**UCID: 2024510001 BATCH: A NAME: Atharva Vasant Angre**

**EXPERIMENT NO: 02**

**EXPERIMENT TITLE:** To implement Matrix multiplication and its variant.

2.1 To Implement Matrix Multiplication

2.2 To Implement Strassen's Matrix Multiplication

2.3 To analyze Time and space complexity of the above with the help of the tool.

Compare the time and space complexity and state what is the difference

between the two methods

**Objective:**

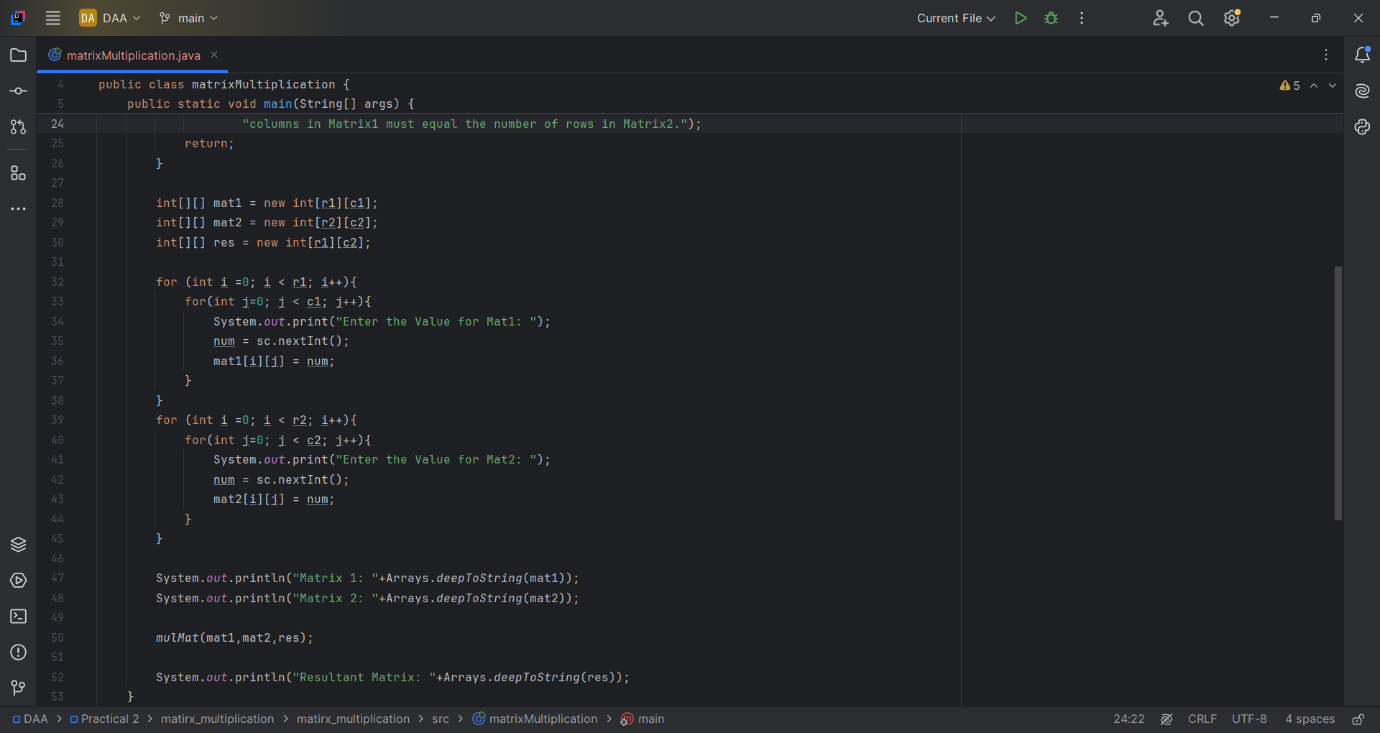
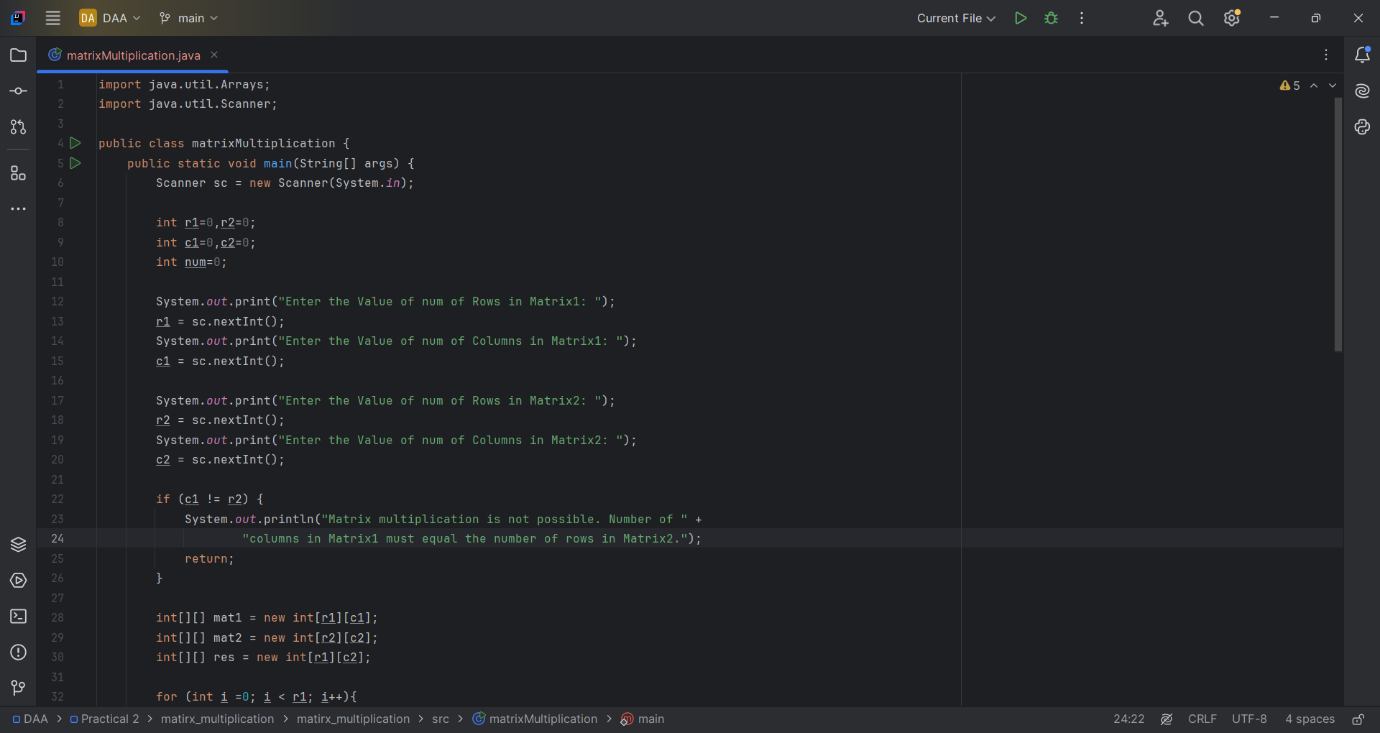
1.To implement matrix multiplication methods to understand the faster

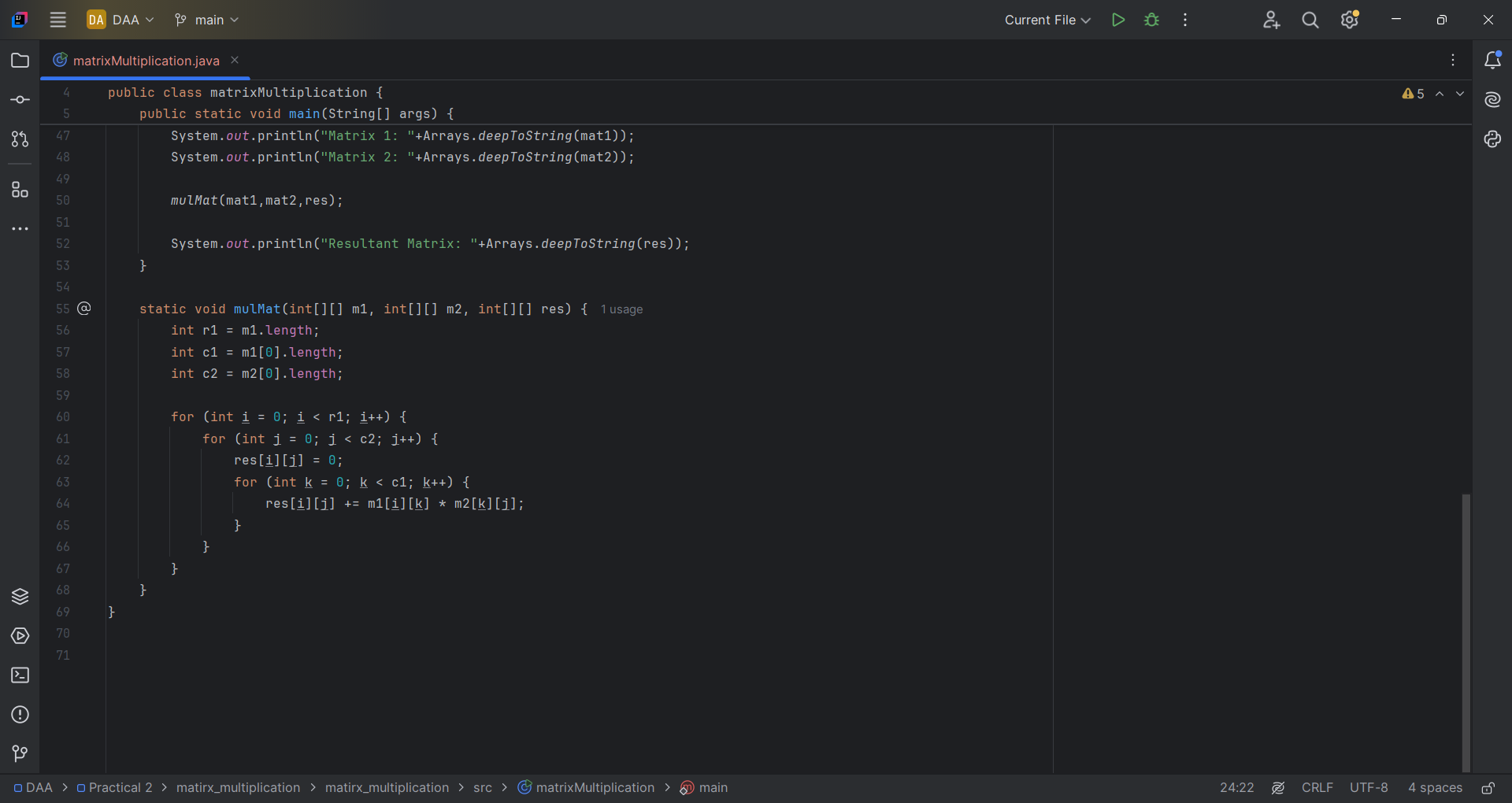
techniques.

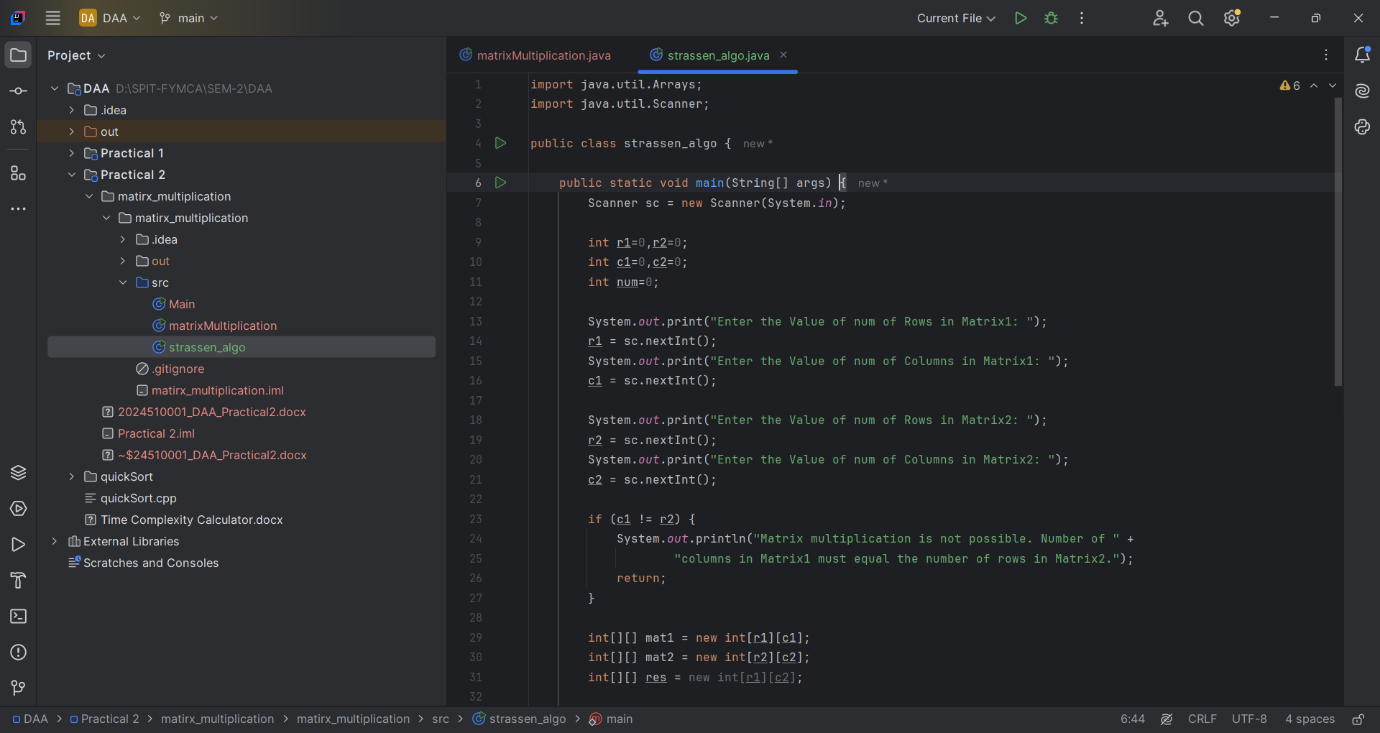
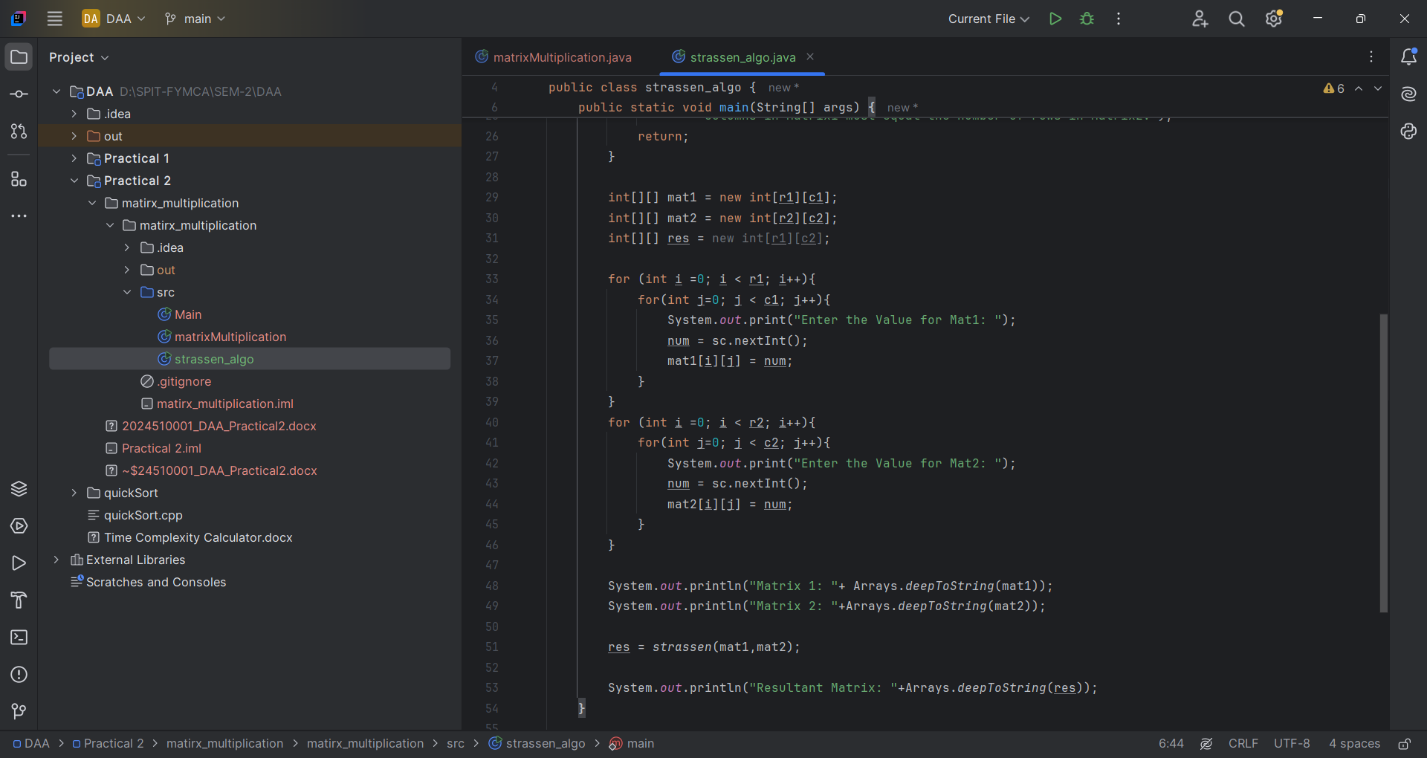
2. To compare which has a better time and space complexity.

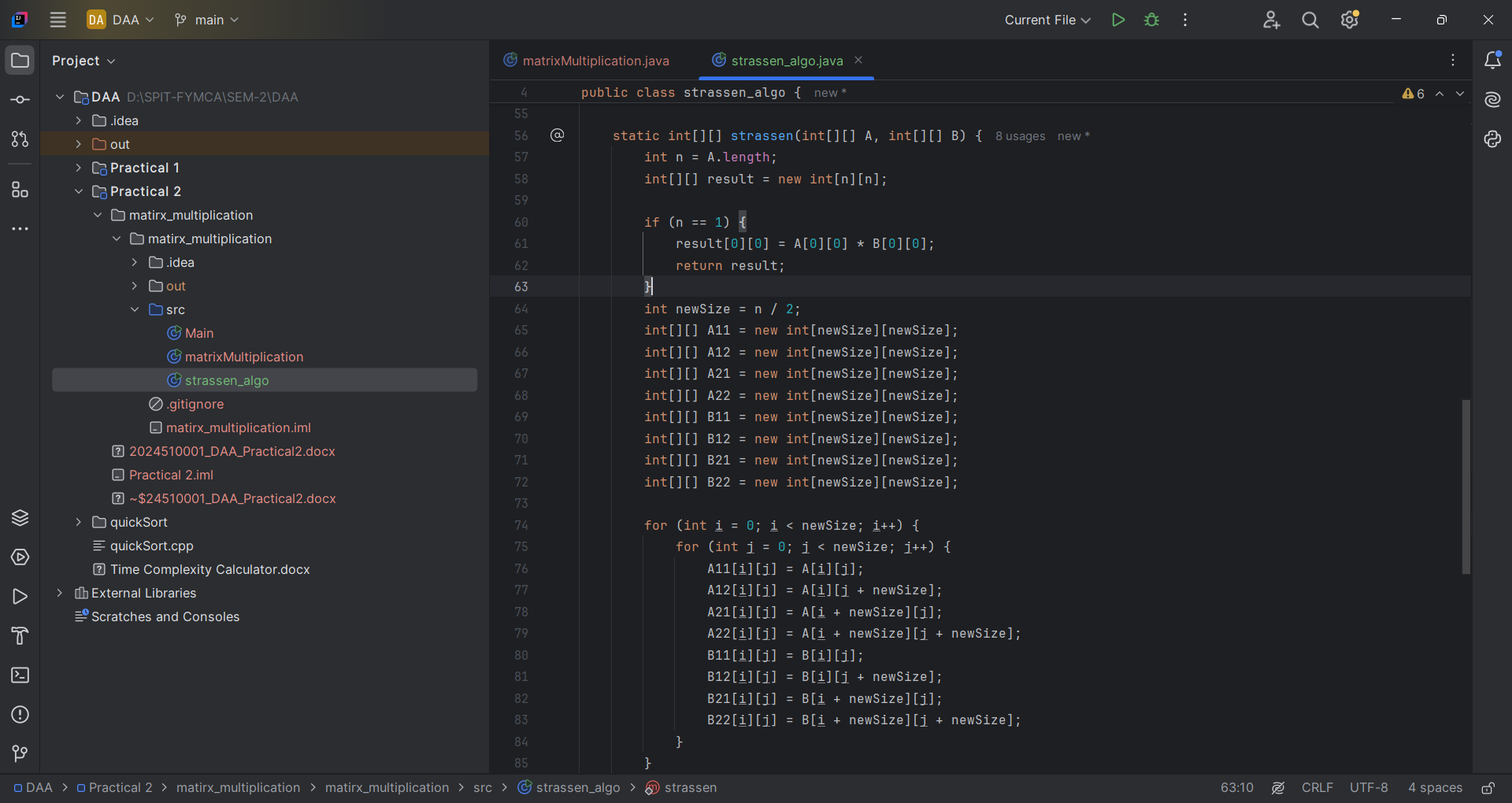
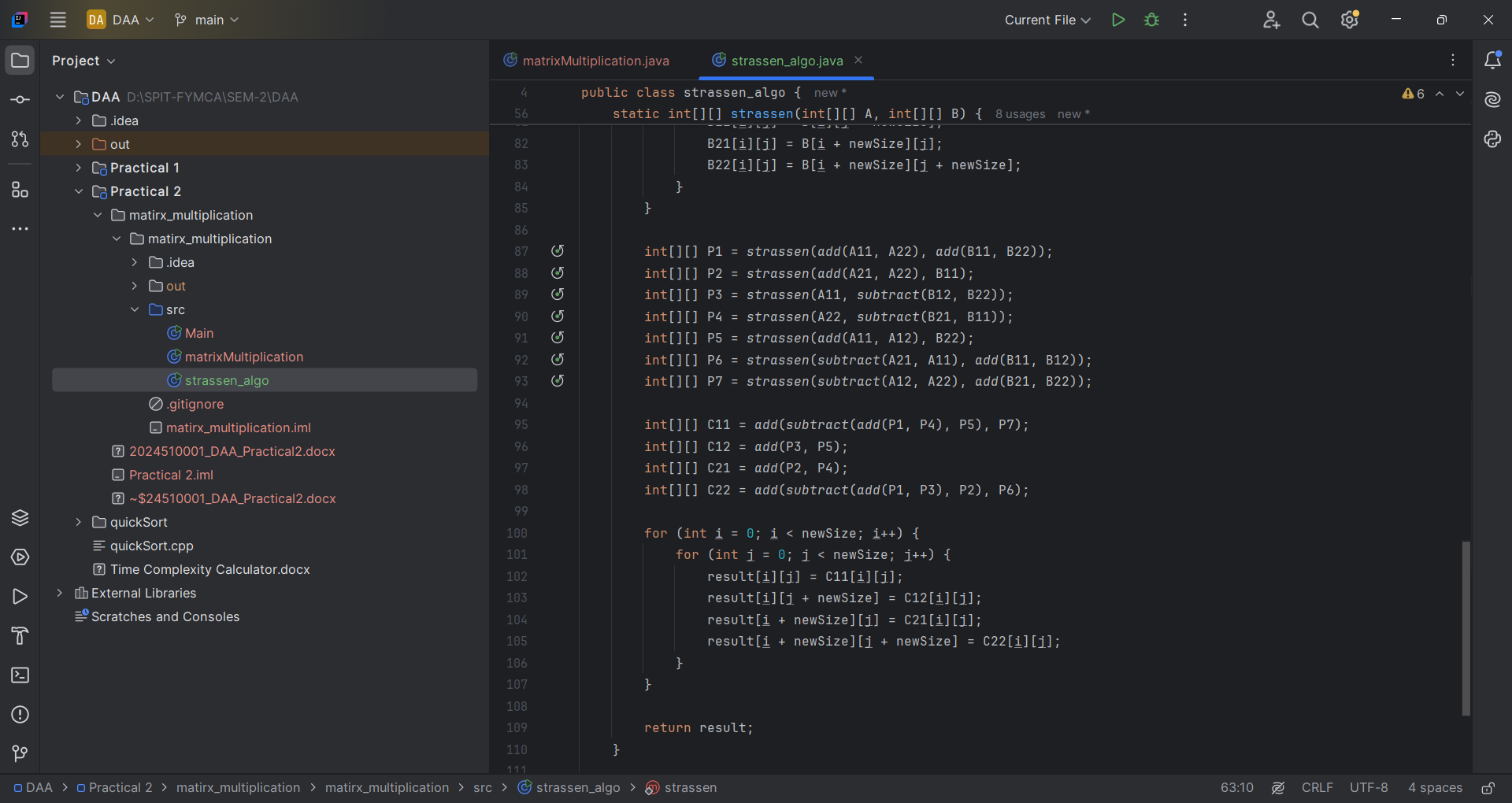
**Program code:** -

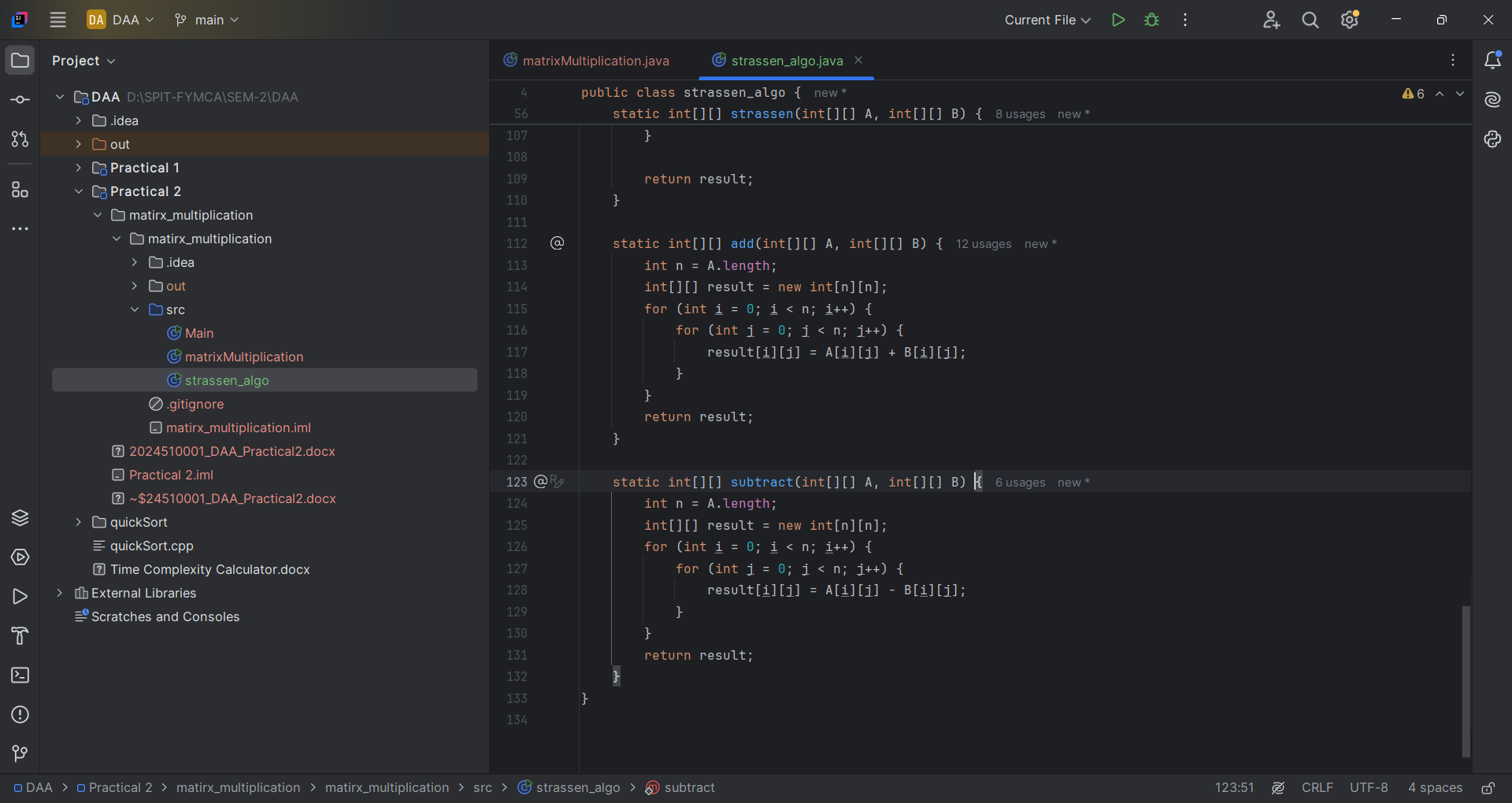
**Matrix Multiplication**

****

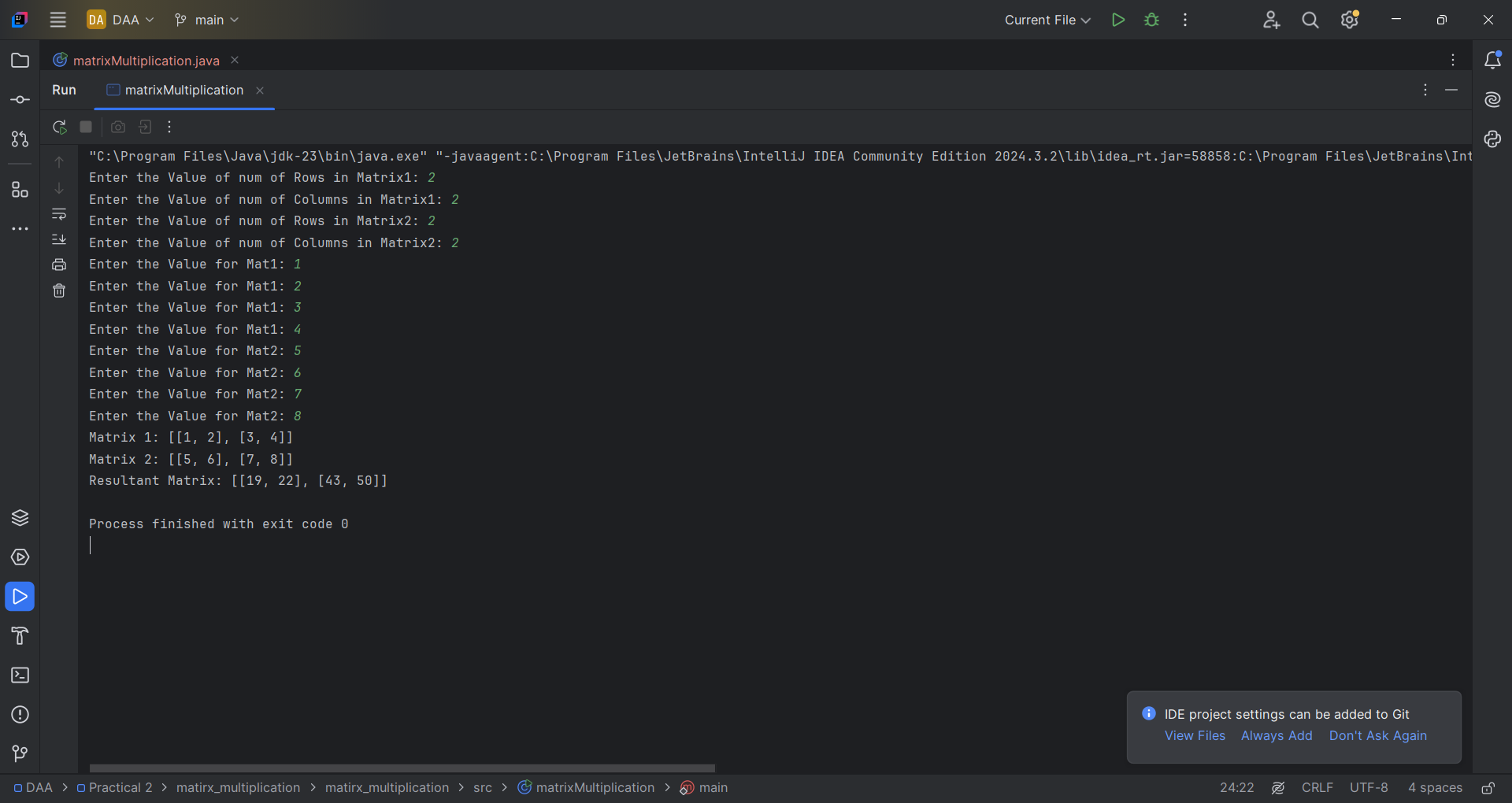
****

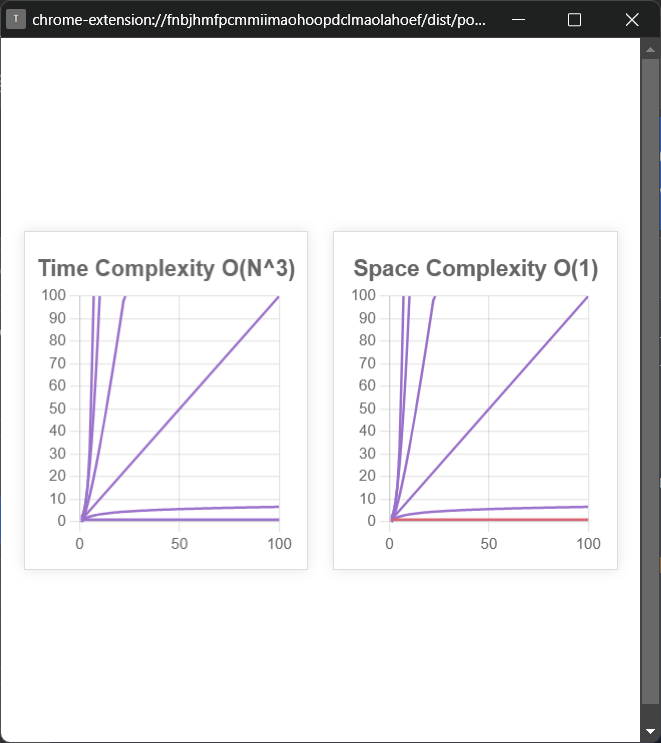
**Strassen’s Matrix Multiplication:**

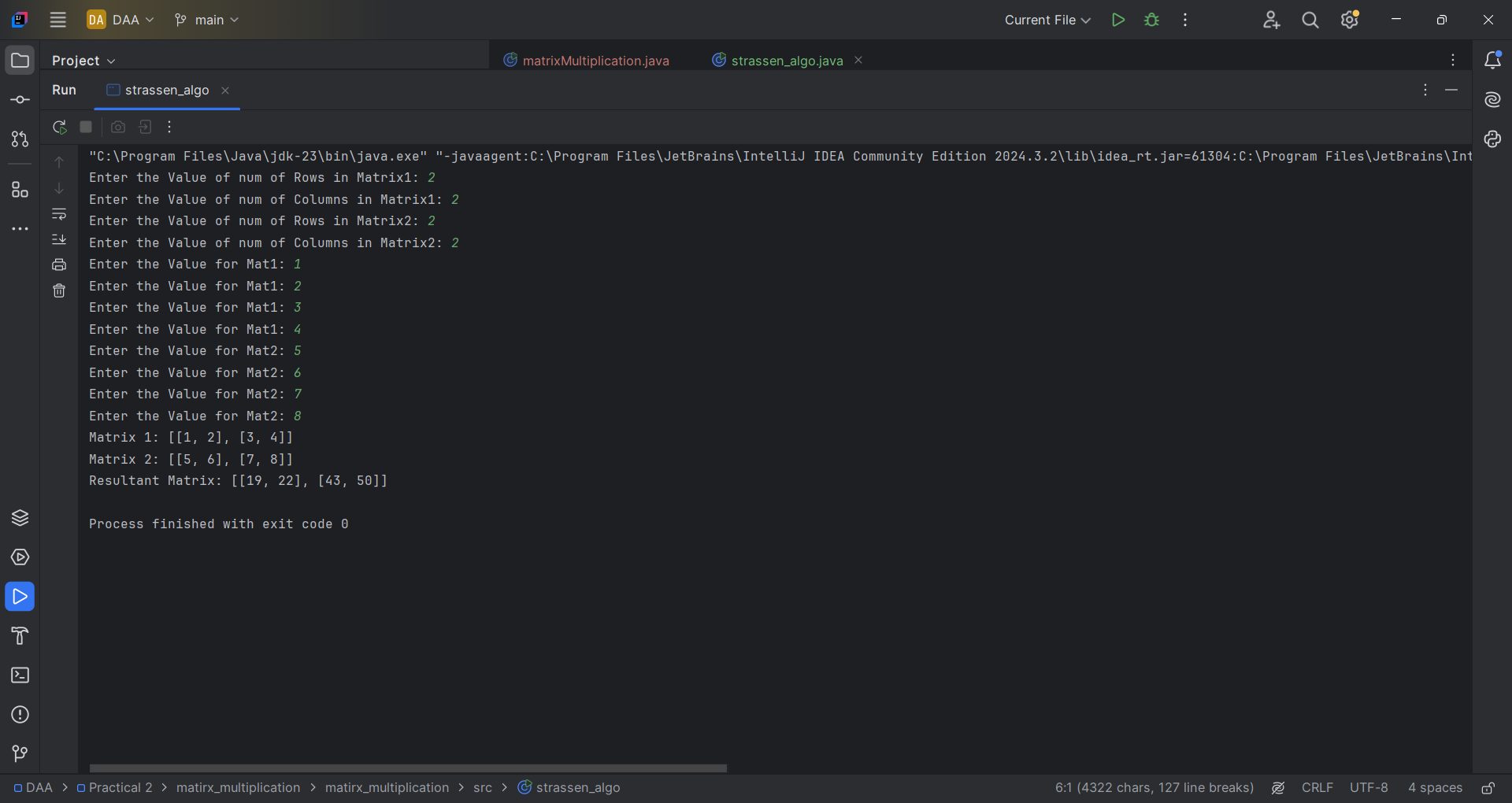
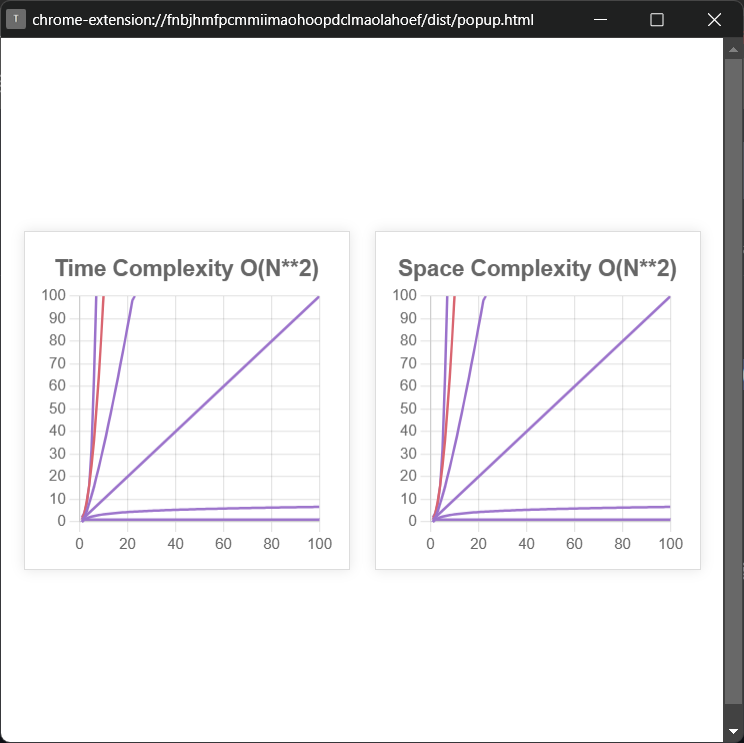
****

****

**Output:**

**Matrix Multiplication**

****

**Strassen’s Matrix Multiplication**

**Result:**

* Best Case: Strassen’s algorithm performs better for large matrices due to fewer multiplication operations, whereas the naive method remains at O(n^3).
* Worst Case: Even in the worst-case scenario, Strassen’s method outperforms the naive approach in terms of time complexity. However, it requires additional space.
* Average Case: Strassen’s method is generally more efficient for large matrices, but for small matrices, the naive approach may be preferable due to lower overhead.

**Conclusion:**

* Strassen’s algorithm provides an optimized approach for matrix multiplication compared to the naive method, especially for large matrices.
* However, due to recursion and additional storage requirements, it may not be suitable for very small matrices.
* Applications**:** This technique is widely used in image processing, machine learning, graphics transformations, scientific computing, and large-scale simulations.