

## Set 1

Write a program to simulate Carrier Sense Multiple Access (CSMA/CD).

### Server.java

```
import java.io.*;
import java.net.*;
public class Server
{
    public static void main(String[] args)
    {
        try
        {
            System.out.println("===== Client 2 =====");
            ServerSocket ss = new ServerSocket(3000);
            System.out.println("Waiting for connection");

            Socket con = ss.accept();
            System.out.println("Connected");
            ObjectInputStream in = new ObjectInputStream(con.getInputStream());
            System.out.println((String) in.readObject());

            in.close();
            ss.close();
        }
        catch (Exception e)
        {
            System.out.println(e);
        }
    }
}
```

### Client1.java

```
import java.io.*;
import java.net.*;
public class client1
{
    public static void main(String[] args)
    {
        try
        {
            System.out.println("===== Client 1 =====");
            client1 cli = new client1();
            int Tp = 2000;
            int R = 0;
            int Tb = 0;

            for(int i=1; i<=15;i++)
            {
                System.out.println("attempt : "+i);
                if(cli.send() == "sent")
                {
                    break;
                }
                else
                {
                    R = 2^i-1;
                    System.out.println("Selected Random number :"+R);
                }
            }
        }
    }
}
```

```

        Tb = R*Tp;
System.out.println("waiting for next attempt in seconds): "+Tb);
        Thread.sleep(Tb);
    }
}
}
catch (InterruptedException e)
{
    System.out.println(e);
}
}
String send()
{
    String str=null;
    try
    {
        Socket soc = new Socket("localhost",3000);
        ObjectOutputStream out = new ObjectOutputStream(soc.getOutputStream());
        String msg = "CNLAB";
        out.writeObject(msg);
        System.out.println("Message sent : "+msg);
        str = "sent";
        soc.close();
    }
    catch (Exception e)
    {
        str = "collision occured";
        String msg = null;
        System.out.println("Message sent : "+msg);
    }
    return str;
}
}
}

```

## OUTPUT

The image shows two side-by-side Windows command prompt windows. The left window, titled 'C:\Windows\System32\cmd.exe', shows the execution of a Java server program. The output is as follows:

```

Microsoft Windows [Version 10.0.22000.675]
(c) Microsoft Corporation. All rights reserved.

C:\Users\bveer\OneDrive\Desktop\CN LAB INT-2\CSMA-CD\src>java Server.java
***** Client 2 *****
Waiting for connection
Connected
CNLAB
C:\Users\bveer\OneDrive\Desktop\CN LAB INT-2\CSMA-CD\src>

```

The right window, also titled 'C:\Windows\System32\cmd.exe', shows the execution of a Java client program. The output is as follows:

```

Microsoft Windows [Version 10.0.22000.675]
(c) Microsoft Corporation. All rights reserved.

C:\Users\bveer\OneDrive\Desktop\CN LAB INT-2\CSMA-CD\src>java client1.java
***** Client 1 *****
attempt : 1
Message sent : CNLAB
C:\Users\bveer\OneDrive\Desktop\CN LAB INT-2\CSMA-CD\src>

```

## Set 2

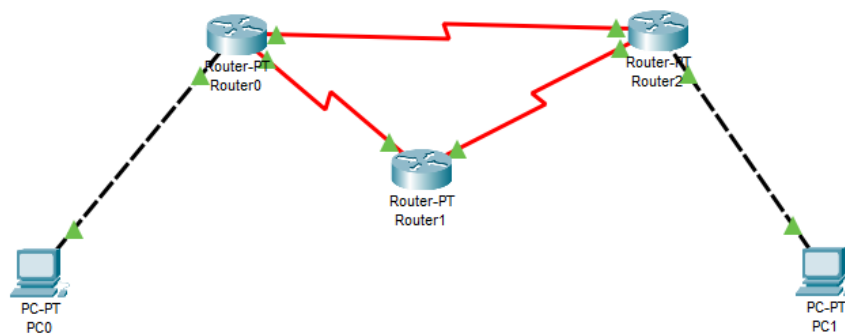
Configure a network using **Distance Vector Routing** protocol using Packet tracer tool.

### Implementation:

Follow the below steps to configure a network using Distance Vector Routing protocol.

1) Open Cisco packet tracer and make an overview of the connection using PC and Router.

2) Consider the below model for DVR protocol using Cisco packet.



- 3) Plug all the PT-routers and PCs as shown in the model without establishing any connection among each others.
- 4) Firstly, take PC<sub>0</sub> connect it with PT-router<sub>0</sub> using FastEthernet 0 in both the cases. Once, its done provide IP address to the PC i.e. 192.168.1.2 and change the Default Gateway to 192.168.1.1. The DGR provided in PC should be the IP address to the PT-router<sub>0</sub>.
- 5) Now consider PT-router<sub>0</sub> and PT-router<sub>1</sub>, connect them using Serial 2/0 in router<sub>0</sub> and Serial 3/0 in router<sub>1</sub>. We need to create a separate class path to all the routers which are interconnected, class IP address can be 10.0.0.1, 20.0.0.1, 30.0.0.1
- 6) In router<sub>0</sub> and router<sub>1</sub>, go into the configuration of Serial 2/0 or 3/0 into their respective routers and give class IP address as listed above. Repeat the same steps to the router<sub>2</sub> and PC<sub>1</sub> from the above steps provided.
- 7) Once, the model is successful created and connected with each other. Now its the task to provide routers with the IP addresses of the connected systems, switches, hubs, etc.
- 8) Go to RIP configuration into the respective router and plug all the other IP address of systems which are connected to it such as PC<sub>0</sub>-192.168.1.1, routers-10.0.0.1, 20.0.0.1, 30.0.0.1.
- 9) Now transmit messages from PC<sub>0</sub> to PC<sub>1</sub>.
- 10) In initial state messages get failed, try another time to activate the connections established and finally message gets transmitted to the other end/destination.

### Set 3

write a program to implement client server communication for chat using TCP.

#### Server.java

```
import java.net.*;
import java.io.*;
public class Server {
    public static void main(String[] args) throws Exception {
        System.out.println("server is connected");
        ServerSocket ss=new ServerSocket(3333);
        System.out.println("Server is waiting for client request");
        Socket s=ss.accept();
        System.out.println("Client is connected, start chatting");

        DataInputStream din=new DataInputStream(s.getInputStream());
        DataOutputStream dout=new DataOutputStream(s.getOutputStream());
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        String str="",str2="";
        while(!str.equals("stop"))
        {
            str=din.readUTF();
            System.out.println("Client Says : "+str);
            str2=br.readLine();
            dout.writeUTF(str2);
            dout.flush();
        }
        din.close();
        ss.close();
    }
}
```

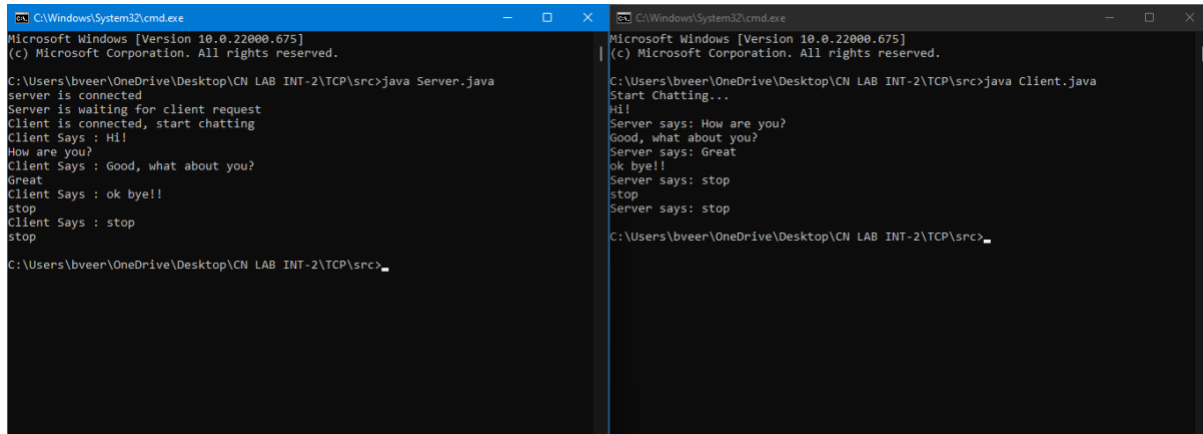
#### Client.java

```
import java.net.*;
import java.io.*;
public class Client {
    public static void main(String[] args) throws Exception {
        try {
            Socket s=new Socket("localhost",3333);
            System.out.println("Start Chatting...");
            DataInputStream din=new DataInputStream(s.getInputStream());
            DataOutputStream dout=new DataOutputStream(s.getOutputStream());
            BufferedReader br=new BufferedReader(new
            InputStreamReader(System.in));

            String str="",str2="";
            while(!str.equals("stop"))
            {
                str=br.readLine();
                dout.writeUTF(str);
                dout.flush();
                str2=din.readUTF();
                System.out.println("Server says: "+str2);
            }
            dout.close();
            s.close();
        }
    }
}
```

```
        catch (ConnectException e) {  
            System.out.println("Server is offline");  
        }  
    }  
}
```

## OUTPUT



The image shows two side-by-side Windows command prompt windows. The left window is titled 'C:\Windows\System32\cmd.exe' and shows the execution of 'java Server.java'. The output indicates the server is waiting for a client request, a client is connected, and the server receives messages from the client: 'Hi!', 'How are you?', 'Good, what about you?', 'Great', 'ok bye!!', and 'stop'. The right window is also titled 'C:\Windows\System32\cmd.exe' and shows the execution of 'java Client.java'. The output indicates the client starts chatting, sends 'Hi!', receives 'Server says: How are you?', sends 'Good, what about you?', receives 'Server says: Great', sends 'ok bye!!', receives 'Server says: stop', and finally sends 'stop'.

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22000.675]
(c) Microsoft Corporation. All rights reserved.

C:\Users\bveer\OneDrive\Desktop\CN LAB INT-2\TCP>java Server.java
server is connected
Server is waiting for client request
Client is connected, start chatting
Client Says : Hi!
How are you?
Client Says : Good, what about you?
Great
Client Says : ok bye!!
stop
Client Says : stop
stop
C:\Users\bveer\OneDrive\Desktop\CN LAB INT-2\TCP>_

C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22000.675]
(c) Microsoft Corporation. All rights reserved.

C:\Users\bveer\OneDrive\Desktop\CN LAB INT-2\TCP>java Client.java
Start Chatting...
Hi!
Server says: How are you?
Good, what about you?
Server says: Great
ok bye!!
Server says: stop
stop
Server says: stop
C:\Users\bveer\OneDrive\Desktop\CN LAB INT-2\TCP>_
```

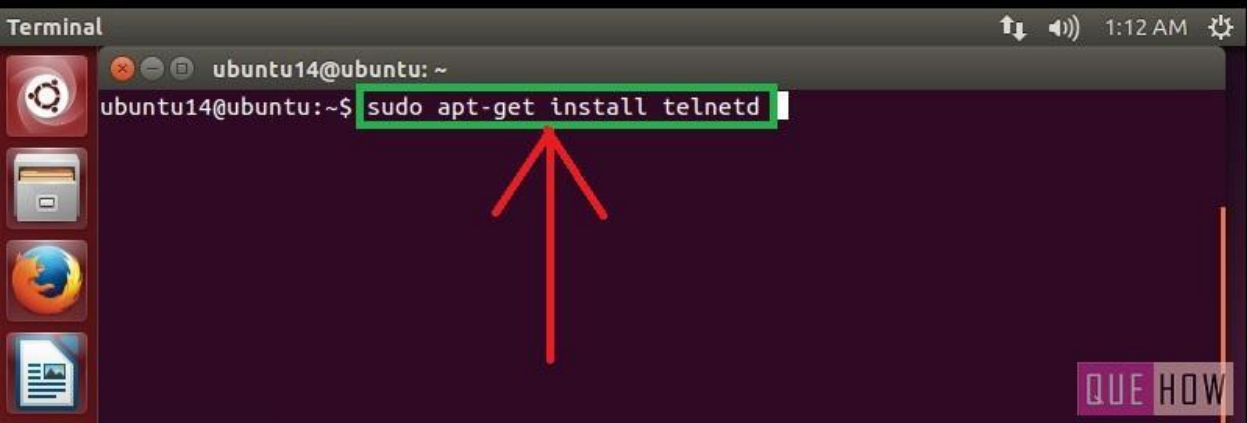
## Set 4

Install **Telnet** on one of the systems connected by a switch and telnet it from other system.

### Installation of Telnet:

#### Step 1: `sudo apt-get install telnetd`

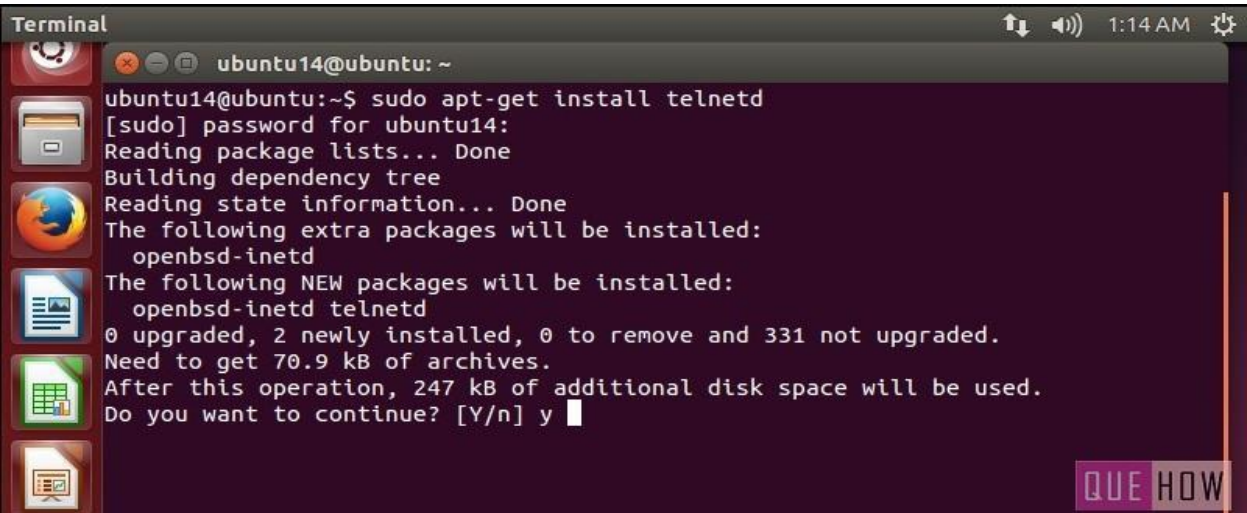
This will complete the installation of Telnet. “**telnetd**” is a daemon that gets invoked by “**inetd**” or its extension “**xinetd**”, both are the internet servers.

A terminal window titled 'Terminal' with a dark background. The prompt is 'ubuntu14@ubuntu: ~'. The command 'sudo apt-get install telnetd' is entered and highlighted with a green box. A red arrow points to the command. The system clock shows 1:12 AM. A 'QUE HOW' logo is in the bottom right corner.

#### Step 2:

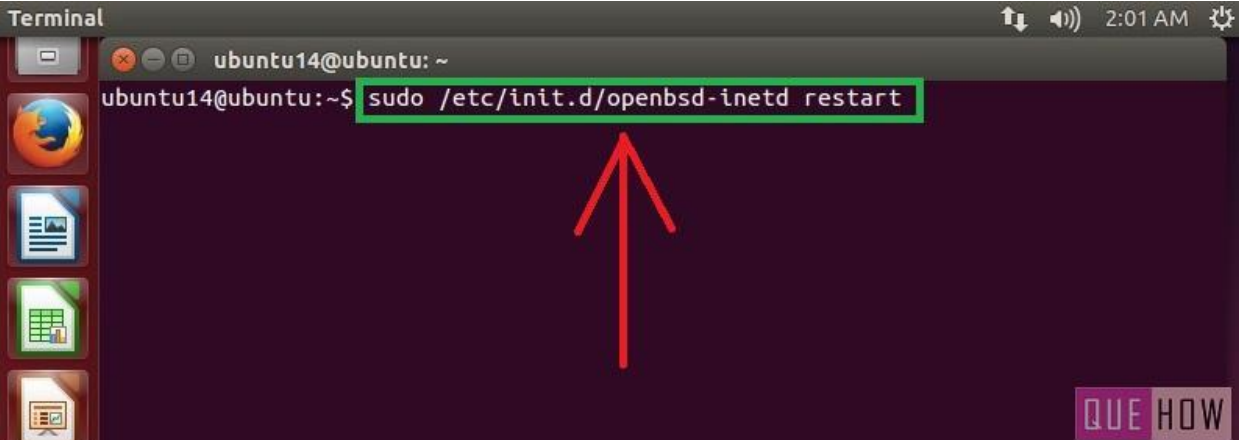
Then you are asked to enter the user password and then press enter. Processing will start as soon as you press enter. After this, I have noticed a line “**274 KB additional disk space will be used**” on the terminal screen.

You may also observe some sort of a message like this and then you’ll be asked to continue or not. Just write “**y**” and then press enter to continue.

A terminal window titled 'Terminal' with a dark background. The prompt is 'ubuntu14@ubuntu: ~'. The command 'sudo apt-get install telnetd' has been executed. The output shows the password prompt, package lists being read, dependency tree building, and state information reading. It lists extra packages (openbsd-inetd) and new packages (openbsd-inetd, telnetd) to be installed. It shows that 0 packages are upgraded, 2 are newly installed, and 0 are to be removed. It states that 70.9 kB of archives are needed and that 247 kB of additional disk space will be used. It asks 'Do you want to continue? [Y/n]' and the user has responded with 'y'. The system clock shows 1:14 AM. A 'QUE HOW' logo is in the bottom right corner.

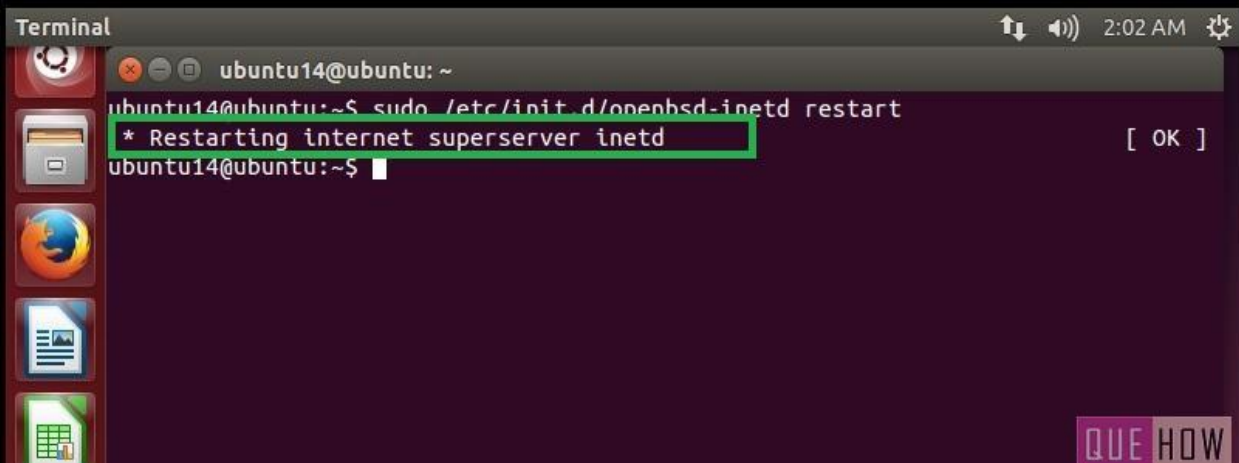
**Step 3:** Now when you are done with it, restart “**inetd**”. Type “**sudo /etc/init.d.open-bsd-inetd restart**”.

“inetd” is daemon used for *dealing with incoming network* and it is responsible for deciding which program to run when a request comes.

A terminal window titled "Terminal" with a dark background. The prompt is "ubuntu14@ubuntu: ~". The command "sudo /etc/init.d/openbsd-inetd restart" is entered and highlighted with a green box. A red arrow points upwards from the bottom center of the terminal towards the command. The system clock in the top right corner shows "2:01 AM". On the left side, there is a vertical dock with icons for a web browser, a document, a spreadsheet, and a presentation. In the bottom right corner, there is a "QUE HOW" logo.

```
Terminal
ubuntu14@ubuntu: ~
ubuntu14@ubuntu:~$ sudo /etc/init.d/openbsd-inetd restart
```

**Step 4:** To ensure “inetd” is started, press enter after writing the above command.

A terminal window titled "Terminal" with a dark background. The prompt is "ubuntu14@ubuntu: ~". The command "sudo /etc/init.d/openbsd-inetd restart" has been executed. The output is "\* Restarting internet superserver inetd" followed by "[ OK ]" on the same line. The text "\* Restarting internet superserver inetd" is highlighted with a green box. The system clock in the top right corner shows "2:02 AM". On the left side, there is a vertical dock with icons for a web browser, a document, a spreadsheet, and a presentation. In the bottom right corner, there is a "QUE HOW" logo.

```
Terminal
ubuntu14@ubuntu: ~
ubuntu14@ubuntu:~$ sudo /etc/init.d/openbsd-inetd restart
* Restarting internet superserver inetd [ OK ]
ubuntu14@ubuntu:~$
```

**To connect with any remote client:**

**Step 5:** Just type: “telnet hostipaddress”. For an example: “telnet 192.168.0.68” and press enter.

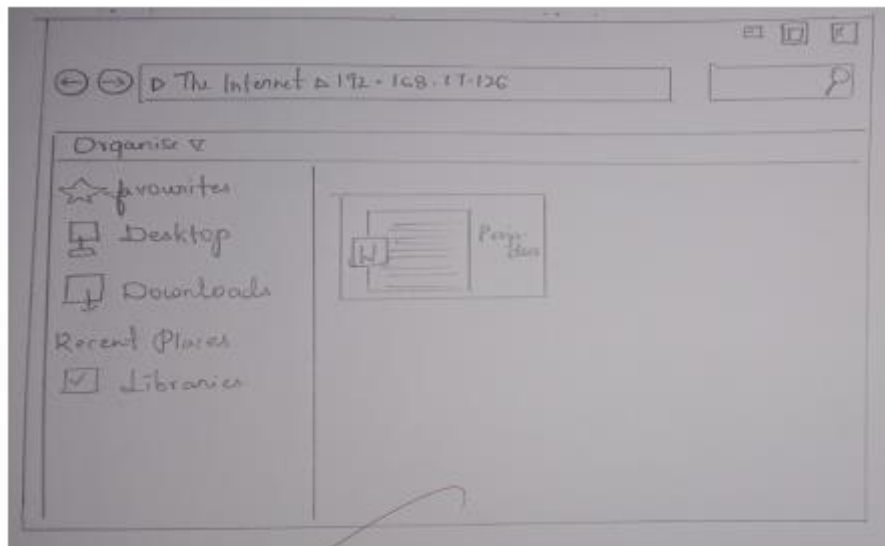
**Step 6:** Then you’ll see, it is connected to “host ip address”. For security reasons, you are required to provide “username” and “password” as well.



## Set 5

Configure **FTP Server** on a Linux or Windows machine using FTP Client.

- 1) Type in cmd ip config to find the ip address of the system
- 2) Create a folder in desktop and store files that you would want to share
- 3) To configure the FTP server go to the control panel, click on programs -> programs and features.
- 4) Click on turn windows features on /off
- 5) Check mark IIS (internet and information services)
- 6) Expand IIS and check mark ftp server
- 7) Expand ftp server check mark ftp extensibility and ftp services
- 8) Make sure web management tools and World Wide Web services are checked.
- 9) Click ok .10) Click on control panel home
- 11) Click on administrative tools (if windows 10 or 11, search for administrative tools).
- 12) Click on IIS MANAGER
- 13) Under connections look for the root (folder name) (available on top right)
- 14) Expand root 15) Expand sites
- 16) Right click on sites and select add ftp site
- 17) Name the ftp site
- 18) Browse the physical path for the folder created in step 2.
- 19) Click on next
- 20) Select the ip address of your computer. And select no ssl under ssl and click on next.
- 21) Select basic authentication
- 22) Select authorization as specific users and type in Gcet (username)
- 23) Check in required permissions.
- 24) Click on finish 25) FTP site is configured.
- 26) Go to MyComputer, type in <ftp://ip> address and click enter2
- 7) Provide user credentials and access the files available in the ftp server. This access is internally with in a network
- 28) To access this ftp outside a network we need to enable port no 21 at the router





## Set 6

Using **RSA Algorithm** Encrypt a Text data and Decrypt the same.

### RSA.java

```
import java.io.DataInputStream;
import java.io.IOException;
import java.math.BigInteger;
import java.util.Random;

public class RSA
{
    private BigInteger p;
    private BigInteger q;
    private BigInteger N;
    private BigInteger phi;
    private BigInteger e;
    private BigInteger d;
    private int bitlength = 1024;
    private Random r;

    public RSA()
    {
        r = new Random();
        p = BigInteger.probablePrime(bitlength, r);
        q = BigInteger.probablePrime(bitlength, r);
        N = p.multiply(q);
        phi = p.subtract(BigInteger.ONE).multiply(q.subtract(BigInteger.ONE));
        e = BigInteger.probablePrime(bitlength / 2, r);

        while (phi.gcd(e).compareTo(BigInteger.ONE) > 0 && e.compareTo(phi) < 0)
        {
            e.add(BigInteger.ONE);
        }
        d = e.modInverse(phi);
    }

    public RSA(BigInteger e, BigInteger d, BigInteger N)
    {
        this.e = e;
        this.d = d;
        this.N = N;
    }

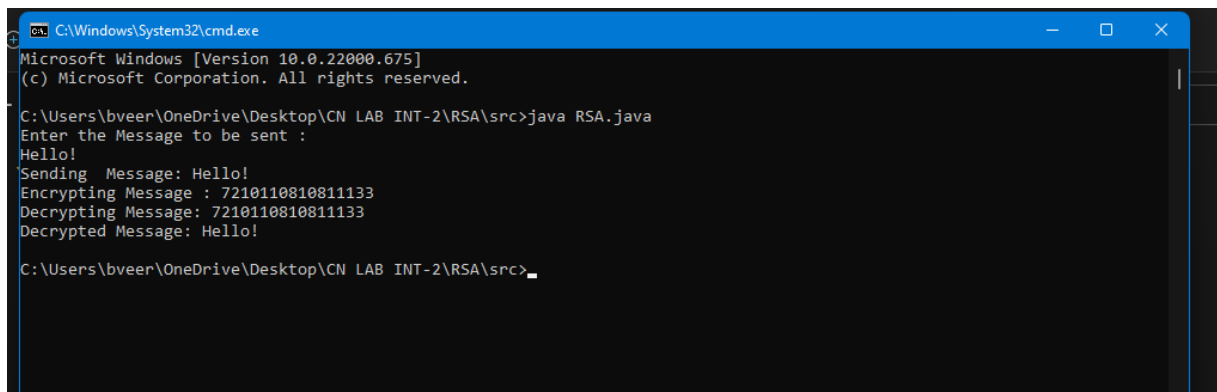
    @SuppressWarnings("deprecation")
    public static void main(String[] args) throws IOException
    {
        RSA rsa = new RSA();
        DataInputStream in = new DataInputStream(System.in);
        String teststring;
        System.out.println("Enter the Message to be sent :");
        teststring = in.readLine();
        System.out.println("Sending Message: " + teststring);
        System.out.println("Encrypting Message : "+
        bytesToString(teststring.getBytes()));
        // encrypt
        byte[] encrypted = rsa.encrypt(teststring.getBytes());
        // decrypt
        byte[] decrypted = rsa.decrypt(encrypted);
        System.out.println("Decrypting Message: " + bytesToString(decrypted));
        System.out.println("Decrypted Message: " + new String(decrypted));
    }
}
```

```

    }
    private static String bytesToString(byte[] encrypted)
    {
        String test = "";
        for (byte b : encrypted)
        {
            test += Byte.toString(b);
        }
        return test;
    }
    // Encrypt message
    public byte[] encrypt(byte[] message)
    {
        return (new BigInteger(message)).modPow(e, N).toByteArray();
    }
    // Decrypt message
    public byte[] decrypt(byte[] message)
    {
        return (new BigInteger(message)).modPow(d, N).toByteArray();
    }
}

```

## OUTPUT



```

C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22000.675]
(c) Microsoft Corporation. All rights reserved.

C:\Users\bveer\OneDrive\Desktop\CN LAB INT-2\RSA\src>java RSA.java
Enter the Message to be sent :
Hello!
Sending Message: Hello!
Encrypting Message : 7210110810811133
Decrypting Message: 7210110810811133
Decrypted Message: Hello!

C:\Users\bveer\OneDrive\Desktop\CN LAB INT-2\RSA\src>

```

## Set 7

Write a program to simulate Carrier Sense Multiple Access (CSMA/CA).

### Server.java

```
import java.net.*;
public class server
{
    public static void main(String[] args)
    {
        try
        {
            System.out.println("===== Client 2 =====");
            while(true)
            {
                ServerSocket ss = new ServerSocket(3000);
                System.out.println("Waiting for connection");
                ss.accept();
                ss.close();
                System.out.println("Connected");
            }
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
    }
}
```

### Client.java

```
import java.net.*;

public class client
{
    public static void main(String[] args)
    {
        try
        {
            System.out.println("===== Client 1 =====");
            client cli = new client();
            int R = 0;
            Boolean bln = false;

            for(int k=1; k<=15;k++)
            {
                System.out.println("Attempt : "+k);
                System.out.println("is Channel idle? ");

                int i = 0;
                while(true)
                {
                    System.out.print(i=i+1);
                    if(cli.isidle())
                    {
                        System.out.println("\n Channel idle");
                        System.out.println("Wait IFS time 5000");
                        Thread.sleep(8000);
                        System.out.println("is still idle?");
                    }
                }
            }
        }
    }
}
```

```

        if(cli.isidle())
        {
            System.out.println("Still idle");
            R = 2^k-1;
            System.out.println("Selected Random number :"+R);
            System.out.println("waiting for R slot time: "+R*6000);
            Thread.sleep(R*6000);
            System.out.println("Message sent");
            System.out.println("Wait for time out : "+10000);
            Thread.sleep(10000);

            if(cli.isidle())
            {
                System.out.println("Ack received");
                bln = true;
                break;
            }
            else
            {
                System.out.println("Ack not received");
                break;
            }
        }
        else
        {
            System.out.println("Busy, goes to channel idle check");
        }
    }

    if(bln == true)
    {
        break;
    }
}

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

}

Boolean isidle()
{
    try
    {
        Socket soc= new Socket("localhost",3000);
        soc.close();
        return true;
    }
    catch (Exception e)
    {
        return false;
    }
}
}

```

## OUTPUT

```
C:\Windows\System32\cmd.exe - java_server.java
Microsoft Windows [Version 10.0.22000.675]
(c) Microsoft Corporation. All rights reserved.

C:\Users\bveer\OneDrive\Desktop\CN LAB INT-2\CSMA-CA\src>java server.java
----- client 2 -----
Waiting for connection
Connected
Waiting for connection
Connected
Waiting for connection
Connected
Waiting for connection

C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22000.675]
(c) Microsoft Corporation. All rights reserved.

C:\Users\bveer\OneDrive\Desktop\CN LAB INT-2\CSMA-CA\src>java client.java
----- client 1 -----
Attempt : 1
is Channel idle?
1
Channel idle
Wait IF5 time 5000
is still idle?
Still idle
Selected Random number :2
waiting for R slot time: 12000
Message sent
Wait for time out : 10000
Ack received

C:\Users\bveer\OneDrive\Desktop\CN LAB INT-2\CSMA-CA\src>
```