1. **Strassen’s matrix multiplication**

**Code:**

#include <stdio.h>

#define N 2

void strassen(int A[][N], int B[][N], int C[][N]);

int main() {

int A[N][N], B[N][N], C[N][N];

printf("Enter elements of matrix A (2x2):\n");

for (int i = 0; i < N; i++) {

for (int j = 0; j < N; j++) {

scanf("%d", &A[i][j]);

}

}

printf("Enter elements of matrix B (2x2):\n");

for (int i = 0; i < N; i++) {

for (int j = 0; j < N; j++) {

scanf("%d", &B[i][j]);

}

}

strassen(A, B, C);

printf("Result Matrix:\n");

for (int i = 0; i < N; i++) {

for (int j = 0; j < N; j++) {

printf("%d ", C[i][j]);

}

printf("\n");

}

return 0;

}

void strassen(int A[][N], int B[][N], int C[][N]) {

int M1 = (A[0][0] + A[1][1]) \* (B[0][0] + B[1][1]);

int M2 = (A[1][0] + A[1][1]) \* B[0][0];

int M3 = A[0][0] \* (B[0][1] - B[1][1]);

int M4 = A[1][1] \* (B[1][0] - B[0][0]);

int M5 = (A[0][0] + A[0][1]) \* B[1][1];

int M6 = (A[1][0] - A[0][0]) \* (B[0][0] + B[0][1]);

int M7 = (A[0][1] - A[1][1]) \* (B[1][0] + B[1][1]);

C[0][0] = M1 + M4 - M5 + M7;

C[0][1] = M3 + M5;

C[1][0] = M2 + M4;

C[1][1] = M1 - M2 + M3 + M6;

}

**Output:**

Enter elements of matrix A (2x2):

5 6 7 8

Enter elements of matrix B (2x2):

1 2 3 4

Result Matrix:

23 34

31 46

--------------------------------

Process exited after 12.03 seconds with return value 0

Press any key to continue . . .

