Simple Matrix Multiplication

Matrix multiplication, a very basic operation in linear algebra, plays an important role in deep learning. In this project, we aim to implement and compare two programs implemented in C and Java.

Requirements

1. Implement a program in C and another in Java for matrix multiplication. You can implement a function for the multiplication. Only `float` is needed since it is the most commonly used data type in deep learning.

2. Measure the time of the computation. As the sizes of the metrics increase, the time should increase. Measuring the time is not easy since many factors will affect it.

3. Compare the performance of the two programs in C and Java, and explain the reason for any observed differences. I am sure the results will not be as you expected.

4. Perform additional comparisons and analysis to identify any interesting insights. You can put your efforts into comparisons and analysis. I am sure you will find many interesting things.

Tips:

1. As efficiency and analysis are the primary focus, random values can be used to generate vectors.

2. There is no need to customize a data type of arbitrary precision.

3. A vector/matrix can be generated using `float * vec1 = (float*)malloc(n * sizeof(float));`. Remember to free it using `free()`.

4. It is meaningless to measure the efficiency of a program that is generated in the DEBUG mode. The DEBUG mode is the default mode of many IDEs. If you compile a C program with the option `-O3`, you will be surprised.

Rules:

1. **Deadline**: Please submit your project report before its deadline. The deadline is **23:59, March 31**. After the deadline (even 1 second), **0 score!** No deadline extension for any students, even for those they enroll in the course late.

2. **Format**: Please submit the files as: *.pdf, xxx.c, xxx.Java. Please do **NOT** put the files into a compressed one. To uncompress a lot of files will cost time for instructors. PDF format can make the layout of the report to be consistent on any computer.

3. **Report**: 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. If you do not know how to write the report, just assume you are a developer in a company and are reporting to your manager that you have developed a good calculator.

4. **Code style**: Please pay more attention to your code style. This is not ACM-ICPC contest. You have enough time to write code with both correct results and a good code style. You will get a 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. You may lose points on your project if your code is ugly.

5. **ChatGPT**: You can use ChatGPT as your personal coach. Such as you can use ChatGPT or some other similar tools to polish the text in your report. But ChatGPT can also generate some rubbish that looks good but contains nothing. It will waste the time of the instructor to review those empty words, and also make you lose points on your project. ChatGPT can also generate a framework of the source code, and help improve the quality of your source code. If you use it for that, please mention it in the report. You are responsible for all materials you submit, not ChatGPT.