The methodology of School 21 makes sense only if peer-to-peer reviews are done seriously. Please read all guidelines carefully before starting the review.
- Please, stay courteous, polite, respectful and constructive in all communications during this review.
- Highlight possible malfunctions of the work done by the person and take the time to discuss and debate it.
- Keep in mind that sometimes there can be differences in interpretation of the tasks and the scope of features. Please, stay open-minded to the vision of the other.
- If you have not finished the project yet, it is compulsory to read the entire instruction before starting the review.

Guidelines

- Evaluate only the files that are in src folder on the GIT repository of the student or group.
- Ensure to start reviewing a group project only when the team is present in full.
- Use special flags in the checklist to report, for example, an “empty work” if repository does not contain the work of the student (or group) in the src folder of the develop branch, or “cheat” in case of cheating or if the student (or group) are unable to explain their work at any time during review as well as if one of the points below is not met. However, except for cheating cases, you are encouraged to continue reviewing the project to identify the problems that caused the situation in order to avoid them at the next review.
- Doublecheck that the GIT repository is the one corresponding to the student or the group.
- Meticulously check that nothing malicious has been used to mislead you.
- In controversial cases, remember that the checklist determines only the general order of the check. The final decision on project evaluation remains with the reviewer.

Main part

Exercise 00 - Still Counts

- Only files 'calculator.c' and 'setup.py' are present
- A resulting Python module can be installed with 'python setup.py install', C code is compiled during installation
- Functions 'add', 'sub', 'mul', 'div' are written in C and implement correspondent mathematical operations
- Zero division is handled and raises a proper built-in Python ZeroDivisionError

0 1 2 3 4 5

Exercise 01 - Split-Second

- File "monotonic.py" is present and includes function "monotonic()"
- Ctypes is used to access monotonic clock function in an OS library. No other third-party w rappers are used
- Function returns a number of seconds from a system monotonic clock

👎 0 1 2 3 4 5 👍

Exercise 02 - Autopilot

- Files "multiply.pyx" and "setup.py" are present
- "mul()" function is implemented and works for matrices up to 100x100 in size
- Multiplication work for integer numbers only and raises a Python exception on other input
- "mul()" handles the case when the number of columns in the first matrix is not equal to t he number of rows in the second matrix and raises a meaningful error

👎 0 1 2 3 4 5 👍

Bonus part

Exercise 00 - Bonus part

- Calculator can handle float values for one or both arguments

👎 0 1 2 3 4 5 👍

Exercise 02 - Bonus part

- Submission includes file `test_mul_perf.py` with a test that compares default and Cython -powered implementations. The latter should win!

👎 👍