



Project 3: Matrix Structure and Multiplication in C

You can only use C, nor C++, in the project. The project is an improvement of Project 2, but only `float` matrix structure is needed to implement.

Requirements

1. The programming language can **only be C, not C++**. Please save your source code into `*.c` files, and compile them using a C compiler such as gcc (not g++).
Try to use Makefile or CMake to manage your source code.
2. Design a **struct for matrices**, and the `struct` should contain the data of a matrix, the number of columns, the number of rows, etc.
3. Implement some functions to
 - create a matrix
 - delete a matrix
 - copy a matrix (copy the data from a matrix to another)
 - multiply two matrices
 - some other functions needed
4. When you run the program as follows, it will output the result into an output file such as `out32.txt`. The data files are the same with those in Project 2.

```
$/matmul mat-A-32.txt mat-B-32.txt out32.txt
```

```
$/matmul mat-A-256.txt mat-B-256.txt out256.txt
```

```
$/matmul mat-A-2048.txt mat-B-2048.txt out2048.txt
```

5. Try to improve the speed of matrix multiplication. Introduce **how you improve it in the report**. **You should explicitly introduce the differences between this one and Project 2.**
6. Compare the speed of your implementation with OpenBLAS (<https://www.openblas.net/>).

Rules:

1. Please submit your project report before its deadline. After the deadline, **23:59 on Oct. 24**, (even 1 second), **0 score!**
2. Please submit the files as: report.pdf, source1.c, source2.c, header.h. I think you do not need more than 5 files for the project. Please do **NOT** put the files into a compressed one.
3. Your score will also depend on the quality of your source code and your report. Your report should be easy to understand and describe your work well, especially the highlights of your work.
4. Please pay more attention to your code style. After all this is not ACM-ICPC contest. You have enough time to write code with both correct result and good code style. You will get deduction if your code style is terrible. You can read Google C++ Style Guide (<http://google.github.io/styleguide/cppguide.html>) or some other guide for code style.