**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 matix

**INPUT -**  A= | 19 17 | B= | -16 21 |

| -13 15 | | 12 -14 |

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

**FORMULAS**

**P = [ A00+ A11] \* [ B00 + B01 ]**

**Q = [A10 + A11 ] \* B 00**

**R = A00 \* [ B10 – B11]**

**S = A11 \* [B10 – B00 ]**

**T = [A00 + A01] \* B11**

**U = [A10 – A00] \* [ B00 + B01]**

**V = [ A01 -A11 ] \* [ B10 + B11]**

**THE RESULTANT MATRIX IS GIVEN AS C 2x2 = | C 00 C01 |**

**| C10 C11 |**

**WHERE C00 = P+S-T+V**

**C01 = R + T**

**C10  = Q + S**

**C11 = 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");

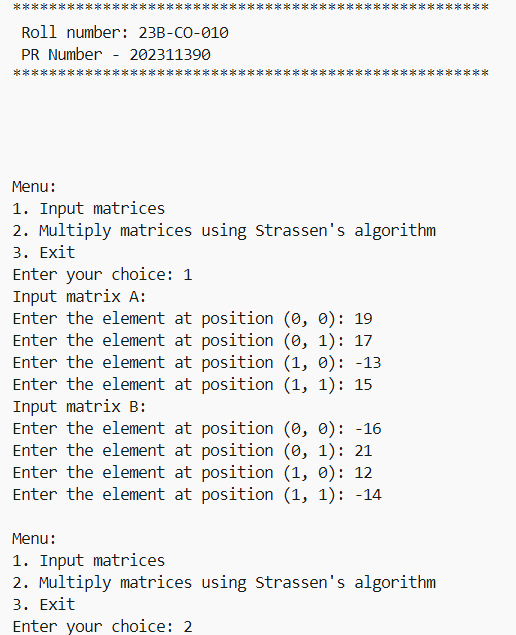
        }

    }

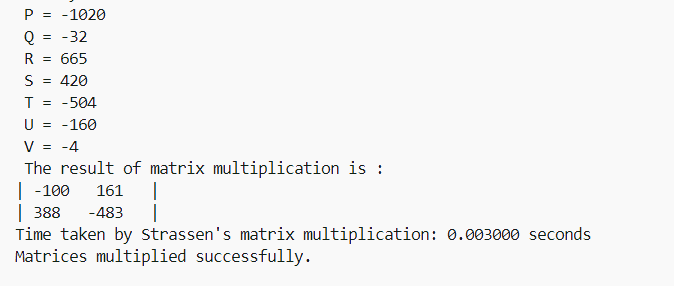
    return 0;

}

**INPUT –**

****

**OUTPUT –**

****

**TIME TAKEN –**

****

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