

Quiz #4

Due: Wednesday September 27 – by 4:10 PM (Before Class)

Submission: Two Files Uploaded to Carmen

This quiz is individual work. You can consult class materials, you can research algorithms, but you are not allowed to collaborate with others including AI. In case you are tempted, I have already asked ChatGPT. YMMV but it gets an F from me – no kidding! Sometimes it does better, this time it tanked.

Part 1: Coding task

Task: Write a short assembly program that reverses the order of all elements in an array. Your program should work on arrays of arbitrary length > 1 (assuming the array fits into the RAM).

Start by downloading Quiz_4.asm from Carmen and open it in CCS (using File>Open File... or by dragging the file from your download folder onto a CCS window).

Quiz_4.asm will create an array for you to reverse. It is a word array and has 18 elements

```
array_1 = {0, 2, 4, 6, ..., 30, 32, 34}
```

You will submit results for `array_1`. However, your program needs to work for any array – not just the given one. Make sure to test your code on arrays of both even and odd lengths.

You can choose to pursue one of the following three levels of increasing difficulty (or all three) but you are allowed to **submit only one solution at your choice of difficulty level**.

Level 1 – For up to 93/100 points

There are **no restrictions** on memory access: Your program can write to any memory location in RAM or FRAM and use all available core registers. You can create a second array, copy the given array in reverse, and copy back to the original array. Gets the job done (you can still get an A) but is very wasteful of resources.

Level 2 – For up to 100/100 points

Your program is **not allowed to write to any addressed memory location outside of the given array**. The above strategy of creating a second array is not allowed at this level. However, you are allowed to use all available core registers.

Level 3 – For 10 bonus points, i.e., up to 110/100 points

Your program is **not** allowed to write to any addressed memory location outside of the array. You are **not** allowed to use any core register other than R4 and R5. At this level, any tricks you employ need to be **fully justified**.

Do not forget to add comments on your code.

Part 2: Preparing two files for submission – Please follow all instructions

PDF File: Screenshot of your code and your memory browser with the reversed array

At the top of your PDF file indicate which level you are attempting to solve. Add a single, legible screenshot of your code that shows the memory initialization and all instructions. Please switch to Light Mode for easy readability. Remember, a good screenshot shows all the code uninterrupted, is properly sized, easy to read, includes line numbers to facilitate easy feedback.

To take a partial screenshot in Windows, press the **Windows Button and Shift and S**. This will bring up a crosshair which allows you to take the screenshot of the selected rectangular area. In macOS, press **command and control and shift and 4** to bring up the crosshair.

After you execute your code, pause it, and take a single screenshot of your memory browser clearly displaying the reversed array at the top of the RAM. Choose “**16-Bit Signed Int**” as the display mode as shown in Fig. 1.

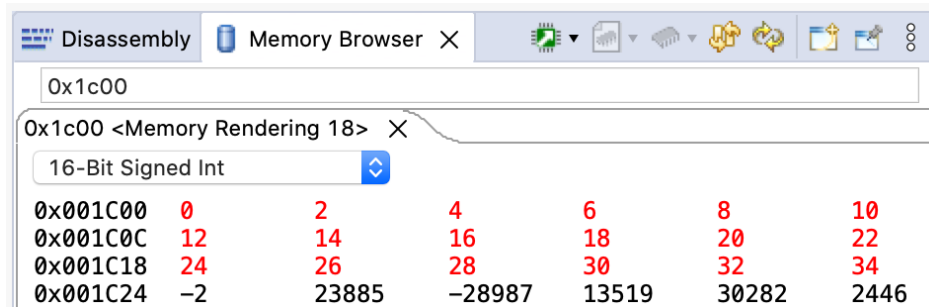


Figure 1 Quiz_4.asm populates the array shown in red. Your program will reverse it.

Text File: Your source code

Save the final version of your source code as txt file so it can be read in Carmen, and make sure that both files reflect your name, e.g., `firstname_lastname.txt` or `name_number.txt`. Easiest way to do this is to use File>Save As in CCS.

I will randomly select and run source code files throughout the semester. If your source code file does not produce the results you demonstrate in your memory browser, you will receive zero points for the assignments – end of story.

Make sure to submit the correct source code in the correct format – no word or PDF files for source codes.

Part 3: Submission

Make sure that your PDF file contains all three parts

1. Your chosen level of difficulty
2. A legible screenshot of your code
3. A good screenshot of your memory browser, displaying the reversed array as 16-bit signed integers

Submit your PDF and text file to Carmen before class on Wednesday September 27, 2023.