RUBRIC
LABORATORY 1

Topic	Period
Introduction to C	22/JAN/2025 - 04/FEB/2025

Welcome to the introductory C lab

This activity aims to review different topics and capabilities of the C language, such as structures, functions, and pointers. It also promotes an organized way of software development by requiring a standardized project file structure, the usage of a compiling configuration framework, and a set of programming styling guides.

Rubric

Category	Task	Notes
Structure 15%	The project has the required folder structure, configuration file, header, and source files in the corresponding folders.	5, 3, 1 OR 0
	All the program’s functions are in separate source files from main.c, and their headers are in the .h files.	5, 3, 1 OR 0

	The project has a cmakelists.txt file that generates a makefile in the build folder. This file can be used to generate an executable with different configurations depending on the compilation mode.	5, 3, 1 OR 0
Program Performance, documentation, and styling 85%		4, 2 OR 0
		4, 2 OR 0
	The **first program of part** A must be organized and legible, and the main function must use a modular structure using functions.	3, 1, OR 0
	The **first program of part** A must be documented with Doxygen. The documentation must explain how to use the function, what it does, what it receives, and what it returns.	2, 1 OR 0
	The **first program of part** A must have comments within the functions that describe the code's operations.	2, 1 OR 0
	In the **first program of part** B, the function **CosineSeries** should allow the calculation of the cosine series between two numbers and return whether the sum of the results is positive or	4, 2 OR 0

	negative.	
	In the **first program of part** B, the main function must ask the user for the necessary data to invoke the **CosineSeries** function.	3 OR 0
	The **first program of part** B must be organized and legible, and the main function must use a modular structure using functions.	3, 1, OR 0
	The **first program of part** B must be documented with Doxygen. The documentation must explain how to use the function, what it does, what it receives, and what it returns.	2, 1 OR 0
	The **first program of part** B must have comments within the functions that describe the code's operations.	2, 1 OR 0
	The **second program of part** B must ask the user three numbers of seconds in the **main function** and print the exact dates accordingly.	4, 3 OR 0
	In the **second program of part** B, FillDate must be outside the main file and can return the calculated date to the main function using three integer pointers.	5, 3 OR 0

	The **second program of part** B must be organized and legible, and the main function must use a modular structure using functions.	3, 1, OR 0
	The **second program of part** B must be documented with Doxygen. The documentation must explain how to use the function, what it does, what it receives, and what it returns.	2, 1 OR 0
	The **second program of part** B must have comments within the functions that describe the code's operations.	2, 1 OR 0
	The **third program of part** B has to return a structure with date, month, and year attributes.	4, 2 OR 0
	In the **third program of part** B, the student declared the date structure correctly, and it is located in the project's .h file.	4, 2 OR 0
	In the **third program of part** B, the **main function** asks the user three numbers of seconds and uses the function to print the exact dates accordingly	4, 2 OR 0
	The **third program of part** B must be organized and legible, and the main function must use a modular structure using functions.	3, 1, OR 0

	The **third program of part** B must be documented with Doxygen. The documentation must explain how to use the function, what it does, what it receives, and what it returns.	2, 1 OR 0
	The **third program of part** B must have comments within the functions that describe the code's operations.	2, 1 OR 0
	In the **fourth program of part** B, the function **UpperRand** fills the base array and stores the modified version in the second array using a random-based strategy to modify the letters.	5, 3 OR 0
	In the **fourth program of part** B, the **PrintArray** function receives pointers to two arrays and correctly prints them on the terminal.	5, 3 OR 0
	The **fourth program of part** B, must have a main program that uses the functions **UpperRand** and **PrintArrays**.	4, 2 OR 0
	The **fourth program of part** B must be documented with Doxygen. The documentation must explain how to use the function, what it does, what it receives, and what it returns.	2, 1 OR 0
	The **fourth program of part** B must have comments	2, 1 OR 0

	within the functions that describe the code’s operations.	
	The **fourth program of part** B must be organized and legible, and the main function must use a modular structure using functions.	3, 1, OR 0

Explanation

Important: All programs must be compiled in order to be graded.

In the structure section:

- **5 points:** All of your programs fully meet the structure requirements.
- **3 points:** One of your programs does not meet the structure requirements.
- **1 point:** Only one of your programs meets the structure requirements.
- **0 points:** None of your programs meet the structure requirements.

In program performance, documentation, and styling:

- For tasks with **three scoring options**:
  - **Max points:** The program fulfills the task exceptionally well.
  - **Middle points:** The task was implemented but was not entirely successful.
  - **0 points:** The task was not implemented at all.
- For **boolean options**:
  - The task is only checked to determine whether it was **completed** or **not completed**.