1. Korisnik je zadao komandu
gpg –sear mi16888 /backup/dat1

a) Koje su posledice zadavanja komande?
b) Šta se sve zahteva od korisnika koji je zadao komandu pre i nakon zadavanja komande?
c) Da li biste tom korisniku preporučili da koristi usluge sa site-a
https://pgp.mit.edu/

Odgovor obrazložiti.

d) Isto pitanje kao prethodno, ali vezano za URL
https://keyserver.pgp.com

ODGOVOR
a) Komanda šifruje datoteku /backup/dat1 javnim kljucem korisnika mi16888 i potpisuje šifrat privatnim ključem korisnika mi16888.

b) Od korisnika koji zadaje komandu se zahteva da unese lozinku svog privatnog ključa koji je prethodno generisao.

c) Da, jer tako korisnik može izvesti svoj javni ključ na bezbedan server javnih ključeva. Takođe s tih servera, korisnik koji je zadao ovu komandu može preuzeti javne ključeve drugih osoba s kojima želi da komunicira.
d Da. Iako su mnogi bili skeptični nakon što je Symantec acquired PGP. Takođe s tih servera, korisnik koji je zadao ovu komandu može preuzeti javne ključeve drugih osoba s kojima želi da komunicira.

2. Napisati mrežni program koji svim računarima u salama Jagić 1 i Jagić 2 šalje veći datagram paket u formi tekstualne poruke:
Домовина се брани лепотом, и чашћу и знањем. Домовина се брани животом и лепим васпитањем. Домовина се брани цветом, И пчелом на цвету, Маком и сунцокретом,И птицом у лету.
a) Pažljivo osmisliti format u kome se nalaze podaci koji se šalju i precizno definisati način na koji pakujete datagram tokom slanja. Taj deo koda jasno iskomentarisati! Voditi računa o ispisu dovoljno deskriptivnih poruka o greškama koje mogu da nastanu IOException? SocketException? SecurityException? …).
b) Metod koji koristite u ovoj aplikaciji bi mogao da koristi DHCP protokol. Objasnite.

RESURS:

Harlod, „Java Network Programming“, chapter 12 „UDP“

The SO_BROADCAST option controls whether a socket is allowed to send packets to and receive packets from broadcast addresses such as 192.168.254.255, the local network broadcast address for the network with the local address 192.168.254**.*.**
UDP broadcasting is often used for protocols such as DHCP that need to communicate with servers on the local net whose addresses are not known in advance. This option is controlled with these two methods:
public void setBroadcast(**boolean** on) **throws** SocketException

public boolean getBroadcast() **throws** SocketException
Routers and gateways do not normally forward broadcast messages, but they can still kick up a lot of traffic on the local network. This option is turned on by default, but if you like you can disable it thusly:
socket.setBroadcast(**false**);
This option can be changed after the socket has been bound. On some implementations, sockets bound to a specific address do not receive broadcast packets. In other words, you should use the Data gramPacket(int port) constructor, not the DatagramPacket(InetAd dress address, int port) constructor to listen to broadcasts. This is necessary in addition to setting the SO_BROADCAST option to true.

```
import java.io.IOException;
import java.io.UnsupportedEncodingException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.net.SocketException;
import java.net.UnknownHostException;

public class Zadatak2 {

        public static void main(String[] args) throws SocketException {

                String s = "Домовина се брани лепотом, и чашћу и знањем. Домовина се
брани животом и лепим васпитањем. Домовина се брани цветом, И пчелом на цвету, Маком и
сунцокретом,И птицом у лету.";
                // datagram, ne koristimo string, vec samo bajtove
                // koristimo utf8 kodiranje da bismo mogli da koristimo cirilicu
                try {
                        byte[] data = s.getBytes("UTF-8");
                        try (DatagramSocket socket = new DatagramSocket(0)){
                                // broadcast za mrežu
                                InetAddress ia =
InetAddress.getByName("192.168.8.255");
                                int port = 1056;
                                DatagramPacket theOutput = new
DatagramPacket(data, data.length, ia, port);
                                socket.send(theOutput);
                        }
                        catch(UnknownHostException e){
                                System.err.println("Unkown host: " + e);
                        }
                        catch (IOException ex) {
                                System.err.println("Cant bind to socket.\n: " +
ex);
                        }
                } catch (UnsupportedEncodingException e1) {
                        System.out.println("Unsuported coding.\n" + e1);
                }


        }

}
```

3. Napisati mrežni program koji se povezuje na server https://petlja.org/, upucuje GET zahtev. Server potom obrađuje GET zahtev i ispisuje rezultat obrade zahteva na standardni izlaz u zasebnim redovima (u formatu koji je čitljiv ljudskom oku).

Resenje:

RESURS:

Harlod, „Java Network Programming“
Chapter 7, URLConnections, Reading the Header

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.MalformedURLException;
import java.net.URL;
import java.net.URLConnection;

public class Client {
    public static String address = "https://petlja.org/";

    public static void main(String[] args) {
        try {
            URL u = new URL(address);
            URLConnection conn = u.openConnection();

            System.out.println(conn.getHeaderField("Server"));

            BufferedReader reader = new BufferedReader(new
InputStreamReader(u.openStream(),"UTF-8"));
            String s;

            while ((s = reader.readLine()) != null) {
                System.out.println(s);
            }
        } catch (MalformedURLException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

RESURS:

Harlod, „Java Network Programming“

Chapter 10, Creating Secure Client Sockets, HHTPS Client

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.io.Writer;

import javax.net.ssl.SSLSocket;
import javax.net.ssl.SSLSocketFactory;

public class Zadatak3 {

        public static void main(String[] args) {
                String host = "petlja.org";
                int port = 443;

                try {
                        SSLSocketFactory factory =
(SSLSocketFactory)SSLSocketFactory.getDefault();
                        SSLSocket socket = (SSLSocket)factory.createSocket(host,
port);


                        String[] supported = socket.getSupportedCipherSuites();
                        socket.setEnabledCipherSuites(supported);
                        Writer out = new
OutputStreamWriter(socket.getOutputStream());

                        out.write("GET https://" + host + "/ HTTP/1.1\r\n");
                        out.write("Host: " + host + "\r\n");
                        out.write("Accept: text/plain\r\n");
                        out.write("\r\n");
                        out.flush();


                        BufferedReader in = new BufferedReader(new
InputStreamReader(socket.getInputStream()));

                        String s;
                        while (!(s = in.readLine()).equals("")){
                                System.out.println(s);
                        }
                        System.out.println();


                        String contentLength = in.readLine();
                        int length = Integer.MAX_VALUE;
                        try {
                                length = Integer.parseInt(contentLength.trim(),
16);
                        } catch (NumberFormatException ex) {

                        }

                        System.out.println(contentLength);


                        for (int i = 0, c; i < length && (c = in.read()) != -1;
i++) {
                                System.out.write((char)c);
                        }
                        System.out.println();

                        out.close();
                        in.close();
                        socket.close();
                } catch (IOException ex) {
                        System.err.println(ex);
                }

        }

}

}
```

4. a) Napisati mrežni program koji pristupa resursu https://petlja.org/biblioteka/r/Problems/BBC3-F-panuql

Program treba da iscita sadrzaj HTML datoteke i ispise na standardni izlaz uz izbacivanje odgovarajućih izuzetaka.
b) Na resursu se nalazi link ka sifrovanoj datoteci (sifra proste zamene) https://petljamediastorage.blob.core.windows.net/root/Media/Default/Problem/BCup3-Problem%20F%20Panuql.pdf

Napisite skript koji parsira gore navedeni URL (uz izbacivanje odgovarajućih izuzetaka) i poredi sadrzaj sa resursom: https://petljamediastorage.blob.core.windows.net/root/Media/Default/Problem/BCup3-Problem F Panuql.pdf

Ispišite rezultat poređenja na standardni ulaz.

RESURS:

Harlod, „Java Network Programming“

Chapter 4-7