EX.NO: 09	Implementation of Sub-netting Application
DATE:	

Aim

To develop a subnetting application in Java that calculates subnet mask, network address, and broadcast address based on the given IP address and number of addresses in each subnet.

Theory:

Subnetting is the process of dividing a large network into smaller subnetworks to improve performance and security. In this experiment, we will create a Java application that takes an IP address and the number of addresses in each subnet as input and calculates the subnet mask, network address, and broadcast address.

ALGORITHM

- Read the IP address from the user.
- Split the IP address into octets and convert each octet to binary.
- Calculate the number of bits required for the given number of addresses in each subnet to determine the subnet mask.
- Calculate the network address by setting the last n bits to 0 in the binary IP address.
- Calculate the broadcast address by setting the last n bits to 1 in the binary IP address.
- Display the subnet mask, network address, and broadcast address in decimal format.

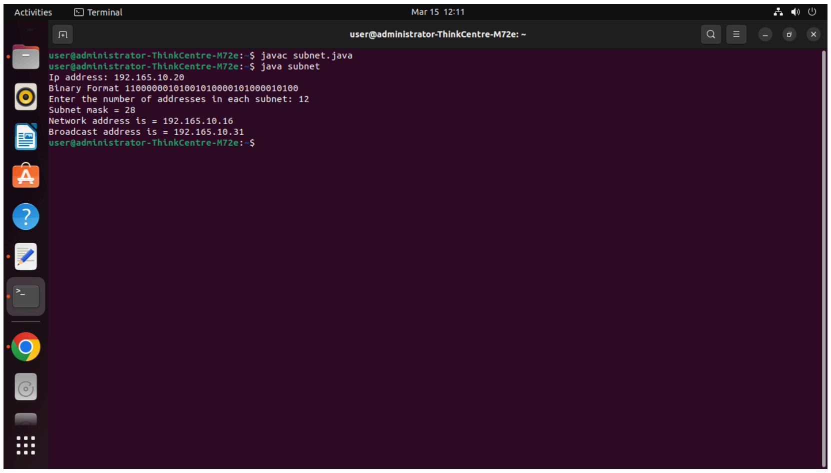
PROGRAM

```
import java.util.Scanner;
class subnet{
public static void main(String args[]){
Scanner sc = new Scanner(System.in);
System.out.print(&quot;Ip address: &quot;);
String ip = sc.nextLine();
String split_ip[] = ip.split(&quot;\\.&quot;); //SPlit the string after every .
String split_bip[] = new String[4]; //split binary ip
String bip = &quot;&quot;;
for(int i=0;i&lt;4;i++){
split_bip[i] = appendZeros(Integer.toBinaryString(Integer.parseInt(split_ip[i])));
bip += split_bip[i];
}
System.out.println(&quot;Binary Format &quot;+bip);
System.out.print(&quot;Enter the number of addresses in each subnet: &quot;);
int n = sc.nextInt();
```

```
//Calculation of mask

int bits = (int)Math.ceil(Math.log(n)/Math.log(2));

int mask = 32-bits;

System.out.println(&quot;Subnet mask = &quot;+mask);


//Calculation of first address and last address

int fbip[] = new int[32];

for(int i=0; i&lt;32;i++) fbip[i] = (int)bip.charAt(i)-48; //convert cahracter 0,1 to integer
0,1

for(int i=31;i&gt;31-bits;i--)//Get first address by ANDing last n bits with 0

fbip[i] &amp;= 0;

String fip[] = {&quot;&quot;,&quot;&quot;,&quot;&quot;,&quot;&quot;};

for(int i=0;i&lt;32;i++)


fip[i/8] = new String(fip[i/8]+fbip[i]);

System.out.print(&quot;Network address is = &quot;);

for(int i=0;i&lt;4;i++){

System.out.print(Integer.parseInt(fip[i],2));

if(i!=3) System.out.print(&quot;.&quot;);

}

System.out.println();

int lbip[] = new int[32];
```

```
0,1
for(int i=0; i&lt;32;i++) lbip[i] = (int)bip.charAt(i)-48; //convert cahracter 0,1 to integer

for(int i=31;i&gt;31-bits;i--)//Get last address by ORing last n bits with 1

lbip[i] |= 1;

String lip[] = {&quot;&quot;,&quot;&quot;,&quot;&quot;,&quot;&quot;};

for(int i=0;i&lt;32;i++)

lip[i/8] = new String(lip[i/8]+lbip[i]);

System.out.print(&quot;Broadcast address is = &quot;);

for(int i=0;i&lt;4;i++){

System.out.print(Integer.parseInt(lip[i],2));

if(i!=3) System.out.print(&quot;.&quot;);

}

System.out.println();

}

static String appendZeros(String s){

String temp = new String(&quot;00000000&quot;);

return temp.substring(s.length())+ s;

}

}
```

OUTPUT



```
user@administrator-ThinkCentre-M72e:~$ javac subnet.java
user@administrator-ThinkCentre-M72e:~$ java subnet
Ip address: 192.165.10.20
Binary Format 11000000101001010000101000010100
Enter the number of addresses in each subnet: 12
Subnet mask = 28
Network address is = 192.165.10.16
Broadcast address is = 192.165.10.31
user@administrator-ThinkCentre-M72e:~$
```

Conclusion

In this experiment, we successfully implemented a client-server architecture using sockets in Java to transfer images. The client was able to send an image to the server, and the server received the image, displayed it, and sent a response back to the client. This experiment demonstrates the use of sockets for communication between client and server applications.