import java.awt.\*;

import java.io.\*;

import java.awt.event.\*;

import javax.swing.JOptionPane.\*;

import javax.swing.\*;

import java.applet.\*;

import java.awt.image.\*;

public class ste12 extends Frame implements ActionListener

{

static String path;

static String path1;

FileDialog fd,fd1;

public Image img=null;

public Image img1=null;

public Image img3=null;

static ste12 f;

//public static bmp b;

int index=0;

int max;

int bytes[]=new int[70000];

int binary[]=new int[2450000];

int maxbinary,maxbinary1,maxbinary2,ivalue;

String msg,msg1=null;

String filename,filename1;

int flag=0;

public int pixels[]=new int[350000];

public int pixels1[]=new int[350000];

static int width,height,padding;

static int index1=0,ch,i=0,l=0,r,g,b,r1,g1,b1;

int p[]=new int[2000000];

int p1[]=new int[2000000];

int p2[]=new int[2000000];

int p3[]=new int[2000000];

int pix[]=new int[350000];

int pix1[]=new int[350000];

int blue[]=new int[300];

int green[]=new int[300];

int red[]=new int[300];

int maxp;

int maxp1;

// static TextArea ta;

TextArea ta=new TextArea(10,100);

MenuBar mb=new MenuBar();

Menu m1=new Menu("File ");

MenuItem mi1=new MenuItem("Open");

MenuItem mi2=new MenuItem("OpenTextFile");

MenuItem mi3=new MenuItem("Save");

MenuItem mi4=new MenuItem("Decrypt");

MenuItem mi5=new MenuItem("Reset");

MenuItem mi6=new MenuItem("Exit");

ste12()

{

setMenuBar(mb);

m1.add(mi1);

mi1.setEnabled(false);

m1.add(mi2);

m1.add(mi3);

mi3.setEnabled(false);

m1.add(mi4);

m1.add(mi5);

m1.add(mi6);

mb.add(m1);

mi1.addActionListener(this);

mi2.addActionListener(this);

mi3.addActionListener(this);

mi4.addActionListener(this);

mi5.addActionListener(this);

mi6.addActionListener(this);

setLayout(new BorderLayout( ));

add(ta,BorderLayout.SOUTH);

addWindowListener(new WindowAdapter()

{

public void WindowClosing(WindowEvent e)

{

System.exit(0);

}

});

}

public static void main(String args[])

{

f=new ste12();

f.setSize(800,800);

f.setTitle("STEGANOGRAPHY ");

f.show();

}

public void actionPerformed(ActionEvent ae)

{

String s=ae.getActionCommand();

if(s.equals("Open"))

{

msg="Encrypted Image";

flag=0;

fd=new FileDialog(f,"Opening Files......");

fd.show();

path=fd.getDirectory()+fd.getFile();

if(path!=null)

{

msg1="original Image";

mi1.setEnabled(false);

mi3.setEnabled(true);

openimagefile(path,11);

}

}

else if(s.equals("OpenTextFile"))

{

fd1=new FileDialog(f,"Opening Files......");

fd1.show();

path1=fd1.getDirectory()+fd1.getFile();

if(path1!=null)

{

mi1.setEnabled(true);

mi2.setEnabled(false);

opentextfile(path1);

ta.setEditable(false);

}

}

else if(s.equals("Save"))

{

flag=0;

FileDialog fd2=new FileDialog(f,"save file",FileDialog.SAVE);

fd2.show();

try

{

if(fd2.getFile()!=null)

{

filename=fd2.getDirectory()+fd2.getFile();

mi3.setEnabled(false);

FileOutputStream fout=new FileOutputStream(filename);

DataOutputStream d=new DataOutputStream(fout);

i=0;

while(i<maxp)

{

d.write(p1[i]);

i++;

}

fout.close();

repaint();

}

}

catch(Exception e)

{

System.out.println(e+"kr");

}

}

else if(s.equals("Decrypt"))

{

ta.setText(" ");

FileDialog fd3=new FileDialog(f,"OPEN file");

fd3.show();

msg="Decrypted Image";

try

{

if(fd3.getFile()!=null)

{

filename1=fd3.getDirectory()+fd3.getFile();

i=0;

FileInputStream fis1 = new FileInputStream(filename1);

DataInputStream dis=new DataInputStream(fis1);

while((ch=dis.readUnsignedByte())!=-1)

{

p2[i]=ch;

p3[i]=ch;

i++;

}

}

}

catch(Exception e)

{

// System.out.println(e);

maxp1=i;

switch(p2[28])

{

case 24:

{

// if(sno==11)

init241();

} //case 24 ends

} // switchp[28] ends

} //catch() ends

img3=createImage(new MemoryImageSource(width,height,pixels,0,width));

img=createImage(new MemoryImageSource(width,height,pixels,0,width));

flag=1;

repaint();

}

else if(s.equals("Reset"))

{

img=null;

img1=null;

img3=null;

msg1=null;

msg=null;

mi2.setEnabled(true);

mi1.setEnabled(false);

mi3.setEnabled(false);

repaint();

}

else if(s.equals("Exit"))

{

System.exit(0);

}

}

public void init241()

{

width= p2[21]<<24 | p2[20]<<16 | p2[19]<<8 | p2[18];

height= p2[25]<<24 | p2[24]<<16 | p2[23]<<8 | p2[22];

//binary=new int[400000];

int extra=(width\*3)%4;

if(extra!=0)

padding=4-extra;

int x,z=54;

l=0;

int j=0;

i=0;

for(int q=0;q<height;q++)

{

x=0;

while(x<width)

{

b=p2[z]&0xff;

binary[j++]=b&0x01;

g=p2[z+1]&0xff;

binary[j++]=g&0x01;

r=p2[z+2]&0xff;

binary[j++]=r&0x01;

pix[l]= 255<<24 | r<<16 | g<<8 | b;

z=z+3;

x++;

l++;

}

z=z+padding;

}

int k;

x=0;

stringcon();

for(i=l-width;i>=0;i=i-width) //l=WIDTH \* height

{

for(k=0;k<width;k++)

{

pixels[x]=pix[i+k];

// pixels1[x]=pix[i+k];

x++;

}

}

} //init241 ends

public void stringcon()

{

int i,j,k;

int temp[]=new int[8];

int a[]=new int[32];

i=0;

j=0;

for(i=0;i<10000;i++)

bytes[i]=0;

i=0;

maxbinary1=0;

for(i=0;i<24;i++)

a[i]=binary[i];

for(i=0;i<24;i++)

maxbinary1+=a[i]\*Math.pow(2,23-i);

maxbinary2=maxbinary1\*8;

ivalue=0;

for(i=24,j=0;i<32;i++,j++)

{

a[j]=binary[i];

ivalue+=a[j]\*Math.pow(2,7-j);

}

if(ivalue==73)

{

i=32;

while(i<maxbinary2)

{

for(k=0;k<=7;k++)

temp[k]=binary[i++];

for(k=0;k<=7;k++)

bytes[j]+=temp[k]\*Math.pow(2,7-k);

j++;

}

String s=JOptionPane.showInputDialog(this,"Enter key with 16 letters");

char ch[]=new char[s.length()];

ch=s.toCharArray();

try

{

FileOutputStream f=new FileOutputStream("key1.txt");

for(i=0;i<ch.length;i++)

f.write(ch[i]);

f.close();

FileOutputStream fout=new FileOutputStream("enc1.txt");

DataOutputStream d=new DataOutputStream(fout);

i=8;

while(i<(maxbinary1))

{

d.write(bytes[i]);

i++;

}

fout.close();

}

catch(Exception e)

{

System.out.println(e+"eneorgouegoieg");

}

ideaalgo b=new ideaalgo();

b.procedure();

b.decryption("enc1.txt");

try

{

BufferedReader d;

StringBuffer sb=new StringBuffer();

d=new BufferedReader(new FileReader("dec.txt"));

String line;

while((line=d.readLine())!=null)

sb.append(line+"\n");

ta.setText(sb.toString());

d.close();

}

catch(Exception e)

{

System.out.println(e);

}

}

}

public void opentextfile(String path1)

{

FileInputStream fin,fin1;

int i,j=0;

String s=JOptionPane.showInputDialog(this,"Enter key with 16 letters");

char ch[]=new char[s.length()];

ch=s.toCharArray();

try

{

FileOutputStream f=new FileOutputStream("key1.txt");

for(i=0;i<ch.length;i++)

f.write(ch[i]);

f.close();

ideaalgo a=new ideaalgo(path1);

a.procedure();

a.encrypt();

BufferedReader d;

StringBuffer sb=new StringBuffer();

d=new BufferedReader(new FileReader(path1));

String line;

while((line=d.readLine())!=null)

sb.append(line+"\n");

ta.setText(sb.toString());

d.close();

fin=new FileInputStream("enc.txt");

do

{

i=fin.read();

if(i!=-1)

{

bytes[j++]=i;

}

}

while(i!=-1);

max=j;

fin.close();

binarycon();

}

catch(Exception e)

{

System.out.println(e+"error");

}

}

public void binarycon()

{

int i,j,k,t;

int temp[]=new int[10];

int m=0;

for(i=0;i<600000;i++)

binary[i]=0;

int b[]=new int[32];

int dum;

dum=max;

i=0;

while(dum!=0)

{

b[i]=dum%2;

i=i+1;

dum/=2;

}

j=24-i;

for(k=j,t=i-1;k<(i+j);k++,t--)

binary[k]=b[t];

dum=73;

i=0;

while(dum!=0)

{

b[i]=dum%2;

i=i+1;

dum/=2;

}

j=32-i;

for(k=j,t=i-1;k<32;k++,t--)

binary[k]= b[t];

m=32;

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

{

j=0;

while( bytes[i]!= 0 )

{

temp[j++]=bytes[i]%2;

bytes[i]=bytes[i]/2;

}

for( k=0;k<8-j ; k++)

binary[m++]=0;

for(k=j-1; k >=0 ; k--)

binary[m++]=temp[k];

}

maxbinary=m;

}

public void paint(Graphics g)

{

if(msg1!=null)

g.drawString (msg1,180,60);

if(msg!=null)

g.drawString (msg,550,60);

if(img!=null)

g.drawImage(img,20,80,this);

if(img1!=null&&flag!=1)

g.drawImage(img1,400,80,this);

if(img3!=null&&flag!=0)

g.drawImage(img3,400,80,this);

}

public void openimagefile(String fn,int sno)

{

try

{

i=0;

FileInputStream fis = new FileInputStream(fn);

DataInputStream dis=new DataInputStream(fis);

while((ch=dis.readUnsignedByte())!=-1)

{

p[i]=ch;

p1[i]=ch;

i++;

}

fis.close();

dis.close();

}

catch(Exception e)

{

maxp=i;

switch(p[28])

{

case 24:

{

if(sno==11)

init24();

break;

} //case 24 ends

} // switchp[28] ends

} //catch() ends

img=createImage(new MemoryImageSource(width,height,pixels,0,width));

img1=createImage(new MemoryImageSource(width,height,pixels1,0,width));

} //openfile ends

public void init24()

{

width= p[21]<<24 | p[20]<<16 | p[19]<<8 | p[18];

height= p[25]<<24 | p[24]<<16 | p[23]<<8 | p[22];

int extra=(width\*3)%4;

if(extra!=0)

padding=4-extra;

int x,z=54;

l=0;

int j=0;

for(int q=0;q<height;q++)

{

x=0;

while(x<width)

{

b=p[z]&0xff;

if(j<maxbinary)

{

if(binary[j]!=0)

{

p1[z]=p[z]&0xff|binary[j++];

b1=p1[z]&0xff;

}

else

{

p1[z]=p[z]&0xff & (binary[j++]|0xfe);

b1=p1[z]&0xff;

}

}

else

b1=p[z]&0xff;

g=p[z+1]&0xff;

if(j<maxbinary)

{

if(binary[j]!=0)

{

p1[z+1]=p[z+1]&0xff|binary[j++];

g1=p[z+1]&0xff;

}

else

{

p1[z+1]=p[z+1]&0xff & (binary[j++]|0xfe);

g1=p1[z+1]&0xff;

}

}

else

g1=p[z]&0xff;

r=p[z+2]&0xff;

if(j<maxbinary)

{

if(binary[j]!=0)

{

p1[z+2]=p[z+2]&0xfe|binary[j++];

r1=p[z+2]&0xff;

}

else

{

p1[z+2]=p[z+2]&0xff & (binary[j++]|0xfe);

r1=p1[z+2]&0xff;

}

}

else

r1=p[z]&0xff;

z=z+3;

pix[l]= 255<<24 | r<<16 | g<<8 | b;

pix1[l]= 255<<24 | r1<<16 | g1<<8 | b1;

l++;

x++;

}

z=z+padding;

}

int k;

x=0;

for(i=l-width;i>=0;i=i-width) //l=WIDTH \* height

{

for(k=0;k<width;k++)

{

pixels[x]=pix[i+k];

pixels1[x]=pix[i+k];

x++;

}

}

} //init24 ends

}

class ideaalgo

{

FileInputStream fin,fkey,fenc1;

DataOutputStream fdec,fenc;

int step1,step2,step3,step4,step5,step6,step7,step8,step9,step10,step11,step12,step13,step14,t;

int index=0,j1,i1,i,mark,k1,k2,k;

int iz[]=new int[52];

int size;

byte buf[],keybuf[];

int ft,ft1,nl,np=0,p;

int x[]=new int[5];

int z[]=new int[52];

byte buf1[];

String file2,file3,file4;

ideaalgo()

{

file2="key1.txt";

}

ideaalgo(String file1)

{

// file1="a.txt";

file2="key1.txt";

file3="enc.txt";

file4="dec.txt";

try

{

fin=new FileInputStream(file1);

fenc1=new FileInputStream(file3);

fenc=new DataOutputStream(new FileOutputStream(file3));

}

catch(Exception e)

{

System.out.println(e);

}

}

void procedure()

{

try

{

fkey=new FileInputStream(file2);

}

catch(Exception e)

{

System.out.println(e+"krishna");

}

//genereating the keys

keybuf= new byte[16];

try

{

fkey.read(keybuf);

z= new int[52];

j1=0;

i1=0;

for(i=0;i<52;i++)

z[i]=0;

while( i1<8)

{

if((j1+1)%2==0)

{

z[i1]|=keybuf[j1]; // dividing 64 bit cypher block into four 16 bit registers

i1++;

}

else

{

z[i1]=keybuf[j1];

z[i1]<<=8;

}

j1++;

}

i=0;

for(j1=1;j1<=5;j1++)

{

i++;

z[i+7]=((z[i]<<9)&0xfe00)|((z[i+1]>>7)&0x1ff);

z[i+8]=((z[i+1]<<9)&0xfe00)|((z[i+2]>>7)&0x1ff);

z[i+9]=((z[i+2]<<9)&0xfe00)|((z[i+3]>>7)&0x1ff);

z[i+10]=((z[i+3]<<9)&0xfe00)|((z[i+4]>>7)&0x1ff);

z[i+11]=((z[i+4]<<9)&0xfe00)|((z[i+5]>>7)&0x1ff);

z[i+12]=((z[i+5]<<9)&0xfe00)|((z[i+6]>>7)&0x1ff);

z[i+13]=((z[i+6]<<9)&0xfe00)|((z[i-1]>>7)&0x1ff);

z[i+14]=((z[i-1]<<9)&0xfe00)|((z[i]>>7)&0x1ff);

i=i+7;

}

i1=41;

z[48]=((z[i1]<<9)&0xfe00)|((z[i1+1]>>7)&0x1ff);

z[49]=((z[i1+1]<<9)&0xfe00)|((z[i1+2]>>7)&0x1ff);

z[50]=((z[i1+2]<<9)&0xfe00)|((z[i1+3]>>7)&0x1ff);

z[51]=((z[i1+3]<<9)&0xfe00)|((z[i1+4]>>7)&0x1ff);

}

catch(Exception e)

{

System.out.println(e+"ravi");

}

}

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Encryption\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void encrypt()

{

try

{

size=fin.available();

p=size%8;

if(p!=0)

np=8-p;

size+=np;

if(p==0)

nl=size/8;

else

nl=size/8+1;

buf=new byte[8];

buf1=new byte[size+10];

fin.read(buf1);

int enc[]=new int[size];

mark=-8;

k2=0;

for(k=0;k<nl;k++)

{

mark+=8;

for(int k1=0;k1<8;k1++)

buf[k1]=buf1[mark+k1];

i=0;

for(i=0;i<4;i++)

x[i]=0;

j1=0;i1=0;

while( i1<=3)

{

if((j1+1)%2==0)

{

x[i1]|=buf[j1]; // dividing 64 bit cypher block into four 16 bit registers

i1++;

}

else

{

x[i1]=buf[j1];

x[i1]<<=8;

}

j1++;

}

// 7 rounds and 14 steps

for(i=0 ; i <48 ; )

{

step1=mul(x[0],z[i+0]);

step2=(x[1]+z[i+1])%65536;

step3=(x[2]+z[i+2])%65536;

step4=mul(x[3],z[i+3]);

step5=step1^step3;

step6=step2^step4;

step7=mul(step5,z[i+4]);

step8=(step6+step7)%65536;

step9=mul(step8,z[i+5]);

step10=(step7+step9)%65536;

step11=step1^step9;

step12=step3^step9;

step13=step2^step10;

step14=step4^step10;

x[0]=step11;

x[1]=step13;

x[2]=step12;

x[3]=step14;

i=i+6;

}

x[0]=mul(x[0],z[48]);

x[1]=(x[1]+z[49])%65536;

x[2]=(x[2]+z[50])%65536;

x[3]=mul(x[3],z[51]);

for(int t=0;t<4;t++)

{

ft1 =x[t]&255;

ft=x[t]>>8;

fenc.write((char)ft);

fenc.write((char)ft1);

}

}

fin.close();

}

catch(Exception e)

{

System.out.println(e);

}

}

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Decryption \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* //

void decryption(String file)

{

try

{

fdec=new DataOutputStream(new FileOutputStream("dec.txt"));

fenc1=new FileInputStream(file);

}

catch(Exception e)

{

System.out.println(e+"ravikrishna");

}

try

{

size=fenc1.available();

np=0;

p=size%8;

if(p!=0)

np=8-p;

size+=np;

if(p==0)

nl=size/8;

else

nl=size/8+1;

buf1=new byte[size+10];

buf=new byte[8];

fenc1.read(buf1);

mark=-8;

int arr[]=new int [8];

for(k1=0;k1<nl;k1++)

{

mark+=8;

for(int k2=0;k2<8;k2++)

buf[k2]=buf1[mark+k2];

for(int k2=0;k2<8;k2++)

{

arr[k2]=0;

if(buf[k2]>=0)

arr[k2]=buf[k2];

else

arr[k2]=buf[k2]+256;

}

j1=0;i1=0;

while( i1<=3)

{

if((j1+1)%2==0)

{

x[i1]|=arr[j1]; // dividing 64 bit cypher block into four 16 bit registers

i1++;

}

else

{

x[i1]=arr[j1];

x[i1]<<=8;

}

j1++;

}

for(int t=0;t<4;t++)

{

ft1 =x[t]&255;

ft=x[t]>>8;

}

step1=mul( x[0] , minverse( z[48] ));

step2=( x[1] + ainverse( z[49] )) % 65536;

step3=( x[2] + ainverse( z[50] )) % 65536;

step4=mul( x[3] , minverse( z[51] ));

step5=step1^step3;

step6=step2^step4;

step7=mul(step5,z[46]);

step8=(step6+step7)%65536;

step9=mul(step8,z[47]);

step10=(step7+step9)%65536;

step11=step1^step9;

step12=step3^step9;

step13=step2^step10;

step14=step4^step10;

x[0]=step11;

x[1]=step12;

x[2]=step13;

x[3]=step14;

// 2nd round

int j2=40;

for(j1=1;j1<=7;j1++)

{

step1=mul( x[0] , minverse( z[j2+2] ));

step2=( x[1] + ainverse( z[j2+4] )) % 65536;

step3=( x[2] + ainverse( z[j2+3] )) % 65536;

t=step2;

step2=step3;

step3=t;

step4=mul( x[3] , minverse( z[j2+5] ));

step5=step1^step3;

step6=step2^step4;

step7=mul(step5,z[j2+0]);

step8=(step6+step7)%65536;

step9=mul(step8,z[j2+1]);

step10=(step7+step9)%65536;

step11=step1^step9;

step12=step3^step9;

step13=step2^step10;

step14=step4^step10;

x[0]=step11;

x[1]=step12;

x[2]=step13;

x[3]=step14;

j2=j2-6;

}

x[0]=mul(x[0],minverse(z[0]));

x[1]=(x[1]+ainverse(z[2]))%65536;

x[2]=(x[2]+ainverse(z[1]))%65536;

x[3]=mul(x[3],minverse(z[3]));

t=x[1];

x[1]=x[2];

x[2]=t;

for(int t=0;t<4;t++)

{

ft1 =x[t]&255;

ft=x[t]>>8;

fdec.write((char)ft);

fdec.write((char)ft1);

}

}

}

catch(Exception e)

{

System.out.println(e+"ergeg");

}

}

int mul( int a , int b)

{

double c,d;

if (a==0)

c=65536;

if(b==0)

d=65536;

c=(double)a;

d=(double)b;

a= (int)((c\*d)%65537);

return a;

}

int minverse(int z)

{

int to,t1;

int q,y;

if(z<=1)

return z;

t1=0x10001/z;

y=0x10001%z;

if(y==1)

return (0xffff&(1-t1));

to=1;

do

{

q=z/y;

z=z%y;

to+=q\*t1;

if(z==1)

return to;

q=y/z;

y=y%z;

t1+=q\*to;

}

while(y!=1);

return (0xffff&(1-t1));

}

int ainverse(int z)

{

return (65536-z);

}

}