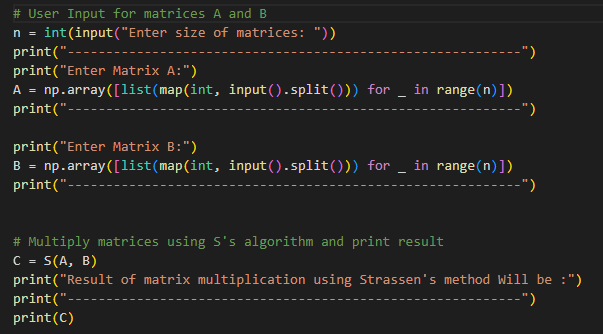
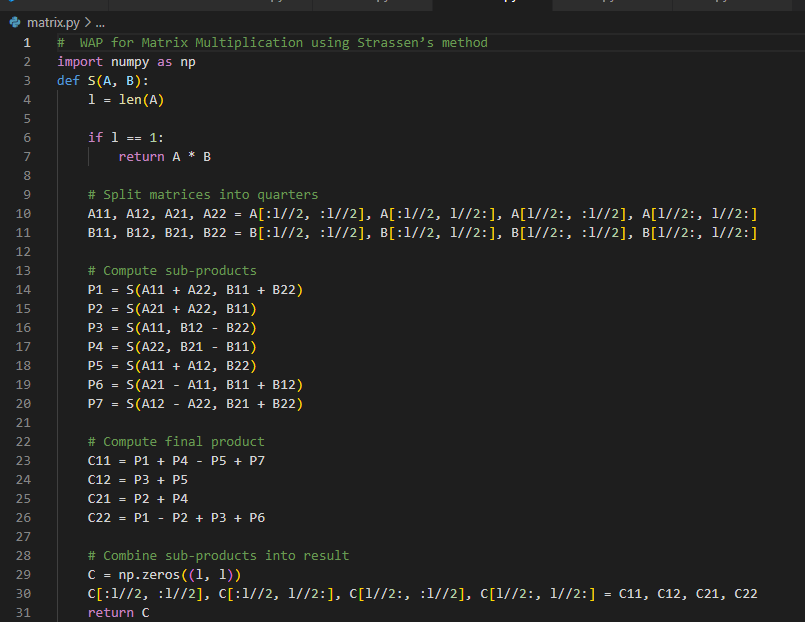
*Name: Aryan Sharma*

*Enrollment no. – BT21GCS161*

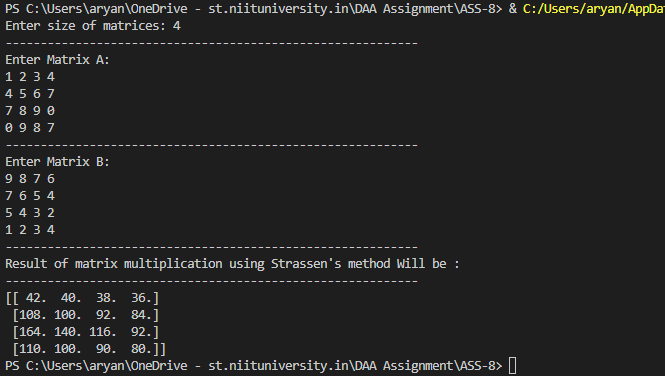
*Assignment no. – 08*

*Q. WAP for Matrix Multiplication using Strassen’s method.*

***Code*** *–*

**

***Output*** *–*



***Analysis-***

* *A well-liked algorithm for multiplying two matrices using the divide-and-conquer strategy is Strassen's algorithm. Recursively breaking up the matrices into smaller sub-matrices, working on them, and then combining them again to get the final output is how the method operates.*
* *Strassen's approach has a worst-case time complexity of O(n^log2(7)), where n is the size of the input matrix. This is due to the algorithm's recursive division of the input matrices into seven smaller submatrices each measuring n/2 by n/2, which it then arithmetically operates on before recombining to get the result. The algorithm's worst-case time complexity is of O(n^log2(7)) because the number of recursive calls is logarithmic to the size of the input matrix.*
* *Strassen's algorithm has an average-case time complexity of O(n^log2(7)), as it employs the same divide-and-conquer strategy for all input matrices. However, the constant factor associated with the technique may differ for different input matrices, resulting in somewhat different execution times.*
* *The best-case time complexity of Strassen's technique is also of O(n^log2(7)), because the programme employs the same strategy for all input matrices and there are no special circumstances or tweaks that can enhance the running time.*