2 DATA STRUCTURES

Lab Exercise 2: Functions, Arrays, Pointers and Dynamic Memory

Task-1: Functions and Arrays **(4 Points)**

Write a C++ program that creates a **dynamic array**. The size of the array should be taken as input from the user through the terminal. Once created, the array should be populated with random values in the range of 0-100 (both inclusive). Complete the function called **'findMinMax()'** that identifies the **minimum** and **maximum** values from the array and returns those values to the main function by updating parameters passed by reference. The starter code lab2_1.cpp is available on Brightspace.
Hints: Use the '**new**' operator to dynamically allocate memory for the array in the **heap**. Don't forget to delete the dynamically allocated memory at the end of the program.

The output of the program should look like the following:

```
Enter Size of the Array: 10
array[0]=99
array[1]=8
array[2]=79
array[3]=43
array[4]=65
array[5]=9
array[6]=66
array[7]=39
array[8]=38
array[9]=31
Min : 8
Max : 99
```

Figure 1

Task-2, 2D Arrays and Pointers (Matrix Arithmetic) **(6 Points)**

Download the starter code called lab2_2.cpp from NYU Class/Brightspace and complete the following functions:

a) **int** add(int** matrix1, int** matrix2, int rows, int cols)**
A function that adds two matrices, ***matrix1***, and ***matrix2*** (2D arrays). The method should store the results in another dynamically created 2D Array and return it to the calling program (main function). Both matrices passed to the function are of the same dimensions.

b) **void cleanup(int** matrix, int rows)**
A procedure that deletes a dynamically created ***2D array*** passed to it.

c) **int** subtract(int** matrix1, int** matrix2, int rows, int cols)**
A method that subtracts the matrix2 from matrix1 and returns the result to the calling program. Both ***matrix1*** and ***matrix2*** are of the same dimensions.

d) **int** multiply(int** matrix1, int** matrix2, int rows1, int cols1, int rows2, int cols2)**
A function that multiplies two matrices (matrix1 and matrix2) and returns the result to the calling program. You can assume that the number of columns of the first matrix is equal to the number of rows of the second matrix (a requirement for matrix multiplication).
How to Multiply Matrices: https://www.mathsisfun.com/algebra/matrix-multiplying.html

e) **void fill(int** matrix, int rows, int cols)**
A procedure that populates the matrix (2D array) with random integer values in the range of 0 to 20 (including 0 and 20).

f) **void display(int** matrix, int rows, int cols)**
A procedure that prints a matrix (2D array) in a two-dimensional form. (see *Figure 2*)

Comments are a crucial part of any program, and you should always write comments in your code. Comments worth 10% of your grade.

Note: Please do not modify the function headers given in the starter code. Keep in mind that the local variables created in a function will be destroyed upon returning from the function. Upload only two cpp files (lab2_1.cpp and lab2_2.cpp) to the Brightspace.

The output of your program should look like the following:

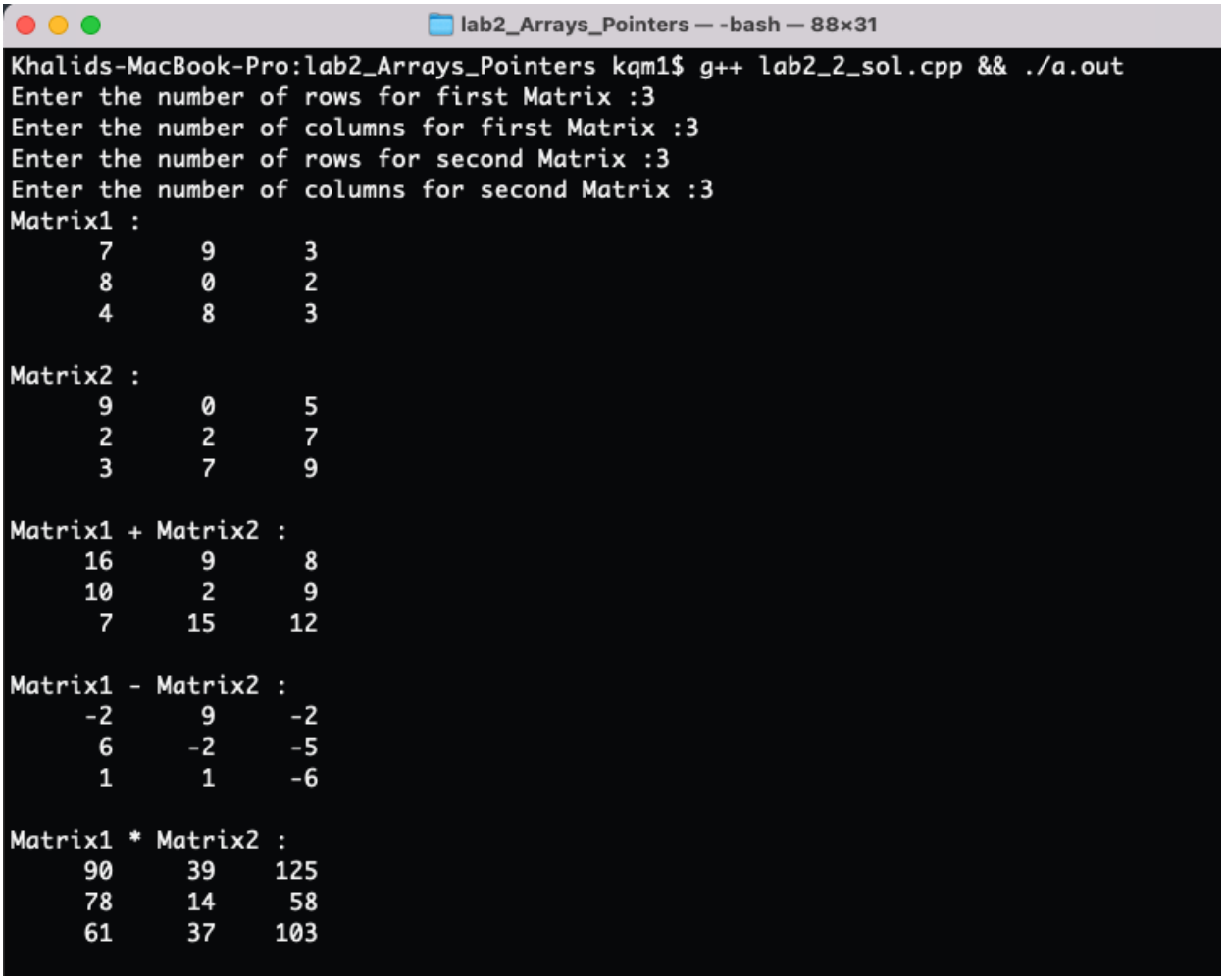

Figure 2

CODE OF CONDUCT

All assignments are graded, meaning we expect you to adhere to the academic integrity standards of NYU Abu Dhabi. To avoid any confusion regarding this, we will briefly state what is and isn't allowed when working on an assignment/lab-task.

Any documents and program code that you submit must be fully written by yourself. You can, of course, discuss your ideas with fellow students, as long as these discussions are restricted to general solution techniques. Put differently, these discussions should not be about concrete code you are writing, nor about specific results you wish to submit. When discussing an assignment with others, this should never lead to you possessing the complete or partial solution of others, regardless of whether the solution is in paper or digital form, and independent of who made the solution, meaning you are also not allowed to possess solutions by someone from a different year or course, by someone from another university, or code from the Internet, etc. This also implies that there is never a valid reason to share your code with fellow students, and that there is no valid reason to publish your code online in any form.

Every student is responsible for the work they submit. If there is any doubt during the grading about whether a student created the assignment themselves (e.g. if the solution matches that of others), we reserve the option to let the student explain why this is the case. In case doubts remain, or we decide to directly escalate the issue, the suspected violations will be reported to the academic administration according to the policies of NYU Abu Dhabi.

https://students.nyuad.nyu.edu/campus-life/community-standards/policies/academic-integrity/