Midterm #1

Due: Wednesday October 4 – 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 other students or AI.
If you are wondering, ChatGPT did not do that well – not an F, but no more than C+/B-.
I am sure you can do better than that.

Part 1: Coding task

Task: Write a short assembly program that finds the maximum value in an array and the number of times this value appears.

Your program should use **indexed mode** when accessing the array; and return the max. value and its frequency in predefined variables called **`max_value`** and **`max_count,`** respectively.

Start by downloading Midterm_1.asm from Carmen and add your code to it as indicated. You will work with the constant LENGTH, and use the labels that are provided in the file. You can reorder the labels as you like and need more if needed (but remember to keep your code compact). Do not remove any lines from the file, it will populate the array you will work with.

You will submit your results for the array created by the file, however, your program needs to work correctly for **any word array containing signed integers**.

Make sure to add comments to your code and use descriptive labels.

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

PDF File: Screenshot of your memory browser and your code

After you execute your code, pause it, and take a single screenshot of your memory browser clearly displaying `max_value`, `max_count`**,** and the given array at the top of the RAM. Choose "**16-Bit Signed Int**" as the display mode.

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.

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 is missing or 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

1. A screenshot of your memory browser displaying `max_value`, `max_count`, and the given array as 16-bit signed integers
2. A screenshot of your code

Submit your PDF and text files to Carmen before class on Wednesday October 4, 2023.

Hints/Checklist

Before you start writing any code, you are encouraged to understand the steps of the problem you are trying to solve.

It is always a good idea to create a flowchart and/or pseudocode of your program. Make sure to simplify your flowchart as discussed in class to minimize the number of conditionals and keep your code tangle free.

Try to keep your code minimal. The task can be completed in 15 lines of code (including the two lined for the infinite loop). Your code could use less or a little more, but if you are using considerably more ask yourself if you can simplify your program flow and/or implementation.

For loops make sure to check that you are addressing elements in your array correctly (remember you are working with a word array) and choose the addressing mode specified in the assignment.

Double check your stopping conditions to ensure that you indeed process all elements in the array and none of the adjacent values in memory.

Test your code at every step. When testing, step through your program to ensure that its flow is correct, i.e., that it is jumping to perform the correct actions under correct conditions.