Computer Networks

# Assignment-5

# Register Number: 23BCE1145

# Name: S. Vishwajith

## Question-1:

### Server:

import java.net.*\**;

import java.io.*\**;

public class Server

{

    private Socket socket = null;

    private ServerSocket server = null;

    private DataInputStream in = null;

    public Server(*int* *port*)

    {

        try

        {

            server = **new** ServerSocket(*port*);

            System.out.println("Server started");

            System.out.println("Waiting for a client ...");

            socket = server.accept();

            System.out.println("Client accepted");

            in = **new** DataInputStream(

**new** BufferedInputStream(socket.getInputStream()));

            String line = "";

            while (!line.equals("End"))

            {

                try

                {

                    line = in.readUTF();

                    //Error creation code

                    line = line.substring(0,14)+"0"+line.substring(15); //Converts E to A

                    for(*int* j=0;j<6;j++){

*int* q=0;

                        for(*int* i=8\*j;i<8\*(j+1);i++){

                            if(line.charAt(i)=='1'){

                                q++;

                            }

                        }

                        if(q%2!=0){

                            System.out.println("Error in transmission of character "+(j+1));

                        }

                    }

                }

                catch(IOException *i*)

                {

                    System.out.println(i);

                }

            }

            System.out.println("Closing connection");

            socket.close();

            in.close();

        }

        catch(IOException *i*)

        {

            System.out.println(i);

        }

    }

    @SuppressWarnings("unused")

    public static *void* main(String *args*[])

    {

        Server server = **new** Server(5000);

    }

}

### Client:

import java.io.*\**;

import java.net.*\**;

public class Client{

    private Socket socket = null;

    private DataInputStream input = null;

    private DataOutputStream out = null;

    @SuppressWarnings("deprecation")

    public Client(String *address*, *int* *port*)

    {

        try {

            socket = **new** Socket(*address*, *port*);

            System.out.println("Connected");

            input = **new** DataInputStream(System.in);

            out = **new** DataOutputStream(

                socket.getOutputStream());

        }

        catch (UnknownHostException *u*) {

            System.out.println(u);

            return;

        }

        catch (IOException *i*) {

            System.out.println(i);

            return;

        }

        String line = "";

        String line2= "";

        String line3= "";

*int* arr[] = {0,0,0,0,0,0,0,0};

        while (!line.equals("End")) {

            try {

                line = input.readLine();

                for(*int* i=0;i<line.length();i++){

                    String b = String.format("%7s",Integer.toBinaryString(line.charAt(i))).replace(' ','0');

*int* q=0;

                    String p="";

                    for(*int* j=0;j<b.length();j++){

                        if(b.charAt(j) == '1'){

                            q++;

                            arr[j] += 1;

                        }

                    }

                    if(q%2==0){

                        p = "0";

                    }

                    else{

                        p = "1";

                    }

                    line2 += (b+p);

                }

                for(*int* i=0;i<arr.length;i++){

                    if(arr[i]%2==0){

                        line3 += "0";

                    }

                    else{

                        line3 += "1";

                    }

                }

                line3 += "0";

                line2 += line3;

                for(*int* j=0;j<6;j++){

                        for(*int* i=8\*j;i<8\*(j+1);i++){

                            System.out.print(line2.charAt(i));

                        }

                        System.out.println("");

                    }

                out.writeUTF(line2);

            }

            catch (IOException *i*) {

                System.out.println(i);

            }

        }

        try {

            input.close();

            out.close();

            socket.close();

        }

        catch (IOException *i*) {

            System.out.println(i);

        }

    }

    @SuppressWarnings("unused")

    public static *void* main(String *args*[])

    {

        Client client = **new** Client("127.0.0.1", 5000);

    }

}

### Outputs:

A screenshot of a computer

AI-generated content may be incorrect.

A computer screen with white text

AI-generated content may be incorrect.

## Question-2:

### Server:

import java.net.*\**;

import java.io.*\**;

public class Server{

    private Socket socket = null;

    private ServerSocket server = null;

    private DataInputStream in = null;

    private DataOutputStream out = null;

    @SuppressWarnings("unused")

    public Server(*int* *port*){

        try{

            server = **new** ServerSocket(*port*);

            System.out.println("Server started");

            System.out.println("Waiting for a client ...");

            socket = server.accept();

            System.out.println("Client accepted");

            out = **new** DataOutputStream(socket.getOutputStream());

            in = **new** DataInputStream(**new** BufferedInputStream(socket.getInputStream()));

            String line = "";

            String msg = "";

            String arr[] = {"127.0.1.1","127.0.1.2","127.0.1.3","127.0.1.4"};

            while (!line.equals("End")){

                try{

                    msg = in.readUTF();

                    line = in.readUTF();

                    msg = xor(msg,"0000000001010");

*int* k=line.length();

*int* z = msg.length();

*int* l = msg.length()-line.length()+1;

                    String r=msg.substring(0,line.length());

                    while(k<z){

                        if(r.charAt(0)=='1'){

                            r = xor(line,r);

                        }else{

                            r = xor("00000",r);

                        }

                        r+=msg.charAt(k);

                        r=r.substring(1);

                        k++;

                    }

                    if(r.charAt(0)=='1'){

                        r = xor(line,r);

                    }else{

                        r = xor("00000",r);

                    }

                    System.out.println(msg);

                    if(r.equals("00000")){

                        System.out.println("No error");

                    }else{

                        System.out.println("Error");

                    }

                }

                catch(IOException *i*)

                {

                    System.out.println(i);

                }

            }

            System.out.println("Closing connection");

            socket.close();

            in.close();

            out.close();

        }

        catch(IOException *i*)

        {

            System.out.println(i);

        }

    }

    public String xor(String *a*,String *b*){

        String c = "";

        for(*int* i=0;i<*a*.length();i++){

            if(*a*.charAt(i)==*b*.charAt(i)){

                c+="0";

            }

            else{

                c+="1";

            }

        }

        return c;

    }

    @SuppressWarnings("unused")

    public static *void* main(String *args*[])

    {

        Server server = **new** Server(5000);

    }

}

### Client:

import java.io.*\**;

import java.net.*\**;

public class Client{

    private Socket socket = null;

    private DataInputStream input = null;

    private DataOutputStream out = null;

    @SuppressWarnings("deprecation")

    public Client(String *address*, *int* *port*){

        try{

            socket = **new** Socket(*address*, *port*);

            System.out.println("Connected");

            input = **new** DataInputStream(System.in);

            out = **new** DataOutputStream(

                socket.getOutputStream());

        }

        catch(UnknownHostException *u*){

            System.out.println(u);

            return;

        }

        catch(IOException *i*){

            System.out.println(i);

            return;

        }

        String line = "";

        String r = "";

        String res="";

        while (!line.equals("End")) {

            try {

                line = input.readLine();

                String msg = "101101101";

*int* g = msg.length();

                for(*int* i=0;i<line.length()-1;i++){

                    msg+="0";

                }

*int* z = msg.length();

*int* k=line.length();

                r=msg.substring(0,line.length());

                while(k<z){

                    if(r.charAt(0)=='1'){

                        r = xor(line,r);

                    }else{

                        r = xor("00000",r);

                    }

                    r+=msg.charAt(k);

                    r=r.substring(1);

                    k++;

                }

                if(r.charAt(0)=='1'){

                    r = xor(line,r);

                }else{

                    r = xor("00000",r);

                }

                res = msg.substring(0,g)+r.substring(1);

                out.writeUTF(res);

                out.writeUTF(line);

                System.out.println(res);

            }

            catch (IOException *i*) {

                System.out.println(i);

            }

        }

        try {

            input.close();

            out.close();

            socket.close();

        }

        catch (IOException *i*) {

            System.out.println(i);

        }

    }

    public String xor(String *a*,String *b*){

        String c = "";

        for(*int* i=0;i<*a*.length();i++){

            if(*a*.charAt(i)==*b*.charAt(i)){

                c+="0";

            }

            else{

                c+="1";

            }

        }

        return c;

    }

    @SuppressWarnings("unused")

    public static *void* main(String *args*[]){

        Client client = **new** Client("127.0.0.1", 5000);

    }

}

### Outputs:

A computer screen shot of a black screen

AI-generated content may be incorrect.

A computer screen with white text

AI-generated content may be incorrect.