

# Assignment - III

Title :- Subnetting

Problem statement :-

Write a program in java or python to demonstrate subnetting & find the subnet masks.

Objective :- To be able to

- Understand the concept of subnetting & subnet masks.
- Understand class-less & class-full addressing.

Outcome :- students will be able to

- Implement code to find subnet mask of the given IP address.
- To understand concept of class-less & class-full addressing scheme.

S/W & H/W requirements :- 64 bit system with Fedora OS, Eclipse IDE, J2SE/python.

Theory :-

Netmask : A netmask is a 32-bit mask used to divide a IP address into subnets & specify the networks ~~and~~ available hosts.



In network mask, 2 bits are always automatically assigned & can't be used (0 & 255)

The no. of networks a netmask can support are

$$\frac{2^{(\text{netmask length} - \text{no. of used segments})}}{2} - 2$$

Ex.:- A 24-bit netmask is

Netmask →	255	255	255	0
Binary →	11111111	11111111	11111111	00000000
Netmask length →	8	8	8	---

$$\text{total no. of networks} = \frac{2^{24-3}}{2}$$

$$= \frac{2^{21}}{2}$$

$$= 2097150$$

~~two~~ 2 are subtracted because of network & broadcast address that are already used.

No. of host a netmask can support are

$$\frac{2^{(\text{no. of zeros})}}{2}$$

ex. For the 24 bit net mask,

$$\text{no. of hosts} = 2^8 - 2 = 254$$

Given below are commonly used network classes they are also called class-full addressing as the no. of bits reserved for network id or the netmask length is fixed.



class	Netmask length	No. of networks	No. of hosts
class A	8	$2^8 - 2$	$2^{24} - 2$
class B	16	$2^{16} - 2$	$2^{16} - 2$
class C	24	$2^{24} - 2$	$2^8 - 2$

Netmask of class A  $\rightarrow$  255.0.0.0

class B  $\rightarrow$  255.255.0.0

class C  $\rightarrow$  255.255.255.0

For class-less addressing, netmask length is mentioned in the address & it is calculated by the binary conversion of no. of bits.

Ex. 192.168.2.0 /20

Binary  $\rightarrow$  11111111 | 11111111 | 11110000 | 00000000  
 Netmask  $\rightarrow$  255 | 255 | 240 | 0

### Subnet Mask :-

- It is used to determine what subnet an IP address belongs to.
- Subnet mask is a data used for bitwise operation on a network of IP address that has been divided into 2 or more groups. This process is called subnetting, which divides IP network into blocks of logical addresses.

Ex. 192.168.2.0 /20

Consider no. of subnet = 4 (binary  $\rightarrow$  100)  
 8 bit from host bits are borrowed



Binary →	11111111	11111111	11111110	00000000
Subnetmask →	255	255	254	0

### Algorithm :-

1. Get the IP address of network & no. of subnets from user.
2. Check which type of addressing, the given address belongs to & if it is valid address.
3. Calculate the network mask, network address, broadcast address & display them.
4. Verify if given no. of subnets can be formed.
5. Calculate no. of bits borrowed, subnet mask, subnet IP's & range of subnets & display them.
6. Get source & destination address from user & verify if they belong to the given network.
7. Check whether the source & destination address belongs to the same subnet or not & display the same.
8. Repeat 1-7 if user wants to continue.
9. Exit.

### Test Cases :-

I/P	O/P	Expected O/P	Result
IP = 192.168.5.0/20 subnet = 7	class-less Netmask 255.255.255.192	—  —	Success

Source

192.168.5.1

Subnet mask

255.255.255.248

Destination

192.168.5.5

Subnet IP's :-

192.168.5.0,

192.168.5.8,

192.168.5.16,

192.168.5.24,

192.168.5.32,

192.168.5.40,

192.168.5.48

192.168.5.56

~~192.168.5~~

They are in same Subnet

IP : 192.168.5.0

-11-

Success

Conclusion :-

We have successfully calculated the subnet mask of given IP address.

```

#include <iostream>
#include<bits/stdc++.h>
using namespace std;
int main()
{
    int a[4],netbits,subnets,bitsub,netmask[4]={0},offset,subnetmask[4]={0};
    int subcal,fixedoct,inp1[4],inp2[4];
    cout<<"\n\n\tEnter the IP address with spaces:: ";
    for(int i=0;i<4;i++)
    {
        cin>>a[i];
    }
    cout<<"\n\n\tEnter the number of network-id bits:: ";
    cin>>netbits;
    if(netbits<8 || netbits>32)
    {
        cout<<"\n\n\t Invalid "<<endl;
    }
    else
    {
        cout<<"\n\n\tEnter the no. of subnets to be formed: ";
        cin>>subnets;
        bitsub = ceil(log2(subnets));
        int subnetids[int(pow(2,bitsub))];
        offset = netbits;
        int i=0;
        while(offset>=8)
        {
            netmask[i]= 255;
            subnetmask[i] = 255;
            i++;
            offset = offset - 8;
        }
        fixedoct =i;
        subcal = bitsub+offset;
        while(offset!=0)
        {
            netmask[i]+=pow(2,8-offset);
            offset--;
        }
        subnetmask[i]=netmask[i];
        offset = subcal;
        while(bitsub!=0)
        {
            subnetmask[i]+=pow(2,8-offset);
            bitsub--;
            offset--;
        }
        cout<<"\n\n\tThe Network Mask is:: "<<endl;
        for(int i=0;i<4;i++)
        {
            cout<<netmask[i];
            if(i!=3)
                cout<<".";
        }
        cout<<"\n\n\tThe Sub-Net Mask is:: "<<endl;
        for(int i=0;i<4;i++)
        {
            cout<<subnetmask[i];

```

```

if(i!=3)
cout<<".";
}
bitsub = ceil(log2(subnets));
for(int i=0;i<pow(2,bitsub);i++)
{
int p = i;
subnetids[i] = p<<(8-subcal);

}
cout<<"\n\n\tThe Subnet Id's are:: "<<endl;
for(int i=0;i<pow(2,bitsub);i++)
{
for(int j=0;j<4;j++)
{
if(j!=fixedoct)
cout<<a[j];
else
cout<<a[j]+subnetids[i];
if(j!=3)
cout<<".";
}
cout<<"\n";
}
cout<<"\n\n\tEnter the 2 subnet-id's(with spaces)"<<endl;
for(int i=0;i<4;i++)
{
cin>>inp1[i];
}
for(int i=0;i<4;i++)
{
cin>>inp2[i];
}
int n=pow(2,bitsub);
for(i=0;i<n-1;i++)
{
if(inp1[fixedoct]>subnetids[i] && inp1[fixedoct]<subnetids[i+1] )
{
if(inp2[fixedoct]>subnetids[i] && inp2[fixedoct]<subnetids[i+1] )
{
cout<<"\n\n\tThey are in the same subnet-id::";
for(int j=0;j<4;j++)
{
if(j!=fixedoct)
cout<<a[j];
else
cout<<subnetids[i];
if(j!=3)
cout<<".";
}
cout<<"\n";
break;
}
}
}
if(i>=n-1)
cout<<"\n\n\tThey are not in the same subnet-id\n";
}
return 0;

```

}



D:\TE\TE sem-1\Lab\CNL\A3\subnet.exe

Enter the IP address with spaces:: 192 168 5 0

Enter the number of network-id bits:: 26

Enter the no. of subnets to be formed: 7

The Network Mask is::  
255.255.255.192

The Sub-Net Mask is::  
255.255.255.248

The Subnet Id's are::  
192.168.5.0  
192.168.5.8  
192.168.5.16  
192.168.5.24  
192.168.5.32  
192.168.5.40  
192.168.5.48  
192.168.5.56

Enter the 2 subnet-id's(with spaces)  
192 168 5 1  
192 168 5 5

They are in the same subnet-id::192.168.5.0

-----  
Process exited after 34.11 seconds with return value 0  
Press any key to continue . . .

D:\TE\TE sem-1\Lab\CNL\A3\subnet.exe

Enter the IP address with spaces:: 192 168 5 0

Enter the number of network-id bits:: 26

Enter the no. of subnets to be formed: 7

The Network Mask is::

255.255.255.192

The Sub-Net Mask is::

255.255.255.248

The Subnet Id's are::

192.168.5.0

192.168.5.8

192.168.5.16

192.168.5.24

192.168.5.32

192.168.5.40

192.168.5.48

192.168.5.56

Enter the 2 subnet-id's(with spaces)

192 168 5 1

192 168 5 9

They are not in the same subnet-id

-----  
Process exited after 32.17 seconds with return value 0  
Press any key to continue . . .