Experiment No.	3
Aim	Strassen Matrix Multiplication
Name	Sakshi D Lonare
UID No.	2021300069
Class & Division	Comps SE A

Algorithm:

Normal Matrix-Multiplication (X, Y, Z)
for i = 1 to p do
  for j = 1 to r do
    Z[i,j] := 0
    for k = 1 to q do
      Z[i,j] := Z[i,j] + X[i,k] × Y[k,j]

Using strassen's technique:-
M1:=(A+C)×(E+F)
M2:=(B+D)×(G+H)
M3:=(A−D)×(E+H)
M4:=A×(F−H)
M5:=(C+D)×(E)
M6:=(A+B)×(H)
M7:=D×(G−E)

For matrices:-
Z=[IKJL]
 X=[ACBD]
 and Y=[EGFH]

Program:-
Taking random input:-

#include <stdio.h>

```c
#include <stdlib.h>

int main()
{
 int n;
 printf("Enter the required dimension:\n");
 scanf("%d",&n);

 int A[n][n],B[n][n],C[n][n];
 int s1,s2,s3,s4,s5,s6,s7,s8,s9,s10;
 int p1,p2,p3,p4,p5,p6,p7;


 for(int i=1;i<n;i++)
 {
   for(int j=1;j<n;j++)
   {
       A[i][j]=rand()%20;
   }

 }


 for(int i=1;i<n;i++)
 {
   for(int j=1;j<n;j++)
   {
      B[i][j]=rand()%20;
   }
 }

 printf("\n\nMatrix A :\n");
 for(int i=1;i<n;i++)
 {
   printf("\n");
   for(int j=1;j<n;j++)
   {
   printf("%d\t",A[i][j]);
   }
 }
```

```c
printf("\n\nMatrix B:\n");
for(int i=1;i<n;i++)
{
  printf("\n");
  for(int j=1;j<n;j++)
  {
  printf("%d\t",B[i][j]);
  }
}

s1=B[1][2]-B[2][2];
s2=A[1][1]+A[1][2];
s3=A[2][1]+A[2][2];
s4=B[2][1]-B[1][1];
s5=A[1][1]+A[2][2];
s6=B[1][1]+B[2][2];
s7=A[1][2]-A[2][2];
s8=B[2][1]+B[2][2];
s9=A[1][1]-A[2][1];
s10=B[1][1]+B[1][2];

p1=A[1][1]*s1;
p2=s2*B[2][2];
p3=s3*B[1][1];
p4=s4*A[2][2];
p5=s5*s6;
p6=s7*s8;
p7=s9*s10;

C[1][1]=p5+p4-p2+p6;
C[1][2]=p1+p2;
C[2][1]=p5+p1+p2;
C[2][2]=p5+p1-p3-p7;

printf("\n\nStrassen Multiplication(Matrix C):\n");
for(int i=1;i<n;i++)
{
  printf("\n");
  for(int j=1;j<n;j++)
```

```
        {
            printf("%d\t",C[i][j]);
        }
    }
   printf("\n\n");

   return 0;
}
```

2] Taking input from user:-
```
#include <stdio.h>

int main()
{
 int n;
 printf("Enter the required dimension:\n");
 scanf("%d",&n);

 int A[n][n],B[n][n],C[n][n];
 int s1,s2,s3,s4,s5,s6,s7,s8,s9,s10;
 int p1,p2,p3,p4,p5,p6,p7;

  printf("\nEnter the first matrix:\n");
  for(int i=1;i<n;i++)
  {
    for(int j=1;j<n;j++)
    {
        scanf("%d",&A[i][j]);
    }

  }

  printf("Enter the second matrix:\n");
  for(int i=1;i<n;i++)
  {
    for(int j=1;j<n;j++)
    {
        scanf("%d",&B[i][j]);
    }
  }
```

```c
printf("\nMatrix A :\n");
for(int i=1;i<n;i++)
{
  printf("\n");
  for(int j=1;j<n;j++)
  {
  printf("%d\t",A[i][j]);
  }
}

 printf("\nMatrix B:\n");
 for(int i=1;i<n;i++)
{
  printf("\n");
  for(int j=1;j<n;j++)
  {
  printf("%d\t",B[i][j]);
  }
}

 s1=B[1][2]-B[2][2];
 s2=A[1][1]+A[1][2];
 s3=A[2][1]+A[2][2];
 s4=B[2][1]-B[1][1];
 s5=A[1][1]+A[2][2];
 s6=B[1][1]+B[2][2];
 s7=A[1][2]-A[2][2];
 s8=B[2][1]+B[2][2];
 s9=A[1][1]-A[2][1];
 s10=B[1][1]+B[1][2];

 p1=A[1][1]*s1;
 p2=s2*B[2][2];
 p3=s3*B[1][1];
 p4=s4*A[2][2];
 p5=s5*s6;
 p6=s7*s8;
 p7=s9*s10;
```

```
  C[1][1]=p5+p4-p2+p6;
  C[1][2]=p1+p2;
  C[2][1]=p5+p1+p2;
  C[2][2]=p5+p1-p3-p7;

 printf("\nStrassen Multiplication:\n");
 for(int i=1;i<n;i++)
 {
   printf("\n");
   for(int j=1;j<n;j++)
   {
      printf("%d\t",C[i][j]);
   }
 }
 printf("\n\n");

 return 0;
}
```

Output:- 1) Input from user

2) Random input

```
Output

/tmp/gLfW5jpSg6.o
Enter the required dimension:
3
Matrix A :

3    6
17   15

Matrix B:

13   15
6    12

Strassen Multiplication(Matrix C):

75   117
567  435
```

Conclusion: I have learnt the implementation of using strassen's method for matrix multiplication successfully. It is useful since its time complexity is less than normal matrix multiplication