```
import java.io.*;
import java.util.ArrayList;
/*   *   * 倪牯踽 震镫镢轶酤?I   *   * Experimental Virtual Lab   *   * Java
virtual modem communications seed code   *   */
public class virtualModem {

  public static void main(String[] param) {
            virtualModem  vM = new virtualModem();
            int [] y= new int[4];
            int k;
            Modem modem;
            modem=new Modem();
            modem.setSpeed(80000);
            modem.setTimeout(2000);
            modem.open("ithaki");
            for (int i=0; i<205; i++) {
                try {
                        k=modem.read();
                        if(k==-1) break;

                System.out.print((char)k);
                y[i%4]=k;
                if(y[1]==10 && y[0]==10 && y[3]==10 && y[2]==13)break;
                }
                catch (Exception x)
                {
                        break;
                }
            }


  vM.EchoPackets(param[0],modem);
  System.out.println("\n");
  vM.ImageClear(param[1],modem);
  System.out.println("\n");
  vM.ImageError(param[2],modem);
  System.out.println("\n");
  vM.GpsTracking(param[3],modem);
  System.out.println("\n");
  vM.Errors(param[4],param[5],modem);
  System.out.print("\n");


  modem.close();
}
//EchoPackets
public  void EchoPackets (String packetcode,Modem modem) {
int k;
FileWriter writer;
final long NANOSEC_PER_SEC = 1000*1000*1000;
long startTime=System.nanoTime();
long t;
int dif;
String mess="";
try {
writer= new FileWriter("C:\\Users\\誊耖矧\\Desktop\\6o 孱犰珥颸\拈牯踽 震镫镢轶
```

```
豉?蓀\G1.txt");
while ((System.nanoTime()-startTime)<5*60*NANOSEC_PER_SEC) {
mess="";
t=System.currentTimeMillis();
modem.write((packetcode+"\r").getBytes());
         for(;;) {
                k=modem.read();
                if(k==-1) break;
                mess=mess+((char)k);
        if(mess.endsWith("PSTOP")) break;
        }
        dif=(int)(System.currentTimeMillis()-t);

        writer.write(String.valueOf(dif)+"  ");
        //System.out.println( "\n "+ mess + "\n");
        }
  writer.close();

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

}
//Image_Clear_Photo
public void ImageClear(String ImageClearCode,Modem modem) {
 int k;
 byte b;
 boolean Byteflag=false;
 FileOutputStream out;
 Integer value;
 String str="";
 ArrayList<Integer> Bytes=new ArrayList<Integer>();
 try {
        modem.write((ImageClearCode+"\r").getBytes());
        out= new FileOutputStream("C:\\Users\\誊耖矧\\Desktop\\6o 孱犰珥颸\拈牯踽 震镫镢轶豉?蓀\Image_Without_Errors.jpg");
        for(;;) {
                k=modem.read();
                str=Integer.toString(k);
                value=new Integer(str);
                b=value.byteValue();
                //System.out.println(value);
                Bytes.add(value);
                if(Bytes.size()>=2) {
                if(Bytes.get(0)==0xFF && Bytes.get(1)==0xD8 && Byteflag==false)
                {
                        //System.out.println("GOOD Start");
                        out.write(Bytes.get(0).byteValue());
                        out.write(Bytes.get(1).byteValue());
                        Byteflag=true;


                }
                else if(Byteflag) {
                        out.write(b);
```

```
                        if(Bytes.get(Bytes.size()-1)==0xD9 &&
Bytes.get(Bytes.size()-2)==0xFF)
                {

                        //System.out.println("GOOD BREAKING");
                        break;
                }

                                                }
                                                        }
                if (k==-1) break;
                        }
        out.close();
 }

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

//Image_Error_Photo
public void ImageError(String ImageErrorCode,Modem modem) {
int k;
byte b;
boolean Byteflag=false;
FileOutputStream out;
Integer value;
String str="";
ArrayList<Integer> Bytes=new ArrayList<Integer>();
try {
        modem.write((ImageErrorCode+"\r").getBytes());
        out= new FileOutputStream("C:\\Users\\誊耖矧\\Desktop\\6o 孱犰珥颸\拈牯踽 震镫镢轶豉?蓀\Image_With_Errors.jpg");
        for(;;) {
                k=modem.read();
                str=Integer.toString(k);
                value=new Integer(str);
                b=value.byteValue();
                //System.out.println(value);
                Bytes.add(value);
                if(Bytes.size()>=2) {
                if(Bytes.get(0)==0xFF && Bytes.get(1)==0xD8 && Byteflag==false)
                {
                        //System.out.println("GOOD Start");
                        out.write(Bytes.get(0).byteValue());
                        out.write(Bytes.get(1).byteValue());
                        Byteflag=true;


                }
                else if(Byteflag) {
                        out.write(b);
                        if(Bytes.get(Bytes.size()-1)==0xD9 &&
Bytes.get(Bytes.size()-2)==0xFF)
                {
```

```
                        0000-972a-169f-4ae1-d7b.txt
                        //System.out.println("GOOD BREAKING");
                        break;
                }

                                                }
                                                        }
                if (k==-1) break;
                        }
        out.close();
}

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

}

//GPS_Tracking
public void GpsTracking(String GpsCode,Modem modem) {
        int k;
  modem.write((GpsCode+"R=1003299\r").getBytes());
  int counter=0;
  int c=0;
  int count2=0;
  String gps_data="";
  String [] Coordinates = new String [99];
  String [] ArrayTimeOfCoordinates= new String[99];
  String  Datata="";
  final double mult=0.006;

  for(;;) {
                        k=modem.read();
                        Datata+=String.valueOf((char)k);
                        //System.out.println(Datata);
                        if(Datata.startsWith("START ITHAKI GPS TRACKING\r\n")) {
                                gps_data+=String.valueOf((char)k);
                                //System.out.println(gps_data);
                                if(gps_data.endsWith("0000*")) {
                                                count2=0;
                                                //System.out.println(gps_data);
                                                String[]
parts=gps_data.split(",");
                                                String[]
SplittedParts=parts[1].split("\\.");
                                                String [] SplittedParts2=
parts[2].split("\\.");
                                                int multiplier1=
Integer.parseInt(SplittedParts2[1]);
                                                String
str1=Integer.toString((int)(multiplier1*mult));
                                                str1=SplittedParts2[0]+str1;
                                                String[] SplittedParts3=
parts[4].split("\\.");
                                                int
multiplier2=Integer.parseInt(SplittedParts3[1]);
                                                String
```

```
str2=Integer.toString((int)(multiplier2*mult));

str2=SplittedParts3[0].substring(1,5)+str2;
                                                    if((c==0)) {

Coordinates[0]=str2+str1;

ArrayTimeOfCoordinates[0]=SplittedParts[0];



}
                                                    else if(c!=0)  {
                                                            int
time_packet1=Integer.parseInt(SplittedParts[0]);
                                                            int
time_packet2=Integer.parseInt(ArrayTimeOfCoordinates[counter]);


if((time_packet1-time_packet2)>15) {

counter++;

Coordinates[counter]=str2+str1;

ArrayTimeOfCoordinates[counter]=SplittedParts[0];


                                                }


}


                                                    c++;
                                                        gps_data="";

                }
                                                }
                                if(count2==2)  {
                                    gps_data="";
                                    }
                                count2++;
                                    if(Datata.endsWith("STOP ITHAKI GPS
TRACKING\r\n"))  {
                                            //System.out.println("GOOD BREAK");
                                            break;

                                                }
                                    if(k==-1) break;



                }
          // System.out.println(Datata);
```

```
  String Message="";
  for(int i=0;i<counter;i++) {

     Message+="T="+Coordinates[i];

  }

  byte b;
  boolean Byteflag=false;
  FileOutputStream out;
  Integer value;
  String str="";
  ArrayList<Integer> Bytes=new ArrayList<Integer>();
  try {
        modem.write((GpsCode+Message+"\r").getBytes());
        out= new FileOutputStream("C:\\Users\\誊耖矧\\Desktop\\6o 孱犰珥颸\拈牯踽 震镫镢轶豉?蓀\GPS_IMAGE.jpg");
        for(;;) {
                k=modem.read();
                str=Integer.toString(k);
                value=new Integer(str);
                b=value.byteValue();
                //System.out.println(value);
                Bytes.add(value);
                if(Bytes.size()>=2) {
                if(Bytes.get(0)==0xFF && Bytes.get(1)==0xD8 && Byteflag==false)
                {
                        //System.out.println("GOOD Start");
                        out.write(Bytes.get(0).byteValue());
                        out.write(Bytes.get(1).byteValue());
                        Byteflag=true;


                }
                else if(Byteflag) {
                        out.write(b);
                        if(Bytes.get(Bytes.size()-1)==0xD9 &&
Bytes.get(Bytes.size()-2)==0xFF)
                {

                        //System.out.println("GOOD BREAKING");
                        break;
                }

                                                }
                                                        }
                if (k==-1) break;
                        }
        out.close();
  }


  catch(IOException e){
                System.out.println(e);
        }
```

```
}
//Errors
public void Errors (String Ackcode,String Nackcode,Modem modem) {
          int k;
          final long NANOSEC_PER_SEC = 1000*1000*1000;
          long startTime1 = System.nanoTime();
          long t1;
          int dif1;
          int error_num=0;
          int XOR;
          boolean flag=true;
          String message="";
          FileWriter writer1;
          FileWriter errorwriter;
          try {
                    writer1= new FileWriter("C:\\Users\\誊耖矧\\Desktop\\6o 孱犰珲颸\拈牯踽 震镫镢轶豉?蓀\G2.txt");
                    errorwriter=new FileWriter("C:\\Users\\誊耖矧\\Desktop\\6o 孱犰珲颸\拈牯踽 震镫镢轶豉?蓀\G3.txt");
                    while ((System.nanoTime()-startTime1)< 5*60*NANOSEC_PER_SEC){
                              t1=System.currentTimeMillis();
                                        if(flag){

modem.write((Ackcode+"\r").getBytes());
                                                  }
                                        else {

modem.write((Nackcode+"\r").getBytes());
                                                  error_num++;
                                         }
                                        for(;;) {
                                        k=modem.read();
                                        if(k==-1) break;
                                        //System.out.println((char)k);
                                        message+=(char)k;
                                                  if(message.endsWith("PSTOP")) {
                                                            String []
parts=message.split(" ");
                                                                      int
fcs=Integer.parseInt(parts[5]);

XOR=parts[4].charAt(1)^parts[4].charAt(2);
                                                                      for(int
i=3;i<=16;i++){

XOR=XOR^parts[4].charAt(i);
                                                                      }
                                                                      if(XOR==fcs){

flag=true;
                                                                      }
                                                                      else
if(XOR!=fcs){

flag=false;
                                                                      }
```

```
                                                                    if(flag) {
dif1=(int)(System.currentTimeMillis()-t1);
writer1.write(String.valueOf(dif1)+"  ");
errorwriter.write(String.valueOf(error_num)+"  ");
error_num=0;
}
                                                                    message="";
                                                                    break;
}
                                                  }
                    }
                                                writer1.close();
                                                errorwriter.close();
          }
          catch(IOException e){
                        System.out.println(e);
                }
  }
  }
```