Date 25/01/24 Expt. No. 03

Expt. Name Page No. 6.

AIM: Implement Strassen Matrix Multiplication using divide & conquer approach.

Algorithm:

→ Matrix Multiplication is based on a divide and conquer-based approach, Here we divide our Matrix and Merge it into larger result for larger Matrix and Merge it into larger result for larger for larger Matrices this approach will continue until we recurse all the Sub Matrices.

01. Start

02. Declare integer arrays a[2][2], b[2][2] and c[2][2] to rep the two matrix and store the result in the other matrix.

03. Declare int variables i & j for loop cycle i.e., for "row & column".

04. Declare int var from M_1 & M_7 to store intermediate multiplication result.

05. Use nested 'For' loop to read the input from the user for both the matrix i.e., a[2][2], b[2][2]

06. Again, use Nested For loop to print the Elements at both the matrices i.e., a[2][2], b[2][2]

07. Calculate the seven intermediate products (M_1 through M7) based in the Strassen matrix multiplication algo using the given formula which is written in the code.

08. Use the intermediate products to calculate the elements of the result matrix ('c').

Teacher's Signature:

09. Now, Display the result which is stored in the matrix c[2][2].

10. End.

Source Code :

```
#include <stdio.h>
  int main()
 {
  int a[2][2], b[2][2], c[2][2], i, j ;
  int m1, m2, m3, m4, m5, m6, m7 ;
  printf ("Enter the 4 Elements of the first
        Matrix : \n");
   {
   for (i=0 ; i<2 ; i++)
     {
      for (j=0 ; j<2 ; j++)
       {
        scanf ("%d", & a[i][j]);
       }
     }
  }
 printf ("Enter the 4 Elements of the Second
        Matrix : \n");
 {
 for (i=0 ; j<2 ; i++)
  {
  for (j=0 ; j<2 ; j++) {
    scanf ("%d", & b[i][j]);
  } }
  printf ("the Elements of 1st Matrix are : \n");
   for (i=0 ; i<2 ; i++) {
```

Output:

```
Enter the 4 Element at the first Matrix : 1 2 3 4
Enter. the 4 Element at the Second Matrix : 4 3 2 1

the Element of 1st Matrix are : 1  2
                                3  4

the Element of 2nd Matrix are   4  3
                                2  1

the Strassen Matrix Multiplication are [ 8   5 ]
                                       [ 20  13 ]
```

Date   Expt. No.

Expt. Name   Page No. 8.

```
    for (j=0 ;j<2 ;j++) {
      printf ("%d", a[i][j]); }
   }}
  printf ("the elements of the 2nd Matrix are : \n");
   }
   for (i=0 ; i<2 ; i++) {
     for (j=0;j<2 ;j++)
       {
        printf ("%d", b[i][j]); }
      }
 }
m1 = (a[0][0] + a[1][1]) * (b[0][0] + b[1][1]) ; =
m2 = (a[1][0] + a[1][1]) * b[0][0]);
m3 = a[0][0] * b[0][1] - b[1][1]);
m4 = (a[1][1] * b[1][0] - b[0][0] );
m5 = (a[0][0] + a[0][1] ) * b[1][1] ;
m6 = a[1][1] -a[0][0]) * (b[0][0] + b[0][1]);
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;

printf ("the Strassen Matrix Multiplication are: ");
 {
   for (i=0 ; i<2 ; i++) {
      for(j=0 ;j<2; j++) {
         printf( "%d\n "c[i][j]);
      }
    }
  }
```

Teacher's Signature: