Name	Kinnari Shah
UID no.	2021700058
Experiment No.	3

<table>
<tr><td>AIM:</td><td>Divide and Conquer -Strassen's Matrix Multiplication</td></tr>
<tr><td colspan="2" align="center">Program 1</td></tr>
<tr><td>PROBLEM STATEMENT :</td><td>Use Strassen’s Matrix Multiplication to multiply two matrices.</td></tr>
<tr><td>ALGORITHM:</td><td>Here is the procedure :
1. Divide a matrix of the order of 2*2 recursively until we get the matrix of order 2*2.
2. To carry out the multiplication of the 2*2 matrix, use Strassen’s Multiplication formulas i.e.

a) D1 = (a11 + a22) * (b11 + b22)
b) D2 = (a21 + a22)*b11
c) D3 = (b12 – b22)*a11
d) D4 = (b21 – b11)*a22
e) D5 = (a11 + a12)*b22
f) D6 = (a21 – a11) * (b11 + b12)
g) D7 = (a12 – a22) * (b21 + b22)

C00= d1 + d4 – d5 + d7
C01 = d3 + d5
C10 = d2 + d4
C11 = d1 + d3 – d2 – d6
Here, C00, C01, C10, and C11 are the elements of the 2*2 matrix

3. Subtraction is also performed within these seven multiplications and four additions.</td></tr>
</table>

4. To find the final product or final matrix combine the result of two matrixes.

PROGRAM:

```
#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;
        // Here we are scanning and printing the first matrix
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]);
          // Here we are scanning and printing the second matrix
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]);
}
        // Here we are applying the above mentioned formulae
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];
```

<table>
<tr><td></td><td>

```
m6= (a[1][0]-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;
        // As we got the value of the elements, we now print them
printf("\n After performing multiplication \n");
for(i=0;i<2;i++){
printf("\n");
for(j=0;j<2;j++)
printf("%d\t",c[i][j]);
}

return 0;
}
```

</td></tr>
<tr><td>OBSERVATION:</td><td>In normal matrix multiplication method, the main component for high time complexity is 8 recursive calls. The idea of Strassen’s method is to reduce the number of recursive calls to 7. Strassen’s method divides matrices to sub-matrices of size N/2 x N/2 and the four sub-matrices of result are calculated using above formulae.

The time complexity for Strassen’s Multiplication is:
$T(N) = 7T(N/2) + O(N^2)$ on solving
$O(N^{Log7})$ which is approximately $O(N^{2.8074})$

While that using normal method is:
$T(N) = 8T(N/2) + O(N^2)$ on solving $O(N^3)$</td></tr>
<tr><td colspan="2">RESULT:</td></tr>
</table>

```
Enter the 4 elements of first matrix: 2
4
3
8
Enter the 4 elements of second matrix: 1
6
9
5

The first matrix is

2       4
3       8
The second matrix is

1       6
9       5
 After performing multiplication

38      32
75      58

...Program finished with exit code 0
Press ENTER to exit console.
```

CONCLUSION:	In this experiment, we implemented Strassen’s Multiplication.