

Name: Harsh Chandra
UID: 2021700013
Experiment no. - 3

AIM: Experiment based on strassen's matrix multiplication.

Program:

```
#include <stdio.h>

int main()
{
    int a[2][2], b[2][2], ans[2][2], i, j;
    int p, q, r, s, t, u, v;
    printf("Enter the 4 elements of first matrix: ");
    for (i = 0; i < 2; i++)
        for (j = 0; j < 2; j++)
            scanf("%d", &a[i][j]);
    printf("Enter the 4 elements of second matrix: ");

    for (i = 0; i < 2; i++)
        for (j = 0; j < 2; j++)
            scanf("%d", &b[i][j]);

    printf("\nThe first matrix is\n");
    for (i = 0; i < 2; i++)
    {
        printf("\n");
        for (j = 0; j < 2; j++)
            printf("%d\t", a[i][j]);
    }

    printf("\nThe second matrix is\n");

    for (i = 0; i < 2; i++)
    {
        printf("\n");
        for (j = 0; j < 2; j++)
```

```

        printf("%d\t", b[i][j]);
    }

    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]);

    ans[0][0] = p + s - t + v;
    ans[0][1] = r + t;
    ans[1][0] = q + s;
    ans[1][1] = p - q + r + u;

    printf("\nAfter multiplication using Strassen's algorithm \n");
    for (i = 0; i < 2; i++)
    {
        printf("\n");
        for (j = 0; j < 2; j++)

            printf("%d\t", ans[i][j]);
    }
    printf("\n");
    return 0;
}

```

Output:

```
students@CE-Lab7-603-U10:~$ cd Desktop
students@CE-Lab7-603-U10:~/Desktop$ gcc exp03.c
students@CE-Lab7-603-U10:~/Desktop$ ./a.out
Enter the 4 elements of first matrix: 1
2
3
4
Enter the 4 elements of second matrix: 5
6
7
8

The first matrix is
1      2
3      4
The second matrix is
5      6
7      8
After multiplication using Strassen's algorithm
19     22
43     50
```

Conclusion: The idea of **Strassen's method** is to reduce the number of recursive calls to 7. Strassen's method is similar to above simple divide and conquer method in the sense that this method also divide matrices to sub-matrices of size $N/2 \times N/2$.