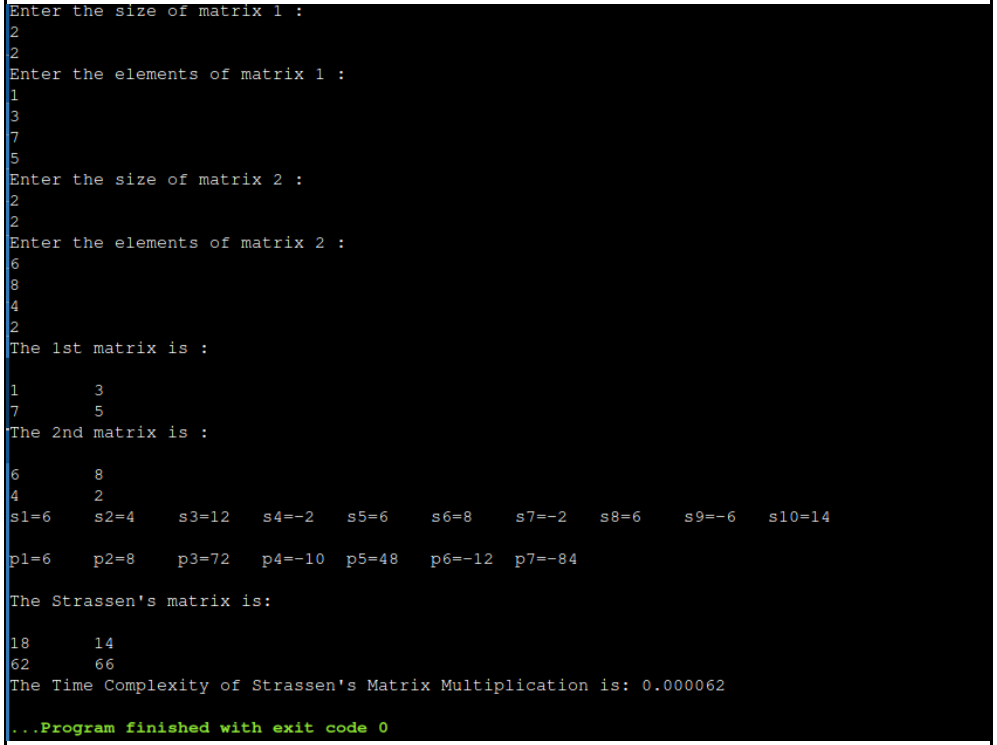
NAME:	**Sanskar Kamble**
UID:	**2021300054**
SUBJECT	**Design and Analysis of Algorithm**
EXPERIMENT NO :	**03**
DATE OF PERFORMANCE	**27/02/2023**
DATE OF SUBMISSION	**05/03/2023**
AIM:	**To implement strassen matrix multiplication and also to compute its time complexity.**
THEORY:	Strassen's Matrix multiplication can be performed only on square matrices where n is a power of 2. Order of both of the matrices are n × n. The algorithms for strassen's Matrix multiplication is as follows: • Divide a matrix of the order of 2*2 recursively until we get the matrix of order 2*2. • To carry out the multiplication of the 2*2 matrix, use the previous set of formulas. • Subtraction is also performed within these eight multiplications and four additions. • To find the final product or final matrix combine the result of two matrixes.

Code:

```
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
int main ()
{
  int a[2][2], b[2][2], c[2][2], i, j;
  int s1, s2, s3, s4, s5, s6, s7, s8, s9, s10;
  int p1, p2, p3, p4, p5, p6, p7;
  int am, an, bm, bn;
  clock_t start,end;
  printf ("Enter the size of matrix 1 :\n");
  scanf ("%d%d", &am, &an);
  printf ("Enter the elements of matrix 1 :\n");
  for (i = 0; i < am; i++)
    {
        for (j = 0; j < an; j++)
            scanf ("%d", &a[i][j]);
    }
printf("Enter the size of matrix 2 :\n");
scanf("%d%d",&bm,&bn);
printf ("Enter the elements of matrix 2 :\n");
  for (i = 0; i < bm; i++)
    {
      for (j = 0; j < bn; j++)
 scanf ("%d", &b[i][j]);
    }
  printf ("The 1st 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 2nd matrix is :\n");
  for (i = 0; i < 2; i++)
    {
      printf ("\n");
      for (j = 0; j < 2; j++)
 printf ("%d\t", b[i][j]);
    }
    start=clock();
  s1 = b[0][1] - b[1][1];
  s2 = a[0][0] + a[0][1];
  s3 = a[1][0] + a[1][1];
  s4 = b[1][0] - b[0][0];
  s5 = a[0][0] + a[1][1];
  s6 = b[0][0] + b[1][1];
  s7 = a[0][1] - a[1][1];
  s8 = b[1][0] + b[1][1];
  s9 = a[0][0] - a[1][0];
  s10 = b[0][0] + b[0][1];
  printf ("\ns1=%d\t", s1);
  printf ("s2=%d\t", s2);
  printf ("s3=%d\t", s3);
  printf ("s4=%d\t", s4);
  printf ("s5=%d\t", s5);
  printf ("s6=%d\t", s6);
  printf ("s7=%d\t", s7);
  printf ("s8=%d\t", s8);
  printf ("s9=%d\t", s9);
  printf ("s10=%d\t\n", s10);


  p1 = a[0][0] * s1;
  p2 = s2 * b[1][1];
  p3 = s3 * b[0][0];
  p4 = s4 * a[1][1];
  p5 = s5 * s6;
  p6 = s7 * s8;
  p7 = s9 * s10;

  printf ("\np1=%d\t", p1);
  printf ("p2=%d\t", p2);
  printf ("p3=%d\t", p3);
  printf ("p4=%d\t", p4);
  printf ("p5=%d\t", p5);
  printf ("p6=%d\t", p6);
  printf ("p7=%d\t\n", p7);

  c[0][0] = p5 + p4 - p2 + p6;
  c[0][1] = p1 + p2;
  c[1][0] = p3 + p4;
  c[1][1] = p5 + p1 - p3 - p7;
  printf ("\nThe Strassen's matrix is: \n");
  for (i = 0; i <am; i++)
    {
      printf ("\n");
      for (j = 0; j < bn; j++)
 printf ("%d\t", c[i][j]);
    }
    end=clock();
    printf("\nThe Time Complexity of Strassen's Matrix
Multiplication is: ");

printf("%lf",(double)(endstart)/CLOCKS_PER_SEC);

}
```

Output:



```
Enter the size of matrix 1 :
2
2
Enter the elements of matrix 1 :
1
3
7
5
Enter the size of matrix 2 :
2
2
Enter the elements of matrix 2 :
6
8
4
2
The 1st matrix is :

1       3
7       5
The 2nd matrix is :

6       8
4       2
s1=6    s2=4    s3=12   s4=-2   s5=6    s6=8    s7=-2   s8=6    s9=-6   s10=14

p1=6    p2=8    p3=72   p4=-10  p5=48   p6=-12  p7=-84

The Strassen's matrix is:

18      14
62      66
The Time Complexity of Strassen's Matrix Multiplication is: 0.000062

...Program finished with exit code 0
```

CONCLUSION:	By performing the above experiment I have successfully understood about the concept of strassen matrix multiplication as well as its implementation.