**PRACTICAL 3**

**AIM:** Develop Web service in Java that returns matrix multiplication by Strassen's algorithm. Two matrices will be entered at run time by client. Server does the matrix multiplication and returns answer to client.

**INPUT:**

**//StrassenServerImpl.java**

```
package Ex01.strassen;
import javax.jws.WebService;
@WebService(endpointInterface = "Ex01.strassen.StrassenServer")
public class StrassenServerImpl implements StrassenServer {
   public int[][] multiply(int[][] A, int[][] B)
   {  int n = A.length;
      int[][] R = new int[n][n];
      if (n == 1)
         R[0][0] = A[0][0] * B[0][0];
      else
      {  int[][] A11 = new int[n/2][n/2];
         int[][] A12 = new int[n/2][n/2];
         int[][] A21 = new int[n/2][n/2];
         int[][] A22 = new int[n/2][n/2];
         int[][] B11 = new int[n/2][n/2];
         int[][] B12 = new int[n/2][n/2];
         int[][] B21 = new int[n/2][n/2];
         int[][] B22 = new int[n/2][n/2];
         split(A, A11, 0 , 0);
         split(A, A12, 0 , n/2);
         split(A, A21, n/2, 0);
         split(A, A22, n/2, n/2);
         split(B, B11, 0 , 0);
         split(B, B12, 0 , n/2);
         split(B, B21, n/2, 0);
         split(B, B22, n/2, n/2);
         int [][] M1 = multiply(add(A11, A22), add(B11, B22));
         int [][] M2 = multiply(add(A21, A22), B11);
         int [][] M3 = multiply(A11, sub(B12, B22));
         int [][] M4 = multiply(A22, sub(B21, B11));
         int [][] M5 = multiply(add(A11, A12), B22);
         int [][] M6 = multiply(sub(A21, A11), add(B11, B12));
         int [][] M7 = multiply(sub(A12, A22), add(B21, B22));
          int [][] C11 = add(sub(add(M1, M4), M5), M7);
         int [][] C12 = add(M3, M5);
         int [][] C21 = add(M2, M4);
         int [][] C22 = add(sub(add(M1, M3), M2), M6);
         join(C11, R, 0 , 0);
```

```
        join(C12, R, 0 , n/2);
        join(C21, R, n/2, 0);
        join(C22, R, n/2, n/2);        }
        return R;    }
public int[][] sub(int[][] A, int[][] B)
{  int n = A.length;
    int[][] C = new int[n][n];
    for (int i = 0; i < n; i++)
        for (int j = 0; j < n; j++)
            C[i][j] = A[i][j] - B[i][j];
    return C;    }
public int[][] add(int[][] A, int[][] B)
{       int n = A.length;
    int[][] C = new int[n][n];
    for (int i = 0; i < n; i++)
        for (int j = 0; j < n; j++)
            C[i][j] = A[i][j] + B[i][j];
    return C;    }
public void split(int[][] P, int[][] C, int iB, int jB)
{       for(int i1 = 0, i2 = iB; i1 < C.length; i1++, i2++)
        for(int j1 = 0, j2 = jB; j1 < C.length; j1++, j2++)
            C[i1][j1] = P[i2][j2];    }
public void join(int[][] C, int[][] P, int iB, int jB)
{       for(int i1 = 0, i2 = iB; i1 < C.length; i1++, i2++)
        for(int j1 = 0, j2 = jB; j1 < C.length; j1++, j2++)
            P[i2][j2] = C[i1][j1];    }
        public String StrassenMessage(String sreq) {
          String MatrixC="";
          System.out.println("Server: StrassenMessage() invoked...");
          System.out.println("Server: Message > " + sreq);
          int idx = sreq.indexOf(",");
          int N = Integer.parseInt(sreq.substring(0,idx));
   System.out.println("N="+N);
          int t=0;
          int[][] A = new int[N][N];
          int[][] B = new int[N][N];
   int[][] C = new int[N][N];
   for (int i = 0; i < N; i++)
          {for (int j = 0; j < N; j++)
                    {int from = sreq.indexOf(',',t);
          int to = sreq.indexOf(',', from+1);
                              A[i][j] = Integer.parseInt(sreq.substring(from+1,to));
                              t=to;   }   }
   for (int i = 0; i < N; i++)
          {         for (int j = 0; j < N; j++)
                    {       int from = sreq.indexOf(',',t);
```

```
        int to = sreq.indexOf(',', from+1);
                        B[i][j] = Integer.parseInt(sreq.substring(from+1,to));
                        t=to;               }       }
        int[][] MAT = multiply(A, B);
    StringBuilder MatC = new StringBuilder();
        MatC.append(N+",");
    for (int i = 0; i < N; i++)
        {   for (int j = 0; j < N; j++)
                {MatC.append(MAT[i][j]);
        if (i==N-1 && j==N-1)
          MatC.append("");
                        else
          MatC.append(",");   }
           System.out.println();       }
    MatrixC=MatC.toString();
        return(MatrixC);      }}
```

**//StrassenServerPublisher.java**

```
package Ex01.strassen;
import javax.xml.ws.Endpoint;
public class StrassenServerPublisher {
        public static void main(String[ ] args) {
        Endpoint.publish("http://localhost:9876/strassen", new StrassenServerImpl());       }}
```

**//StrassenServer.java**

```
package Ex01.strassen;
import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.soap.SOAPBinding;
import javax.jws.soap.SOAPBinding.Style;
@WebService // This signals that this is a Service Endpoint Interface (SEI)
@SOAPBinding(style = Style.RPC)
public interface StrassenServer {
        @WebMethod // This signals that this method is a service operation
        String StrassenMessage(String strMsg); }
```

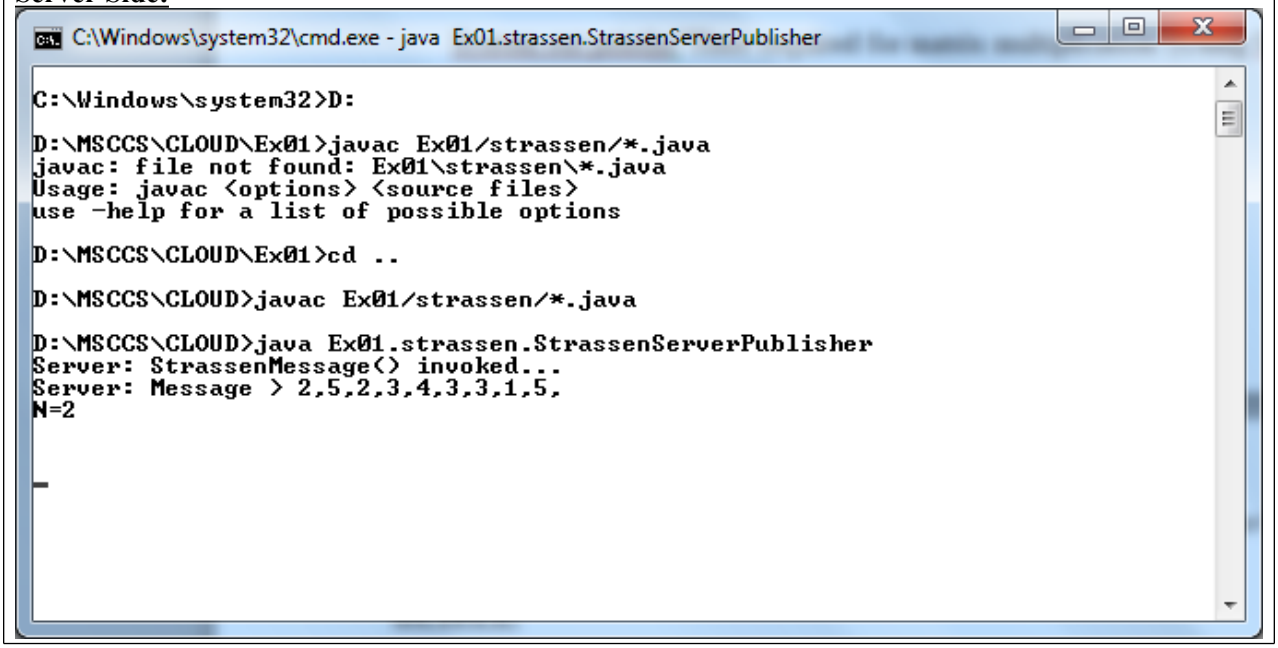
**//StrassenClient.java**

```
package Ex01.strassen;
import javax.xml.namespace.QName;
import javax.xml.ws.Service;
import java.net.URL;
class StrassenClient { public static void main(String argv[ ]) throws Exception {
                long start,finish, difference;
      if (argv.length < 1) {
System.out.println("Usage: java StrassenClient \"N,matrix elements separated by ,\"");
                        System.exit(1);                 }
```

```
                String strMsg = argv[0];
                URL url = new URL("http://localhost:9876/strassen?wsdl");
QName qname = new QName("http://strassen.Ex01/", "StrassenServerImplService");
                Service service = Service.create(url, qname);
                StrassenServer eif = service.getPort(StrassenServer.class);
              start=System.currentTimeMillis();
                String sreq=eif.StrassenMessage(strMsg);
                finish=System.currentTimeMillis();
                sreq=sreq+",";
                int idx = sreq.indexOf(",");
         int N = Integer.parseInt(sreq.substring(0,idx));
         int t=0;
         int[][] C = new int[N][N];
     for (int i = 0; i < N; i++)
         {       for (int j = 0; j < N; j++)
                 {      int from = sreq.indexOf(',',t);
        int to = sreq.indexOf(',', from+1);
                        C[i][j] = Integer.parseInt(sreq.substring(from+1,to));
                        t=to;              }       }
     System.out.println("\nMatrix Multiplication is ...\n");
      for (int i = 0; i < N; i++)
     {                  for (int j = 0; j < N; j++)
                    System.out.print(C[i][j] + "  ");
                   System.out.println();             }
                difference=finish-start;
     System.out.println("Time required for matrix multiplication (Using Strassen algorithm) :");
     System.out.println(difference + " milli seconds");        }         }
```

**OUTPUT:**

**Server Side:**




**Client Side:**