Experiment-1

AIM: Write a program to implement fiestal cipher structure

PROGRAM:

```
import java.io.*;

class functions

{

  String Xor(String s1,String s2)

   {

          char s1array[]=s1.toCharArray();

          char s2array[]=s2.toCharArray();

          char resultarray[]=new char[s1.length()];

          for(int i=0;i<s1.length();i++)

             resultarray[i]=computexor(s1array[i],s2array[i]);

          String result=new String(resultarray);

          return result;

   }

   char computexor(char c1,char c2)

   {

          if(c1==c2)

             return '0';

          else

             return '1';

   }


}
```

```
class fiestal
{
  public static void main(String args[])throws Exception
  {
       DataInputStream dis=new DataInputStream(System.in);
       System.out.println("Enter your choice\n1.Encryption\n2.Decryption");
       int choice=Integer.parseInt(dis.readLine());
       System.out.println("Enter the number of rounds");
       int rnd=Integer.parseInt(dis.readLine());
       System.out.println("Enter the plain text");
       String ptext=dis.readLine();
       System.out.println("Enter the key");
       String key=dis.readLine();
       String tempkey=key;
       System.out.println("Enter the permutation matrix one by one");
       int p[]=new int[key.length()];
       for(int m=0;m<key.length();m++)
         p[m]=Integer.parseInt(dis.readLine());
       String func;
       int i,j=0,k,rndct;
       char parray[]=ptext.toCharArray();
       char leftarray[]=new char[ptext.length()/2];
       char rightarray[]=new char[ptext.length()/2];
       for(i=0;i<(ptext.length()/2);i++)
```

```
        leftarray[i]=parray[i];
      for(k=i;k<ptext.length();k++)
        rightarray[j++]=parray[k];
      String left=new String(leftarray);
      String right=new String(rightarray);
      functions fn=new functions();
      String temp;
      char permut[]=new char[key.length()];
      rndct=1;
      if(choice==2)
      {

            int pinv[]=new int[key.length()];
            //Finding Inverse Permutation
            for(i=0;i<key.length();i++)
            {
                  int tem=p[i]-1;
                  pinv[tem]=i+1;
            }

            //Finding the key used for last round in encryption
            for(i=1;i<=rnd;i++)
            {
                  char keyarray[]=tempkey.toCharArray();
                  for(int l=0;l<key.length();l++)
```

```
              {
                int s=p[l]-1;
                permut[l]=keyarray[s];
              }
            tempkey=new String(permut);
        }
        //making the inverse matrix as permutation matrix
        for(i=0;i<key.length();i++)
            p[i]=pinv[i];
    //doing the first round of decryption using the last round key found
    func=fn.Xor(right,tempkey);
    temp=fn.Xor(left,func);
    left=right;
    right=temp;
    rndct=2;
    }
//for encryption loop starts from first round
    //for decryption loop starts from next round
    for(i=rndct;i<=rnd;i++)
    {
        char keyarray[]=tempkey.toCharArray();

        for(int l=0;l<key.length();l++)
          {
            int s=p[l]-1;
            permut[l]=keyarray[s];
          }
        tempkey=new String(permut);
        func=fn.Xor(right,tempkey);
        temp=fn.Xor(left,func);
        left=right;
        right=temp;
     }
    String result=right+left;
    System.out.println("RESULT IS "+result);
 }
}
```

OUTPUT:

(Find yourself..)

Experiment-2

AIM: Write a program to implement hill cipher algorithm

PROGRAM:

```
import java.io.*;

class hill

{

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

{

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

 int i,j,k,buffer=0;

            int key[][]=new int[3][3];

            System.out.println("Enter the Key matrix:");

            for(i=0;i<3;i++)

            {

                  for(j=0;j<3;j++)

                  {

                        key[i][j]=Integer.parseInt(br.readLine());

                  }
```

```
}
System.out.println("Enter the plaintext:");
String plaintext=br.readLine().toUpperCase();
int pt[][]=new int[10][3];
k=0;
int s;
boolean set=false;
for(i=0;;i++)
{
    for(j=0;j<3;j++)
    {
        if(k==plaintext.length())
        {
            if(j==2)
            {
                set=true;
                pt[i][j]=23;
                break;
            }
            pt[i][j]=23;
                       buffer++;
        }
        else
        {
           pt[i][j]=((int)plaintext.charAt(k)-65);
```

```
            k++;
        }
    }
    if(set==true)
    {
        s=i;
        break;
    }
}
int ct[][]=new int[10][3];
System.out.println("Cipher text:");
for(i=0;i<=s;i++)
{
    ct[i][0]=0;
    for(j=0;j<3;j++)
    {
        for(k=0;k<3;k++)
        {
            ct[i][j]=(ct[i][j]+key[j][k]*pt[i][k]) %26;
        }
    }
}
    for(i=0;i<=s;i++)
    {
          for(j=0;j<3;j++)
          {
                System.out.print((char)(ct[i][j]+'A'));
    }
    }
System.out.println();
    int detk=0,x,y,a,b;
for(i=0;i<3;i++)
{
    for(j=0;j<1;j++)
    {
        x=(i+1)%3;
        y=(i+2)%3;
        a=(j+1)%3;
        b=(j+2)%3;
        detk=detk+key[i][j] * (key[x][a]*key[y][b]-key[y][a]*key[x][b]);
    }
}
if(detk<0)
{
    detk=-detk;
    detk=detk%26;
    detk=-detk+26;
}
else
    detk=detk%26;
```

```
        System.out.println("Determinant="+detk);
        if(detk==0)
            {
                    System.out.println("Inverse not exist");
                    System.exit(0);
            }
            int adj[][]=new int[3][3];
        for(i=0;i<3;i++)
        {
            for(j=0;j<3;j++)
            {
                adj[i][j]=0;
                x=(i+1)%3;
                y=(i+2)%3;
                a=(j+1)%3;
                b=(j+2)%3;
                adj[i][j]= (key[x][a]*key[y][b]-key[y][a]*key[x][b]);
                if(adj[i][j]<0)
                {
                    adj[i][j]=-adj[i][j];
                    adj[i][j]=adj[i][j]%26;
                    adj[i][j]=-adj[i][j]+26;
                }
                else
                    adj[i][j]=adj[i][j]%26;
```

```
          }
      }
      int ans[][]=new int[3][3];
      String qq;
          for(i=0;i<3;i++)
      {
          for(j=0;j<3;j++)
          {
              for(a=0;;a++)
              {
                  if((detk*a-adj[i][j]) % 26==0)
                  {
                      ans[i][j]=a;
                                break;
                  }
                  }
          }
          }
          System.out.println("Inverse MAtrix:");
          for(i=0;i<3;i++)
          {
                for(j=0;j<3;j++)
                {
                      System.out.print(ans[j][i]+"\t");
                }
```

```
                    System.out.println();
                }
            int dt[][]=new int[10][3];
            String text="";
                System.out.println("Decrypted text:");
            for(i=0;i<=s;i++)
            {
                dt[i][0]=0;
                for(j=0;j<3;j++)
                {
                    for(k=0;k<3;k++)
                    {
                        dt[i][j]=(dt[i][j]+ans[k][j]*ct[i][k]) %26;
                    }
                }
            }
            for(i=0;i<=s;i++)
                {
                    for(j=0;j<3;j++)
                    {
                        text=text+((char)(dt[i][j]+'A'));
                    }
                }
            System.out.println(text.substring(0,text.length()-buffer-1));
        }
```

```
}
```

OUTPUT:

Experiment-3

AIM: Write a program to implement playfair cipher algorithm

PROGRAM:

```
import java.io.*;
class playfairk
{
    public static void main(String arg[])throws IOException
    {
        int i,j,len=1,temp=0,row,col,plen=0,r1=0,c1=0,r2=0,c2=0;
        int space[]=new int[20];
        char mat[][]=new char[25][25];
        char keyarr[]=new char[25];
        char a,b;
        char alpha[]={'A','B','C','D','E','F','G','H','I','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z'};
        char ptext[]=new char[100];
        char pltext[]=new char[110];
        char cipher[]=new char[110];
        char dcipher[]=new char[110];
        char dplain[]=new char[110];
        String key,plain;
        DataInputStream dis=new DataInputStream(System.in);
        System.out.println("ENTER THE KEY: ");
        key=dis.readLine();
        key=key.toUpperCase();
        key=key.replace('J','I');
        keyarr[0]=key.charAt(0);
```

```
for(i=1;i<key.length();i++)
{
    for(j=0;j<i;j++)
    {
        if(key.charAt(i)==key.charAt(j))

            break;
    }
    if(i==j)
    {
        keyarr[len]=key.charAt(j);
        len++;
    }
}
for(i=0;i<5;i++)
{
    for(j=0;j<5;j++)
    {
        if(temp==len)
            break;
        mat[i][j]=keyarr[temp];
        temp++;
    }
    if(temp==len)
        break;
}
row=len/5;
col=len%5;
for(i=0;i<25;i++)
{
    for(j=0;j<len;j++)
    {
        if(alpha[i]==keyarr[j])
            break;
    }
    if(j==len)
    {
        if(col==5)
        {
            row++;
            col=0;
        }
        mat[row][col]=alpha[i];
        col++;
    }
}
System.out.println("\nPLAYFAIR MATRIX:");
for(i=0;i<5;i++)
{
    for(j=0;j<5;j++)
```

```
            System.out.print("\t"+mat[i][j]);
        System.out.println();
    }
    System.out.println("ENTER THE PLAINTEXT: ");
    plain=dis.readLine();
    plain=plain.toUpperCase();
    plain=plain.replace('J','I');
    temp=0;
    j=0;
    for(i=0;i<plain.length();i++)
    {
        if(plain.charAt(i)!=' ')
        {
            ptext[temp]=plain.charAt(i);
            temp++;
        }
        else
        {
            space[j]=i;
            j++;
        }
   }
    for(i=0;i<j;i++)
    space[i]=space[i]-i;
    for(i=0;i<temp;i++)
```

```
{
    pltext[plen]=ptext[i];
    plen++;
    if(ptext[i]==ptext[i+1])
    {
        pltext[plen]='X';
        plen++;
    }
}
if(plen%2!=0)
{
    pltext[plen]='X';
    plen++;
}
System.out.println(pltext);
for(temp=0;temp<plen-1;temp=temp+2)
{
    a=pltext[temp];
    b=pltext[temp+1];
    for(i=0;i<5;i++)
    {
        for(j=0;j<5;j++)
        {
            if(mat[i][j]==a)
            {
```

```
                        r1=i;
                        c1=j;
                    }
                    if(mat[i][j]==b)
                    {
                        r2=i;
                        c2=j;
                    }
                }
            }
            if(r1==r2)
            {
                if(c1==4)
                {
                    cipher[temp]=mat[r1][0];
                    cipher[temp+1]=mat[r2][c2+1];
                }
                else if(c2==4)
                {
                    cipher[temp+1]=mat[r2][0];
                    cipher[temp]=mat[r1][c1+1];
                }
                else
                {
                    cipher[temp]=mat[r1][c1+1];
                    cipher[temp+1]=mat[r2][c2+1];
                }
            }
            else if(c1==c2)
            {
                if(r1==4)
                {
                    cipher[temp]=mat[0][c1];
                    cipher[temp+1]=mat[r2+1][c2];
                }
                else if(r2==4)
                {
                    cipher[temp]=mat[r1+1][c1];
                    cipher[temp+1]=mat[0][c2];
                }
                else
                {
                    cipher[temp]=mat[r1+1][c1];
                    cipher[temp+1]=mat[r2+1][c2];
                }
            }
            else
            {
                cipher[temp]=mat[r1][c2];
                cipher[temp+1]=mat[r2][c1];
            }
        }
        System.out.println("CIPHER TEXT: ");
        System.out.println(cipher);//DECRYPTOIN
        for(temp=0;temp<plen-1;temp=temp+2)
        {
            a=cipher[temp];
            b=cipher[temp+1];
            for(i=0;i<5;i++)
            {
                for(j=0;j<5;j++)
                {
                    if(mat[i][j]==a)
                    {
                        r1=i;
                        c1=j;
                    }
                    if(mat[i][j]==b)
                    {
                        r2=i;
                        c2=j;
                    }
                }
            }
            if(r1==r2)
```

```
{
    if(c1==0)
    {
        dcipher[temp]=mat[r1][4];
        dcipher[temp+1]=mat[r2][c2-1];
    }
    else if(c2==0)
    {
        dcipher[temp+1]=mat[r2][4];
        dcipher[temp]=mat[r1][c1-1];
    }
    else
    {
        dcipher[temp]=mat[r1][c1-1];
        dcipher[temp+1]=mat[r2][c2-1];
    }
}
else if(c1==c2)
{
    if(r1==0)
    {
        dcipher[temp]=mat[4][c1];
        dcipher[temp+1]=mat[r2-1][c2];
    }
    else if(r2==0)
```

```
                {
                    dcipher[temp]=mat[r1-1][c1];
                    dcipher[temp+1]=mat[4][c2];
                }
                else
                {
                    dcipher[temp]=mat[r1-1][c1];
                    dcipher[temp+1]=mat[r2-1][c2];
                }
            }
            else
            {
                dcipher[temp]=mat[r1][c2];
                dcipher[temp+1]=mat[r2][c1];
            }
        }
        System.out.println("DECRYPTED TEXT");
        System.out.println(dcipher);
        r1=0;
        temp=0;
        for(i=0;i<plen;i++)
        if(dcipher[i]!='X')
            dplain[temp++]=dcipher[i];
        for(i=0;i<plen;i++)
        {
            if(i==space[r1])
            {
                if(space[r1]!=0)
                System.out.print(" ");
                r1++;
           }
            j=dplain[i]+32;
            System.out.print((char)j);
        }
    }
}
```

OUTPUT:

Experiment-4

AIM: Write a program to implement transposition method

PROGRAM:

```
import java.io.*;
class trans
{
    public static void main(String args[]) throws Exception
    {
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        System.out.println("Enter the Plain Text:");
        String pt=br.readLine();
        System.out.println("Enter the depth:");
        int m = Integer.parseInt(br.readLine());
        int n=pt.length()/m;
        int i,j,index=0,buffer=0;
        System.out.println("Length of plain text:"+pt.length());
        if(pt.length() % m !=0)
            n=n+1;
        char mat[][] = new char [m][n];
        int key[]=new int[n];
        for(i=0;i<m;i++)
        {
            for(j=0;j<n;j++)
            {
                if(index<pt.length())
                {
                    mat[i][j]=pt.charAt(index);
                    index++;
                }
                else
                {
                    mat[i][j]='x';
                    buffer++;
                }
                if(index==pt.length())
                {
                    if(i!=m-1)
                        m=i+1;
                    //mat[i][j]='x';
                }

            }
        }
        for(i=0;i<m;i++)
        {
            for(j=0;j<n;j++)
            {
                System.out.print(mat[i][j]+"\t");
```

```
    }
    System.out.println();
}
System.out.println("Enter the key");
for(i=0;i<n;i++)
{
    key[i]=Integer.parseInt(br.readLine());
    if(key[i] < 1 || key [i]>n)
    {
        System.out.println("Invalid key");
        i--;
    }
}
System.out.println("Key Matrix is");
for(i=0;i<n;i++)
{
    System.out.print(key[i]+"\t");
}
System.out.println();
index=0;
String ct="";
for(i=0;i<n;i++)
{
    for(index=0;index<n;index++)
    {
        if(key[index]==i+1)
            break;
    }
    for(j=0;j<m;j++)
    {
        ct=ct+mat[j][index];
    }
}
System.out.println("Encrypted Text:"+ct);
        /* decryption  */
index=0;
char dmat[][]=new char[m][n];
String dt="";
for(i=0;i<n;i++)
{
    for(j=0;j<m;j++)
    {
        dmat[j][i]=ct.charAt(index);
        index++;
    }
}
for(i=0;i<m;i++)
{
for(j=0;j<n;j++)
    {
```

```
                                        dt=dt+dmat[i][key[j]-1];
                              }
                              //System.out.println();
                    }
                    dt=dt.substring(0,dt.length()-buffer);
                    System.out.println("Decrypted Text:"+dt);
          }
}
```

OUTPUT:

Experiment-5

AIM: Write a chatting program using TCP/IP protocol

PROGRAM:

```
//server.java

import java.net.*;

import java.io.*;

public class server

{

        public static void main(String args[])throws IOException

        {

                ServerSocket s1 = null;

                try

                {

                        s1 = new ServerSocket(98);

                }
                catch(IOException u1)
                {
                        System.err.println("Could not found port 98");
                        System.exit(1);
                }
                Socket c = null;
                try
                {
                        c = s1.accept();
                        System.out.println("Connection from" + c);
                }
                catch(IOException e)
                {
                        System.out.println("accept failed");
                        System.exit(1);
                }
                PrintWriter out = new PrintWriter(c.getOutputStream(),true);
                BufferedReader in = new BufferedReader(new InputStreamReader(c.getInputStream()));
                String l;
                BufferedReader sin = new BufferedReader(new InputStreamReader(System.in));
                System.out.println("I m ready type now");
                while((l = sin.readLine())!= null)
                {
                        out.println( l );
                }
```

```
                out.close();
                sin.close();
                c.close();
                s1.close();
        }
}

//client.java
import java.net*;
import java.io*;
public class client
{
        public static void main(String args[])throws IOException
        {
                Socket s = null;
                BufferedReader b = null;
                try
                {
                        s = new Socket(InetAddress.getLocalHost(),98);
                //If you want to connect another machine in the network,specify the IP Address
                // s = new Socket(IP address of anoher machine,98);
                b = new BufferedReader(new InputStreamReader (s.getInputStream()));
                }
                catch(UnknownHostException u)
                {
                        System.err.println("I don't know host");

                        System.exit(1);

                }

                String inp;

                while((inp = b.readLine())!= null)

                {

                        System.out.println(inp);

                }

                b.close();

                s.close();

        }

}
```

OUTPUT:

Experiment-6

AIM: Write a chatting program using UDP/IP protocol

PROGRAM:

```
//server.java

import java.net.*;

import java.io.*;

class server

{

        public static DatagramSocket ds;

        public static byte buffer[]= new byte[1024];

        public static void Myserver() throws Exception

        {
```

```
        int pos=0;
        while(true)
        {
            int c = System.in.read();
            switch(c)
            {
                case 1: System.out.println("Server quits");
                return;
                case '\r': break;
                case '\n': ds.send(new
DatagramPacket(buffer,pos,InetAddress.getLocalHost(),777));
                pos = 0;
                break;
                default: buffer[pos++] = (byte)c;
            }

        }
    }
    public static void main(String args[]) throws Exception
    {
        System.out.println("Server ready.. \n please type here");
        ds = new DatagramSocket (888);
        Myserver();
    }
}
//client.java
```

```
import java.net.*;

import java.io.*;

class client

{

        public static DatagramSocket ds;

        public static byte buffer[] = new byte[1024];

        public static void MyClient() throws Exception

        {


                while(true)

                {

                        DatagramPacket p = new DatagramPacket(buffer,buffer.length);

                        ds.receive(p);

                        System.out.println(new String(p.getData(),0,p.getLength()));

                }


        }

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

        {

                System.out.println("Client-For quit press Ctrl+C");

                ds = new DatagramSocket(777);

                MyClient();

        }

}
```

OUTPUT