

UNIVERSITY OF THE PHILIPPINES VISAYAS  
COLLEGE OF ARTS AND SCIENCES  
DIVISION OF PHYSICAL SCIENCES AND MATHEMATICS

CMSC 131

Introduction to Computer Organization and Machine Level Computing  
A.Y. 2022 - 2023

Assignment Guide

Prepared by:

Jayvee B. Castañeda  
Instructor

*ACADEMIC INTEGRITY*

*As a student of the University of the Philippines, I pledge to act ethically and uphold the value of honor and excellence. I understand that suspected misconduct on given assignments/examinations will be reported to the appropriate office and if established, will result in disciplinary action in accordance with University rules, policies and procedures. I may work with others only to the extent allowed by the Instructor.*

## Laboratory Exercise #10

### Reading

- Read [Section 5.1 of Paul Carter's PC Assembly Book](#)

### Practice Exercise:

- Execute “**array1.asm**” and interface it with “**array1c.c**”. What is the purpose of *array1.asm*? What is the purpose of *array1c.c*? Explain the relationship between them.
- Analyze the sample codes **array1.asm** and **array1c.c**. How are arrays being implemented in assembly?

# Problem #10.

## Definitions of “Array”

*Array (n.) – An exclamation said by the Filipinos when they are in pain.*

*Array (n.) – A part of the sun that is being emitted through light*

*Array (n.) – A popular brand of biscuit/cookie sometimes eaten with milk*

**If you didn't get any of them, then go array.**

*Note: Please don't take these seriously as these are not actual definitions of the word “array”.*

- Write an assembly program that *sorts the values in an array*.
- Create a program named “*sortArray.asm*” that sorts the values in an array **taken from the user input**, which should interface with “*array1c.c*”. Please refer to the file used in the practice exercise.
- You are free to choose what type of sorting algorithm you will use, just make sure to **write that in the documentation** of your code. You must ask the user for the size of the array and each of the values individually. The program should **print the sequence of the elements first** before it is sorted.
- The output of your program should be something like this:

```
Enter array size: 7
Enter value @ array[0]: 3
Enter value @ array[1]: 7
Enter value @ array[2]: 2
Enter value @ array[3]: 9
Enter value @ array[4]: 4
Enter value @ array[5]: 1
Enter value @ array[6]: 5
```

```
The array contains:
3 7 2 9 4 1 5
```

```
Sorting array...
The sorted array is:
1 2 3 4 5 7 9
```

```
Done.
```

- A good programming practice is to *write comments on important line of codes* for readability and documentation.
- Save your program in a file called *SurnameFirstLetterOfFirstName\_lab10.asm* in camel case. For instance, if your surname is “Juan Dela Cruz”, submit it as follows:

*DelaCruzJ\_lab10.asm*

- Take a screen recording of your working code and make sure to **record a video explaining each line of your code** as well as showing the correct output of your code. Use screen recorder application in Ubuntu (<https://itsfoss.com/best-linux-screen-recorders/>) or Windows (<https://atomisystems.com/screencasting/record-screen-windows-10/>)

### Submission Requirements:

1. Program Code (.asm file)
2. Screen Recorded Defense Video

**DEADLINE: January 5, 2022, 11:59 PM**

Rubric for Programming Exercises				
Program (50 pts)	Excellent	Good	Fair	Poor
<b><i>Program Execution</i></b>	Program executes correctly with no syntax or runtime errors (9-10)	Program executes with minor (easily fixed) error (4-8)	Program executes with a major (not easily fixed) error (2-3)	Program does not execute (0-1)
<b><i>Correct Output</i></b>	Program displays correct output with no errors (9- 10)	Output has minor errors (6-8)	Output has multiple errors (3-5)	Output is incorrect (0- 2)
<b><i>Design of Output</i></b>	Program displays more than expected (7-8)	Program displays minimally expected output (5-6)	Program does not display the required output (3-4)	Output is poorly designed (0-2)
<b><i>Design of Logic</i></b>	Program is logically well-designed (9-10)	Program has slight logic errors that do not significantly affect the results (6-8)	Program has significant logic errors (3-5)	Program is incorrect (0-2)
<b><i>Standards</i></b>	Program is stylistically well designed (6-7)	Few inappropriate design choices (i.e., poor variable names, improper indentation) (4-5)	Several inappropriate design choices (i.e., poor variable names, improper indentation) (2-3)	Program is poorly written (0-1)
<b><i>Documentation</i></b>	Program is well documented (5)	Missing one required comment (4)	Missing two or more required comments (2- 3)	Most or all documentation missing (0-1)