Program:

Remote.java

```java
import java.rmi.*;

import java.net.*;

import java.io.*;

import java.rmi.registry.*;

public class remote

{

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

{

InetAddress ip=InetAddress.getLocalHost();

 rev_in obj =new rev_im();

 LocateRegistry.createRegistry(1900);

  Naming.rebind("rmi://localhost:1900/"+ip.getHostName(),obj);

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

}}
```

Interface class:

```java
import java.rmi.*;

public interface rev_in extends java.rmi.Remote

{

 public int reverse(int x) throws RemoteException;

}
```

Implementation class:

```java
import java.rmi.server.*;

import java.io.*;

public class rev_im extends UnicastRemoteObject implements rev_in
```

```java
{
 public rev_im() throws Exception
 {
  super();
 }
 public int reverse(int x)
 {
  int res = 0;
  while(x!=0)
  {
   int d=x%10;
   res=(res*10)+d;
   x=x/10;
  } return res;
 }
}
```

Output:

Program:

UPD server:

```java
import java.io.*;
import java.net.*;
public class udps
{      public static void main(String[] args) throws IOException
       {          DatagramSocket ds = new DatagramSocket(8000);
              byte[] receive = new byte[99999];
              FileWriter fw=new FileWriter("udpin.txt");
              while (true)
              {
                      DatagramPacket DpReceive = new DatagramPacket(receive,
receive.length);
                      ds.receive(DpReceive);
                      System.out.println("Client:-" + data(receive));
                  String ttt=data(receive).toString();
                  //fw.write(data(receive).toString());
                  for(int kk=0;kk<ttt.length();kk++)
                          fw.write(ttt.charAt(kk));
                      if (data(receive).toString().equals("exit"))
                      {
                              System.out.println("EXITING");
                              break;
                      }
                      receive = new byte[99999];
              }
              //System.out.println("EXITING");

       }
```

```java
        public static StringBuilder data(byte[] a)
        {
                if (a == null)
                        return null;
                StringBuilder ret = new StringBuilder();
                int i = 0;
                while (a[i] != 0)
                {
                        ret.append((char) a[i]);
                        i++;
                }
                return ret;
        }
}
```

UDP client:

```java
import java.io.*;
import java.net.*;
import java.util.*;
public class udpcl
{       public static void main(String args[]) throws IOException
        {       File f=new File("udpout.txt");
                BufferedReader br=new BufferedReader(new FileReader(f));
                DatagramSocket ds = new DatagramSocket();
                InetAddress ip = InetAddress.getLocalHost();
                byte buf[] = null;
                //String inp;
                        while (true)
                        {
```

```
String inp = br.readLine();

buf = inp.getBytes();

DatagramPacket DpSend =new DatagramPacket(buf, buf.length,
ip, 8000);

ds.send(DpSend);

if (inp.equals("exit"))

        break;}}}
```

Output:

Open ▾ | udpin.txt ~/nwlab   Save   ≡   ⊖ ▢ ✕

```
this
is
selvamani
from
tce
exit
```

in Text ▾   Tab Width: 8 ▾   Ln 1, Col 1   ▾   INS

Open ▾ | udpout.txt ~/nwlab   Save   ≡   ⊖ ▢ ✕

```
this
is
selvamani
from
tce
exit
```

n Text ▾   Tab Width: 8 ▾   Ln 1, Col 1   ▾   INS

```
sachin@sachin: ~/Downloads
sachin@sachin:~/nwlab$ javac udpclient.java
sachin@sachin:~/nwlab$ javac udpserver.java
sachin@sachin:~/nwlab$ java udpserver
Client:-this
Client:-is
Client:-selvamani
Client:-from
Client:-tce
Client:-exit
Client sent bye.....EXITING
sachin@sachin:~/nwlab$ █
```

```
sachin@sachin: ~/Download
sachin@sachin:~/nwlab$ java udpclient
File Sent..! Exiting!
sachin@sachin:~/nwlab$ █
```

Program:

TCP Client:

```java
import java.io.*;
import java.net.*;
public class tcpclient{
  public static void main(String args[]) throws Exception{
      Socket socket =new Socket(InetAddress.getByName("LocalHost"),5000);
      byte[] contents=new byte[10000];
      FileOutputStream fos=new FileOutputStream("tcpin.txt");
      BufferedOutputStream bos=new BufferedOutputStream(fos);
      InputStream is=socket.getInputStream();
      int bytesread=0;
      while((bytesread=is.read(contents))!=-1)
         bos.write(contents,0,bytesread);
      bos.flush();
      socket.close();
      System.out.println("File saved successfully");
}}
```

Server:

```java
import java.io.*;
import java.net.*;
public class tcpserver{
public static void main(String args[]) throws Exception{
  ServerSocket sock=new ServerSocket(5000);
  Socket socket=sock.accept();
  InetAddress ia=InetAddress.getByName("LocalHost");
  File file =new File("tcpout.txt");
  FileInputStream fis=new FileInputStream(file);
```

```java
BufferedInputStream bis=new BufferedInputStream(fis);

OutputStream os=socket.getOutputStream();

byte[] contents;

long filelength=file.length();

long current=0;

long start=System.nanoTime();

while(current!=filelength){


int size=10000;

if(filelength-current>=size)

    current+=size;

else{

  size=(int)(filelength-current);

  current=filelength;

  }

 contents=new byte[size];

 bis.read(contents,0,size);

 os.write(contents);

 System.out.println("sending file..."+(current+100)/filelength+"% complete");

}

 os.flush();

 socket.close();

 sock.close();

 System.out.println("file sent successfully");}}
```

Output:

tcpin.txt
~/nwlab

```
this
is
selvamani
from
tce
exit
```


tcpout.txt
~/nwlab

```
this
is
selvamani
from
tce
exit
```

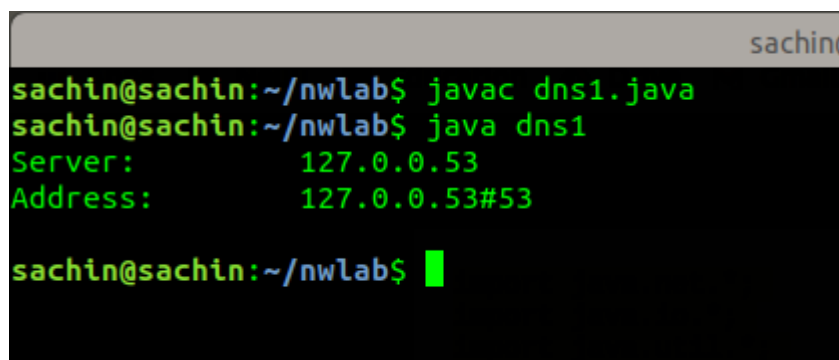

```
sachin@sachin: ~/Downloads
sachin@sachin:~/nwlab$ java tcpclient
File saved successfully
sachin@sachin:~/nwlab$
```



```
sachin@sachin: ~/Downloads
sachin@sachin:~/nwlab$ javac tcpserver.java
sachin@sachin:~/nwlab$ javac tcpclient.java
sachin@sachin:~/nwlab$ java tcpserver
sending file...4% complete
file sent successfully
sachin@sachin:~/nwlab$
```

Program:

```java
import java.net.*;

import java.io.*;

import java.util.*;

public class dns1{

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

   Process process = Runtime.getRuntime().exec("nslookup www.yahoo.com");

   BufferedReader output = new BufferedReader(new
InputStreamReader(process.getInputStream()));

   String s=output.readLine();

    System.out.println(s);

  s=output.readLine();

   System.out.println(s);

   s=output.readLine();     System.out.println(s);

}}
```
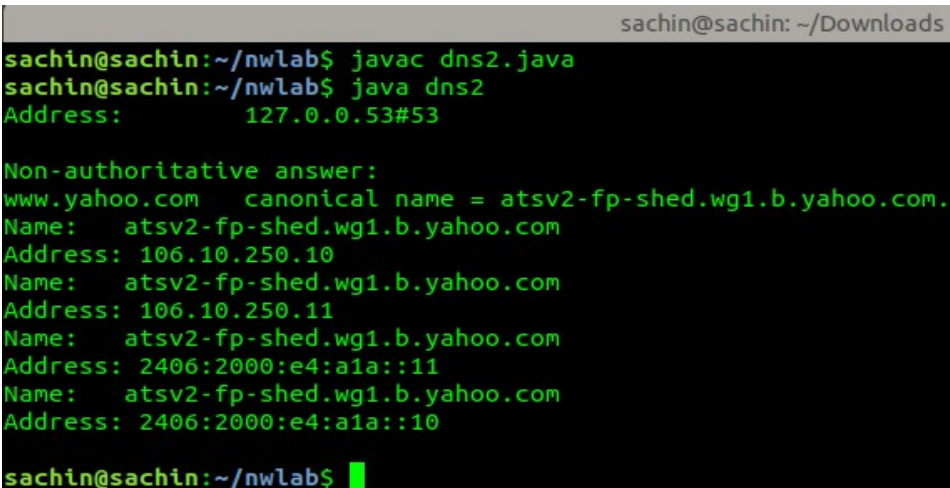
Output:

Program:

```java
import java.net.*;

import java.io.*;

import java.util.*;

public class dns2{

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

  Process process = Runtime.getRuntime().exec("nslookup www.yahoo.com");

   BufferedReader output = new BufferedReader(new
InputStreamReader(process.getInputStream()));

  String s=output.readLine();

   while((s=output.readLine())!=null)

    System.out.println(s);

}}
```

Output:

```
                                    sachin@sachin: ~/Downloads
sachin@sachin:~/nwlab$ javac dns2.java
sachin@sachin:~/nwlab$ java dns2
Address:        127.0.0.53#53

Non-authoritative answer:
www.yahoo.com   canonical name = atsv2-fp-shed.wg1.b.yahoo.com.
Name:   atsv2-fp-shed.wg1.b.yahoo.com
Address: 106.10.250.10
Name:   atsv2-fp-shed.wg1.b.yahoo.com
Address: 106.10.250.11
Name:   atsv2-fp-shed.wg1.b.yahoo.com
Address: 2406:2000:e4:a1a::11
Name:   atsv2-fp-shed.wg1.b.yahoo.com
Address: 2406:2000:e4:a1a::10

sachin@sachin:~/nwlab$
```