

# COMP26020 Part 1– Assignment 1:

## Matrix Multiplication in C

The goal of this assignment is to write a C program performing a matrix multiplication operation. The program takes 3 command line parameters that are  $M$ ,  $N$  and *seed* and is launched as follows:

```
./mat-mult <M> <N> <seed>
```

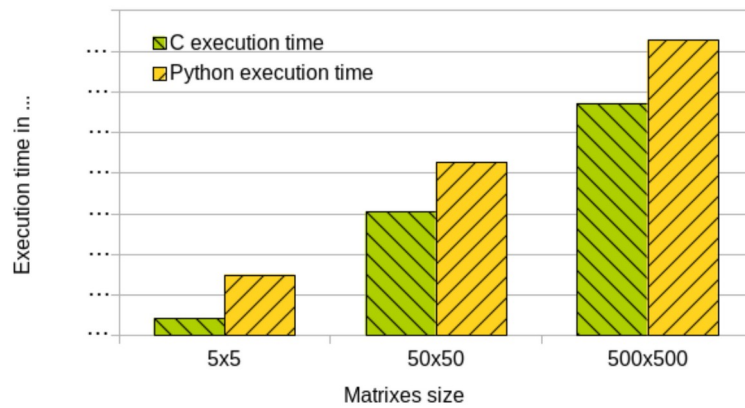
The program generates 2 matrixes *matrix 1* and *matrix 2*. *Matrix 1* has a size of  $M \times N$  and *matrix 2* has a size of  $N \times M$ . Both are filled with random integer numbers – each taking a value between 0 and 99 – through a generator initialized with *seed*. Both matrixes are printed on the standard output. Then the program multiplies A by B and prints the resulting matrix on the standard output. The program also measures the execution time of the multiplication operation and prints it, in seconds, on the standard output after the result matrix.

Here is a complete execution over one example:

```
./mat-mult 5 3 0
MATRIX 1:
83 86 77
15 93 35
86 92 49
21 62 27
90 59 63
MATRIX 2:
26 40 26 72 36
11 68 67 29 82
30 62 23 67 35
RESULT:
5414 13942 9691 13629 12735
2463 9094 7426 6122 9391
4718 12734 9527 12143 12355
2038 6730 5321 5119 6785
4879 11518 7742 12412 10283
EXECUTION TIME: 0.000004
```

Can download a program written in python that behaves exactly as described above at the following address: <https://olivierpierre.github.io/comp26020/labs/lab1/mat-mult.py>.

Once your C program is done, use the execution time reported to produce a graph comparing the performance of your C code vs the Python implementation for matrixes of sizes 5x5, 50x50, and 500x500. An example of such graph is below:



The requirements for this assignment are as follows:

- Given the variable size of the matrixes, they should be stored in memory using dynamic memory allocation, i.e. within buffers allocated through malloc. Keep in mind the important rules about dynamic memory allocation that were covered in the related video;
- The code should be well organized within multiple functions, correctly indented, and should compile without warnings;
- Using a wrong number of command line arguments, or incorrect values for these arguments should lead to a graceful exit and not a crash;
- This assignment will be graded in part using automated methods and because of this it is crucial that the output format described above is respected to the letter.

The deadline for this assignment is **November 10**, however it is advised to finish it sooner as lab assignment 2's deadline is also set on that date. To submit your assignment, push your code and the graph to the repository named "26020-lab1-S-Matrix Multiplication in C" present in the department GitLab. Make sure both files are present and pushed. Any modification to the repository passed November 10 will be considered a late submission.