1. **Matrix multiplication using STARSSEN METHOD**

```
def add_matrix(A, B):
    return [[A[i][j] + B[i][j] for j in range(len(A[0]))] for i in range(len(A))]

def sub_matrix(A, B):
    return [[A[i][j] - B[i][j] for j in range(len(A[0]))] for i in range(len(A))]

def strassen(A, B):
    if len(A) == 1:
        return [[A[0][0] * B[0][0]]]


    mid = len(A) // 2
    A11 = [row[:mid] for row in A[:mid]]
    A12 = [row[mid:] for row in A[:mid]]
    A21 = [row[:mid] for row in A[mid:]]
    A22 = [row[mid:] for row in A[mid:]]

    B11 = [row[:mid] for row in B[:mid]]
    B12 = [row[mid:] for row in B[:mid]]
    B21 = [row[:mid] for row in B[mid:]]
    B22 = [row[mid:] for row in B[mid:]]


    P1 = strassen(add_matrix(A11, A22), add_matrix(B11, B22))
    P2 = strassen(add_matrix(A21, A22), B11)
    P3 = strassen(A11, sub_matrix(B12, B22))
    P4 = strassen(A22, sub_matrix(B21, B11))
    P5 = strassen(add_matrix(A11, A12), B22)
    P6 = strassen(sub_matrix(A21, A11), add_matrix(B11, B12))
    P7 = strassen(sub_matrix(A12, A22), add_matrix(B21, B22))


    C11 = add_matrix(sub_matrix(add_matrix(P1, P4), P5), P7)
    C12 = add_matrix(P3, P5)
    C21 = add_matrix(P2, P4)
    C22 = add_matrix(sub_matrix(add_matrix(P1, P3), P2), P6)


    C = []
    for i in range(mid):
        C.append(C11[i] + C12[i])
    for i in range(mid):
        C.append(C21[i] + C22[i])

    return C

A = [
```

```
    [1, 2, 3,4],
    [5, 6, 7, 8],
    [1,2,3,4],
    [5,6,7,8]
]

B = [
    [1,2,1,3],
    [1,4,1,5],
    [1,6,1,7],
    [1,8,1,9]
]

C = strassen(A, B)
for row in C:
    print(row)
```