Fast Fourier Transform

v1.0 — December 4th 2023

We provide you with a `fft.c` sequential C program that:

- Generate some white noise
- Computes a Fourier transform
- Applies some transformation to the Fourier coefficients
- Compute the inverse Fourier transform
- Outputs (a part of) the result as an audio file

This yields a "damped" sound reminiscent of what one can hear underwater.

Your primary goal is to make a parallel version of this code that runs as fast as possible *on a single compute node* — using MPI is allowed but it is really going to make your life harder.

You may use the external libraries that are usually available in computing centers.

Some things that you can do:

- Parallelize using OpenMP.
- Use vectorization.
- Try to mind the memory hierarchy.
- Produce a roofline analysis — how close are you to peak performance?

Whatever you do, you **must**:

1. *Describe* what you have done.
2. *Measure* the performance of your implementation and its scalability (both strong- and weak-scaling are fair game).
3. *Comment* these results: are they good or not? If not, why? Can you design an experiment that would confirm your opinion? Would it have been possible to tell in advance what has happened?

Your work must be submitted by Sunday, January 7th (2024) before 23:59 (using Moodle). You must submit your source code, a `Makefile` that compiles your code (without errors nor warnings, even with `-Wall`), and a ≈ 5 pages report in `.PDF` format that explains what you have done.

Please follow our guidelines about writing reports (on Moodle).

You must work in pairs (submit one report with both names)

If you believe that you have found an error in our programs (it does happen), or if you and your classmates are facing a common technical problem, don't hesitate to contact us.