Shirish
C3 C32
2103164

EXPERIMENT :- 7

CN - C32 - 2103164

Aim :- WAP to implement IPv4 addressing concept along with subnet masking.

Theory :-

IPV4 addressing is a fundamental concept in networking in that involves assigning unique numerical label to devices on a network. These addresses are used for indentification and communication between devices in the IP network. IPv4 addresses are respented as a series of four decimal numbers, each ranging from 0 to 255, seperated by periods. Each numbers is an octet representing 8 bits of 32 bit IPv4 address.

Here how IPv4 addressing works :-

32-bit Address :- IPv4 uses a 32-bit address space, allowing for approximately, 4.3 billon unique addresses.

Network and Host Portion :- An IPv4 address is divided into two parts : the network portion. and the host portion. The boundary between two pointems is determined by a subnet mask.

Subnet Mask :- The subnet mask is 32 bit nu
that defines the boundary between the ne
and host portions of an IPV4 address. It
consists of series of contigous. 1's followed by a
of contigous 0s. The 1s indicate the network
bits, and 0s indicate the host bits.

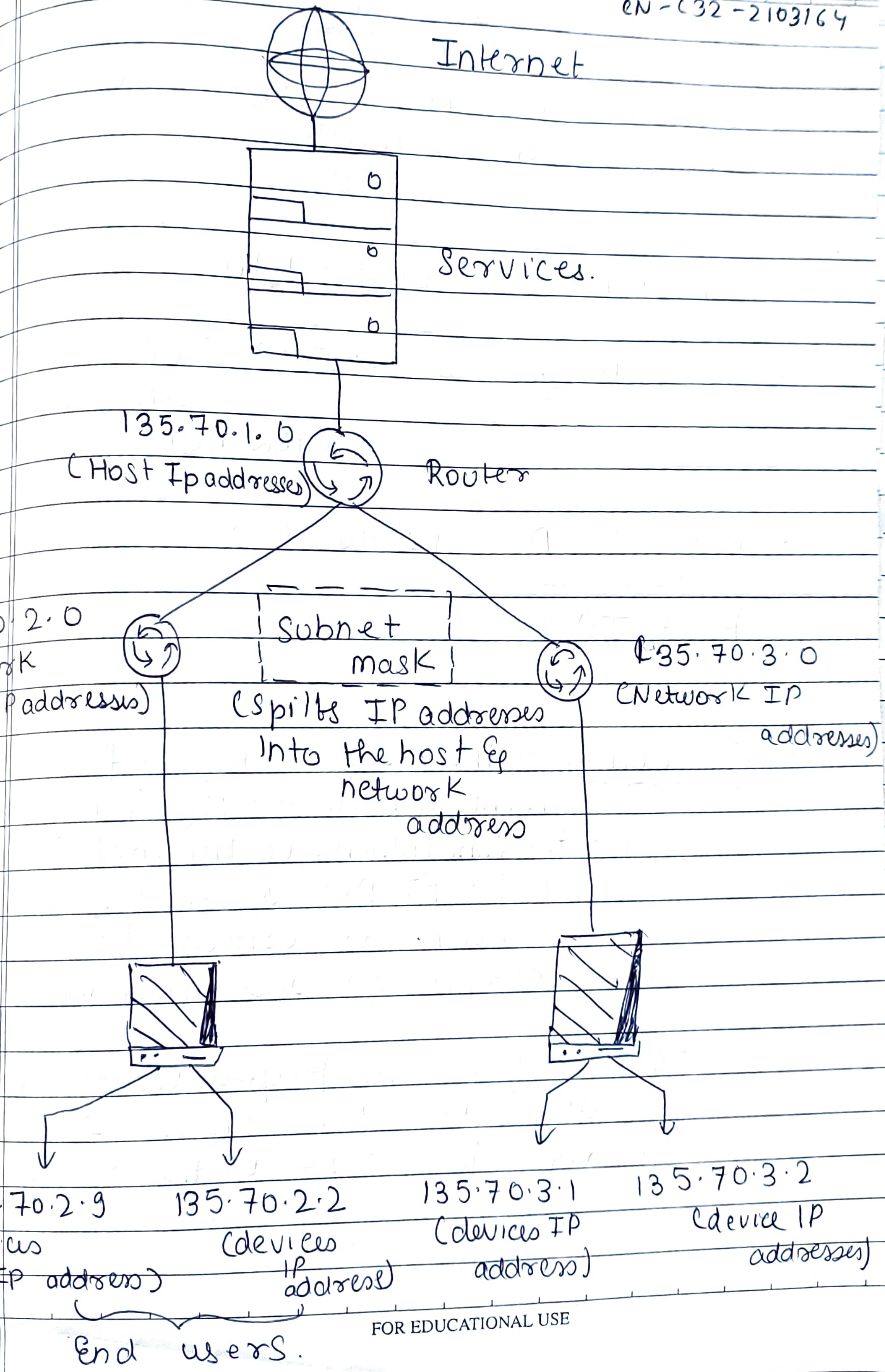
Network ID and Host ID :- The network portion
of address identifies the network to whic
devices belongs, while the host portion id
specific devics within that network. Devic
within the same network share share the
same network ID.

Classes of IP addresses :- In past, IPV4 ad
were divided into five classes (A, B, C, D a
based on their first octet range. However
classful addressing has largely been rep
by Classless InterDomain Routing (CIDR) whic
allows for more flexible allocation of I
addresses space.

Private and Public Addresses :- IPV4 addresses a
categorized into private and public addre
ranges. Private addresses are reserved f
use within private network and are not
routable on the public internet. Public ad
are globally unique and are used for devices
accessible from the internet.

Subnet Masking

CN-C32-2103164



CN-C32-2103264

Subnet Masking

* **Example of IPv4 addresses with their associated subnet mask**

- IPv4 Address : 192.168.1.10
- Subnet Mask : 255.255.255.0 (also represented as /24) (CIDR notation.)
- Network ID : 192.168.1.0
- Host ID : 10.

* Conclusion :- IPv4 addressing is fundamental concept in networking & is essential for the functioning of internet & local networks. Proper addresses allocation & subnetting are critical for efficient network, design and management.

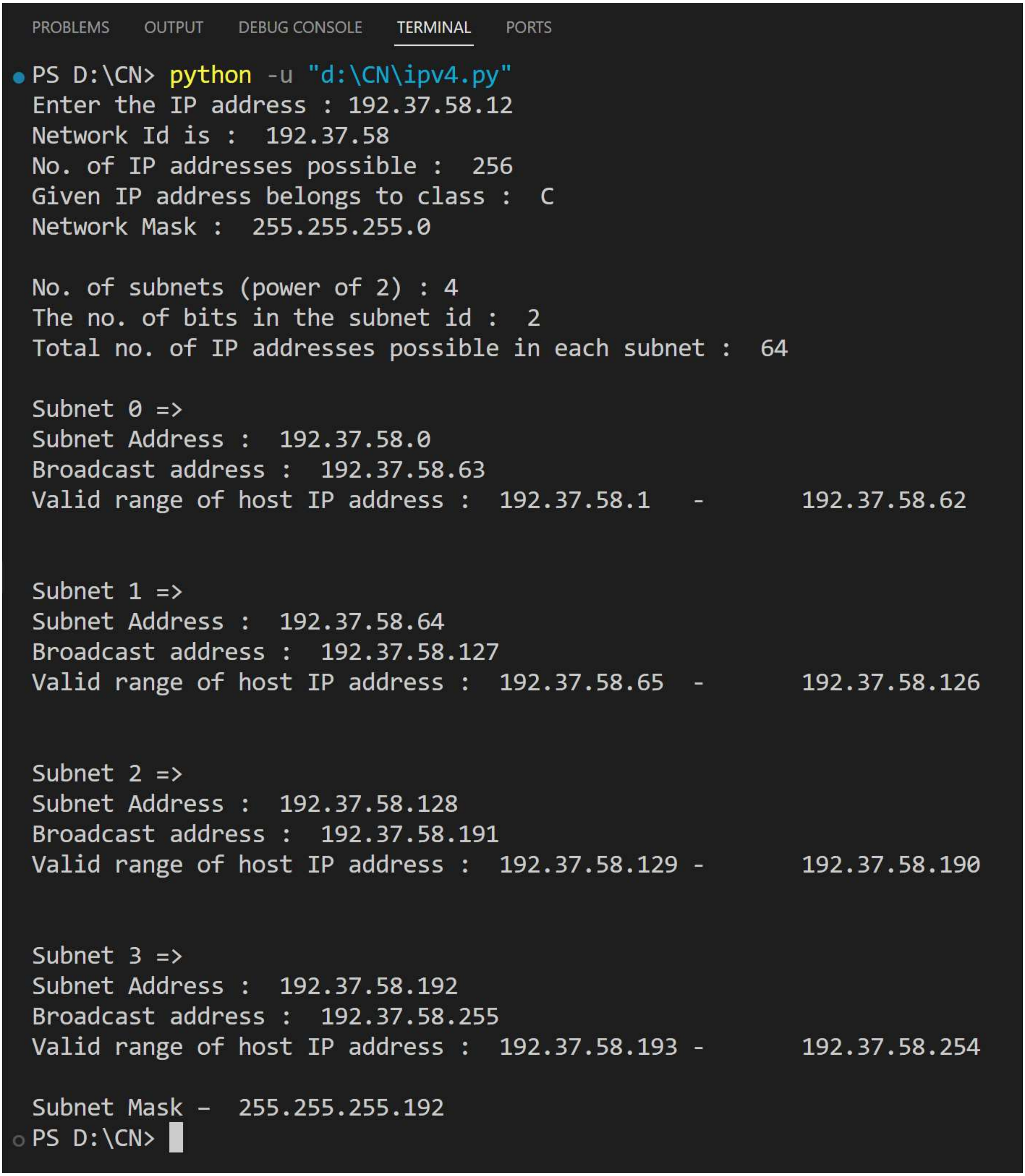
This experiment Helped us to shrunk & built its logic which further enchanced the clarity of this concept.

A 03/10/23.

```
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:



```
PS D:\CN> python -u "d:\CN\ipv4.py"
Enter the IP address : 192.37.58.12
Network Id is :  192.37.58
No. of IP addresses possible :  256
Given IP address belongs to class :  C
Network Mask :  255.255.255.0

No. of subnets (power of 2) : 4
The no. of bits in the subnet id :  2
Total no. of IP addresses possible in each subnet :  64

Subnet 0 =>
Subnet Address :  192.37.58.0
Broadcast address :  192.37.58.63
Valid range of host IP address :  192.37.58.1   -       192.37.58.62


Subnet 1 =>
Subnet Address :  192.37.58.64
Broadcast address :  192.37.58.127
Valid range of host IP address :  192.37.58.65  -       192.37.58.126


Subnet 2 =>
Subnet Address :  192.37.58.128
Broadcast address :  192.37.58.191
Valid range of host IP address :  192.37.58.129 -       192.37.58.190


Subnet 3 =>
Subnet Address :  192.37.58.192
Broadcast address :  192.37.58.255
Valid range of host IP address :  192.37.58.193 -       192.37.58.254

Subnet Mask -  255.255.255.192
PS D:\CN>
```