prompt. At this prompt, end the following command:

```
[user@venus] /# edit add-two.s
```

That will bring you to the `Editor` tab with an empty text editor. **Copy the sample source code from the previous tab and paste it into this editor.** Then click the `Save` button.

Examine and then run the program: *We will walk through this part together during lab*, first examining the source code, and then seeing its translation into machine code steps as we step through the execution of the program.

3 Finding the max

Now open a different assembly source code file:

```
https://bit.ly/COSC-175-2526f-find-max
```

Go back the *Venus Simulator*, and return to the `Venus` tab and its terminal. Open a new file in the editor:

```
[user@venus] /# edit find-max.s
```

Once again, **copy and paste** the source code from the earlier tab and into this editor, then click `Save`.

This file contains the skeleton of an assembly program that traverses an array of values, remembering the maximum value so far. **Your task** is to write the loop that finds the maximum value in that array. Once your loop is done, the code already at the end of the program will store that array into a main memory location.

Remember to download your code: When you have written code within the `find-max.s` file that is worth keeping, go back to the `Venus` tab terminal. Then, download it, like so:

```
[user@venus] /# download find-max.s
```

4 How to submit your work

Copy your completed `find-max.s` to the `lab-6` folder within your Google Drive folder.

This assignment is due on Oct-30, 11:59 pm.