

4.16 Lab Assignment 3

[Start Assignment](#)

Due Monday by 11:59pm **Points** 95 **Submitting** a file upload
File Types asm and s **Available** after Feb 22 at 10am


Purpose: Write your a MIPS assembly language program using an array made with a label

Program

Make an array of word values that has 10 total elements. The array should be made using the label a.

The elements should be these integer values: 5, 6, 7, 10, 14, 17, 21, 22, 25, 36

After the array a is made, access the first element of a and store the array's first value in the register \$t0. Then access the last element of a and store the array's last value in the register \$t1.

[Use The MIPS Technical Document To See The MIPS Assembly Language instructions](https://ccsf.instructure.com/courses/47907/files/7405493/download?download_frd=1)  [\(https://ccsf.instructure.com/courses/47907/files/7405493/download?download_frd=1\)](https://ccsf.instructure.com/courses/47907/files/7405493/download?download_frd=1) . You may wish to download the document and refer to it as you are writing the program.

Write comments in your program that state your name, the programming logic, and any assembly language instruction line level details you feel you'd like to explain how you are using the assembly language instructions to perform the calculation of the sum.

Note: This program does not require any output. Just store the array's first value in the register \$t0 and the last element of a and store the array's last value in the register \$t1.

This program is to be saved in its own file, with a .asm or a .s file extension. You may only use a .asm or .s file extension (MIPS assembly language programs are named using these files extensions). Once your program is working correctly, submit the program file to this assignment to receive a grade for your program.

Example Programs

Below is the source code of some example MIPS assembly language programs that can be used to help you write your programs. Feel free to download and run them in MARS to get yourself used to running assembly language programs and seeing how MARS's viewing areas, menus, and controls work.

Experiment with as much of MARS as you can, and pay close attention to what is displayed as you run the programs.

Try to run each program step by step, using MARS's Run > Step option so you can see what part of the MIPS architecture is changing as each instruction is being run. Then try to run each program fully using MARS's Run > Go option making sure you see in all of MARS's viewing areas and controls what happens in MARS during a program's normal running.

[intarrayelementaccess1.asm](https://ccsf.instructure.com/courses/47907/files/7444996/download?download_frd=1) ↓ (https://ccsf.instructure.com/courses/47907/files/7444996/download?download_frd=1) [intarrayelementaccess2.asm](https://ccsf.instructure.com/courses/47907/files/7445070/download?download_frd=1) ↓ (https://ccsf.instructure.com/courses/47907/files/7445070/download?download_frd=1)