Inside the Operation Optimization notebook, there are 4 problems. You need to use any resource at your disposal to optimize that code targeting either running performance or memory. You will run the same notebook targeting a Jetson Nano and a workstation (a personal computer or Google Colab). Following these steps:

1. Run the base code in both platforms and record the memory profiler and the time profiler information and save it
2. Optimize each function, I recommend you start by optimizing it on the workstation and then on the Nano.
3. If you try different approaches, keep a record of how much improvement you get (record both the raw number and the percentage of improvement)
4. Write a report with the following information:
   1. A graph and/or table comparing the performance for each of the problems between the optimized version and the non-optimized version separated by platform (nano vs workstation)
   2. A detailed explanation of the approach you took and the tools you used in your optimization
   3. A reflection of why there are different results between the two platforms that include insight on how knowledge about the platform's characteristics helped in making decisions on what type of performance you were prioritizing
5. Submit a zip file containing two jupyter notebooks (one with the code you optimized for the workstation and another one with the code for the nano) and your report.

Tips:

* We recommend running the code for each challenge separately so there is no influence between what happened in one function over the other. To achieve this, run the code for one of the challenges, reset your jupyter notebooks engine (without deleting the output for the one you already ran), and then run the other.
* You are free to pursue approaches that are outside of python/jupyter notebooks if you wish to write the function using a different programming language. You have to explain, however, how your code is equivalent and make sure that you are generating the same output.
* **Important:** on jetson nano, or platforms that do not support jupyter notebook well, run the code outside the notebook, copy and paste it to run directly/natively on ubuntu.

**Rubric**

For each of the 4 challenges (loop 1, loop 2, matrix multiplication, and laplacian operator), the points are distributed as follows:

* The code was transformed in some way that benefits memory or running performance but doesn't lose correctness (2 points)
* The code uses a tool for improving performance (e.g. Numba, Cython, etc.) (2 points)
* The code is tailored to each platform: nano or laptop. If the code is the same, the report clearly explains why the code yields the best results in both platforms (2 points)
* The report shows a clear understanding as to why the tools and transformations implemented made a difference (2 points)
* The code and the report are overall well organized, clean, and easily understandable (2 points)