Name :- Rushikesh Karbhari Palve
Roll NO. 31258

Date : / /20 Page No

Assignment NO. 5

DOS :- 31-10-2021

Title :- Write a program to demonstrate subnetting and find subnet masks.

Problem statement :-

Write a program to demonstrate sub-netting and find subnet masks.

Objectives :-

(i) To calculate subnet mask.
(ii) To implement a program for calculating subnet mask for number of networks.

Outcomes :-

students will be able to understand subnet masking.

THEORY :-

Address :-
The unique number ID assigned to one host or interface in a network.

Subnet mask –

A 32-bit combination used to describe which portion of an address refers to the subnet and which part refers to the host.

IP Addresses –

An IP address is an address used in order to uniquely identify a device on an IP network. the address is made up of 32 binary bits, which can be divisible into a network portion and host portion with the help of a subnet mask. the 32 binary bits are broken into four octets (1 octet = 8 bits).

Each octet is converted to decimal and seperated by a period (dot). For this reason, an IP address is said to be expressed in dotted decimal format (for example, 172.16.81.100). The value in each octet ranges from 0 to 255 decimal, or 00000000 – 11111111 binary.

Class A	Network	Host	Host	Host
Subnet Mask	255	0	0	0

Class B	Network	Network	Host	Host
Subnet Mask	255	255	0	0

Class C	Network	Network	Network	Host
Subnet Mask	255	255	255	0

Network Masks :-

A network mask helps you know which portion of the address identifies the network and which portion of the address identifies the node. Class A, B and C networks have default masks, also known as natural masks, as shown here :

Class A : 255.0.0.0
Class B : 255.255.0.0
Class C : 255.255.255.0

An IP address on a class A network that has not been subnetted would have an address/mask pair similar to : 8.20.15.1 255.0.0.0. In order to see how the mask helps you identify the network and node parts of the address, convert the address and mask to binary numbers.

8.20.15.1 = 00001000.00010100.00001111.00000001
255.0.0.0 = 11111111.00000000.00000000.00000000

Once you have the address and the mask represented in binary, then identification of the network and host ID is easier. Any address bits which have corresponding mask bits set to 1 represent the network ID. Any address bits that have corresponding mask bit set to 0 represent the node ID.

8.20.15.1 = 00001000.00010100.00001111.00000001
255.0.0.0 = 11111111.00000000.00000000.00000000

net id | host id

net Id = 00001000 = 8
host id = 00010100.00001111.00000001 = 20.15.1

Subnet :-

A portion of a network that shares a particular subnet address. Each data link on a network must have a unique network ID, with every node on that link being a member of the same network. If you break a major network (class A, B or C) into smaller subnetworks, it allows you to create a network of interconnecting subnetworks. Each data link on this network would then have a unique network/subnetwork ID. Any device, or gateway that connects n networks/subnetworks has n distinct IP addresses, one for each network/subnetwork that it interconnects.

In order to subnet a network, extend the natural mask with some of the bits from the host ID portion of the address in order to create a subnetwork ID. For example, given a class C network of 204.17.5.0 which has a natural mask of 255.255.255.0, you can create subnets in this manner :

```
204.17.5.0 = 11001100.00010001.00000101.00000000
255.255.255.224= 11111111.11111111.11111111.11100000
                 --------------------------|sub|----
```

By extending the mask to be 255.255.255.224, you have taken three bits (indicated by "sub") from the original host portion of the address and used them to make subnets. With these three bits, it is possible to create eight subnets. With the remaining five host ID bits, each subnet can have up to 32 host addresses, 30 of which can actually be assigned to a device since host ids of all zeros or all ones are not allowed (it is very important to remember this). so, with this in mind, these subnets have been created.

204.17.5.0 255.255.255.224 host address range 1 to 30

204.17.5.32 255.255.255.224 host address range 33 to 62

204.17.5.64 255.255.255.224 host address range 65 to 94

204.17.5.96 255.255.255.224 host address range 97 to 126

204.17.5.128 255.255.255.224 host address range 129 to 158

204.17.5.160 255.255.255.224 host address range 161 to 190

204.17.5.192 255.255.255.224 host address range 193 to 222

204.17.5.224 255.255.255.224 host address range 225 to 254

ALGORITHM :-

1. Take a input as a IP address as a string.
2. Split the string after every '.'.
3. Convert string to integer, integer to binary and append 0 to make it 8 bit binary number.
4. Take a input of number of addresses.
5. Calculation of mask using ceil and log functions -

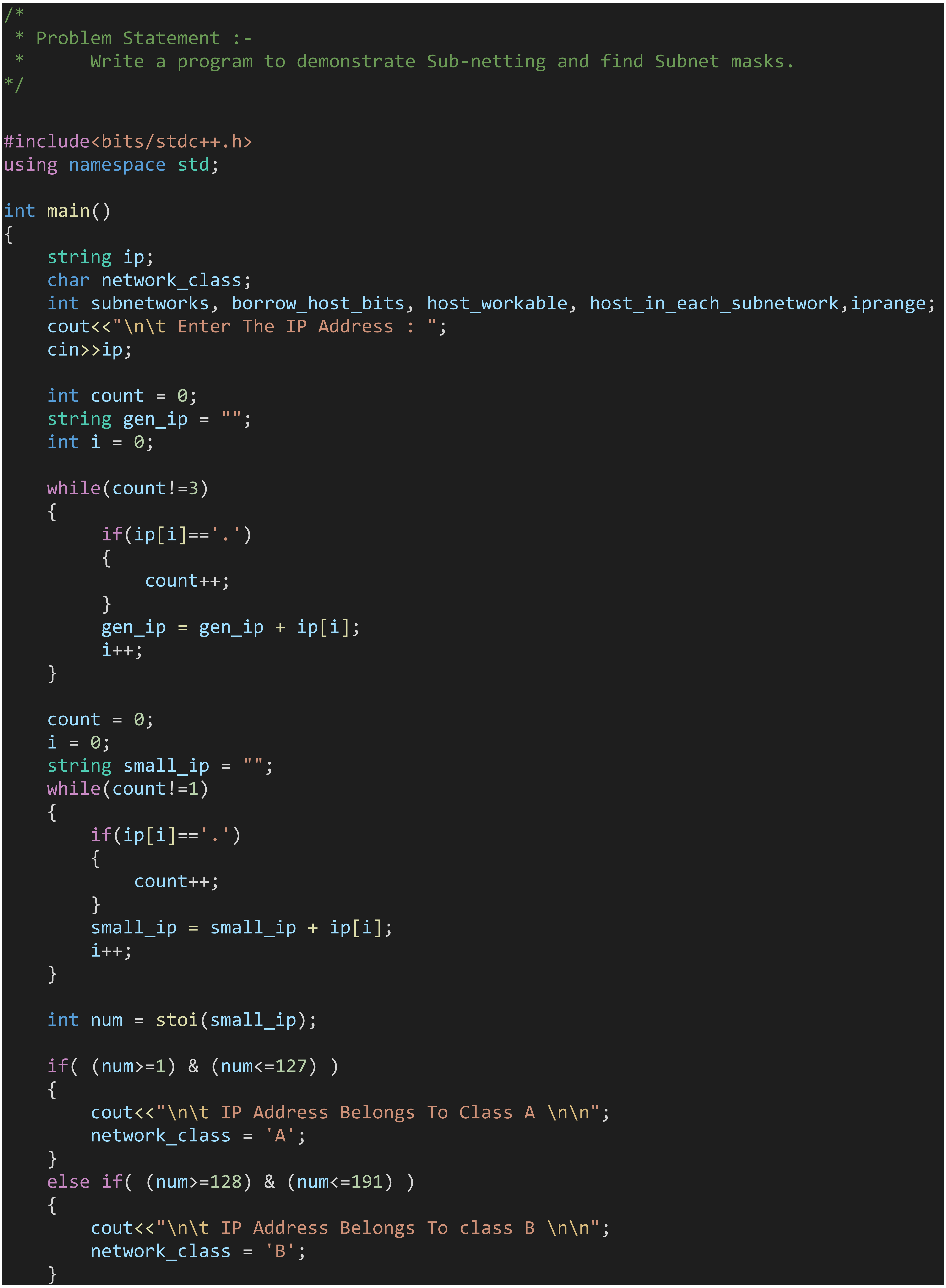
```
int bits = (int)Math.ceil(Math.log(n)/Math.log(2));
mask = 32-bits.
```

6. Convert character 0, 1 to integer 0, 1 and Get first address by ANDing last n bits with 0.
7. Convert that binary number into decimal number by dividing by 8.
8. Get last address by ORing last n bits with 1 and convert it into decimal number.

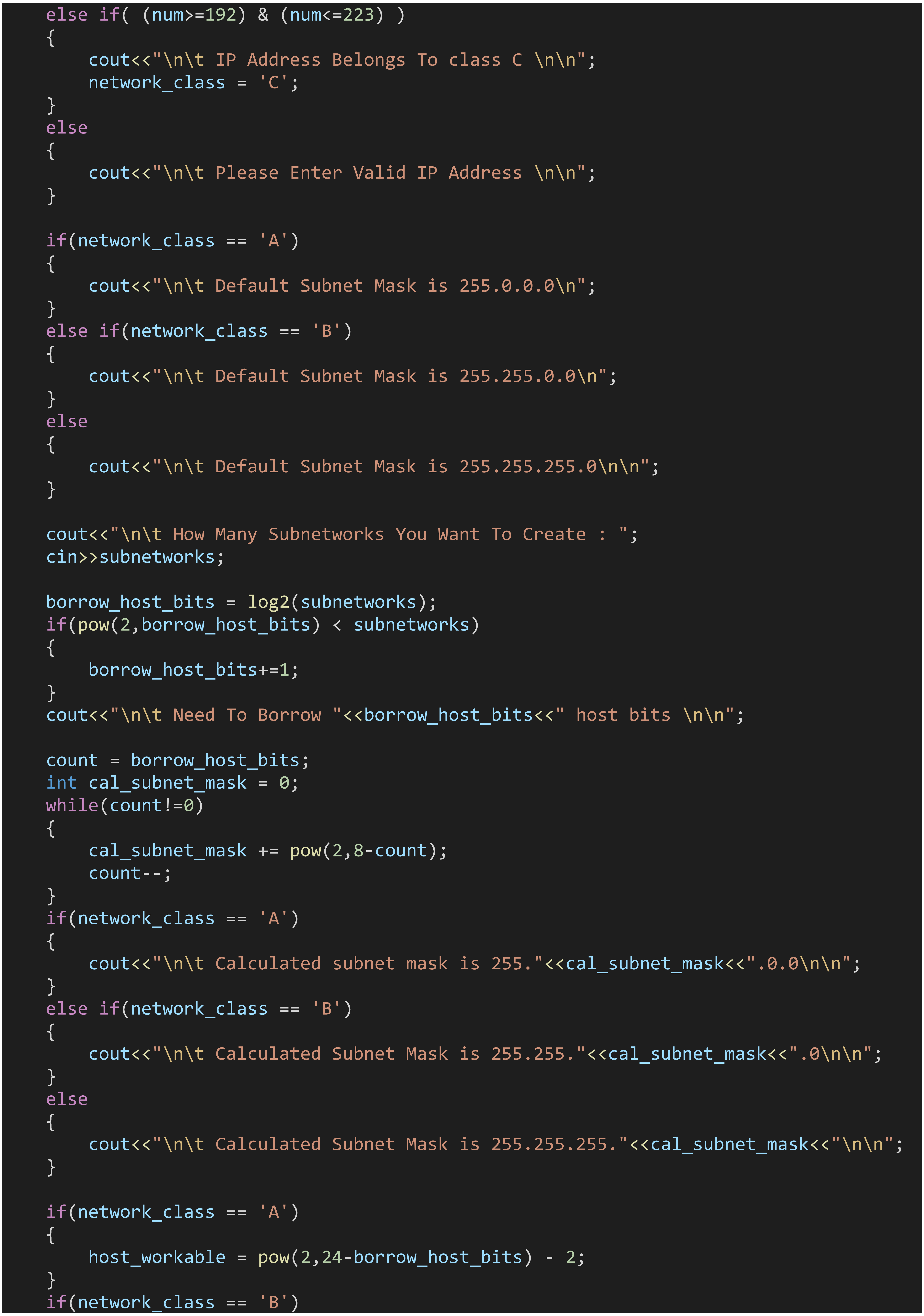
CONCLUSION :-

Thus, we have studied subnetting and the subnet masks.

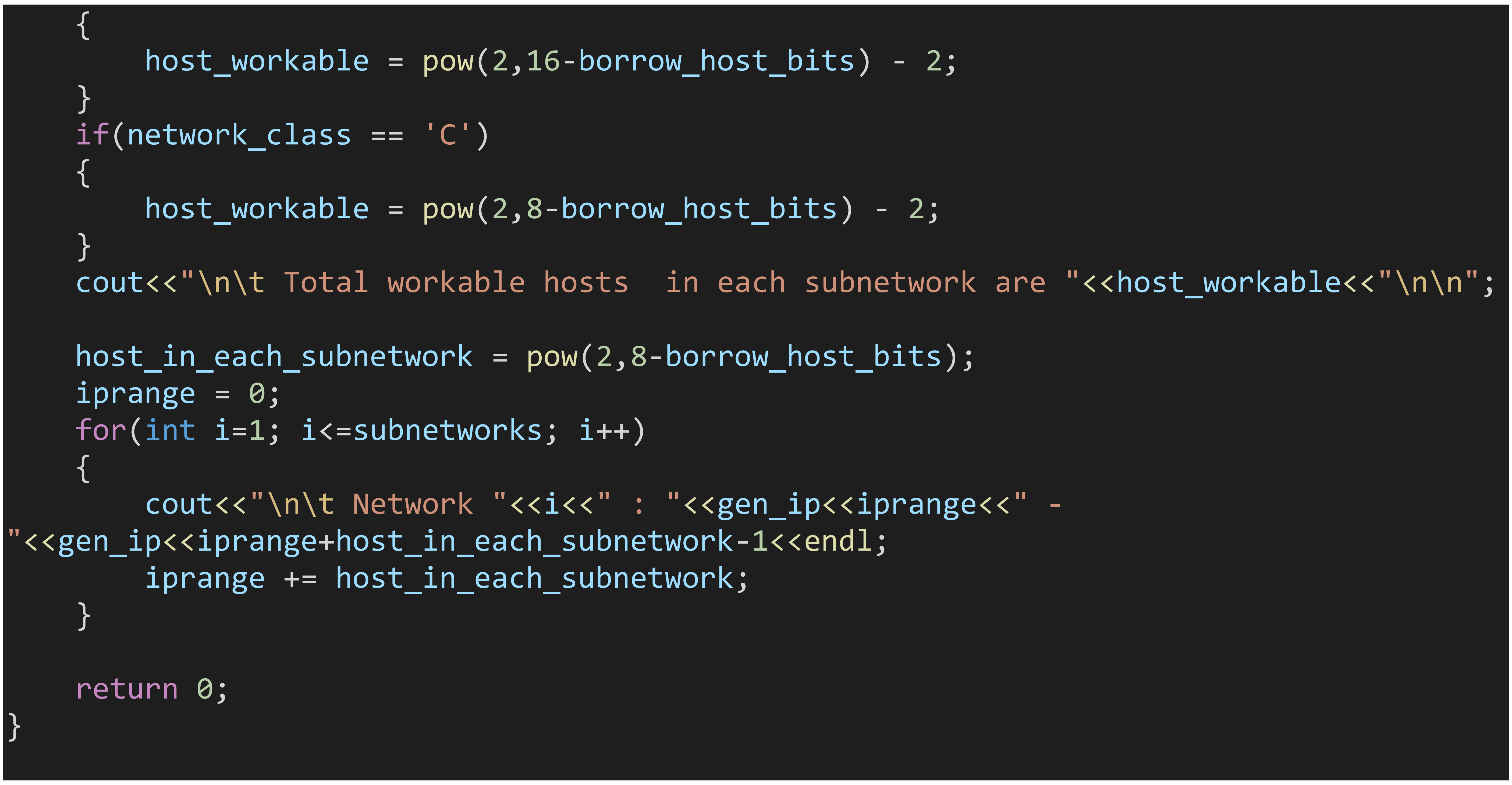
CODE :-



```
/*
 * Problem Statement :-
 *      Write a program to demonstrate Sub-netting and find Subnet masks.
*/

#include<bits/stdc++.h>
using namespace std;

int main()
{
    string ip;
    char network_class;
    int subnetworks, borrow_host_bits, host_workable, host_in_each_subnetwork,iprange;
    cout<<"\n\t Enter The IP Address : ";
    cin>>ip;

    int count = 0;
    string gen_ip = "";
    int i = 0;

    while(count!=3)
    {
         if(ip[i]=='.')
         {
             count++;
         }
         gen_ip = gen_ip + ip[i];
         i++;
    }

    count = 0;
    i = 0;
    string small_ip = "";
    while(count!=1)
    {
        if(ip[i]=='.')
        {
            count++;
        }
        small_ip = small_ip + ip[i];
        i++;
    }

    int num = stoi(small_ip);

    if( (num>=1) & (num<=127) )
    {
        cout<<"\n\t IP Address Belongs To Class A \n\n";
        network_class = 'A';
    }
    else if( (num>=128) & (num<=191) )
    {
        cout<<"\n\t IP Address Belongs To class B \n\n";
        network_class = 'B';
    }
```



```
    else if( (num>=192) & (num<=223) )
    {
        cout<<"\n\t IP Address Belongs To class C \n\n";
        network_class = 'C';
    }
    else
    {
        cout<<"\n\t Please Enter Valid IP Address \n\n";
    }

    if(network_class == 'A')
    {
        cout<<"\n\t Default Subnet Mask is 255.0.0.0\n";
    }
    else if(network_class == 'B')
    {
        cout<<"\n\t Default Subnet Mask is 255.255.0.0\n";
    }
    else
    {
        cout<<"\n\t Default Subnet Mask is 255.255.255.0\n\n";
    }

    cout<<"\n\t How Many Subnetworks You Want To Create : ";
    cin>>subnetworks;

    borrow_host_bits = log2(subnetworks);
    if(pow(2,borrow_host_bits) < subnetworks)
    {
        borrow_host_bits+=1;
    }
    cout<<"\n\t Need To Borrow "<<borrow_host_bits<<" host bits \n\n";

    count = borrow_host_bits;
    int cal_subnet_mask = 0;
    while(count!=0)
    {
        cal_subnet_mask += pow(2,8-count);
        count--;
    }
    if(network_class == 'A')
    {
        cout<<"\n\t Calculated subnet mask is 255."<<cal_subnet_mask<<".0.0\n\n";
    }
    else if(network_class == 'B')
    {
        cout<<"\n\t Calculated Subnet Mask is 255.255."<<cal_subnet_mask<<".0\n\n";
    }
    else
    {
        cout<<"\n\t Calculated Subnet Mask is 255.255.255."<<cal_subnet_mask<<"\n\n";
    }

    if(network_class == 'A')
    {
        host_workable = pow(2,24-borrow_host_bits) - 2;
    }
    if(network_class == 'B')
```



```
    {
        host_workable = pow(2,16-borrow_host_bits) - 2;
    }
    if(network_class == 'C')
    {
        host_workable = pow(2,8-borrow_host_bits) - 2;
    }
    cout<<"\n\t Total workable hosts  in each subnetwork are "<<host_workable<<"\n\n";

    host_in_each_subnetwork = pow(2,8-borrow_host_bits);
    iprange = 0;
    for(int i=1; i<=subnetworks; i++)
    {
        cout<<"\n\t Network "<<i<<" : "<<gen_ip<<iprange<<" - 
"<<gen_ip<<iprange+host_in_each_subnetwork-1<<endl;
        iprange += host_in_each_subnetwork;
    }

    return 0;
}
```

OUTPUT :-

```
    Enter The IP Address : 192.168.1.0

    IP Address Belongs To class C


    Default Subnet Mask is 255.255.255.0


    How Many Subnetworks You Want To Create : 16

    Need To Borrow 4 host bits


    Calculated Subnet Mask is 255.255.255.240


    Total workable hosts  in each subnetwork are 14


    Network 1 : 192.168.1.0 - 192.168.1.15

    Network 2 : 192.168.1.16 - 192.168.1.31

    Network 3 : 192.168.1.32 - 192.168.1.47

    Network 4 : 192.168.1.48 - 192.168.1.63

    Network 5 : 192.168.1.64 - 192.168.1.79

    Network 6 : 192.168.1.80 - 192.168.1.95
```

```
Network 7 : 192.168.1.96 - 192.168.1.111

Network 8 : 192.168.1.112 - 192.168.1.127

Network 9 : 192.168.1.128 - 192.168.1.143

Network 10 : 192.168.1.144 - 192.168.1.159

Network 11 : 192.168.1.160 - 192.168.1.175

Network 12 : 192.168.1.176 - 192.168.1.191

Network 13 : 192.168.1.192 - 192.168.1.207

Network 14 : 192.168.1.208 - 192.168.1.223

Network 15 : 192.168.1.224 - 192.168.1.239

Network 16 : 192.168.1.240 - 192.168.1.255
```