**CODING:**

**SERVER SIDE:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Roll.no:20UIT021

Name: R.SRIDEVI

Ex.No: 3

Program name: Simulate error correction mechanism.

Date: 27/07/2022

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

import java.net.\*;

import java.io.\*;

class crcserver

{

public static void main(String args[])

{

//An exception should be handled because we used DataInputStream class to get input

try

{

//creating server socket

ServerSocket ss = new ServerSocket(3192);

//Binding

Socket s1 = ss.accept();

//Read the data from the socket

DataInputStream dis = new DataInputStream(s1.getInputStream());

DataOutputStream writeData = new DataOutputStream(s1.getOutputStream());

System.out.println("\t\t------- Server Side --------\n");

//read data from socket (UTF encoding method) and typecast to string

String code = (String)dis.readUTF();

String generator = (String)dis.readUTF();

StringBuffer string = new StringBuffer(code);

//int index = (code.length())/2;

string.setCharAt(2,'1');

code = string.toString();

System.out.println("Received code from client is : "+code);

System.out.println("Generator : "+generator);

String result = divide(code,generator);

if(result.equals("000"))

{

System.out.println("Message is received without errors");

}

else

{

System.out.println("Message is received with errors");

}

//close the connection

writeData.close();

s1.close();

}

catch(Exception e)

{

System.out.println("Exception occured"+e);

}

}

static String divide(String code,String gen)

{

int length = gen.length();

String sub = code.substring(0,length);

String remainder = "",temp="";

if(sub.charAt(0) == '1')

{

temp = gen;

}

else

{

temp = "0000";

}

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

{

if(sub.charAt(i) == temp.charAt(i))

{

remainder+='0';

}

else

{

remainder+='1';

}

}

while(length<code.length())

{

if(remainder.charAt(0)=='0')

{

remainder = remainder.substring(1,remainder.length());

remainder += code.charAt(length);

//System.out.println(remainder);

length++;

if(remainder.charAt(0) == '1')

{

temp = gen;

}

else

{

temp = "0000";

}

}

sub = remainder;

remainder = "";

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

{

if(sub.charAt(i)==temp.charAt(i))

{

remainder+='0';

}

else

{

remainder+='1';

}

}

}

return remainder.substring(1,remainder.length());

}

}

**CLIENT SIDE:**

import java.net.\*;

import java.io.\*;

public class crcclient

{

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

{

System.out.println("\t\t------- Client Side --------\n");

//creating the socket

Socket client = new Socket("localhost",3192);

//get the message from client

DataInputStream input = new DataInputStream(System.in);

DataOutputStream writeData = new DataOutputStream(client.getOutputStream());

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

String generator,data,code;

System.out.println("Enter generator : ");

generator = input.readLine();

System.out.println("Enter data : ");

data = input.readLine();

code = data;

int data\_length = data.length();

int generator\_length = generator.length();

for(int i=data\_length;i<(generator\_length+data\_length)-1;i++)

{

code+='0';

}

String result = divide(code,generator);

//Send the message to server

result = data+result;

System.out.println("Code Sent to server is : "+result);

writeData.writeUTF(result);

writeData.writeUTF(generator);

//close the connection

client.close();

writeData.close();

}

static String divide(String code,String gen)

{

int length = gen.length();

String sub = code.substring(0,length);

String remainder = "",temp="";

if(sub.charAt(0) == '1')

{

temp = gen;

}

else

{

temp = "0000";

}

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

{

if(sub.charAt(i) == temp.charAt(i))

{

remainder+='0';

}

else

{

remainder+='1';

}

}

while(length<code.length())

{

if(remainder.charAt(0)=='0')

{

remainder = remainder.substring(1,remainder.length());

remainder += code.charAt(length);

//System.out.println(remainder);

length++;

if(remainder.charAt(0) == '1')

{

temp = gen;

}

else

{

temp = "0000";

}

}

sub = remainder;

remainder = "";

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

{

if(sub.charAt(i)==temp.charAt(i))

{

remainder+='0';

}

else

{

remainder+='1';

}

}

}

return remainder.substring(1,remainder.length());

}

}

**OUTPUT:**

[fystudent@a3fy27 ~]$ javac crcclient.java

Note: crcclient.java uses or overrides a deprecated API.

Note: Recompile with -Xlint:deprecation for details.

[fystudent@a3fy27 ~]$ java crcclient

------- Client Side --------

Enter generator :

1011

Enter data :

10001000111100000111

Code Sent to server is : 10001000111100000111011

[fystudent@a3fy27 ~]$

output for crcserver  
[fystudent@a3fy27 ~]$ javac crcserver.java

[fystudent@a3fy27 ~]$ java crcserver

------- Server Side --------

Received code from client is : 10001000111100000111011

Generator : 1011

Message is received without errors

[fystudent@a3fy27 ~]$

[fystudent@a3fy29 ~]$ javac crcserver.java

[fystudent@a3fy29 ~]$ java crcserver

------- Server Side --------

Received code from client is : 101100001

Generator : 1101

Message is received with errors

[fystudent@a3fy29 ~]$

[fystudent@a3fy29 ~]$ javac crcclient.java

Note: crcclient.java uses or overrides a deprecated API.

Note: Recompile with -Xlint:deprecation for details.

[fystudent@a3fy29 ~]$ java crcclient

------- Client Side --------

Enter generator :

1101

Enter data :

100100

Code Sent to server is : 100100001

[fystudent@a3fy29 ~]$