2.To write a socket program for implementation of echo.

ALGORITHM

CLIENT SIDE

1. Start the program.
2. Create a socket which binds the Ip address of server and the port address to acquire service.
3. After establishing connection send a data to server.
4. Receive and print the same data from server.
5. Close the socket.
6. End the program.

SERVER SIDE

1. Start the program.
2. Create a server socket to activate the port address.
3. Create a socket for the server socket which accepts the connection.
4. After establishing connection receive the data from client.
5. Print and send the same data to client.
6. Close the socket.
7. End the program.

PROGRAM

```
import java.io.*;
import java.net.*;
public class eclient
{
public static void main(String args[])
{
Socket c=null; String line;
DataInputStream is,is1;
PrintStream os;
try
{
c=new Socket("localhost",8080);
}
catch(IOException e)
{
System.out.println(e);
}
try
{
os=new PrintStream(c.getOutputStream());
is=new DataInputStream(System.in);
is1=new DataInputStream(c.getInputStream());
do
{
System.out.println("client");
line=is.readLine();
os.println(line);
if(!line.equals("exit"))
System.out.println("server:"+is1.readLine());
}while(!line.equals("exit"));
}
catch(IOException e)
{
System.out.println("socket closed");
}
}
}
```

Echo Server:

```
import java.io.*;
import java.net.*;
import java.lang.*;
public class eserver
{
public static void main(String args[])throws IOException
{
ServerSocket s=null;
String line;
DataInputStream is;
PrintStream ps;
Socket c=null;
try
{
s=new ServerSocket(8080);
}
catch(IOException e)
{
System.out.println(e);
}
try
{
c=s.accept();
is=new DataInputStream(c.getInputStream());
ps=new PrintStream(c.getOutputStream());
while(true)
{
line=is.readLine();
System.out.println("msg received and sent back to client");
ps.println(line);
}
}
catch(IOException e)
{
System.out.println(e);
}
}
}
```

OUTPUT:

```
E:\network>javac eclient.java
Note: eclient.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

E:\network>java eclient
client
hello
server:hello
client
hai
server:hai
client
exit

E:\network>

E:\network>javac eserver.java
Note: eserver.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

E:\network>java eserver
msg received and sent back to client
msg received and sent back to client
msg received and sent back to client
java.net.SocketException: Connection reset
```

3. To write a client-server application for chat.

Chatclient.java

```
import java.net.*;
import java.io.*;
public class Chatclient
{
public static void main(String arg[])throws Exception
{
Socket c=null;
String line;
DataInputStream is,is1;
PrintStream os;
try
{
c=new Socket("127.0.0.1", 3000);
}
catch(IOException e)
{
System.out.println(e);
}
try
{
os=new PrintStream(c.getOutputStream());
is=new DataInputStream(System.in);
is1=new DataInputStream(c.getInputStream());
do
{
System.out.println("Client:");
line=is.readLine();
os.println(line);
System.out.println("Server:" + is1.readLine());
}while(line.equalsIgnoreCase("quit")==false);
is1.close();
os.close();
}
catch(IOException e)
{
System.out.println("Socket Closed!Message Passing is over");
}
}
}
```

Chatserver.java

```
import java.net.*;
import java.io.*;

public class Chatserver
{
 public static void main(String arg[])throws Exception
{
ServerSocket s=null; String line;
DataInputStream is=null,is1=null;
PrintStream os=null;
Socket c=null;
try
{
s=new ServerSocket(3000);
}
catch(IOException e)
{
System.out.println(e);
 }
try
{
c=s.accept();
is=new DataInputStream(c.getInputStream());
is1=new DataInputStream(System.in);
os=new PrintStream(c.getOutputStream());
do
{
line=is.readLine();
System.out.println("Client:"+line);
System.out.println("Server:");
line=is1.readLine();
os.println(line);
}
while(line.equalsIgnoreCase("quit")==false);
is.close();
os.close();
}
catch(IOException e)
{
System.out.println(e);
}
}
}
```

Client side

```
E:\network>java Chatclient
Client:
hello
Server:hai
Client:
what are doing
Server:do some work
Client:
okay
Server:how r u
Client:
good whats about you
Server:very good
Client:
okay...bye
Server:bye
Client:
quit
Server:quit

E:\network>
```

Server side

```
E:\network>java Chatserver
Client:hello
Server:
hai
Client:what are doing
Server:
do some work
Client:okay
Server:
how r u
Client:good whats about you
Server:
very good
Client:okay...bye
Server:
bye
Client:quit
Server:
quit

E:\network>
```

4. Write a program to compute CRC code for the polynomials.

```
import java.io.*;
class CRC
{
public static void main(String args[]) throws IOException
{
BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
System.out.println("Enter Generator:");
String gen = br.readLine();
System.out.println("Enter Data:");
String data = br.readLine();
String code = data;
while(code.length() < (data.length() + gen.length() - 1))
code = code + "0";
code = data + div(code,gen);
System.out.println("The transmitted Code Word is: " + code);
System.out.println("Please enter the received Code Word: ");
String rec = br.readLine();
if(Integer.parseInt(div(rec,gen)) == 0)
System.out.println("The received code word contains no errors.");
else
System.out.println("The received code word contains errors.");
}
static String div(String num1,String num2)
{
int pointer = num2.length();
String result = num1.substring(0, pointer);
String remainder = "";
for(int i = 0; i < num2.length(); i++)
{
if(result.charAt(i) == num2.charAt(i))
remainder += "0";
else
remainder += "1";
}
while(pointer < num1.length())
{
if(remainder.charAt(0) == '0')
{
remainder = remainder.substring(1, remainder.length());
remainder = remainder + String.valueOf(num1.charAt(pointer));
pointer++;
}
result = remainder;
```

OUTPUT

slidsender side:

```
[net@ItLab-01-63 d5]$ javac slidsender.java
Note: slidsender.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
[net@ItLab-01-63 d5]$ java slidsender
Enter the no. of frames : 4
Enter 4 Messages to be send

hai
how r you
what r you doing
see you soon
Acknowledgment received for 4 frames

Do you wants to send some more frames : no
[net@ItLab-01-63 d5]$
```

slidreceiver side:

```
[net@ItLab-01-63 d5]$ javac slidreceiver.java
Note: slidreceiver.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
[net@ItLab-01-63 d5]$ java slidreceiver

The received Frame 0 is : hai
The received Frame 1 is : how r you
The received Frame 2 is : what r you doing
The received Frame 3 is : see you soon

Acknowledgment sent

[net@ItLab-01-63 d5]$
```

6. To write a java program to perform stop and wait protocol.

StopandwaitSender.java

```
import java.io.*;
import java.net.*;
public class stopandwaitSender
{
Socket sender;
ObjectOutputStream out;
ObjectInputStream in;
String packet,ack,str, msg;
int n,i=0,sequence=0;
stopandwaitSender(){}
public void run(){
try{
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
System.out.println("Waiting for Connection....");
sender = new Socket("localhost",2004);
sequence=0;
out=new ObjectOutputStream(sender.getOutputStream());
out.flush();
in=new ObjectInputStream(sender.getInputStream());
str=(String)in.readObject();
System.out.println("reciver     > "+str);
System.out.println("Enter the data to send....");
packet=br.readLine();
n=packet.length();
do{
try{
if(i<n){
msg=String.valueOf(sequence);
msg=msg.concat(packet.substring(i,i+1));
}
else if(i==n){
msg="end";out.writeObject(msg);break;
}
out.writeObject(msg);
sequence=(sequence==0)?1:0;
out.flush();
System.out.println("data sent>"+msg);
ack=(String)in.readObject();
System.out.println("waiting for ack.....\n\n");
if(ack.equals(String.valueOf(sequence))){
i++;
System.out.println("receiver   >  "+" packet recieved\n\n");
}
else{
System.out.println("Time out resending data....\n\n");
sequence=(sequence==0)?1:0;
}
}catch(Exception e){}
}while(i<n+1);
System.out.println("All data sent. exiting.");
}catch(Exception e){}
finally{
try{
in.close();
out.close();
sender.close();
}
catch(Exception e){}
}
}
public static void main(String args[]){
stopandwaitSender s=new stopandwaitSender();
s.run();
}
}
```

StopandwaitReciever.java

```
import java.io.*;
import java.net.*;
public class stopandwaitReciever{
ServerSocket reciever;
Socket connection=null;
ObjectOutputStream out;
ObjectInputStream in;
String packet,ack,data="";
int i=0,sequence=0;
stopandwaitReciever(){}
public void run(){
try{
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
reciever = new ServerSocket(2004,10);
System.out.println("waiting for connection...");
connection=reciever.accept();
sequence=0;
System.out.println("Connection established   :");
out=new ObjectOutputStream(connection.getOutputStream());
out.flush();
in=new ObjectInputStream(connection.getInputStream());
out.writeObject("connected    .");
do{
try{
packet=(String)in.readObject();
if(Integer.valueOf(packet.substring(0,1))==sequence){
data+=packet.substring(1);
sequence=(sequence==0)?1:0;
System.out.println("\n\nreceiver         >"+packet);
}
else
{
System.out.println("\n\nreceiver         >"+packet +"   duplicate data");
}
if(i<3){
out.writeObject(String.valueOf(sequence));i++;
}
else{
out.writeObject(String.valueOf((sequence+1)%2));
i=0;
}
}
catch(Exception e){}
}while(!packet.equals("end"));
System.out.println("Data recived="+data);
out.writeObject("connection ended    .");
}
catch(Exception e){}
finally{
try{
in.close();
out.close();
reciever.close();
}
catch(Exception e){}
} }
public static void main(String args[]){
stopandwaitReciever s=new stopandwaitReciever();
while(true){
s.run();
}
}
}
```

Output

```
[net@ItLab-01-63 d5]$ javac GoBackNServer.java
[net@ItLab-01-63 d5]$ java GoBackNServer
waiting for connection
The number of packets sent is:8
[net@ItLab-01-63 d5]$

[net@ItLab-01-63 d5]$ javac GoBackNClient.java
[net@ItLab-01-63 d5]$ java GoBackNClient
Localhost/127.0.0.1
No of frame is:8
30
40
50
60
70
80
90
100
Received frame is: 30
Received frame is: 40
Received frame is: 50
Received frame is: 60
Received frame is: -1
Received frame is: 80
Received frame is: 90
Received frame is: 100
Request to retransmit packet no 5 again!!

Received frame is: 70
quiting
[net@ItLab-01-63 d5]$
```

// second program for GoBackN algorithm

GoBackN.java

```
import java.io.*;
public class GoBackN
{
public static void main(String args[]) throws IOException
{
  BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
   System.out.println("Please enter the Window Size: ");
   int window = Integer.parseInt(br.readLine());
    boolean loop = true;
    int sent = 0;
   while(loop)
  {
 for(int i = 0; i < window; i++)
   {
    System.out.println("Frame " + sent + " has been transmitted.");
    sent++;
    if(sent == window)
     break;
   }
   System.out.println("Please enter the last Acknowledgement received.");
   int ack = Integer.parseInt(br.readLine());
    if(ack == window)
    loop = false;
   else
    sent = ack;
  }
  }
 }
```

OUTPUT

```
[net@ItLab-01-63 d5]$ java GoBackN
Please enter the Window Size:
8
Frame 0 has been transmitted.
Frame 1 has been transmitted.
Frame 2 has been transmitted.
Frame 3 has been transmitted.
Frame 4 has been transmitted.
Frame 5 has been transmitted.
Frame 6 has been transmitted.
Frame 7 has been transmitted.
Please enter the last Acknowledgement received.
4
Frame 4 has been transmitted.
Frame 5 has been transmitted.
Frame 6 has been transmitted.
Frame 7 has been transmitted.
Please enter the last Acknowledgement received.
7
Frame 7 has been transmitted.
Please enter the last Acknowledgement received.
8
[net@ItLab-01-63 d5]$
```

8. To implement Address Resolution Protocol(ARP) and Reverse Address Resolution Protocol (RARP) .

Before start this program most should be execute arp –a commends and other commends. Save your arp table. This program generates the arp table.txt file

.

```
import java.io.*;
import java.util.*;
public class arp_rarp
{
    private static final String Command = "arp -a";
    public static void getARPTable(String cmd) throws Exception
    {
        File fp = new File("ARPTable.txt");
        FileWriter fw = new FileWriter(fp);
        BufferedWriter bw = new BufferedWriter(fw);
        Process P = Runtime.getRuntime().exec(cmd);
        Scanner S = new Scanner(P.getInputStream()).useDelimiter("\\A");
        while(S.hasNext())
        bw.write(S.next());
        bw.close();
        fw.close();
    }
    public static void findMAC(String ip) throws Exception
    {
        File fp = new File("ARPTable.txt");
        FileReader fr = new FileReader(fp);
        BufferedReader br = new BufferedReader(fr);
        String line;
        while ((line = br.readLine()) != null)
        {
            if (line.contains(ip))
           {
               System.out.println("Internet Address     Physical Address     Type");
               System.out.println(line);
               break;
           }
        }
        if((line == null))
           System.out.println("Not found");
        fr.close();
        br.close();
    }
    public static void findIP(String mac) throws Exception
    {
        File                 fp                 =                 new                 File("ARPTable.txt");
```

```
          FileReader fr = new FileReader(fp);
          BufferedReader br = new BufferedReader(fr);
          String line;
          while ((line = br.readLine()) != null)
          {
             if (line.contains(mac))
             {
                 System.out.println("Internet Address      Physical Address      Type");
                 System.out.println(line);
                 break;
             }
          }
          if((line == null))
             System.out.println("Not found");
          fr.close();
          br.close();
      }
      public static void main(String as[]) throws Exception
      {
          getARPTable(Command);
          Scanner S = new Scanner(System.in);
          System.out.println("ARP Protocol.");
          System.out.print("Enter IP Address: ");
          String IP = S.nextLine();
          findMAC(IP);
          System.out.println("RARP Protocol.");
          System.out.print("Enter MAC Address: ");
          String MAC = S.nextLine();
          findIP(MAC);
      }
  }
```

OUTPUT

```
javac arp_rarp.java
>java arp_rarp
ARP Protocol.
Enter IP Address: 10.0.15.253
Internet Address        Physical Address       Type
  10.0.15.253           00-16-76-bd-41-27      dynamic

RARP Protocol.
Enter MAC Address: 01-00-5e-00-00-fc
Internet Address        Physical Address       Type
  224.0.0.252           01-00-5e-00-00-fc      static
```

```
Enter IP Address : 92.25.12.123/9
Given address = 92.25.12.123/9
Mask = 255.128.0.0
network address = 92.0.0.0
host address = 0.25.12.123

D:\cnlab\Subnetting>java subnetting
Enter IP Address : 132.45.67.34/20
Given address = 132.45.67.34/20
Mask = 255.255.240.0
network address = 132.45.64.0
host address = 0.0.3.34
```

12. To write a java program for congestion control using Leaky bucket algorithm.

```
import java.io.*;
import java.util.*;

class  LeakyBucketalg
{
 private static int    no_of_packet,bucket_capacity,array_size,current_bucket,over_flow,
          fixed_data_flow;
    private static int array[];
    public void LeakyBucket()
    {
        current_bucket=0;
        System.out.print("\nCurrent Bucket size :" +current_bucket );
        for(int i=0;i<array_size;i++)
        {
            int input=array[i];
            System.out.print("\n---------------------\nInput to bucket is " +input );
            over_flow=0;
            current_bucket=current_bucket+input;
            System.out.print("\nCurrent Bucket size :"+current_bucket);
            if(current_bucket<=fixed_data_flow)
                current_bucket=0;
            else
            {
                current_bucket=current_bucket-fixed_data_flow;
            }

            if(current_bucket<=bucket_capacity)
            {
                System.out.print("\nNO OverFLOW");
            }
            else
            {
                over_flow=(current_bucket-bucket_capacity);
                current_bucket=bucket_capacity;
                System.out.print("\nOver Flow Occured :"+over_flow);
            }
        System.out.print("\nAfter Processing Bucket size is :"+current_bucket);
        }
    }
    public static void main(String args[])
    {
        LeakyBucketalg lb = new LeakyBucketalg();
        Scanner scan = new Scanner(System.in);
        System.out.print("Enter the Bucket Capacity : ");
        bucket_capacity = scan.nextInt();
        System.out.print("\nEnter the Bucket Fixed Data Flow : ");
        fixed_data_flow = scan.nextInt();
        System.out.print("\nEnter the Array Size : ");
        array_size=scan.nextInt();
        System.out.println("\nEnter the Input values of size : "+array_size);
        array = new int[array_size];
        for(int i=0;i<array_size;i++)
        {
            array[i]=scan.nextInt();
        }
        System.out.println("The Input for LeakyBucket is ");
        for(int i=0;i<array_size;i++)
        {
            System.out.print(array[i]+ " " );
        }
        lb.LeakyBucket();
        System.out.print("\n\nPROGRAM TERMINATING SUCCESSFULLY...\n");
        scan.close();
    }
}
```

```
E:\network>java LeakyBucketalg
Enter the Bucket Capacity : 8

Enter the Bucket Fixed Data Flow : 4

Enter the Array Size : 5

Enter the Input values of size : 5
1
50
30
60
40
The Input for LeakyBucket is
1 50 30 60 40
Current Bucket size :0
----------------------
Input to bucket is 1
Current Bucket size :1
NO OverFLOW
After Processing Bucket size is :0
----------------------
Input to bucket is 50
Current Bucket size :50
Over Flow Occured :38
After Processing Bucket size is :8
----------------------
Input to bucket is 30
Current Bucket size :38
Over Flow Occured :26
After Processing Bucket size is :8
```

```
----------------------
Input to bucket is 60
Current Bucket size :68
Over Flow Occured :56
After Processing Bucket size is :8
----------------------
Input to bucket is 40
Current Bucket size :48
Over Flow Occured :36
After Processing Bucket size is :8

PROGRAM TERMINATING SUCCESSFULLY...

E:\network>
```

14. To write a java program for DNS application.

DNS_Server.java

```
import java.io.*;
import java.net.*;
public class DNS_Server
{
    public static void main(String as[]) throws IOException
    {
        DatagramSocket ds;
        DatagramPacket dp_send;
        DatagramPacket dp_rec;
        final int buf_size = 512;
        final int port = 1500;
        byte msg_rec[] = new byte[buf_size];
        byte msg_send[] = new byte[buf_size];
        try
        {
            ds = new DatagramSocket(port);
            dp_rec = new DatagramPacket(msg_rec, buf_size);
            ds.receive(dp_rec);
            String dname = new String(dp_rec.getData()).trim();
            InetAddress ipaddr = InetAddress.getByName(dname);
            InetAddress DA = dp_rec.getAddress();
            int destport = dp_rec.getPort();
            System.out.println(ipaddr);
            msg_send = ipaddr.toString().getBytes();
             dp_send = new DatagramPacket(msg_send, msg_send.length, DA, destport);
            ds.send(dp_send);
        }
        catch(Exception e)
        {
            System.out.println("Exception " + e);
        }
    }
}
```

DNS_Client.java

```
import java.io.*;
import java.net.*;
import java.util.*;
public class DNS_Client
{
    public static void main(String as[]) throws IOException
    {
        DatagramSocket ds;
        DatagramPacket dp_send;
        DatagramPacket dp_rec;
        final int port = 1500;
        final int buf_size = 512;
        Scanner s = new Scanner(System.in);
        byte[] msg_send = new byte[buf_size];
        byte[] msg_rec = new byte[buf_size];
        try
        {
            ds = new DatagramSocket();
            System.out.print("Enter domain name : ");
            String data = s.nextLine();
            msg_send = data.getBytes();
            InetAddress DA = InetAddress.getByName("localhost");
            dp_send = new DatagramPacket(msg_send, msg_send.length, DA, port);
            ds.send(dp_send);
            dp_rec = new DatagramPacket(msg_rec, buf_size);
            ds.receive(dp_rec);
            String ipaddr = new String(dp_rec.getData()).trim();
            System.out.println(ipaddr);
        }
        catch(Exception e)
        {
            System.out.println("Exception " + e);
        }
    }
}
```