

Assignment No.

**Write a program for finding the
First Address and last address in
Subnet**

```
//subnet.....
```

```
import java.util.Scanner; import  
java.net.InetAddress;
```

```
class subnet  
{
```

```
public static void main(String args[])  
{  
Scanner sc= new Scanner(System.in);  
System.out.print("Enter the ip address=");  
String ip=sc.nextLine();
```

```
//----Split the Ip Address-----
```

```
String split_ip[] = ip.split("\\.");
```

```
//----- Converting the Ip Address to binary----
```

```
String split_bip[]= new String[4];
```

```
String bip = "";
```

```
for(int i=0;i<4;i++)
```

```
{
```

```
split_bip[i]=appendZeroes(Integer.toString(Integer.parseInt(split_ip[i]))); bip+=split_bip[i];
```

```
Int(split_ip[i])); bip+=split_bip[i];
```

```
}
```

```
System.out.println("The binary IpAddress is="+bip);
```

```
//----- Finding the Subnet mask-----
```

```
System.out.println("Enter the number of address"); int
```

```
n=sc.nextInt();
```

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

```
System.out.println("The number of bits required="+bits);
```

```
int mask=32-bits; int
```

```
total_address=(int)Math.pow(2,bits);
```

```
System.out.println("Subnet mask is "+mask);
```

```
//---- Finding the first and last address----
```

```
//---- First address Calculation----- int
```

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

```
for(int i=0;i<32;i++)
```

```
{
```

```
//Convert to the character 1,0 to integer 1,0
```

```
fbip[i]=(int)bip.charAt(i)-48;
```

```
}
```

```
for(int i=31;i>31-bits;i--)
```

```
{
```

```
//Get first address by anding the last bits with 0
```

```
fbip[i] &=0;
```

```
}
```

```
String fip[]={"", "", "", ""}; for(int
```

```
i=0;i<32;i++)
```

```
{
```

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

```
}
```

```

int first_offset=0; int
ipAddr[]=new int[4];    ;
System.out.println("Group 1 \nThe First Address is:"); for(int
i=0;i<4;i++)
{
System.out.print(ipAddr[i]=first_offset=Integer.parseInt(fip[i],2)); if(i!=3)
System.out.print(".");
}
System.out.println();

```

```

//--- Last address Calculation---- int

```

```

lbip[]=new int [32];

```

```

for(int i=0;i<32;i++)

```

```

{

```

```

// Convert the character 1,0 to integer 1,0

```

```

lbip[i]=(int)bip.charAt(i)-48;

```

```

}

```

```

for(int i=31;i>31-bits;i--)

```

```

{

```

```

// Get last address by oring with last bits with 1

```

```

lbip[i] |= 1;

```

```

}

```

```

String lip[]={"","","",""},for(int
i=0;i<32;i++)
{
lip[i/8]=new String(lip[i/8]+lbip[i]);
}
int ipLast[]=new int[4];
System.out.println("The Last Address is:"); for(int
i=0;i<4;i++)
{
System.out.print(ipLast[i]=Integer.parseInt(lip[i],2)); if(i!=3)
System.out.print(".");
}
System.out.println();
System.out.println("How many subnets do you want to form?"); int
scount=sc.nextInt();

for(int j=1;j<scount;j++)
{
System.out.println(" GROUP "+ (j+1)+" FIRST ADDRESS:");
for(int i=0;i<4;i++)
{ if(i<3)
{
System.out.print(ipAddr[i]+".");
}
else
System.out.println(ipAddr[i]=ipAddr[i]+total_address);
}
System.out.println(" GROUP "+ (j+1)+" LAST ADDRESS:"); for(int
i=0;i<4;i++)

```

```
{ if(i<3)
{
System.out.print(ipLast[i]+".");
}
else
System.out.println(ipLast[i]=ipLast[i]+total_address);

}
System.out.println();
} try
{

System.out.println("Enter the Ip address to ping="); Scanner
s=new Scanner(System.in);

String ip_add=s.nextLine();

InetAddress inet = InetAddress.getByName(ip_add);

if(inet.isReachable(5000))
{
System.out.println("The ip address is reachable"+ip_add);
}
else
{
System.out.println("The ip address is not reachable"+ip_add);
}
}
catch( Exception e)
{
System.out.println("Exception:"+e.getMessage());
}
```

```
}
```

```
static String appendZeroes(String s)
{
String temp= new String("00000000"); return
temp.substring(s.length()+ s;
}
}
```

```
//subnetmask..... import
java.util.Scanner; public class
subnetmask { public static
void main(String[] args) {
    Scanner scan = new Scanner(System.in);
String ip = new String();    int
mask=0,defaultmask=0;
    System.out.println("enter ip address");
ip = scan.nextLine();
    System.out.println("enter mask");
mask = scan.nextInt();
    String[] splitip = ip.split("\\.");
    System.out.println(splitip[0]+" "+splitip[1]+" "+splitip[2]+" "+splitip[3]+" ");
int first = Integer.parseInt(splitip[0]);    if(first>=0&&first<=127){
System.out.println("CLASS A");    defaultmask = 8;
    }
    else if(first>127&&first<=191){
System.out.println("CLASS B");    defaultmask=16;
    }
}
```



```

        else if(first>191&&first<=223){
System.out.println("CLASS C");
defaultmask=24;
        }
        else if(first>223){
            System.out.println("CLASS D");
defaultmask=32;
        }
        String binip = new String();
        String defmask = new String();    for(int i=0;i<4;i++){        binip = binip +
appendZeroes(Integer.toBinaryString(Integer.parseInt(splitip[i])));
        }
        System.out.println("IP in binary : "+binip);
        System.out.println("Default Mask : "+defaultmask);

        for(int i=0;i<32;i++){
if(i<mask){        defmask =
defmask + "1";
        }
else{
            defmask = defmask + "0";
        }
    }
    System.out.println(defmask);    String netid = new String();    for(int i=0;i<32;i++){
netid = netid + (Integer.parseInt(""+binip.charAt(i))&Integer.parseInt(""+defmask.charAt(i)));
    }
    int p=-1;
    System.out.println(netid);

```

```

String[] net = new String[4];
String[] def = new String[4];
for(int i=0;i<32;i++){
if(i%8==0){      p++;
net[p] = "";      def[p]="";

        net[p] = net[p] + netid.charAt(i);
def[p] = def[p] + defmask.charAt(i);
        }      else{      net[p] = net[p]
+ netid.charAt(i);      def[p] = def[p] +
defmask.charAt(i);
        }
}

System.out.println("Given IP : "+ip);
System.out.print("subnet mask :");
for(int i=0;i<4;i++){

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

    if(i!=3)

        System.out.print(".");
}

System.out.println();
System.out.print("NetId : ");
for(int i=0;i<4;i++){

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

    if(i!=3)

        System.out.print(".");
}
}

```

```
private static String appendZeroes(String binaryString) {  
    // TODO Auto-generated method stub  
    /*if(binaryString.length()<8){        for(int  
i=0;i<8-binaryString.length();i++){  
binaryString = "0" + binaryString;  
  
        }  
    }*/  
    String temp = new String("00000000");    return  
temp.substring(binaryString.length()+ binaryString;  
  
    }  
}
```

Output.....

varsha@varsha-Inspiron-3542: ~/Documents/EXECUTED CN

```
varsha@varsha-Inspiron-3542:~$ cd Documents
varsha@varsha-Inspiron-3542:~/Documents$ cd EXECUTED\ CN/
varsha@varsha-Inspiron-3542:~/Documents/EXECUTED CN$ gedit subnet.java
varsha@varsha-Inspiron-3542:~/Documents/EXECUTED CN$ javac subnet.java
varsha@varsha-Inspiron-3542:~/Documents/EXECUTED CN$ java subnet
Enter the ip address=192.168.4.10
The binary IPAddress is=11000000101010000000010000001010
Enter the number of address
26
The number of bits required=5
Subnet mask is 27
The First Address is:
192.168.4.0
The Last Address is:
192.168.4.31
Enter the Ip address to ping=
192.168.4.1
The ip address is not reachable192.168.4.1
varsha@varsha-Inspiron-3542:~/Documents/EXECUTED CN$
```

- ① Subnet the class C IP address 195.1.1.0
that you have 10 subnets each with a
maximum 12 hosts on each subnet list
the address on host 1 on subnet 0, 1, 2, 3, 10

→ Current mask: 255.255.255.0

Bits need for 10 subnets = 4
= 2^4

= 16 possible subnets.

Bits need for 12 hosts = 4
= 2^4

= $16 - 2$

= 14 hosts.

So our mask in binary
= 11 1100 00

= 240 decimal.

Final mask: 255.255.255.240

Host on subnet 0, 1, 2, 3, 10

Subnet 0 host 1 IP = 195.1.1.1

Subnet 1 host 1 IP = 195.1.1.17

Subnet 2 host 1 IP = 195.1.1.33

Subnet 3 host 1 IP = 195.1.1.49

Subnet 10 host 1 IP = 195.1.1.151

Q2 Subnet the class C IP address 195.1.1.0 so that you have at least 2 subnet each subnet must have room for 48 host what are the two possible subnet mask?

Current mask:

255. 255. 255. 0

Bit need for 48 hosts.

$$= 6$$

$$= 2^6$$

$$= 64 - 2$$

$$= 62 \text{ host.}$$

Bit need for 2 subnet

$$= 1$$

$$= 2^1 = 2 \text{ possible subnet.}$$

total of 7 bit needed so therefore we can use either one bit or 2 bit for subnet so we could have.

1 bit Subnet 7 bit host or

2 bit Subnet 6 bit host.

mask are 10000000 & 11000000 =

128 decimal & 192 decimal.

final possible mask are

255. 255. 255. 128

255. 255. 255. 192.