Government College of Engineering, Jalgaon

(An Autonomous Institute of Govt. of Maharashtra)
Department of Computer Engineering

Student Name :____ **PRN:** ______________

Course Teacher **:** Mrs Priyanka H. Gadade, Government College of Engg., Jalgaon

Experiment No. ____

Title: - To find Class of IP address

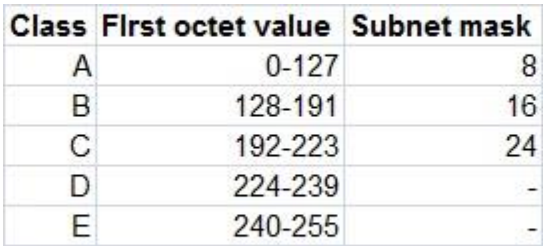
Aim: - Program to find Class and to display Mask of given IP address.

Theory:

IP Addresses:

TCP/IP defines five classes of IP addresses: class A, B, C, D, and E. Each class has a range of valid IP addresses. The value of the first octet determines the class. IP addresses from the first three classes (A, B and C) can be used for host addresses. The other two classes are used for other purposes (class D for multicast and class E for experimental purposes).

Classes of IP addresses:



Class	First octet value	Subnet mask
A	0-127	8
B	128-191	16
C	192-223	24
D	224-239	-
E	240-255	-

Special IP address ranges:

0.0.0.0/8 – addresses used to communicate with the current network

127.0.0.0/8 – loopback addresses

Since these are used for routing data across the Internet and most other networks, these are important. An IP address is basically 32 bits long, for the current IP version 4. The upcoming version 6 uses 128-bit long IP addresses. The main reason that this change is probably going to happen is that the world is running out of version 4 IP addresses.

Here is a typical IP address, written first in dotted-decimal notation and then in binary.

128.21.41.3 = 10000000 00010101 00101001 00000011

(That's a binary version of 128, then 21, 41, and 3.) Note that each of the 4 decimal numbers must fit in the range from 0 to 255, since that is all that will fit in an 8-bit binary number.

Now, the first part of an IP address is the network address, and the second part is the host address, the address of a particular computer. How do we tell where to break the IP address apart to get these two pieces? The usual method is the netmask (sometimes called a subnet mask). Let's suppose that our network uses a netmask of 255.255.0.0,
which in binary is 11111111 11111111 00000000 00000000. What we do is to AND (bitwise boolean AND) the IP address and netmask.

In case AND is unfamiliar, here is how it works on pairs of bits:

0 AND 0 = 0
0 AND 1 = 0
1 AND 0 = 0
1 AND 1 = 1

So, let's AND our IP address and netmask:

```
128.21.41.3 = 10000000 00010101 00101001 00000011
255.255.0.0 = 11111111 11111111 00000000 00000000
-----------------------------------------------------------------------
              10000000 00010101 00000000 00000000 = 128.21.0.0
```

What this shows is that the network address is essentially the first half, written either as 128.21 or 128.21.0.0, while the host address is the second half, the 41.3. Networks with a netmask of 255.255.0.0 are class B networks. Here is a short summary of the IP addresses used on class A, B, and C networks. The class D multicasting IP address format is also shown. In multicasting a number of hosts join a group, which is represented by the multicast address. Data send to that address is sent to all hosts in the group.

Class	Network Address	Host Address	Default Netmask
A	8-bit starting with 0	24-bit	255.0.0.0
B	16-bit starting with 10	16-bit	255.255.0.0
C	24-bit starting with 110	8-bit	255.255.255.0
D	32-bit multicast address starting with 1110		

Obviously a class C network can only have a few hosts, whereas a class A network can have a huge number of hosts on it. However, there are only a few class A addresses.

Let’s look at some more examples of IP addressing and include our previous example as well. These examples use typical CIDR notation, the /n, which means that the netmask consists of n ones followed by all zeros. The netmask is shown below the IP address in each case.

120.0.0.0/8	Network address	120.0.0.0	
Netmask	11111111 00000000 00000000 00000000		255.0.0.0
Example IP address:	120.21.41.125	01111000 00010101 00101001 01110101	
128.21.0.0/16	Network address	128.21.0.0	
Netmask	11111111 11111111 00000000 00000000		255.255.0.0
Example IP address:	128.21.41.3	10000000 00010101 00101001 00000011	
195.223.139.0/24	Network address	195.223.139.0	

Netmask	11111111 11111111 11111111 00000000	255.255.255.0
Example IP address:	195.223.139.27	11000011 11011111 10001011 00011011

Note that in any of these examples we can AND a particular host IP address with the netmask in order to extract the network address. (This just zeros out the host address portion of the IP address.) What should we make of a network address of 128.21.0.16/28 in CIDR notation? The /28 indicates a network mask of 11111111 11111111 11111111 11110000 = 255.255.255.240. The IP address itself is 10000000 00010101 00000000 00010000 in binary, which is clearly a class B address due to the initial bits of 10. Using this IP address as a network address is strange, since we would normally have all zeros for the low order 16 bits. The netmask shows us that only the low order 4 bits can be used for the host number. With 4 bits we can only have 16 numbers. So, this scheme allows us to put up to 16 computers on a network with network address 128.21.0.16. Typically we say that the network ID is the 128.21.0.0 and that a subnet number of 1 is being used. We get the subnet number by taking the host portion of this class B address, the 00000000 00010000 and throwing away the 4 low-order bits that we really use to specify the hosts. This leaves 00000000 0001, which is binary for 1. In essence a class B network is being subdivided into little subnetworks. The particular subnetwork described here includes just the IP addresses from

10000000 00010101 00000000 00010000 = 128.21.0.16

through

10000000 00010101 00000000 00011111 = 128.21.0.31

Another small network could use a different portion of the same class B network, perhaps 128.21.0.32/28. Let's analyze this in the same way. The netmask is the same, so only the bottom 4 bits are available to specify a host. That allows us to put up to 16 hosts on this subnetwork, just like the previous one. The network address is 128.21.0.32 = 10000000 00010101 00000000 00100000 in binary. Throwing away the low-order 4 bits used to specify the hosts from the host portion of the IP address (host portion is 00000000 00100000) yields a subnet number of 00000000 0010, which is 2. The 16 possible IP addresses on this little subnetwork run from

10000000 00010101 00000000 00100000 = 128.21.0.32

through

10000000 00010101 00000000 00101111 = 128.21.0.47

Note that in these complicated examples one can still start with the IP address of any host and use the netmask to figure out the network ID, subnet, and host number. Let’s try as an example the host IP address of 128.21.0.40 on the network 128.21.0.32/28 that we just looked at. In binary we have the following.

```
             = 10000000 00010101 00000000 00101000
AND
netmask        11111111 11111111 11111111 11110000
               ----------------------------------------------------
               10000000 00010101 00000000 00100000
```

Since overall this is a class B address, the network ID is given by the first 16 bits. We would probably use dotted decimal notation and give 128.21.0.0 as the network ID. The netmask masks off the low-order 4 bits, so the host number is given by the last 4 bits, 1000 = 8. Leaving out these last 4 bits in the host section of this class B address gives 00000000 0010 = 2, so the subnet number is 2. Thus we get network ID 128.21.0.0, subnet number 2, host number 8.

Special addresses

The IP address 255.255.255.255 = 11111111 11111111 11111111 11111111 is a broadcast address for sending something to all computers on the local network.

Placing all 1’s in the host portion of an IP address gives a broadcast address for sending data to all hosts on the particular network. For example, 128.21.255.255 = 10000000 00010101 11111111 11111111 would be the address to use to send something to all computers on the network with network ID 128.21.

Loopback addresses use the network address 127.0.0.0 with some valid host number, typically 1. Thus 127.0.0.1 = 01111111 00000000 00000000 00000001 is usually used for loopback purposes, such as testing communications software with both sender and receiver on the same computer. Data sent to this address appears as incoming data at this same address.

The address 0.0.0.0 is used by a computer to indicate itself at bootup, before it has learned its true IP address (perhaps from a DHCP server).

Implementation:

```
/*
Function : extractIpAddress
Arguments :
1) sourceString - String pointer that contains ip address
2) ipAddress - Target variable short type array pointer that will store ip address octets
*/
#include <stdio.h>
#include <string.h>
#include<stdlib.h>
void extractIpAddress(unsigned char *sourceString,short *ipAddress)
{       unsigned short len=0;
        unsigned char  oct[4]={0},cnt=0,cnt1=0,i,buf[5];

        len=strlen(sourceString);
        for(i=0;i<len;i++)
        {       if(sourceString[i]!='.')
                {       buf[cnt++] =sourceString[i];
                }
                if(sourceString[i]=='.' || i==len-1)
                {
                        buf[cnt]='\0';
                        cnt=0;
                        oct[cnt1++]=atoi(buf);
                }
        }
        ipAddress[0]=oct[0];
        ipAddress[1]=oct[1];
        ipAddress[2]=oct[2];
        ipAddress[3]=oct[3];
}

int main()
{       unsigned char ip[20]={0};
        short ipAddress[4];
        printf("Enter IP Address (xxx.xxx.xxx.xxx format): ");
        scanf("%s",ip);
        extractIpAddress(ip,&ipAddress[0]);
        printf("\nIp Address: %03d. %03d. %03d. %03d\n", ipAddress[0], ipAddress[1],
                ipAddress[2], ipAddress[3]);

        if(ipAddress[0]>=0 && ipAddress[0]<=127)
        printf("Class A Ip Address.and Mask is : 255.0.0.0\n");
        if(ipAddress[0]>127 && ipAddress[0]<191)
        printf("Class B Ip Address and Mask is : 255.255.0.0\n");
        if(ipAddress[0]>191 && ipAddress[0]<224)
        printf("Class C Ip Addressand Mask is : 255.255.0.0\n");
        if(ipAddress[0]>224 && ipAddress[0]<=239)
        printf("Class D Ip Address and Mask is : 255.255.0.0\n");
        if(ipAddress[0]>239)
        printf("Class E Ip Address and Mask is : 255.255.0.0\n");

        return 0;
}
```

Mrs. Priyanka H. Gadade

Course Teacher