Name: Neeraj Varma

Roll. No: 8

Batch: A1

PRN: 1032210651

Title: Subnetting

Aim: Write a program to implement subnetting to find subnet mask

Objectives:

1. To understand and learn the concept of IP address, subnet mask and subnetting

Neeraj Varma PRN: 10322106S1
Roll.No: 08
Batch: A1

CN Lab Assignment

Subnetting

Theory:

i) Introduction to IPv4 and IPv6: IPv4 and IPv6 are the actual protocols tasked with data transmission in the form of packet and datagrams IPv4 protocol is mainly used with ethernet during packet switching in the link-layer networks.

IPv4	IPv6
• 32 bit address	• 128 bit address
• Supports manual and DHCP	• Supports Auto Config
• 4.29×10^9 address space	• 3.4×10^{38} address space
• IPv4 is decimal	• IPv6 is hexadecimal

ii) CIDR (Classless Inter Domain Routing): It is a method of assigning protocol address that improve the efficiency of address distribution and replaces the previous system based on Class A, Class B and Class C network

iii) Default Subnet Mask for all Classes:

- Class A - 255.0.0.0
- Class B - 255.255.0.0
- Class C - 255.255.255.0

iv) Subnetting Example:

172.16.0.0 — Class B network

Observation: Successfully implemented Subnetting program

FAQ's

1) Describe classful and classless IP addressing scheme with an example

→ i) Classful addressing: It is a IPv4 addressing architechture that sends address into 5 groups A, B, C, D, E. It is split into 4 sections adding up to 32 bits of data.

ii) Classless addressing: To reduce wastage of IP addresses in a block, we use subnetting. It is an IPv4 addressing architechture that use variable length subnet mask

2) What are different reserved IPv4 addresses?

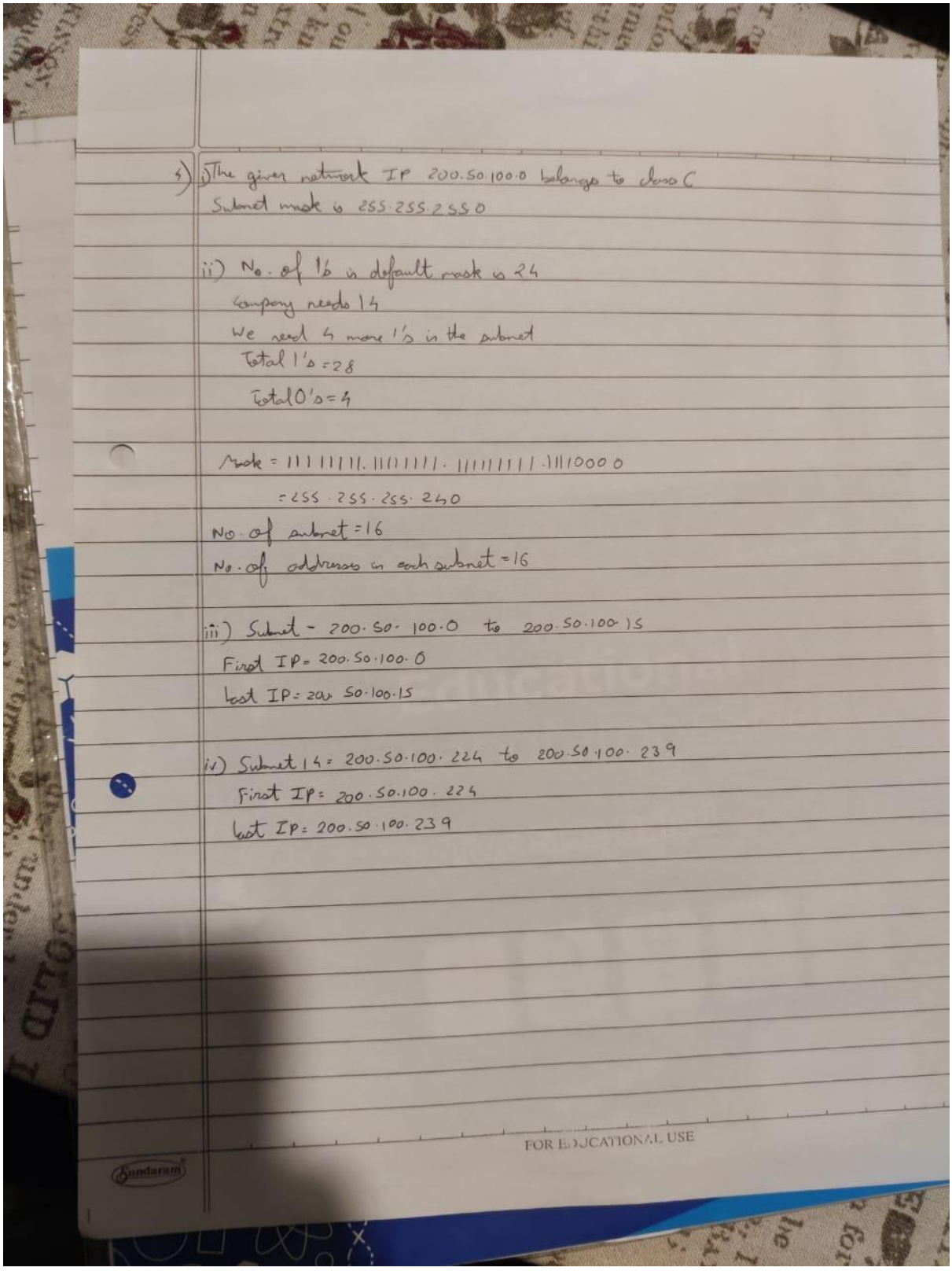
→ 10.0.0.0 — 10.255.255.255 . Private network

127.0.0.0 — 127.255.255.255 Host

169.254.0.0 - 169.255.255.255 Subnet

3) What are the uses of subnetting?

→ • Organizes a network in an efficient way
• Used for large firms and companies with huge scale
• IPs can be kept geographically localised



4) i) The given network IP 200.50.100.0 belongs to class C
Subnet mask is 255.255.255.0

ii) No. of 1's in default mask is 24
Company needs 14
We need 4 more 1's in the subnet
Total 1's = 28
Total 0's = 4

Mask = 11111111.11111111.11111111.11110000
= 255.255.255.240
No. of subnet = 16
No. of addresses in each subnet = 16

iii) Subnet - 200.50.100.0 to 200.50.100.15
First IP = 200.50.100.0
Last IP = 200.50.100.15

iv) Subnet 14 = 200.50.100.224 to 200.50.100.239
First IP = 200.50.100.224
Last IP = 200.50.100.239

Code:

```java
import java.util.*;

public class sub {
    Run | Debug
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        int[] arr = new int[4];
        System.out.println(x: "Enter each byte of the IP Address in the decimal format: ");
        for (int i = 0; i < arr.length; i++) {
            arr[i] = in.nextInt();
        }
        if (arr[0] <= 127 && arr[0] >= 0) {
            System.out.println(x: "The class of the given IP Address is A");
            System.out.println(x: "The default Subnet Mask of the given IP Address is 255.0.0.0");
        } else if (arr[0] <= 191 && arr[0] >= 128) {
            System.out.println(x: "The class of the given IP Address is B");
            System.out.println(x: "The default Subnet Mask of the given IP Address is 255.255.0.0");
        } else if (arr[0] <= 223 && arr[0] >= 192) {
            System.out.println(x: "The class of the given IP Address is C");
            System.out.println(x: "The default Subnet Mask of the given IP Address is 255.255.255.0");
        } else if (arr[0] <= 239 && arr[0] >= 224) {
            System.out.println(x: "The class of the given IP Address is D");
            System.out.println(x: "The default Subnet Mask of the given IP Address is not defined");
        } else if (arr[0] <= 255 && arr[0] >= 240) {
            System.out.println(x: "The class of the given IP Address is E");
            System.out.println(x: "The default Subnet Mask of the given IP Address is not defined");
        } else {
            System.out.println(x: "The inputted IP Address is not valid");
        }
        int[] binary = new int[8];
        for (int i = 0; i < binary.length; i++) {
            binary[i] = 0;
```

```
31          binary[i] = 0;
32        }
33        System.out.println(x: "Enter the required number of Subnets: ");
34        int subnets = in.nextInt();
35        if (subnets <= 1 && subnets >= 0) {
36          arr[3] = 0;
37          System.out.print(Arrays.toString(arr) + " to ");
38          arr[3] = 127;
39          System.out.println(Arrays.toString(arr));
40          arr[3] = 128;
41          System.out.print(Arrays.toString(arr) + " to ");
42          arr[3] = 255;
43          System.out.println(Arrays.toString(arr));
44        } else if (subnets <= 4 && subnets >= 2) {
45          arr[3] = 0;
46          System.out.print(Arrays.toString(arr) + " to ");
47          arr[3] = 63;
48          System.out.println(Arrays.toString(arr));
49          arr[3] = 64;
50          System.out.print(Arrays.toString(arr) + " to ");
51          arr[3] = 127;
52          System.out.println(Arrays.toString(arr));
53          arr[3] = 128;
54          System.out.print(Arrays.toString(arr) + " to ");
55          arr[3] = 191;
56          System.out.println(Arrays.toString(arr));
57          arr[3] = 192;
58          System.out.print(Arrays.toString(arr) + " to ");
59          arr[3] = 255;
60          System.out.println(Arrays.toString(arr));
61        } else if (subnets <= 8 && subnets >= 5) {
62          arr[3] = 0;
```

```
61        } else if (subnets <= 8 && subnets >= 5) {
62          arr[3] = 0;
63          System.out.print(Arrays.toString(arr) + " to ");
64          arr[3] = 31;
65          System.out.println(Arrays.toString(arr));
66          arr[3] = 32;
67          System.out.print(Arrays.toString(arr) + " to ");
68          arr[3] = 63;
69          System.out.println(Arrays.toString(arr));
70          arr[3] = 64;
71          System.out.print(Arrays.toString(arr) + " to ");
72          arr[3] = 95;
73          System.out.println(Arrays.toString(arr));
74          arr[3] = 96;
75          System.out.print(Arrays.toString(arr) + " to ");
76          arr[3] = 127;
77          System.out.println(Arrays.toString(arr));
78          arr[3] = 128;
79          System.out.print(Arrays.toString(arr) + " to ");
80          arr[3] = 159;
81          System.out.println(Arrays.toString(arr));
82          arr[3] = 160;
83          System.out.print(Arrays.toString(arr) + " to ");
84          arr[3] = 191;
85          System.out.println(Arrays.toString(arr));
86          arr[3] = 192;
87          System.out.print(Arrays.toString(arr) + " to ");
88          arr[3] = 223;
89          System.out.println(Arrays.toString(arr));
90          arr[3] = 224;
91          System.out.print(Arrays.toString(arr) + " to ");
92          arr[3] = 255;
```

```
92              arr[3] = 255;
93              System.out.println(Arrays.toString(arr));
94          }
95      }
96  }
```

Output:

```
  }
 Enter each byte of the IP Address in the decimal format:
 100
 120
 1
 1
 The class of the given IP Address is A
 The default Subnet Mask of the given IP Address is 255.0.0.0
 Enter the required number of Subnets:
 4
 [100, 120, 1, 0] to [100, 120, 1, 63]
 [100, 120, 1, 64] to [100, 120, 1, 127]
 [100, 120, 1, 128] to [100, 120, 1, 191]
 [100, 120, 1, 192] to [100, 120, 1, 255]
 PS D:\Code\JAVA\Java\src>
```

```
  }
Enter each byte of the IP Address in the decimal format:
128
150
10
1
The class of the given IP Address is B
The default Subnet Mask of the given IP Address is 255.255.0.0
Enter the required number of Subnets:
4
[128, 150, 10, 0] to [128, 150, 10, 63]
[128, 150, 10, 64] to [128, 150, 10, 127]
[128, 150, 10, 128] to [128, 150, 10, 191]
[128, 150, 10, 192] to [128, 150, 10, 255]
PS D:\Code\JAVA\Java\src>
PS D:\Code\JAVA\Java\src>
```

```
Enter each byte of the IP Address in the decimal format:
192
168
1
1
The class of the given IP Address is C
The default Subnet Mask of the given IP Address is 255.255.255.0
Enter the required number of Subnets:
4
[192, 168, 1, 0] to [192, 168, 1, 63]
[192, 168, 1, 64] to [192, 168, 1, 127]
[192, 168, 1, 128] to [192, 168, 1, 191]
[192, 168, 1, 192] to [192, 168, 1, 255]
PS D:\Code\JAVA\Java\src>
```