Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра Обчислювальної Техніки

Лабораторна робота №1

з дисципліни "Проектування складних систем"

Тема: " Дослідження протоколу когерентності кешів MESIF"

Виконав: Перевірив:

студент групи ІП-93 Долголенко О. М.

Домінський В.О.

Київ 2023

**Зміст**

[Мета: 3](#_Toc128516692)

[Вихідні дані: 3](#_Toc128516693)

[Хід роботи 4](#_Toc128516694)

[Висновок: 11](#_Toc128516695)

[Посилання: 12](#_Toc128516696)

# Мета:

Розробіть два екземпляра свого варіанта багатопотокової програми на Java. Перший екземпляр багатопотокової програми розробіть за додержанням принципу локалізації, при цьому як скалярні так і векторні змінні не повинні розділюватися між потоками – механізм когерентності кешів MESIF при цьому працювати не буде. Другий екземпляр багатопотокової програми розробіть за використанням скалярних та векторних глобальних змінних та механізмів синхронізації потоків при зверненні до цих змінних – пошук актуального значення змінної при цьому буде відбуватися, відповідно до механізму когерентності кешів MESIF, у кешах L1D всіх ядер мікропроцесора. Побудуйте порівняльні графіки й зробіть висновки по відносній швидкодії цих екземплярів програми

# Вихідні дані:

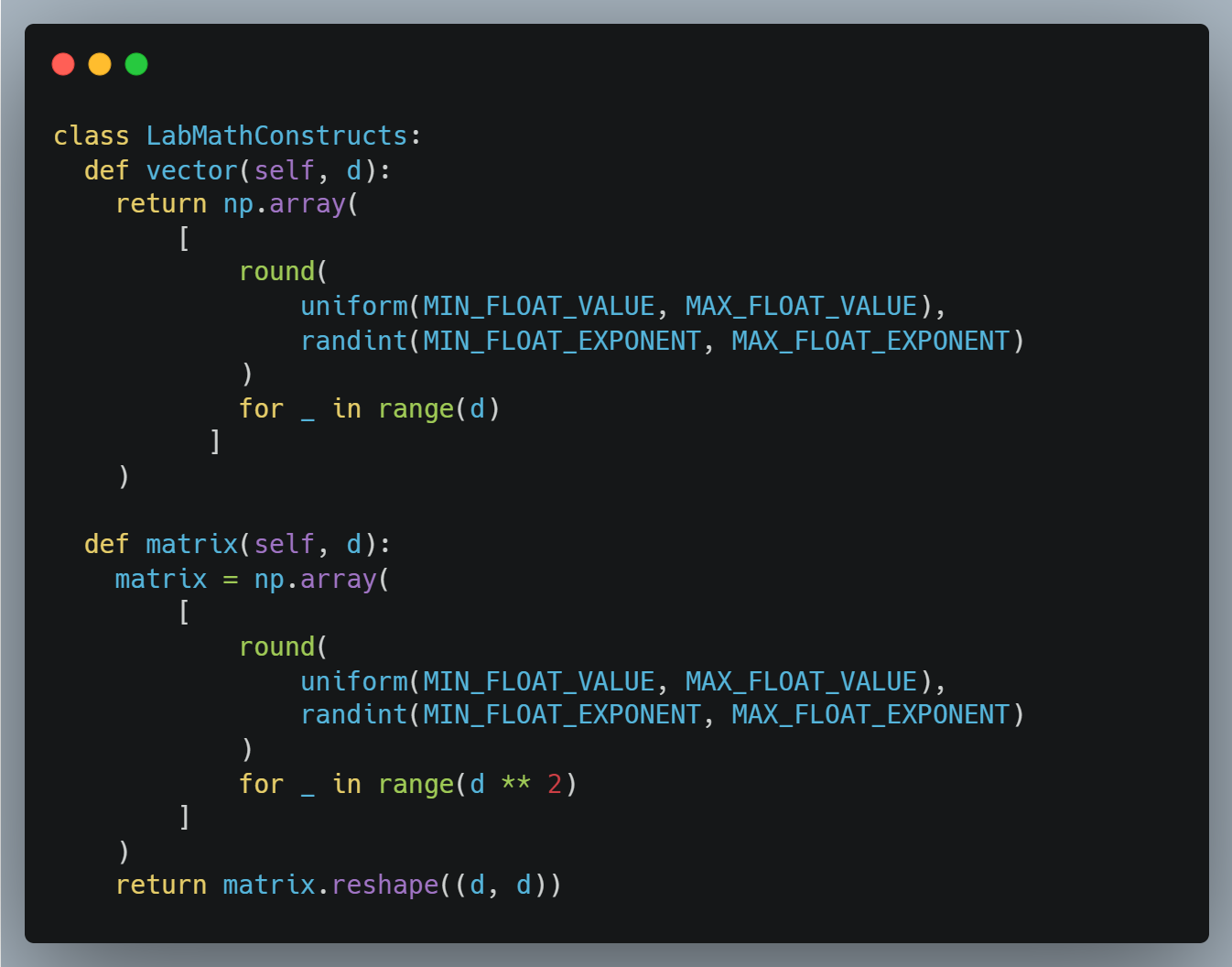
Варіант – 12

Функції:

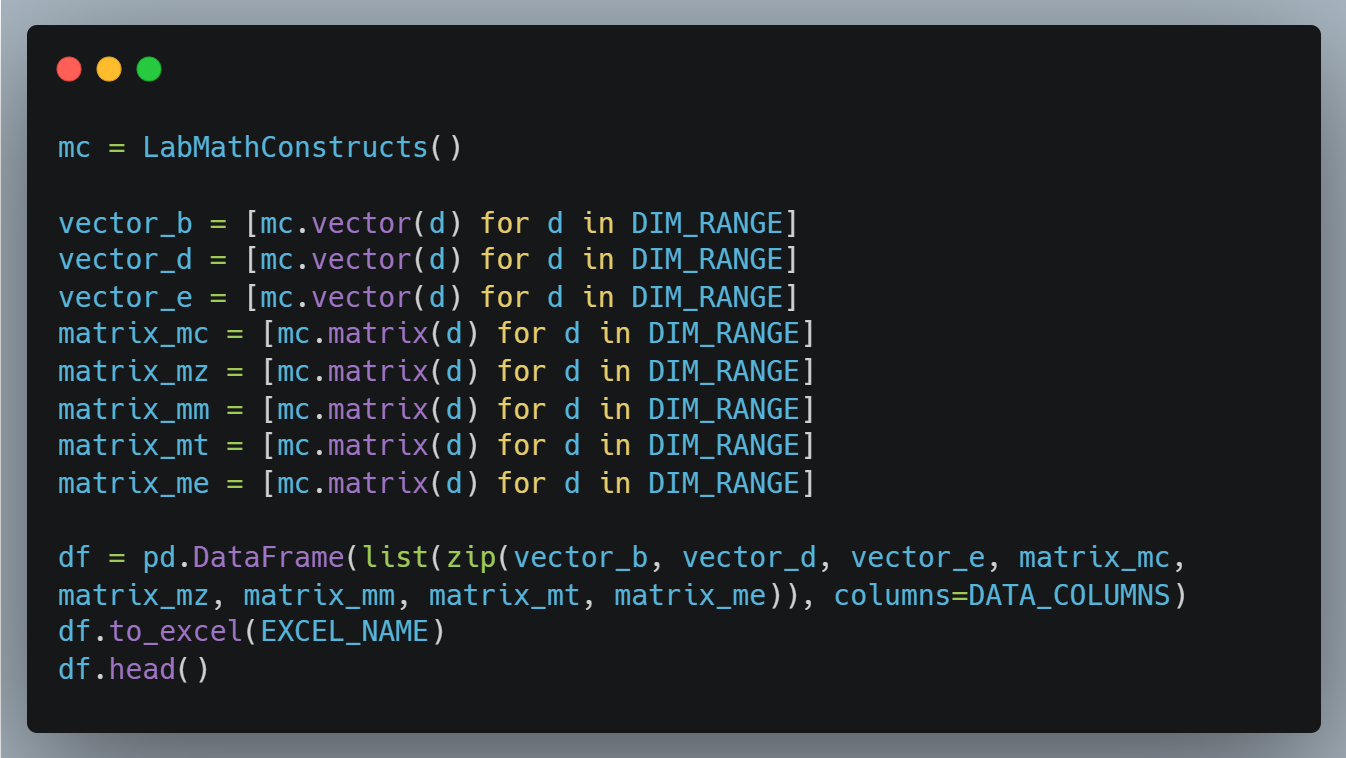
* А=В\*МС+D\*MZ+E\*MM
* MG=min(D+E)\*MM\*MT-MZ\*ME

## Хід роботи

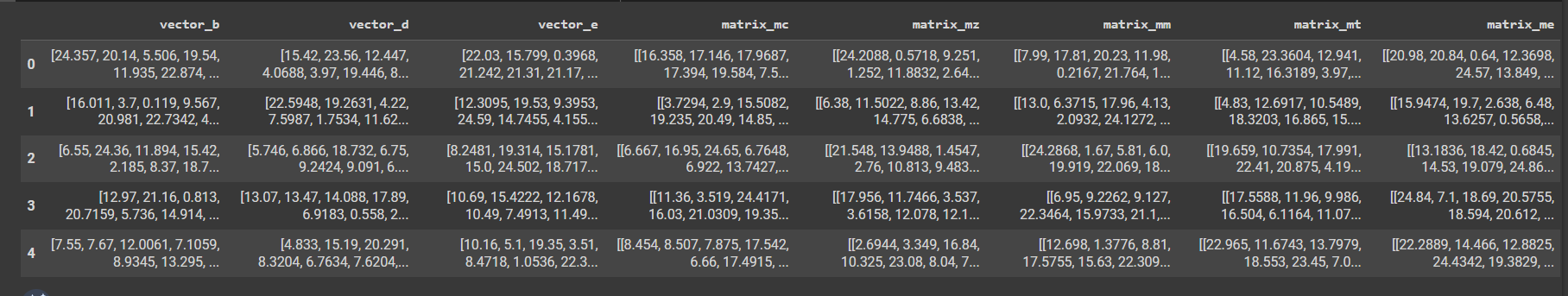
Для початку Ми створюємо Наші власні конструкції для вектору та матриці:



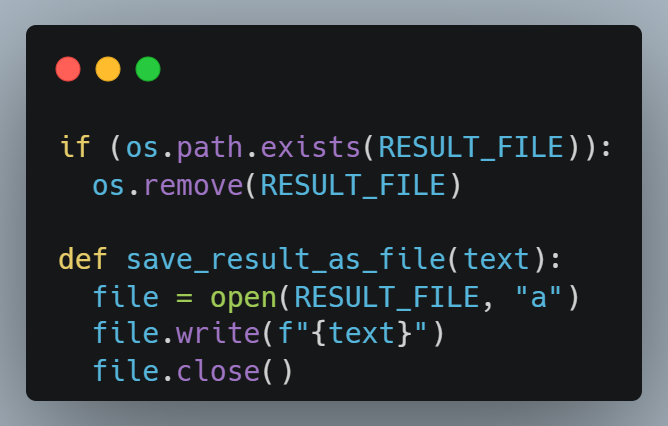
Наступний крок – записати їх до якогось місця зберігання. У даному випадку – excel файл:



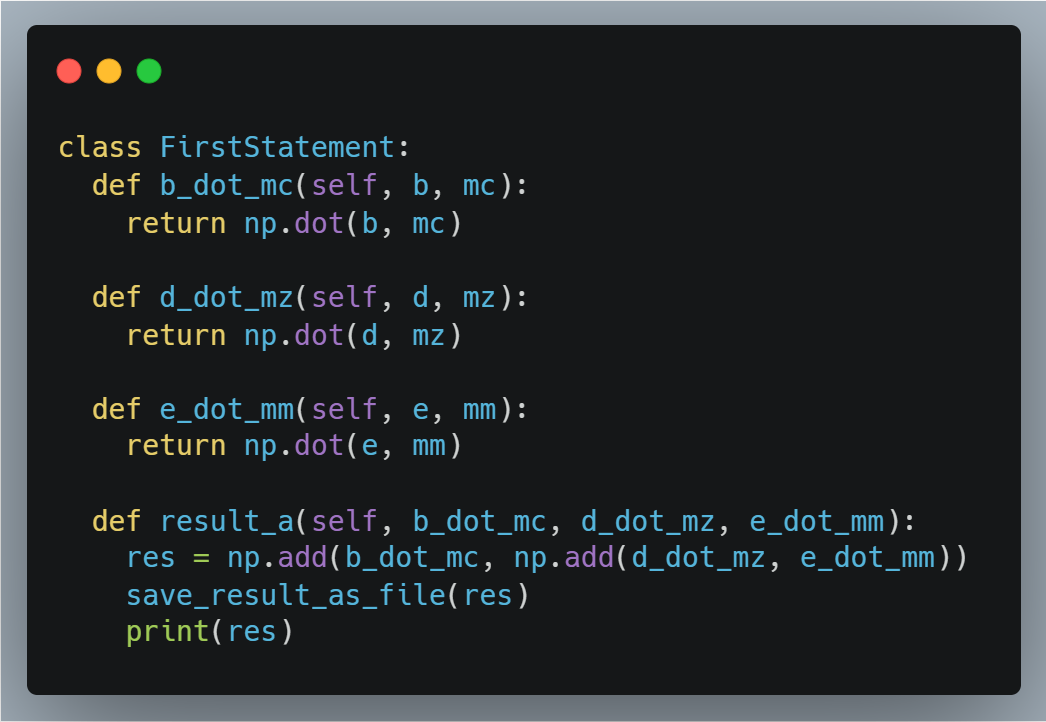
Вивід в Google colab:

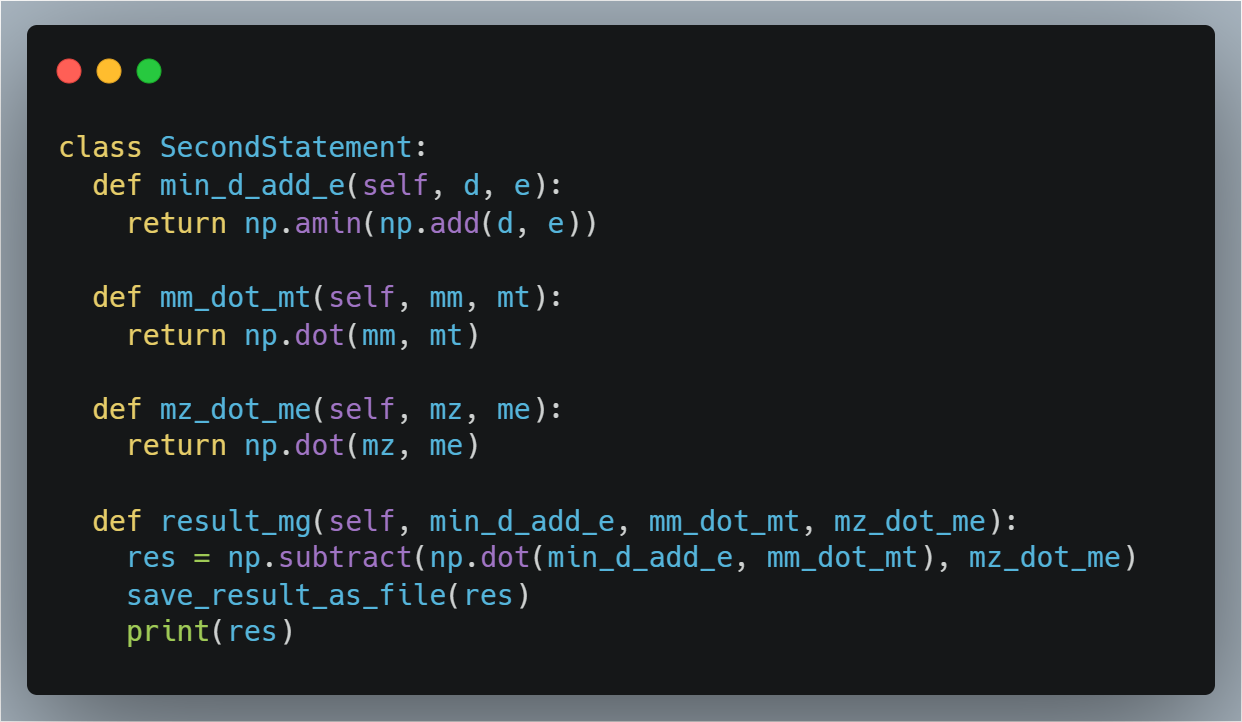


Але ж треба ще кудись виписувати результати. Для цього варіанту будемо використовувати звичайний .txt формат. Спочатку потрібно перевірити чи файл уже існує: якщо так, то його треба видалити:



Також потрібно створити функції для обрахування Наших виразів. Оформити їх Я вирішив у вигляді 2-ох класів для легшого тестування обчислень в різних потоках - у кожному з них є метод, котрий відповідає за власний функціонал: хтось множить 2 матриці, хтось – вектор на матрицю і т.д.:





Подальші класи будуть використовувати функціонал вище і записуватимуть результати до глобальних змінних, аби їх бачили всі потоки



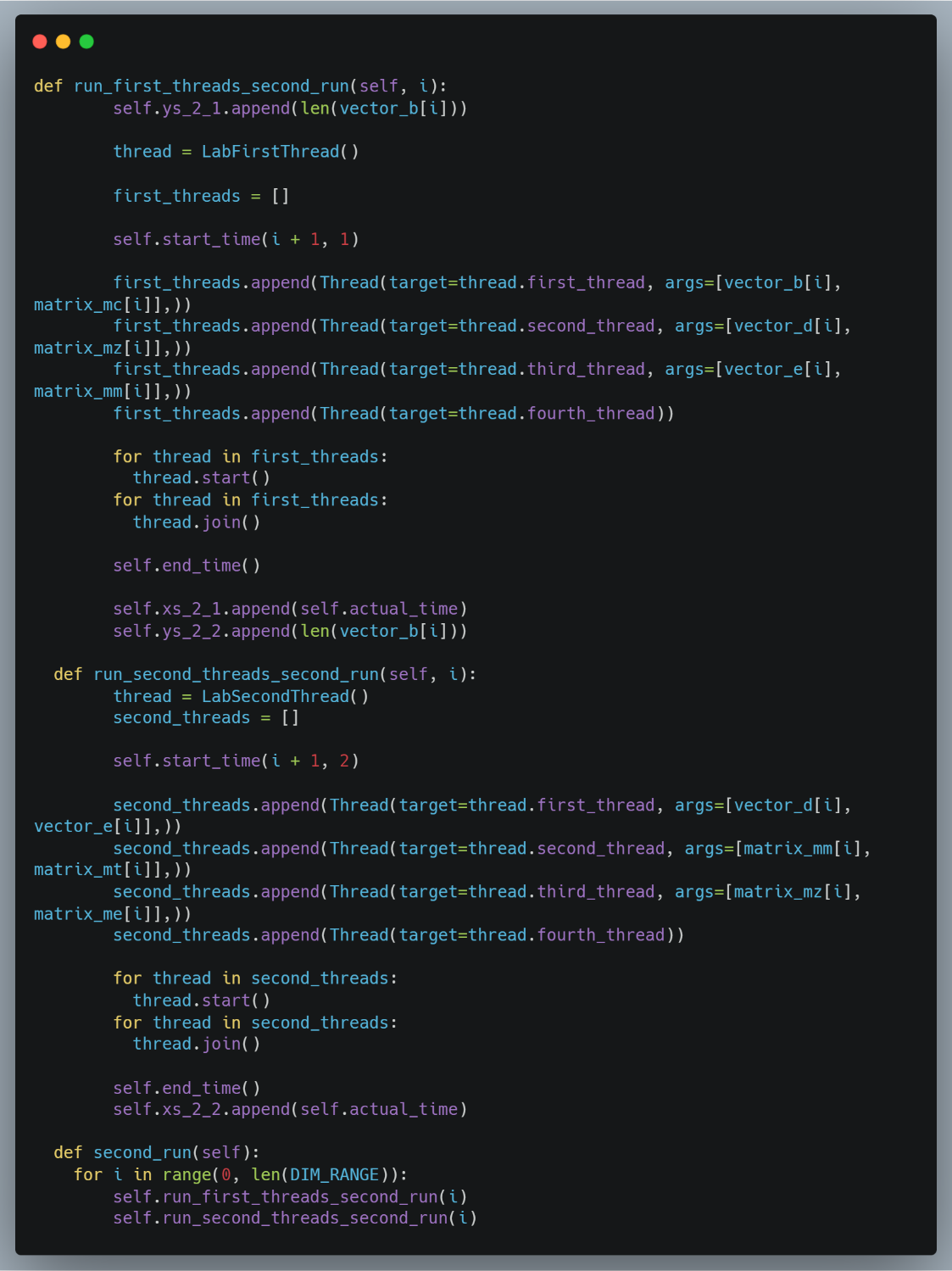


Тепер найголовніший та найбільший клас усього проекту – LabRuns.

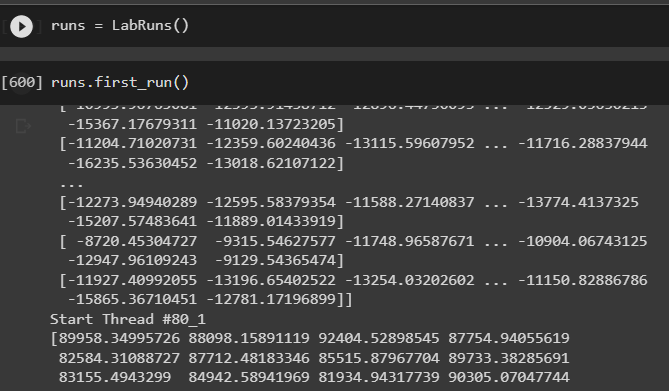
Його ціль - вираховування швидкості обчислень у потоках, який має такі елементи:

* Масиви точок для будування графіків
* Початок та кінець відліку часу та виведення до консолі часу виконання
* Методи first\_run та second\_run, котрі запускають обчислення виразів та часу на їх обробку
* run\_first\_thread\_first\_run та run\_second\_thread\_first\_run – для вираховування виразів в одному потоці
* run\_first\_threads\_first\_run та run\_second\_threads\_first\_run, котрі компонують кілька потоків з різними частинами рівнянь



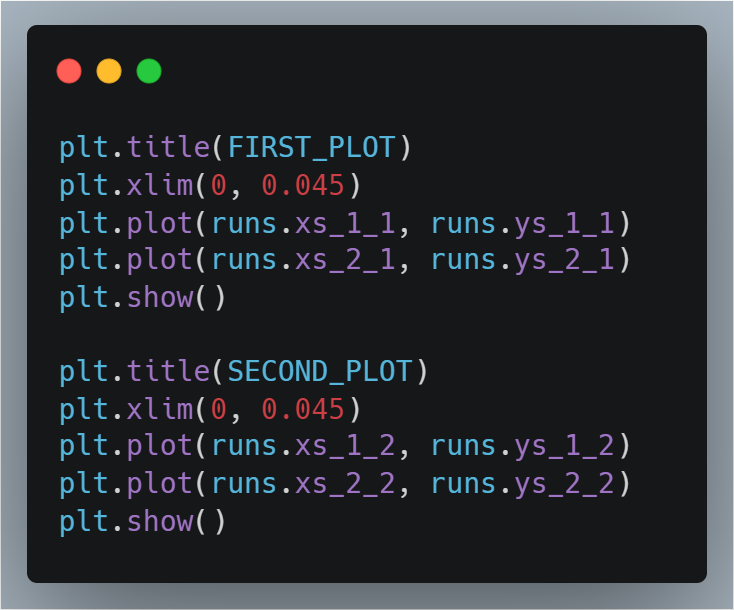


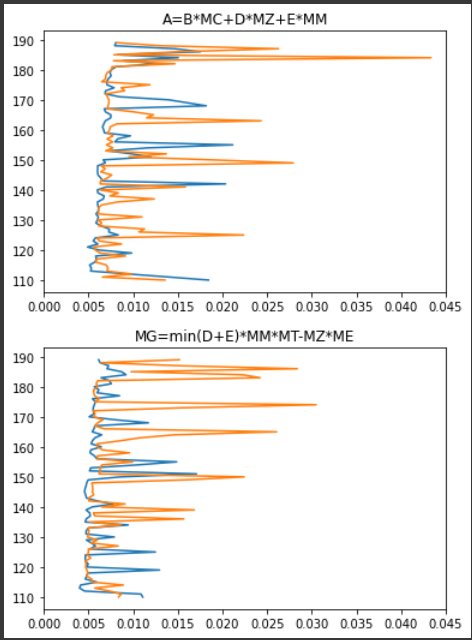
При запуску обох методів first\_run та second\_run Нам виводить ось таку інформацію:



Як видно, то Ми маємо результати виразів, та позначки для розуміння, коли починається новий потік

Ну і в самому кінці Нам треба відобразити показники Наших потоків:





# Висновок:

На жаль пророблена робота не дала бажаних результатів, а саме пришвидшення обрахування всіх виразів, а навпаки – зробила тільки гірше. Можливо було не зовсім коректно проведено синхронізацію; можливо це пов’язано з несправедливим розподілом задач (деякі потоки роблять більш складні операції, ніж інші) - точно Я сказати в даному випадку не можу.

У ході виконання роботи Я створив 2 варіанти багатопотокової програми:

1. Перший – мав в собі принципи локалізації - при цьому як скалярні так і векторні змінні не повинні розділюватися між потоками – механізм когерентності кешів MESIF при цьому працювати не буде
2. У другий ж навпаки – було використано глобальні змінні для скалярів та матриць за допомогою механізмів синхронізації - пошук актуального значення змінної при цьому буде відбуватися, відповідно до механізму когерентності кешів MESIF, у кешах L1D всіх ядер мікропроцесора

Побудував кілька графіків, котрі порівнюють обидві варіації та зробив невтішні висновки щодо швидкодії багатопотокового методу.

Додатково створив запис значень в окремі файли – excel та txt

# Посилання:

1. Робота на Google Colab – [посилання](https://colab.research.google.com/drive/1rwjOYMOAeb3BiBhhC4iHz5a8dASqY9Gk?usp=sharing)
2. Робота на GitHub – [посилання](https://github.com/VsIG-official/Designing-Complex-Systems/tree/master/Lab1)