

Workshop 07

In this workshop, we will practice working with functions, arrays, and structs. Furthermore, we will break down our code into multiple files. To achieve this, we will create a module that calculates basic statistics from an array of floating-point numbers.

Instructions

Coding

For this workshop, you will use as a starting point the provided **main.c** file. This file has all the required code for the main function already implemented. Your task will be to create a module called **stats** containing the definition of a structure, as well as a series of functions. The main file should not be altered! Your module should be broken down into two files: one called *stats.h* (header) and another one called *stats.c* (implementation).

- Inside your header *stats.h*, you should define **SIZE** with a value of 8 (this is a `#define`, not a variable), as well as a struct called **stats** with the following member variables: an int called **size**, and floats called **max**, **min**, **mean**, and **std**
- In your header file, you should also declare (but not define) all the functions described below. In your implementation file, you should provide the definitions of all these functions:
 - **get_max**: which takes as arguments an array of float and an integer (for the size of the array) and returns a float containing the maximum value in the array.
 - **get_min**: which takes as arguments an array of float and an integer (for the size of the array) and returns a float containing the minimum value in the array.
 - **get_mean**: which takes as arguments an array of float and an integer (for the size of the array) and returns a float containing the mean (average) value of the array.
 - **get_std**: which takes as arguments an array of float and an integer (for the size of the array) and returns a float containing the standard deviation of the array. The standard deviation is calculated using this formula:

$$\sigma = \sqrt{\frac{\sum (x_i - \mu)^2}{N}}$$

σ = population standard deviation

N = the size of the population

x_i = each value from the population

μ = the population mean

*Hint: You will need to use functions from the `math.h` library. **pow(x, 2.0)** returns the square of a variable x . And, **sqrt(x)**, returns the square root of a variable x . Note*

*that this function should not calculate the mean directly, but rather, call the previously defined function (**get_mean()**) to do so.*

- **get_stats**: which takes as arguments an array of float and an integer (for the size of the array) and returns a struct of the type **stats** containing all statistics about the array: its *size*, its *maximum*, its *min*, its *mean*, and its *standard deviation*. Note that this function should not calculate these values directly, but rather call other previously defined functions to do so.

Reflection

Breaking the code into a proper module will be considered as your reflection for this workshop. I.e., if you decide to define all structs and functions in your main file and submit it, instead of breaking it down in a module as specified, you will get full marks for coding, but lose marks for reflection. Make sure to use proper macros to avoid redefinition.

Submission

You should submit your source code files, **stats.c** and **stats.h**, here on Blackboard. Make sure to add a comment at the top of the source code files containing the name of the workshop, and your name. For example:

```
// Workshop 07 – Marcel Jar
```

Evaluation

get_max and get_min	0.5 marks
get_mean	0.5 marks
get_std	0.5 marks
get_stats	0.5 marks
reflection (header)	0.5 marks
reflection (implementation)	0.5 marks