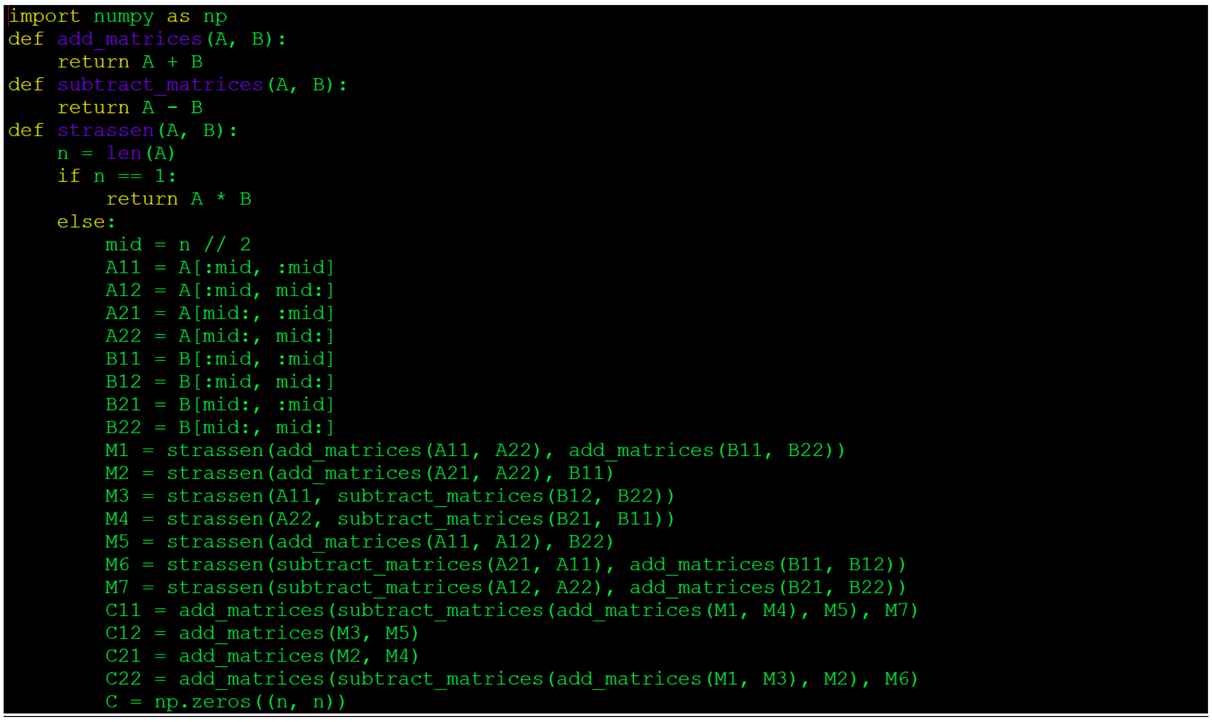
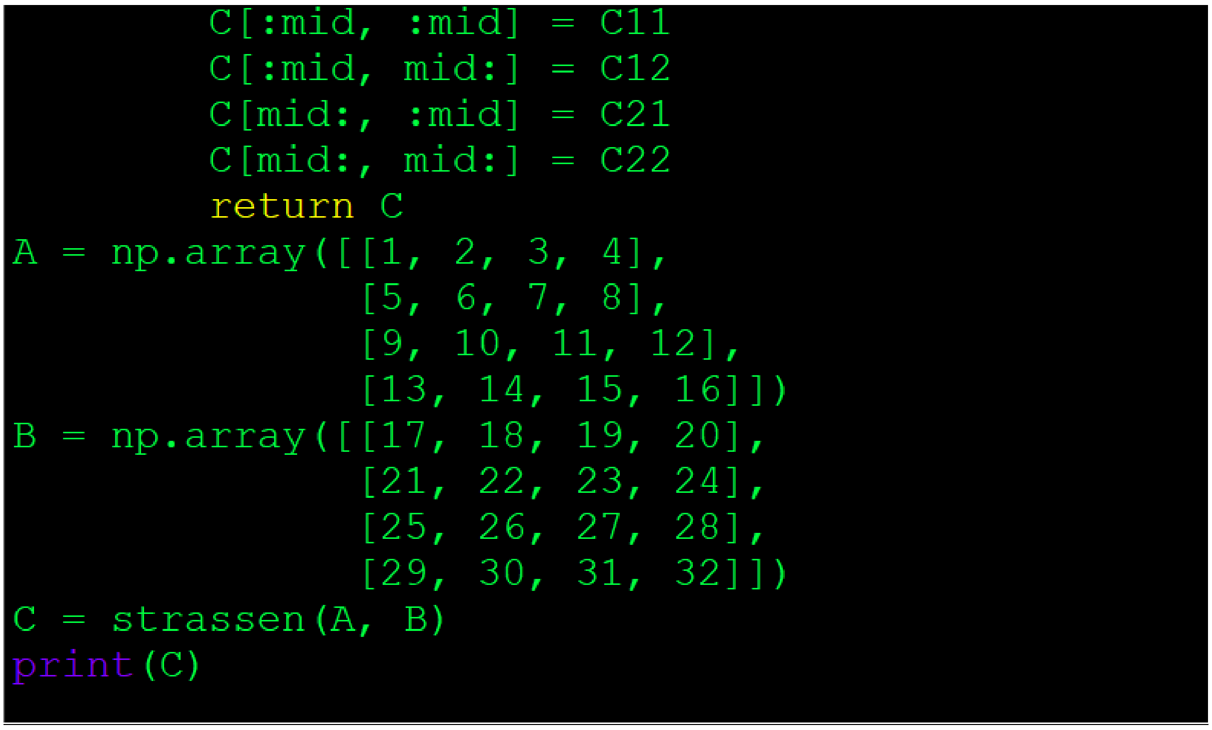
83. Strassens matrix multiplication

Code:

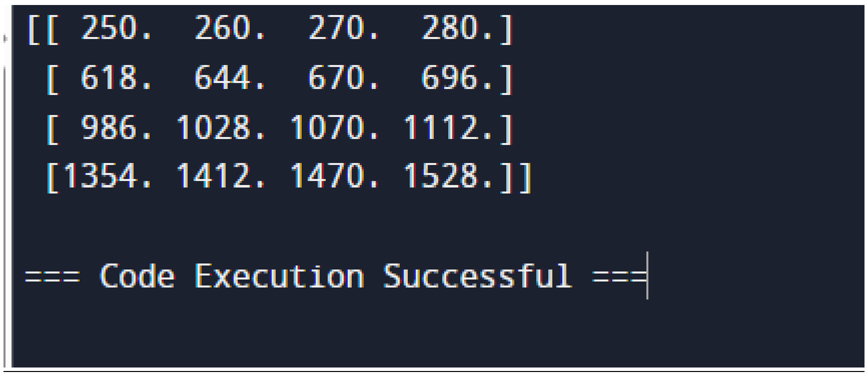


```
import numpy as np
def add_matrices(A, B):
    return A + B
def subtract_matrices(A, B):
    return A - B
def strassen(A, B):
    n = len(A)
    if n == 1:
        return A * B
    else:
        mid = n // 2
        A11 = A[:mid, :mid]
        A12 = A[:mid, mid:]
        A21 = A[mid:, :mid]
        A22 = A[mid:, mid:]
        B11 = B[:mid, :mid]
        B12 = B[:mid, mid:]
        B21 = B[mid:, :mid]
        B22 = B[mid:, mid:]
        M1 = strassen(add_matrices(A11, A22), add_matrices(B11, B22))
        M2 = strassen(add_matrices(A21, A22), B11)
        M3 = strassen(A11, subtract_matrices(B12, B22))
        M4 = strassen(A22, subtract_matrices(B21, B11))
        M5 = strassen(add_matrices(A11, A12), B22)
        M6 = strassen(subtract_matrices(A21, A11), add_matrices(B11, B12))
        M7 = strassen(subtract_matrices(A12, A22), add_matrices(B21, B22))
        C11 = add_matrices(subtract_matrices(add_matrices(M1, M4), M5), M7)
        C12 = add_matrices(M3, M5)
        C21 = add_matrices(M2, M4)
        C22 = add_matrices(subtract_matrices(add_matrices(M1, M3), M2), M6)
        C = np.zeros((n, n))
```



```
        C[:mid, :mid] = C11
        C[:mid, mid:] = C12
        C[mid:, :mid] = C21
        C[mid:, mid:] = C22
        return C
A = np.array([[1, 2, 3, 4],
              [5, 6, 7, 8],
              [9, 10, 11, 12],
              [13, 14, 15, 16]])
B = np.array([[17, 18, 19, 20],
              [21, 22, 23, 24],
              [25, 26, 27, 28],
              [29, 30, 31, 32]])
C = strassen(A, B)
print(C)
```

Output:



```
[[ 250.  260.  270.  280.]
 [ 618.  644.  670.  696.]
 [ 986. 1028. 1070. 1112.]
 [1354. 1412. 1470. 1528.]]

=== Code Execution Successful ===
```

Time Complexity:

- T(n)= O(m*n*p)