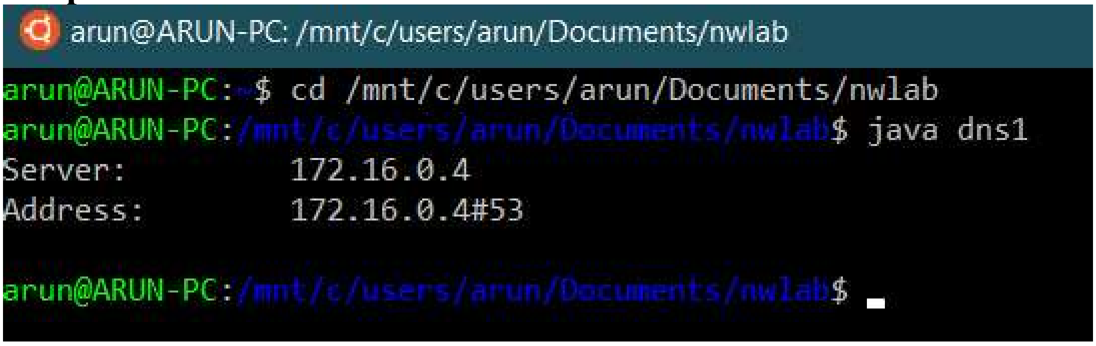
Program: DNS server's host name and IP address.

```
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(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:

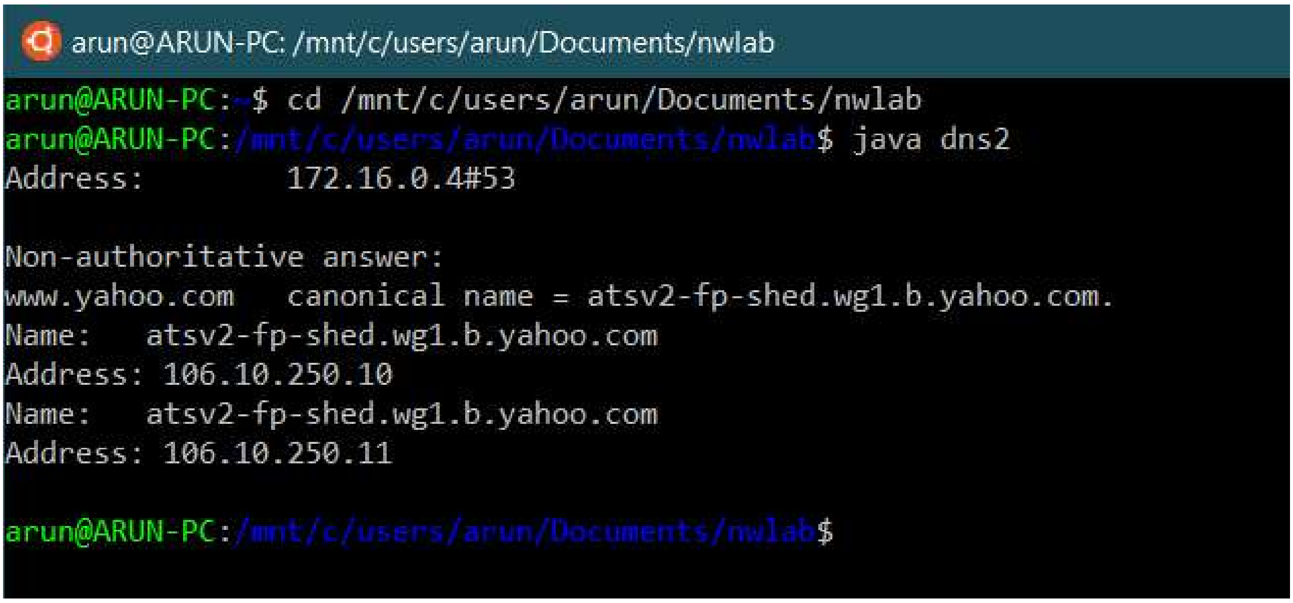


```
arun@ARUN-PC: /mnt/c/users/arun/Documents/nwlab
arun@ARUN-PC:~$ cd /mnt/c/users/arun/Documents/nwlab
arun@ARUN-PC:/mnt/c/users/arun/Documents/nwlab$ java dns1
Server:         172.16.0.4
Address:        172.16.0.4#53

arun@ARUN-PC:/mnt/c/users/arun/Documents/nwlab$
```

Program: DNS server to resolve a host name.

```
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(newInputStreamReader(process.getInputStream()));
   String s=output.readLine();
    while((s=output.readLine())!=null)
     System.out.println(s);
 }
}
```

Output:



```
arun@ARUN-PC: /mnt/c/users/arun/Documents/nwlab
arun@ARUN-PC:~$ cd /mnt/c/users/arun/Documents/nwlab
arun@ARUN-PC:/mnt/c/users/arun/Documents/nwlab$ java dns2
Address:        172.16.0.4#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

arun@ARUN-PC:/mnt/c/users/arun/Documents/nwlab$
```

Program: Remote Interface:

```
import java.rmi.*;
public interface rev_in extends java.rmi.Remote
{
  public int reverse(int x) throws RemoteException;
}
```

Program: Implemention of Remote Interface:

```
import java.rmi.server.*;
import java.io.*;
public class rev_im extends UnicastRemoteObject implements rev_in
{
  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;
 }
}
```

Program: **Remote Server**

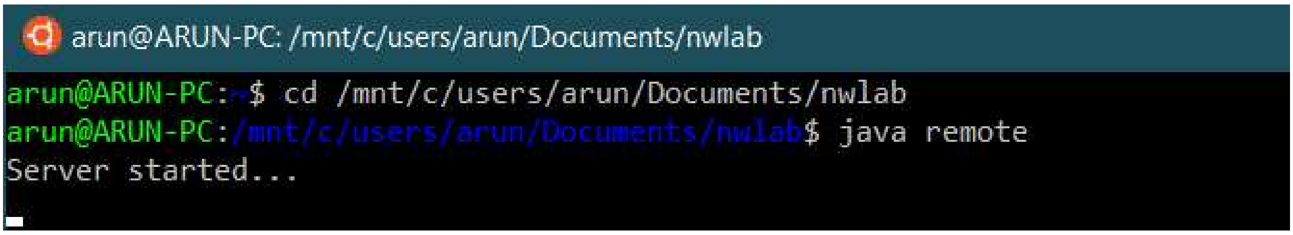
```
import java.rmi.*;
import java.net.*;
import java.io.*;
import java.rmi.registry.*;
public class remote
{
public static void main(String a[]) throws Exception
{
InetAddress ip=InetAddress.getLocalHost();
  rev_in obj =new rev_im();
 LocateRegistry.createRegistry(1900);
  Naming.rebind("rmi://localhost:1900/"+ip.getHostName(),obj);
  System.out.println("Server started...");
 }
}
```

Program: **Remote Client**

```
import java.rmi.*;
import java.io.*;
import java.net.*;
import java.util.*;
public class client
{
 public static void main(String a[]) throws Exception
 {
   InetAddress ip=InetAddress.getLocalHost();

   rev_in obj =(rev_in) Naming.lookup("rmi://localhost:1900/"+ip.getHostName());
   Scanner input = new Scanner(System.in);
   System.out.println("Enter a Number:");
   int in = Integer.parseInt(input.nextLine());
   int n= obj.reverse(in);
   System.out.println("Reversed Number:"+n);
}
}
```

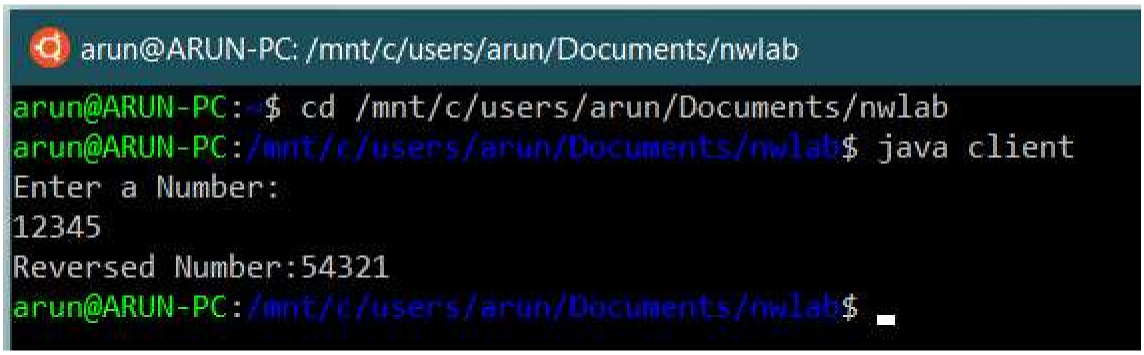
Output:

Server



```
arun@ARUN-PC:~$ cd /mnt/c/users/arun/Documents/nwlab
arun@ARUN-PC:/mnt/c/users/arun/Documents/nwlab$ java remote
Server started...
```

Client



```
arun@ARUN-PC:~$ cd /mnt/c/users/arun/Documents/nwlab
arun@ARUN-PC:/mnt/c/users/arun/Documents/nwlab$ java client
Enter a Number:
12345
Reversed Number:54321
arun@ARUN-PC:/mnt/c/users/arun/Documents/nwlab$
```

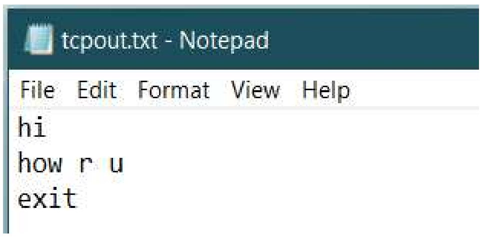
Program: **TCP-Server**

```
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);
   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");
  }
}
```

Program: **TCP-Client**

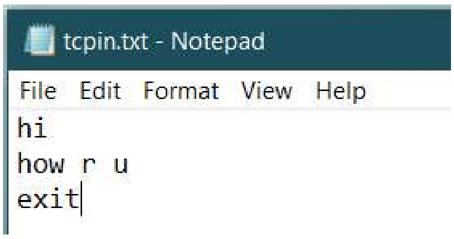
```
import java.io.*;
import java.net.*;
public class tcpclient{
  public static void main(String args[]) throws Exception{
       Socket socket =new Socket(InetAddress.getByName("LocalHost"),5000);
       byte[] contents=new byte[10000];
       FileOutputStream fos=new FileOutputStream("tcpin.txt");
       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");
 }
}
```

FIlE: Server - tcpout.txt



```
hi
how r u
exit
```

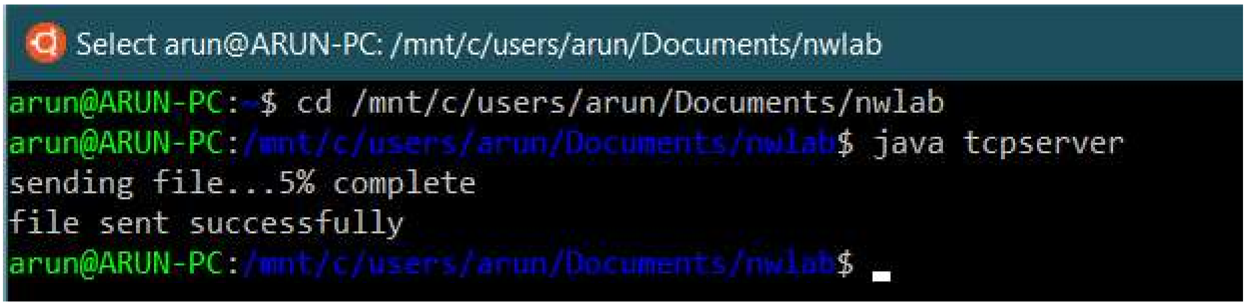
FILE: Client - tcpin.txt



```
hi
how r u
exit
```

Output:

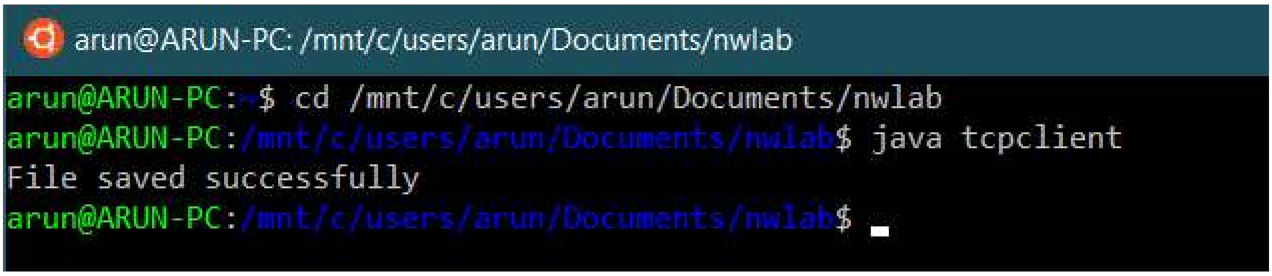
Server



Select arun@ARUN-PC: /mnt/c/users/arun/Documents/nwlab

```
arun@ARUN-PC:~$ cd /mnt/c/users/arun/Documents/nwlab
arun@ARUN-PC:/mnt/c/users/arun/Documents/nwlab$ java tcpserver
sending file...5% complete
file sent successfully
arun@ARUN-PC:/mnt/c/users/arun/Documents/nwlab$
```

Client



arun@ARUN-PC: /mnt/c/users/arun/Documents/nwlab

```
arun@ARUN-PC:~$ cd /mnt/c/users/arun/Documents/nwlab
arun@ARUN-PC:/mnt/c/users/arun/Documents/nwlab$ java tcpclient
File saved successfully
arun@ARUN-PC:/mnt/c/users/arun/Documents/nwlab$
```

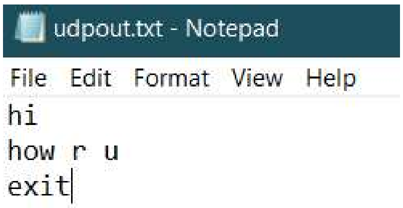
Program: **UDP-Server**

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

public class udpserver
{
	public static void main(String[] args) throws IOException
	{
		DatagramSocket ds = new DatagramSocket(8000);
		byte[] receive = new byte[99999];
		FileWriter fw=new FileWriter("tcpin.txt");
		while (true)
		{
			DatagramPacket DpReceive = new DatagramPacket(receive, receive.length);
			ds.receive(DpReceive);
			System.out.println("Client:-" + data(receive));
			if (data(receive).toString().equals("exit"))
			{
				System.out.println("Client sent bye.....EXITING");
				break;
			}
			receive = new byte[99999];
		}
	}
	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;
	}
}
```

Program: **UDP-Client**

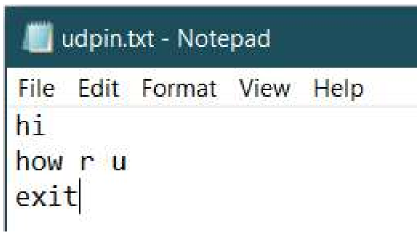
```
import java.io.*;
import java.net.*;
import java.util.*;

public class udpclient
{
        public static void main(String args[]) throws IOException
        {
                File f=new File("tcpout.txt");
                BufferedReader br=new BufferedReader(new FileReader(f));
                DatagramSocket ds = new DatagramSocket();
                InetAddress ip = InetAddress.getLocalHost();
                byte buf[] = null;
                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")){
                                System.out.println("File Sent..! Exiting!");
                                break;
                        }
                }
        }
}
```

FIlE: Server - udpout.txt



```
hi
how r u
exit
```

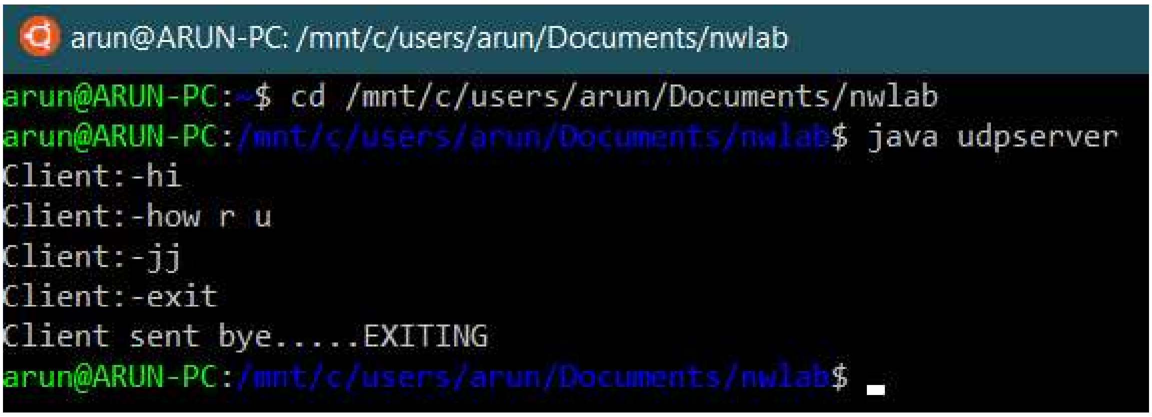
FILE: Client - udpin.txt



```
hi
how r u
exit
```

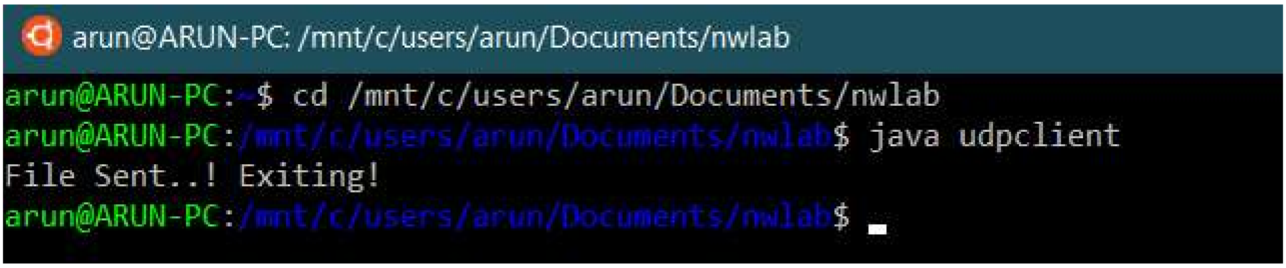
Output:

Server



```
arun@ARUN-PC:~$ cd /mnt/c/users/arun/Documents/nwlab
arun@ARUN-PC:/mnt/c/users/arun/Documents/nwlab$ java udpserver
Client:-hi
Client:-how r u
Client:-jj
Client:-exit
Client sent bye.....EXITING
arun@ARUN-PC:/mnt/c/users/arun/Documents/nwlab$ _
```

Client:



```
arun@ARUN-PC:~$ cd /mnt/c/users/arun/Documents/nwlab
arun@ARUN-PC:/mnt/c/users/arun/Documents/nwlab$ java udpclient
File Sent..! Exiting!
arun@ARUN-PC:/mnt/c/users/arun/Documents/nwlab$ _
```