Computer Networks

Experiment - 7

Code:-

```
import math
def findClass(ip):
    if 0 <= ip[0] <= 127:
        print("Network Address is : ", ip[0])
        print('No. of IP addresses possible : ', 2 ** 24)
        return "A", '255.0.0.0'
    elif 128 <= ip[0] <= 191:
        ip = [str(i) for i in ip]
        print("Network Address is : ", ".".join(ip[0:2]))
        print('No. of IP addresses possible : ', 2 ** 16)
        return "B", '255.255.0.0'
    elif 192 <= ip[0] <= 223:
        ip = [str(i) for i in ip]
        print("Network Id is : ", ".".join(ip[0:3]))
        print('No. of IP addresses possible : ', 2 ** 8)
        return "C", '255.255.255.0'
    elif 224 <= ip[0] <= 239:
        print("In this Class, IP address is not divided into Network and Host ID")
        return "D"
    else:
        print("In this Class, IP address is not divided into Network and Host ID")
        return "E"

def Subnetting(ip, num, className, ip_addresses):
    temp = 0
    if className == "A":
        place2 = ip_addresses / (256 ** 2)
        for i in range(num):
            print(f"Subnet {i} => ")
            print(temp)
            print("Subnet Address : ", ip[0] + '.' + str(temp) + '.0' + '.0')
            temp += int(place2)
            print("Broadcast address : ", ip[0] + '.' + str(temp - 1) + '.255' + '.255')
            print("Valid range of host IP address : ", ip[0] + '.' + str(temp -
int(place2)) + '.' + '0' + '.1' + '\t-\t' + ip[0] + '.' + str( temp - 1) + '.254' +
'.254')
            print()
    elif className == "B":
```

```
        place2 = ip_addresses / 256
        for i in range(num):
            print(f"\nSubnet {i} => ")
            print("Subnet Address : ", ".".join(ip[0:2]) + '.' + str(temp) + '.0')
            temp += int(place2)
            print("Broadcast address : ", ".".join(ip[0:2]) + '.' + str(temp - 1) +
'.255')
            print("Valid range of host IP address : ",  ".".join(ip[0:2]) + '.' +
str(temp - int(place2)) + '.1\t-\t' + ".".join(ip[0:2]) + '.' + str( temp - 1) + '.254')
            print()
    elif className == "C":
        for i in range(num):
            print(f"\nSubnet {i} => ")
            print("Subnet Address : ", ".".join(ip[0:3]) + '.' + str(temp))
            temp += int(ip_addresses)
            print("Broadcast address : ", ".".join(ip[0:3]) + '.' + str(temp - 1))
            print("Valid range of host IP address : ",  ".".join(ip[0:3]) + '.' +
str(temp - int(ip_addresses) + 1) + '\t-\t' + ".".join(ip[0:3]) + '.' + str( temp - 2))
            print()
    else:
        print("In this Class, IP address is not divided into Network and Host ID")

def subnetmask(num, network_mask):
    var = '1' * int(math.log(num, 2))
    var1 = '0' * (8 - int(math.log(num, 2)))
    binary_num = var + var1
    network_mask = network_mask.split('.')
    network_mask = [i for i in network_mask if i != '0']
    network_mask.append(str(int(binary_num, 2)))
    while len(network_mask) < 5:
        network_mask.append('0')
        print('Subnet Mask  ', ".".join(network_mask[0:4]))

ip = input("Enter the IP address : ")
ip = ip.split(".")
ip = [int(i) for i in ip]
lst = findClass(ip)
networkClass = lst[0]
print("Given IP address belongs to class : ", networkClass)
ip = [str(i) for i in ip]
network_mask = lst[1]
print('Network Mask : ', network_mask)
num_subnet = int(input('\nNo. of subnets(power of 2) : '))
num_ip = int(2 ** (8 * (68 - ord(networkClass))) / num_subnet)
print('The no. of bits in the subnet id : ', int(math.log(num_subnet, 2)))
```

```
if ord(networkClass) < 68:
    print('Total no. of IP addresses possible in each subnet : ', num_ip)
    Subnetting(ip, num_subnet, networkClass, num_ip)
    subnetmask(num_subnet, network_mask)
```

Output:-