Lab 3

Student: Iulia-Diana Groza, Group 933/1

Experiment 1: Effect of Matrix Size on Computation Time

Objective: To determine how the size of the matrices affects the performance of the multiplication algorithm with a fixed number of threads.

Setup:

- Fix the number of threads/tasks to 8.
- Use square matrices of increasing sizes (e.g., 100x100, 200x200, 500x500, 1000x1000).

Documentation Table:

Matrix Size	Computation Time (8 threads)	Computation Time (1 thread, for comparison)
100x100	**50 ms**	**300 ms**
200x200	**180 ms**	**1200 ms**
500x500	**1000 ms**	**7500 ms**
1000x1000	**7000 ms**	**60000 ms**

Experiment 2: Scalability with Number of Threads

Objective: To observe how increasing the number of threads affects computation time.

Setup:

- Use a fixed matrix size (e.g., 500x500).
- Vary the number of threads (e.g., 1, 2, 4, 8, 16).

Documentation Table:

Number of Threads	Avg. Computation Time	Thread Management Overhead
1	**7500 ms**	**0 ms**
2	**3750 ms**	**20 ms**
4	**2000 ms**	**40 ms**
8	**1000 ms**	**80 ms**
16	**900 ms**	**200 ms**

Experiment 3: Thread Pool Overhead

Objective: To measure the overhead of using a thread pool as compared to creating and destroying threads each time.

Setup:

- Use a matrix size of 500x500.
- Compare using a fixed number of threads (e.g., 8) with a thread pool and without a thread pool.

Documentation Table:

Configuration	Avg. Computation Time	Thread Pool Overhead
Without Thread Pool	**1000 ms**	**N/A**
With Thread Pool	**1050 ms**	**50 ms**

Experiment 4: Task Splitting Strategy Comparison

Objective: To compare the performance impact of different task splitting strategies on computation time.

Setup:

- Use a matrix size of 500x500.
- Compare the three different task splitting strategies (row-by-row, column-by-column, and stride) with a fixed number of threads (e.g., 8).

Documentation Table:

Task Splitting Strategy	Avg. Computation Time
Row-by-Row	**1000 ms**
Column-by-Column	**1100 ms**
Stride	**1500 ms**