

## STRASSEN'S METHOD OF MATRIX MULTIPLICATION

**Aim-** Write a C program to implement strassen's method of multiplication to multiply two 2x2 matrices

**Problem Statement** – Given two matrices A and B ,multiply both the matrices using Strassen's method and calculate the resultant matrix

**INPUT** - A=  $\begin{bmatrix} 19 & 17 \\ -13 & 15 \end{bmatrix}$                       B=  $\begin{bmatrix} -16 & 21 \\ 12 & -14 \end{bmatrix}$

**OUTPUT** – Display P,Q,R,S,T,U,V and also display the resultant matrix

### FORMULAS

$$P = [A_{00} + A_{11}] * [B_{00} + B_{01}]$$

$$Q = [A_{10} + A_{11}] * B_{00}$$

$$R = A_{00} * [B_{10} - B_{11}]$$

$$S = A_{11} * [B_{10} - B_{00}]$$

$$T = [A_{00} + A_{01}] * B_{11}$$

$$U = [A_{10} - A_{00}] * [B_{00} + B_{01}]$$

$$V = [A_{01} - A_{11}] * [B_{10} + B_{11}]$$

THE RESULTANT MATRIX IS GIVEN AS  $C_{2 \times 2} = \begin{bmatrix} C_{00} & C_{01} \\ C_{10} & C_{11} \end{bmatrix}$

WHERE  $C_{00} = P + S - T + V$

$$C_{01} = R + T$$

$$C_{10} = Q + S$$

$$C_{11} = P + R - Q + U$$

## PROGRAM –

```
#include <stdio.h>
```

```
#include <time.h>
```

```
void display_matrix(int matrix[2][2]) {
```

```
    for (int i = 0; i < 2; i++) {
```

```
        printf(" | ");
```

```
        for (int j = 0; j < 2; j++) {
```

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

```
        }
```

```
        printf(" | ");
```

```
        printf("\n");
```

```
    }
```

```
}
```

```
void strassen_multi(int a[2][2], int b[2][2], int c[2][2]) {
```

```
    clock_t start, end;
```

```
    double cpu_time_used;
```

```
    start = clock();
```

```
    int p, q, r, s, t, u, v;
```

```
    p = (a[0][0] + a[1][1]) * (b[0][0] + b[1][1]);
```

```
    q = (a[1][0] + a[1][1]) * b[0][0];
```

```
    r = a[0][0] * (b[0][1] - b[1][1]);
```

```
    s = a[1][1] * (b[1][0] - b[0][0]);
```

```
    t = (a[0][0] + a[0][1]) * b[1][1];
```

```
    u = (a[1][0] - a[0][0]) * (b[0][0] + b[0][1]);
```

```
    v = (a[0][1] - a[1][1]) * (b[1][0] + b[1][1]);
```

```
    printf ( " P = %d \n Q = %d \n R = %d \n S = %d \n T =  
%d \n U = %d \n V = %d \n", p, q, r, s, t, u, v);
```

```
    c[0][0] = p + s - t + v;
```

```
    c[0][1] = r + t;
```

```
    c[1][0] = q + s;
```

```
    c[1][1] = p - q + r + u;
```

```
    printf ( " The result of matrix multiplication is : \n");
```

```
    display_matrix(c);
```

```
    end = clock();
```

```
    cpu_time_used = ((double) (end - start)) /  
CLOCKS_PER_SEC;
```

```
    printf("Time taken by Strassen's matrix multiplication:  
%f seconds\n", cpu_time_used);
```

```
}
```

```
void input_matrix(int matrix[2][2]) {
```

```
    for (int i = 0; i < 2; i++) {
```

```
        for (int j = 0; j < 2; j++) {
```

```
            printf("Enter the element at position (%d, %d): ",  
i, j);
```

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

```
        }
```

```
    }
```

```
}
```

```
int main() {
```

```
    printf
```

```
("*****  
*****");
```

```
    printf ("\n Roll number: 23B-CO-010\n");
```

```
    printf (" PR Number - 202311390\n");
```

```
    printf("*****  
*****\n\n\n");
```

```
    int a[2][2], b[2][2], c[2][2];
```

```
    int choice;
```

```
    while (1) {
```

```
        printf("\nMenu:\n");
```

```
printf("1. Input matrices\n");  
  
printf("2. Multiply matrices using Strassen's  
algorithm\n");  
  
printf("3. Exit\n");  
  
printf("Enter your choice: ");  
  
scanf("%d", &choice);
```

```
switch (choice) {
```

```
case 1:
```

```
    printf("Input matrix A:\n");  
    input_matrix(a);  
    printf("Input matrix B:\n");  
    input_matrix(b);  
    break;
```

```
case 2:
```

```
    strassen_multi(a, b, c);  
    printf("Matrices multiplied successfully.\n");  
    break;
```

```
case 3:
```

```
    return 0;
```

```
default:
```

```
    printf("Invalid choice. Please try again.\n");
```

```
    }
```

```
}
```

```
return 0;
```

```
}
```

## INPUT –

```
*****
Roll number: 23B-CO-010
PR Number - 202311390
*****

Menu:
1. Input matrices
2. Multiply matrices using Strassen's algorithm
3. Exit
Enter your choice: 1
Input matrix A:
Enter the element at position (0, 0): 19
Enter the element at position (0, 1): 17
Enter the element at position (1, 0): -13
Enter the element at position (1, 1): 15
Input matrix B:
Enter the element at position (0, 0): -16
Enter the element at position (0, 1): 21
Enter the element at position (1, 0): 12
Enter the element at position (1, 1): -14

Menu:
1. Input matrices
2. Multiply matrices using Strassen's algorithm
3. Exit
Enter your choice: 2
```

## OUTPUT –

```
P = -1020
Q = -32
R = 665
S = 420
T = -504
U = -160
V = -4
The result of matrix multiplication is :
| -100  161 |
| 388  -483 |
Time taken by Strassen's matrix multiplication: 0.003000 seconds
Matrices multiplied successfully.
```

## TIME TAKEN –

Time taken by Strassen's matrix multiplication: 0.003000 seconds

**CONCLUSION –** Two matrices were successfully multiplied and it's result was correctly calculated using strassen's method of matrix multiplication .

