DCN

Assignment - 5

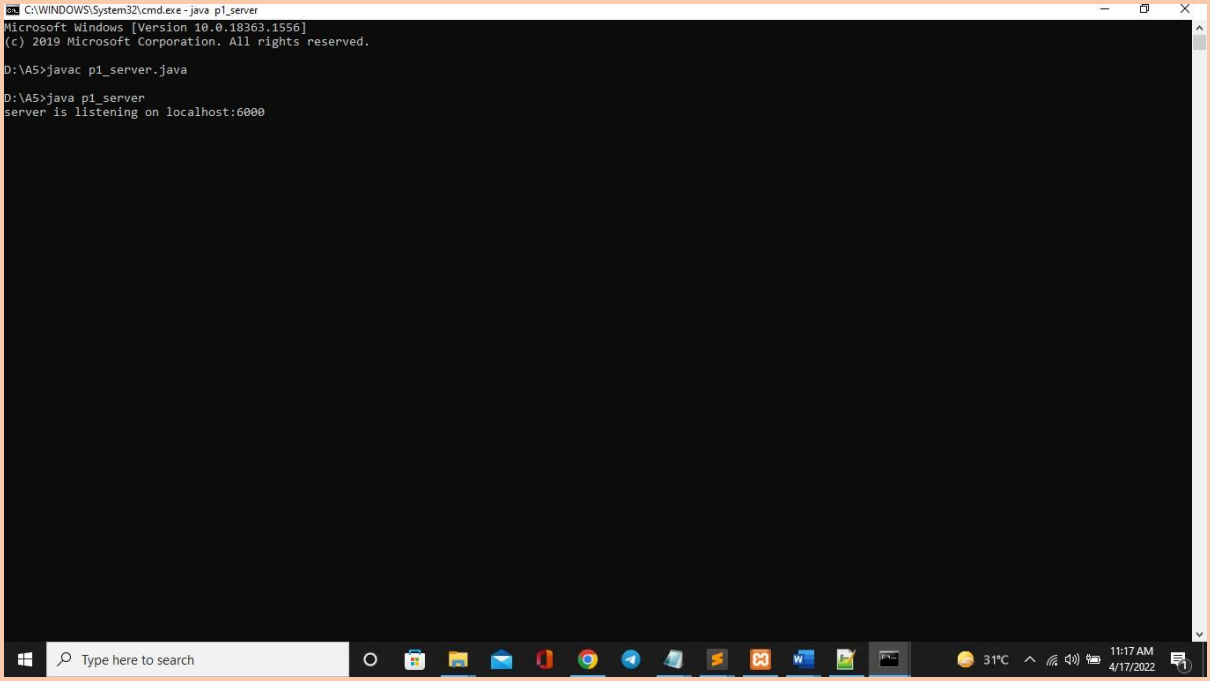
3152 HARDIK TOGADIYA

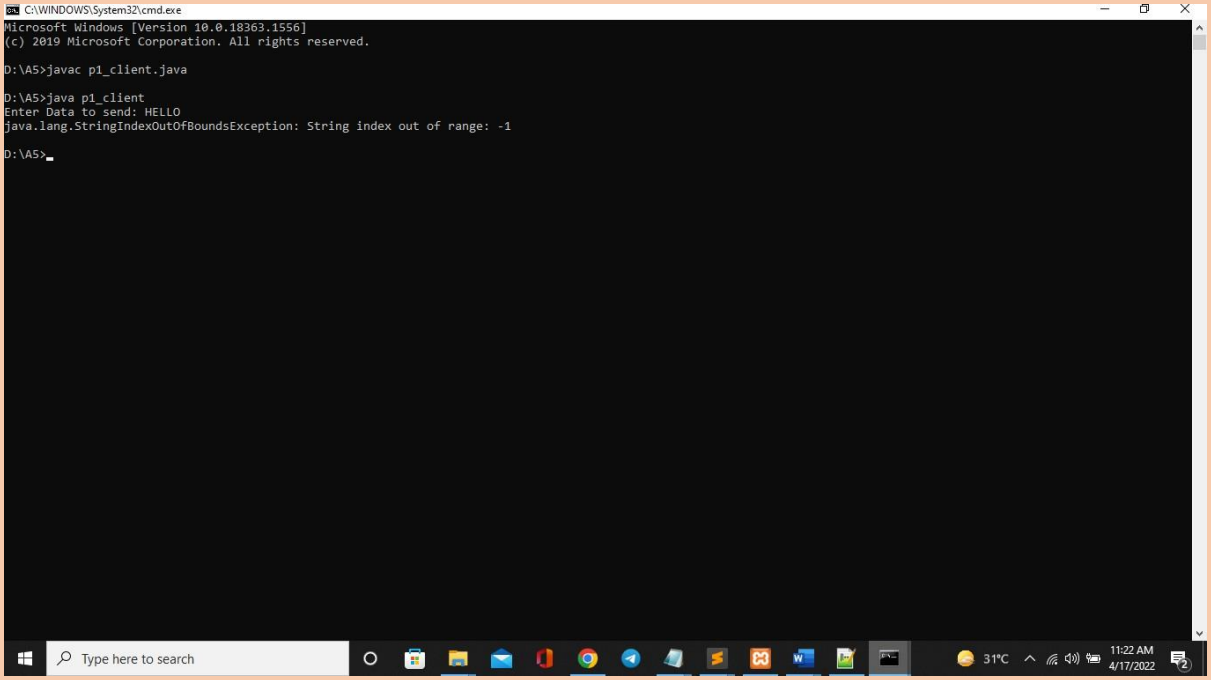
1. Write a java program to implement Mono-Alphabetic Cipher with Key-String using TCP.

```
import java.io.*;
import java.net.*;
import java.util.Scanner;

class p1_client {
    public static void main(String[] args) throws Exception {
        try {
            Socket socket = new Socket("localhost", 6000);
            DataOutputStream ostream = new
                    DataOutputStream(socket.getOutputStream());
            DataInputStream instream = new
                    DataInputStream(socket.getInputStream());
            Scanner sc = new Scanner(System.in);
            String data = "abcdefghijklmnopqrstuvwxyz";
            String key = "qwertyuiopasdfghjklzxcvbnm";
            System.out.print("Enter Data to send: ");
            String msg = sc.nextLine();
            String sendmsg = "";
            char ch;
            for (int i = 0; i < msg.length(); i++) {
                ch = msg.charAt(i);
                int index = data.indexOf(ch);
                sendmsg += key.charAt(index);
            }
            ostream.writeUTF(sendmsg);
            System.out.println("Encrypted = " + sendmsg);
            instream.close();
            ostream.close();
            socket.close();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

```
import java.io.*;
import java.net.*;
import java.util.Scanner;

class lass p1_server {
    public static void main(String[] args) throws Exception {
        try {
            ServerSocket serversocket = new ServerSocket(6000);
            System.out.println("server is listening on localhost:6000");
            Socket socket = serversocket.accept();
            DataInputStream instream = new
DataInputStream(socket.getInputStream());
            DataOutputStream ostream = new
```

```
DataOutputStream(socket.getOutputStream());
            String data = "abcdefghijklmnopqrstuvwxyz";
            String key = "qwertyuiopasdfghjklzxcvbnm";
            String rmsg = instream.readUTF();
            System.out.println("\nReceived msg = " + rmsg);
            String final_msg = new String();
            for (int i = 0; i < rmsg.length(); i++) {
                char ch = rmsg.charAt(i);
                int index = key.indexOf(ch);
                final_msg += data.charAt(index);
            }
            System.out.println("Decrypted msg = " + final_msg);
            ostream.close();
            instream.close();
            socket.close();
            serversocket.close();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```



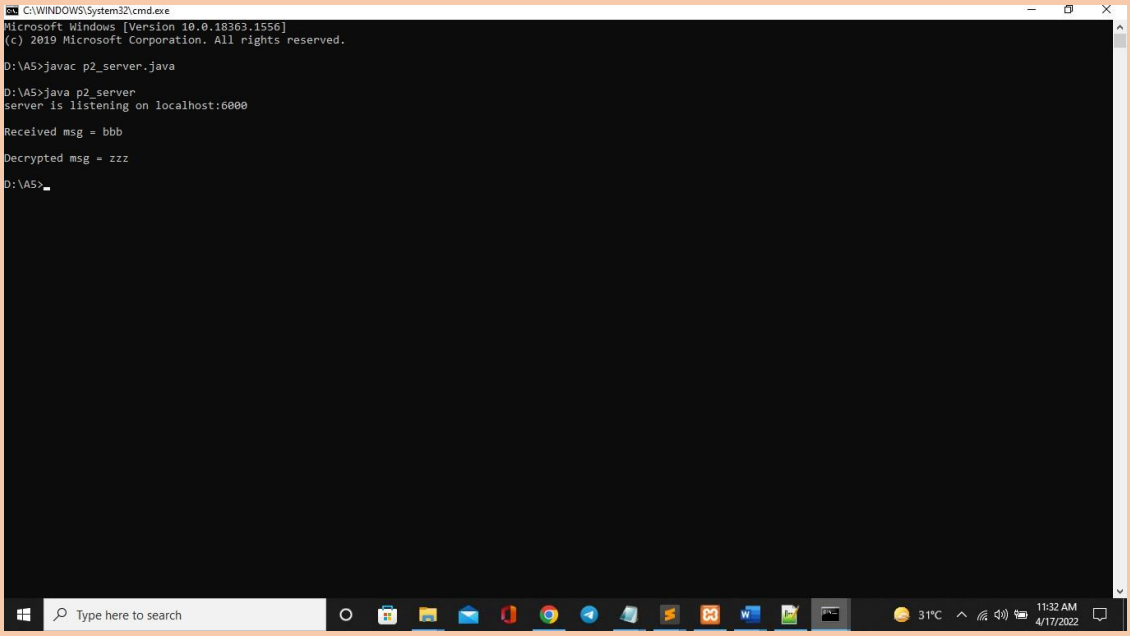


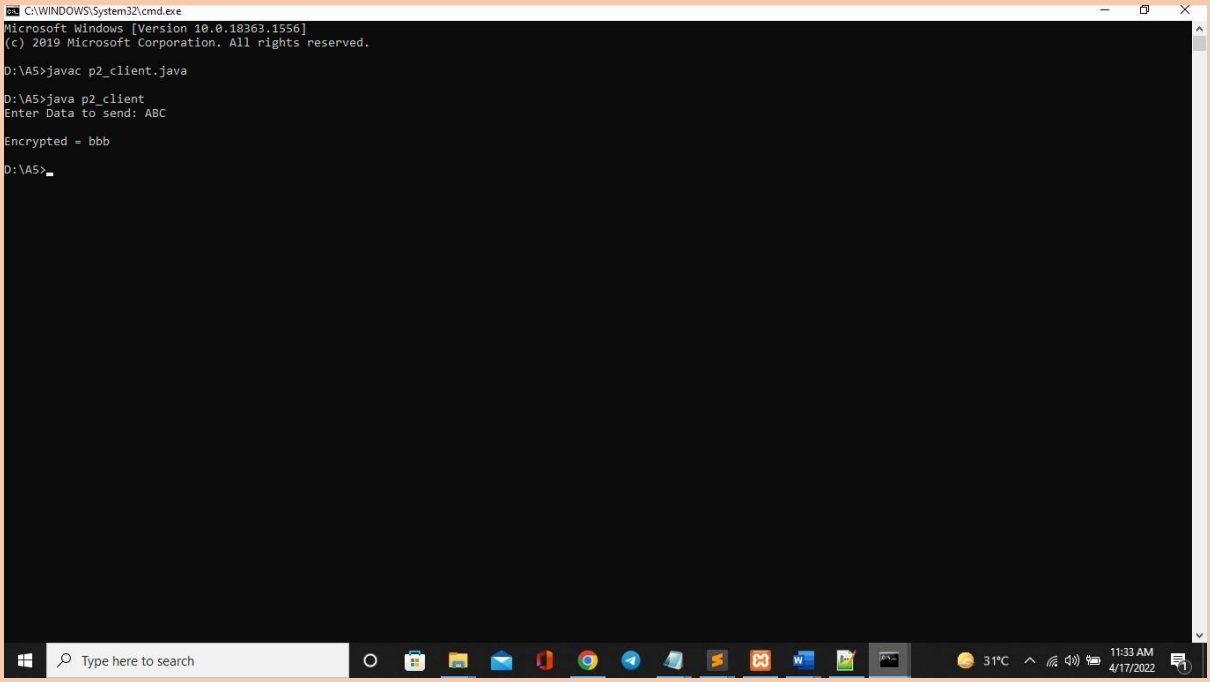
2. Write a java program to implement Caesar Cipher using TCP.

```
import java.io.*;
import java.net.*;
import java.util.Scanner;

class p2_client {
    public static void main(String[] args) throws Exception {
        try {
            Socket socket = new Socket("localhost", 6000);
            DataOutputStream ostream = new
DataOutputStream(socket.getOutputStream());
            DataInputStream instream = new
DataInputStream(socket.getInputStream());
            Scanner sc = new Scanner(System.in);
            String data = "abcdefghijklmnopqrstuvwxyz";
            int key = 2;
            System.out.print("Enter Data to send: ");
            String msg = sc.nextLine();
            String sendmsg = "";

            for (int i = 0; i < msg.length(); i++) {
                char ch = msg.charAt(i);
                if (ch == 'y') {
                    sendmsg += 'a';
                } else if (ch == 'z') {
```

```
                } else {


                }
            }

            sendmsg += 'b';


            int index = data.indexOf(ch);
            sendmsg += data.charAt(index + key);

            ostream.writeUTF(sendmsg);
            System.out.println("\nEncrypted = " + sendmsg);

            3111 - Kumbhani Sanket


            instream.close();
            ostream.close();
            socket.close();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}



        import java.io.*;import java.net.*;
                import java.util.Scanner;

class p2_server {
    public static void main(String[] args) throws Exception {

        try {


            ServerSocket serversocket = new ServerSocket(6000);
            System.out.println("server is listening on localhost:6000");
            Socket socket = serversocket.accept();
            DataInputStream instream = new
DataInputStream(socket.getInputStream());
            DataOutputStream ostream = new
DataOutputStream(socket.getOutputStream());

            String data = "abcdefghijklmnopqrstuvwxyz";
            int key = 2;

            String rmsg = instream.readUTF();
            System.out.println("\nReceived msg = " + rmsg);
            String final_msg = new String();
            for (int i = 0; i < rmsg.length(); i++) {
                char ch = rmsg.charAt(i);
                if (ch == 'a') {
                    final_msg += 'y';
```

```
                } else if (ch == 'b') {


                } else {


                }
            }

            final_msg += 'z';
            int index = data.indexOf(ch);
            final_msg += data.charAt(index - key);
            System.out.println("\nDecrypted msg = " + final_msg);
            ostream.close();
            instream.close();
            socket.close();
            serversocket.close();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```



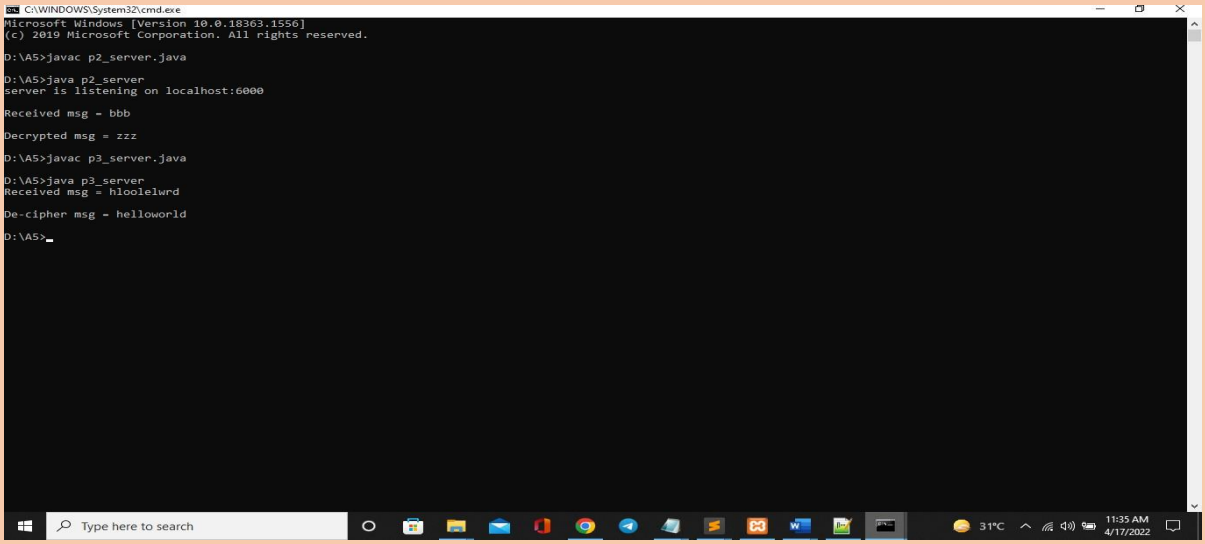


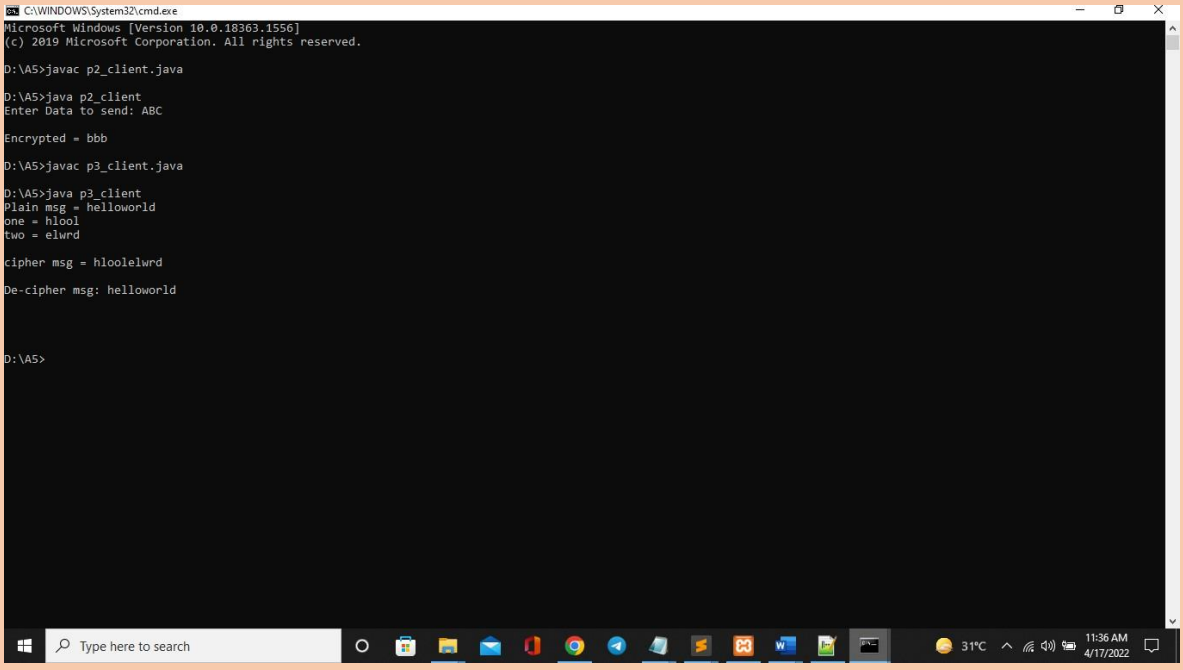
3. Write a java program to implement Rail-Fence Cipher using UDP.

```
import java.net.*; import java.io.*;
import java.util.Scanner;
class p3_client
{
   public static void main(String args[]) throws Exception
   {
      try{
         DatagramSocket ds1 = new DatagramSocket();

         String msg = "helloworld";
         System.out.println("Plain msg = "+ msg); String one = "";
         String two = "";
         for(int i=0;i<msg.length();i++)
         {
            if(i%2==0)
               one += msg.charAt(i); else
               two += msg.charAt(i);
         }
         System.out.println("one = "+one); System.out.println("two = "+two);
String cmsg = one + two ;
         System.out.println("\ncipher msg = "+ cmsg); InetAddress ip =
```

```
InetAddress.getByName("localhost");

         DatagramPacket dp1 = new
DatagramPacket(cmsg.getBytes(),cmsg.length(),ip,6363);
         ds1.send(dp1);

         DatagramSocket ds2 = new DatagramSocket(6565); byte[] buf = new
byte[500];
         DatagramPacket dp2 = new DatagramPacket(buf,500); ds2.receive(dp2);

         String msg1 = new String(buf); System.out.println("\nDe-cipher msg: "
+ msg1);

      }catch(Exception e)
      {System.out.println(e);}
   }
}

      import java.net.*; import java.io.*;
      import java.util.Scanner;

class p3_server
{
   public static void main(String args[]) throws Exception
   {
      try{
         DatagramSocket ds1 = new DatagramSocket(6363); byte[] buf = new
byte[500];
         DatagramPacket dp1 = new DatagramPacket(buf,500); ds1.receive(dp1);
         ds1.close();

         String cmsg = new String(dp1.getData(),0,dp1.getLength());
System.out.println("Received msg = " + cmsg);
         DatagramSocket ds2 = new DatagramSocket(); String a =
cmsg.substring(0,cmsg.length()/2);
         String b = cmsg.substring(cmsg.length()/2,cmsg.length());
         String dcmsg = "";
         for(int i=0,x=0,y=0; i<cmsg.length();i++)
         {
            if(i%2 == 0)
            {
               dcmsg += a.charAt(x); x++;
            }
            else
            {
               dcmsg += b.charAt(y); y++;
            }
         }
         System.out.println("\nDe-cipher msg = "+ dcmsg);
         InetAddress ip = InetAddress.getByName("localhost"); DatagramPacket
dp2 = new
               DatagramPacket(dcmsg.getBytes(),dcmsg.length(),ip,6565);
ds2.send(dp2);
      }catch(Exception e)

      {System.ou
```

```
t.println(e);}
    }
}
```

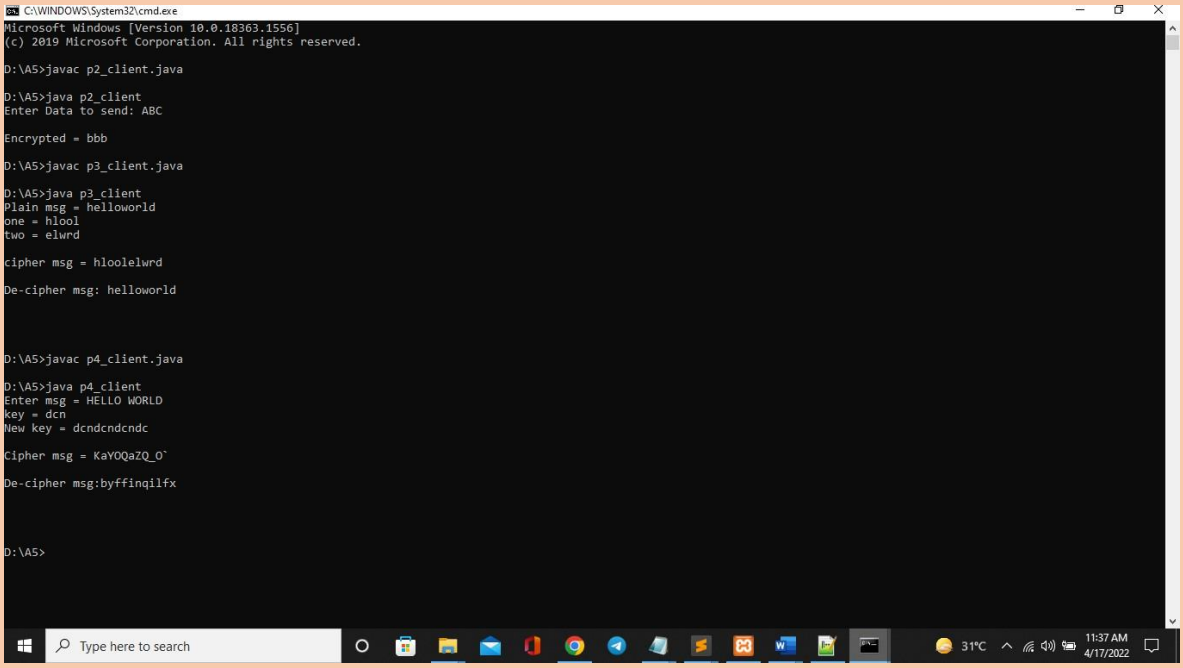


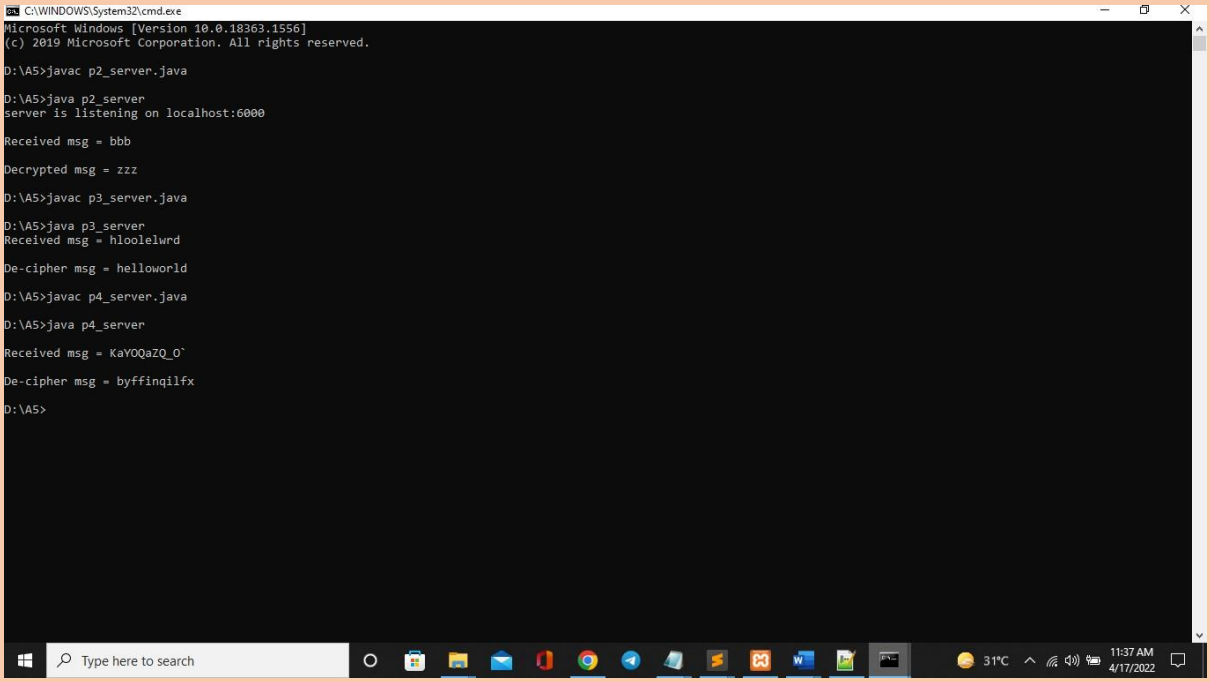


4. Write a java program to implement Vigenere Cipher using UDP.

```
import java.net.*; import java.io.*;
import java.util.Scanner;

class p4_client
{

   public static void main(String args[]) throws Exception
   {
      try{
         DatagramSocket ds1 = new DatagramSocket();

         Scanner sc = new Scanner(System.in);

         System.out.print("Enter msg = "); String msg = sc.nextLine();
// System.out.print("Enter Key = ");
// String key = sc.nextLine(); String key = "dcn";
// System.out.println("msg = "+ msg); System.out.println("key = "+ key);
         int x=0; while(key.length()<msg.length())
         {
            key += key.charAt(x); x++;
         }
         System.out.println("New key = "+ key);


// cipher process

         String cmsg = "";
         for(int i=0;i<msg.length();i++)
         {
            int cno = (msg.charAt(i)-97 + key.charAt(i)-97) % 26 ; cno += 97;
            cmsg += (char)cno;
         }
         System.out.println("\nCipher msg = "+ cmsg); InetAddress ip =
InetAddress.getByName("localhost");
         DatagramPacket dp1 = new
DatagramPacket(cmsg.getBytes(),cmsg.length(),ip,6363);
         ds1.send(dp1);

         DatagramSocket ds2 = new DatagramSocket(6565);

         byte[] buf = new byte[500];
         DatagramPacket dp2 = new DatagramPacket(buf,500); ds2.receive(dp2);

         String msg1 = new String(buf); System.out.println("\nDe-cipher msg:"
+ msg1);



      }catch(Exception e)
      {System.out.println(e);}
   }
}

      import java.net.*; import java.io.*;
      import java.util.Scanner;
```

```
class p4_server
{
   public static void main(String args[]) throws Exception
   {

      try{
         DatagramSocket ds1 = new DatagramSocket(6363); byte[] buf = new
byte[500];
         DatagramPacket dp1 = new DatagramPacket(buf,500); ds1.receive(dp1);
         ds1.close();
         String rmsg = new String(dp1.getData(),0,dp1.getLength());
System.out.println("\nReceived msg = " + rmsg);

         DatagramSocket ds2 = new DatagramSocket(); String key = "dcn";
         int x=0; while(key.length()<rmsg.length())
         {
            key += key.charAt(x); x++;
         }
         String dcmsg = "";
         for(int i=0;i<rmsg.length();i++)
         {
            int cno = ((rmsg.charAt(i)-97) - (key.charAt(i)-97) + 26) % 26 ;
cno += 97;

            dcmsg += (char)cno;
         }
         System.out.println("\nDe-cipher msg = "+dcmsg);

         InetAddress ip = InetAddress.getByName("localhost"); DatagramPacket
dp2 = new
               DatagramPacket(dcmsg.getBytes(),dcmsg.length(),ip,6565);
ds2.send(dp2);
      }catch(Exception e)
      {System.out.println(e);}
   }
}
```





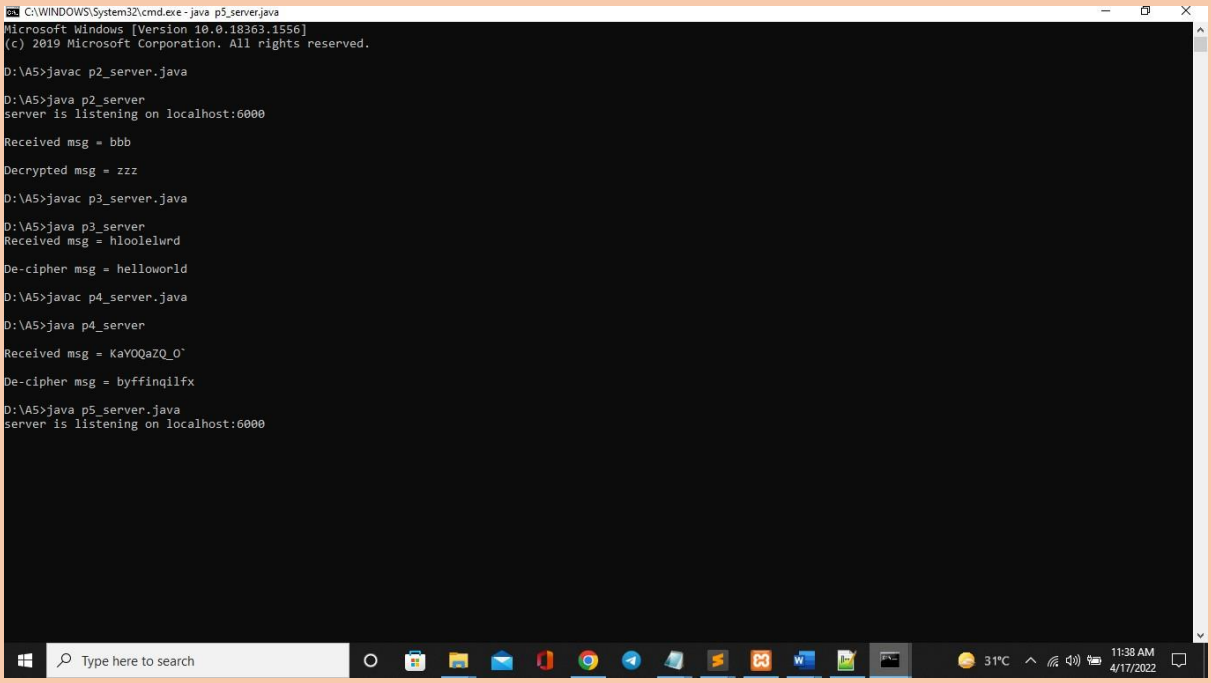
5. Write a java program to implement DES Algorithm with help of Java Cryptography packages using TCP.

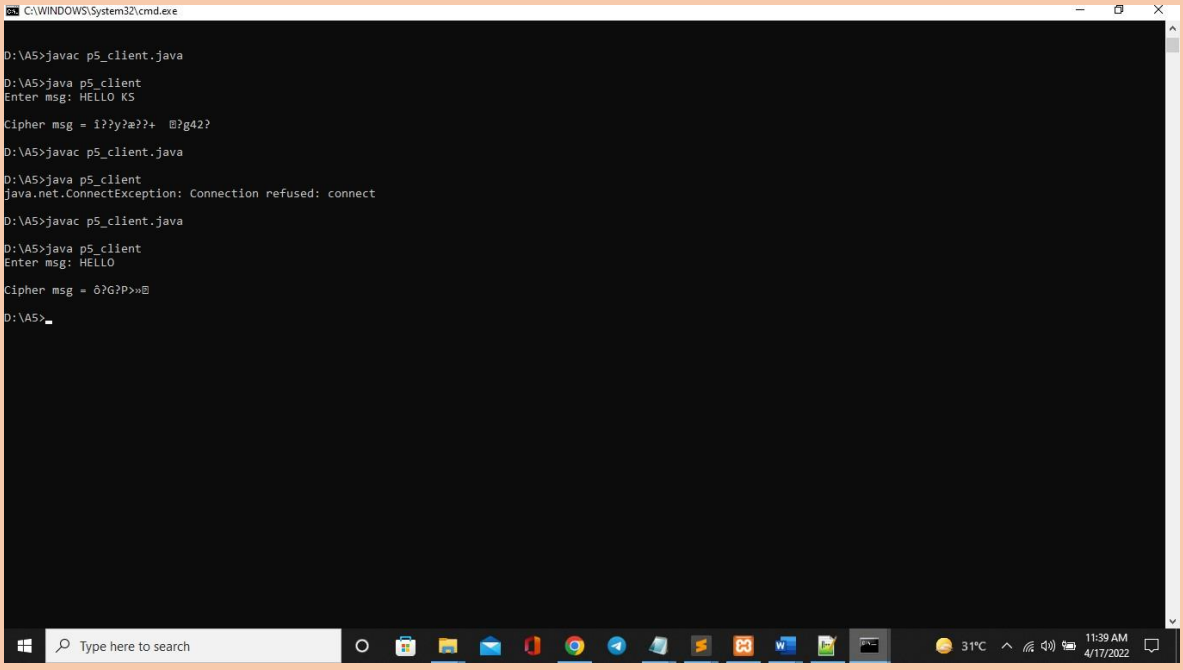
```
import java.io.*; import java.net.*;
import java.util.Scanner; import java.security.*; import javax.crypto.*;
import javax.crypto.spec.*;

class p5_client
{
   public static void main(String[] args) throws Exception

   {
      try
      {
         Socket socket = new Socket("localhost",6000);
         DataOutputStream ostream = new
DataOutputStream(socket.getOutputStream());
         DataInputStream instream = new
DataInputStream(socket.getInputStream());
         Scanner sc = new Scanner(System.in); System.out.print("Enter msg: ");
String msg = sc.nextLine();
//key generate
         String kstring = "sahildcn";

         SecretKey key = new SecretKeySpec(kstring.getBytes(),"DES");

         Cipher cipher = Cipher.getInstance("DES/ECB/PKCS5Padding");
         cipher.init(Cipher.ENCRYPT_MODE,key);

         String cmsg = new String(cipher.doFinal(msg.getBytes()));
ostream.writeUTF(cmsg);
         System.out.println("\nCipher msg = "+ cmsg);

         instream.close(); ostream.close(); socket.close();
      }
      catch(Exception e)

      {
         System.out.println(e);
      }

   }
}
```

Server..

```
      import java.io.*; import java.net.*;
      import java.util.Scanner; import java.security.*;
      import javax.crypto.*; import javax.crypto.spec.*;

class p5_server
{
   public static void main(String[] args) throws Exception
   {

      try
```

```
      {
         ServerSocket serversocket = new ServerSocket(6000);
System.out.println("server is listening on localhost:6000");


         Socket socket = serversocket.accept(); DataInputStream instream = new
            DataInputStream(socket.getInputStream());
         DataOutputStream ostream = new
DataOutputStream(socket.getOutputStream());


         String cmsg = instream.readUTF(); System.out.println("Received msg =
"+cmsg);
//key generate

         String kstring = "sahildcn"; SecretKey key = new
            SecretKeySpec(kstring.getBytes(),"DES");

         Cipher cipher = Cipher.getInstance("DES/ECB/PKCS5Padding");

         cipher.init(Cipher.DECRYPT_MODE,key);

         String dcmsg = new String(cipher.doFinal(cmsg.getBytes()));
System.out.println("\nDe-Cipher msg = "+ dcmsg);

         ostream.close(); instream.close(); socket.close();
serversocket.close();
      }
      catch(Exception e)
      {
         System.out.println(e);
      }
   }
}
```





6. Write a java program to implement AES Algorithm with help of Java Cryptography packages using UDP.

```
import java.net.*;
import java.io.*;
import java.util.Scanner;
import java.security.*;
import javax.crypto.*;
import javax.crypto.spec.*;

class p6_client {

    public static void main(String args[]) throws Exception {
        try {
            DatagramSocket ds1 = new DatagramSocket();

            Scanner sc = new Scanner(System.in);
            System.out.print("Enter msg: ");
            String msg = sc.nextLine();

            String kstring = "sahildcnsahildcn";
            SecretKey key = new SecretKeySpec(kstring.getBytes(), "AES");

            Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5Padding");
            cipher.init(Cipher.ENCRYPT_MODE, key);

            String cmsg = new String(cipher.doFinal(msg.getBytes()));
            System.out.println("\nCipher msg = " + cmsg);


            InetAddress ip = InetAddress.getByName("localhost");
            DatagramPacket dp1 = new
                    DatagramPacket(cmsg.getBytes(), cmsg.length(), ip, 6363);
            ds1.send(dp1);

            DatagramSocket ds2 = new DatagramSocket(6565);
            byte[] buf = new byte[500];
            DatagramPacket dp2 = new DatagramPacket(buf, 500);
            ds2.receive(dp2);

            String msg1 = new String(buf);
            3111 - Kumbhani Sanket

            System.out.println("server:" + msg1);


        } catch (Exception e) {
            System.out.println(e);
        }
    }
}

      import java.net.*;import java.io.*;
                import java.util.Scanner;import java.security.*;
                import javax.crypto.*;import javax.crypto.spec.*;

class p6_server {
    public static void main(String args[]) throws Exception {
```

```java
        try {
            DatagramSocket ds1 = new DatagramSocket(6363);
            byte[] buf = new byte[500];
            DatagramPacket dp1 = new DatagramPacket(buf, 500);
            ds1.receive(dp1);
            ds1.close();

            String cmsg = new String(dp1.getData(), 0, dp1.getLength());
            System.out.println("Received msg = " + cmsg);
            DatagramSocket ds2 = new DatagramSocket();
            String kstring = "sahildcnsahildcn";
            SecretKey key = new SecretKeySpec(kstring.getBytes(), "AES");

            Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5Padding");
            cipher.init(Cipher.DECRYPT_MODE, key);

            String dcmsg = new String(cipher.doFinal(cmsg.getBytes()));
            System.out.println("\nDe-Cipher msg = " + dcmsg);


            InetAddress ip = InetAddress.getByName("localhost");
            DatagramPacket dp2 = new
                    DatagramPacket(dcmsg.getBytes(), dcmsg.length(), ip,
6565);
            ds2.send(dp2);
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```