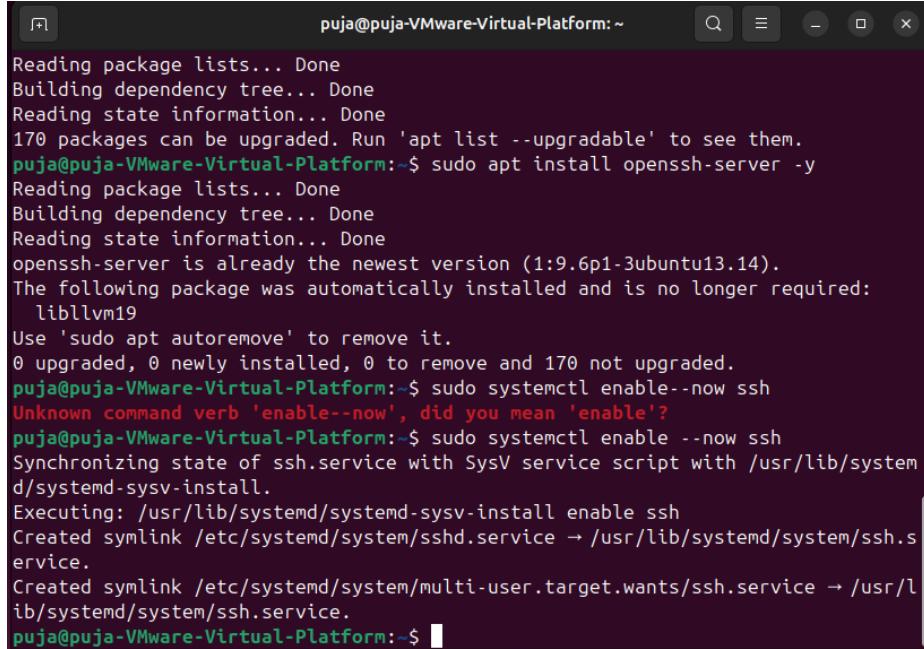


Template Week 6 – Networking

Student number: 577029

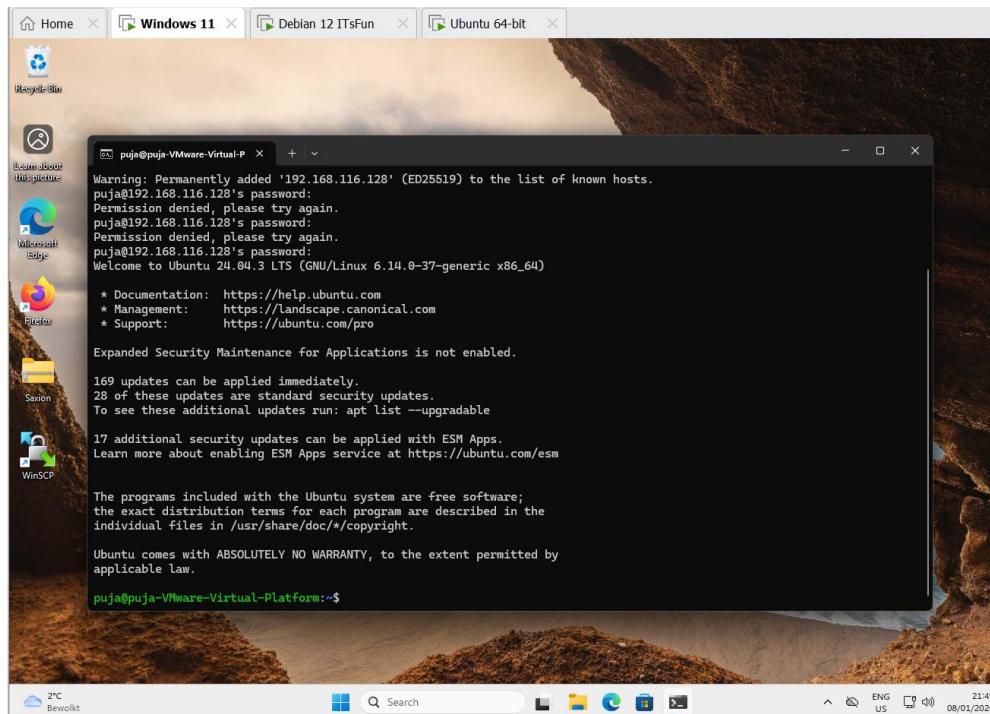
Assignment 6.1: Working from home

Screenshot installation openssh-server:

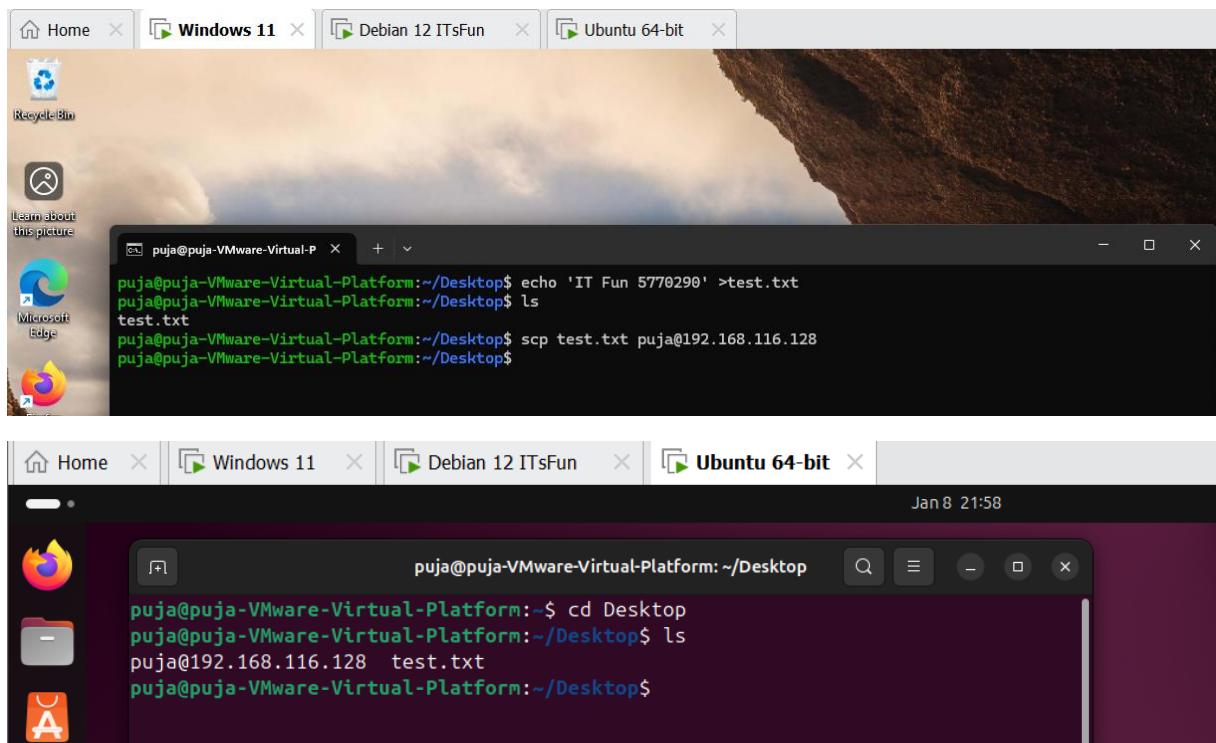


```
puja@puja-VMware-Virtual-Platform: ~
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
170 packages can be upgraded. Run 'apt list --upgradable' to see them.
puja@puja-VMware-Virtual-Platform:~$ sudo apt install openssh-server -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openssh-server is already the newest version (1:9.6p1-3ubuntu13.14).
The following package was automatically installed and is no longer required:
  liblvm19
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 170 not upgraded.
puja@puja-VMware-Virtual-Platform:~$ sudo systemctl enable--now ssh
Unknown command verb 'enable--now', did you mean 'enable'?
puja@puja-VMware-Virtual-Platform:~$ sudo systemctl enable --now ssh
Synchronizing state of ssh.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable ssh
Created symlink /etc/systemd/system/sshd.service → /usr/lib/systemd/system/sshd.service.
Created symlink /etc/systemd/system/multi-user.target.wants/ssh.service → /usr/lib/systemd/system/ssh.service.
puja@puja-VMware-Virtual-Platform:~$
```

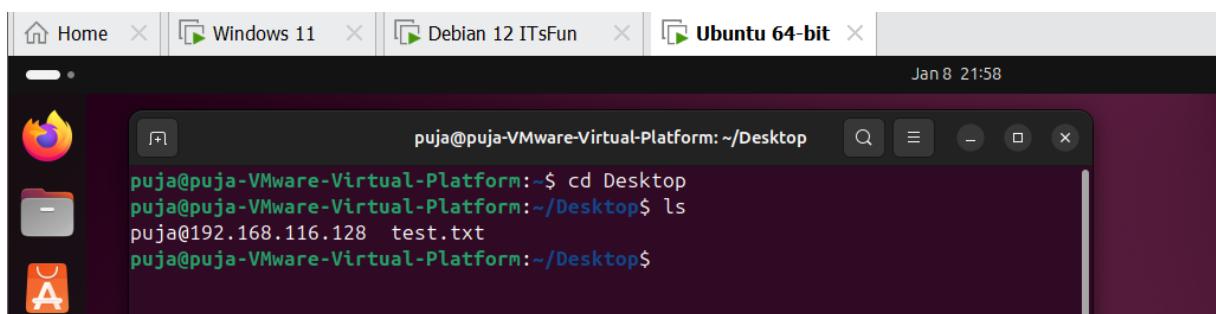
Screenshot successful SSH command execution:



Screenshot successful execution SCP command:

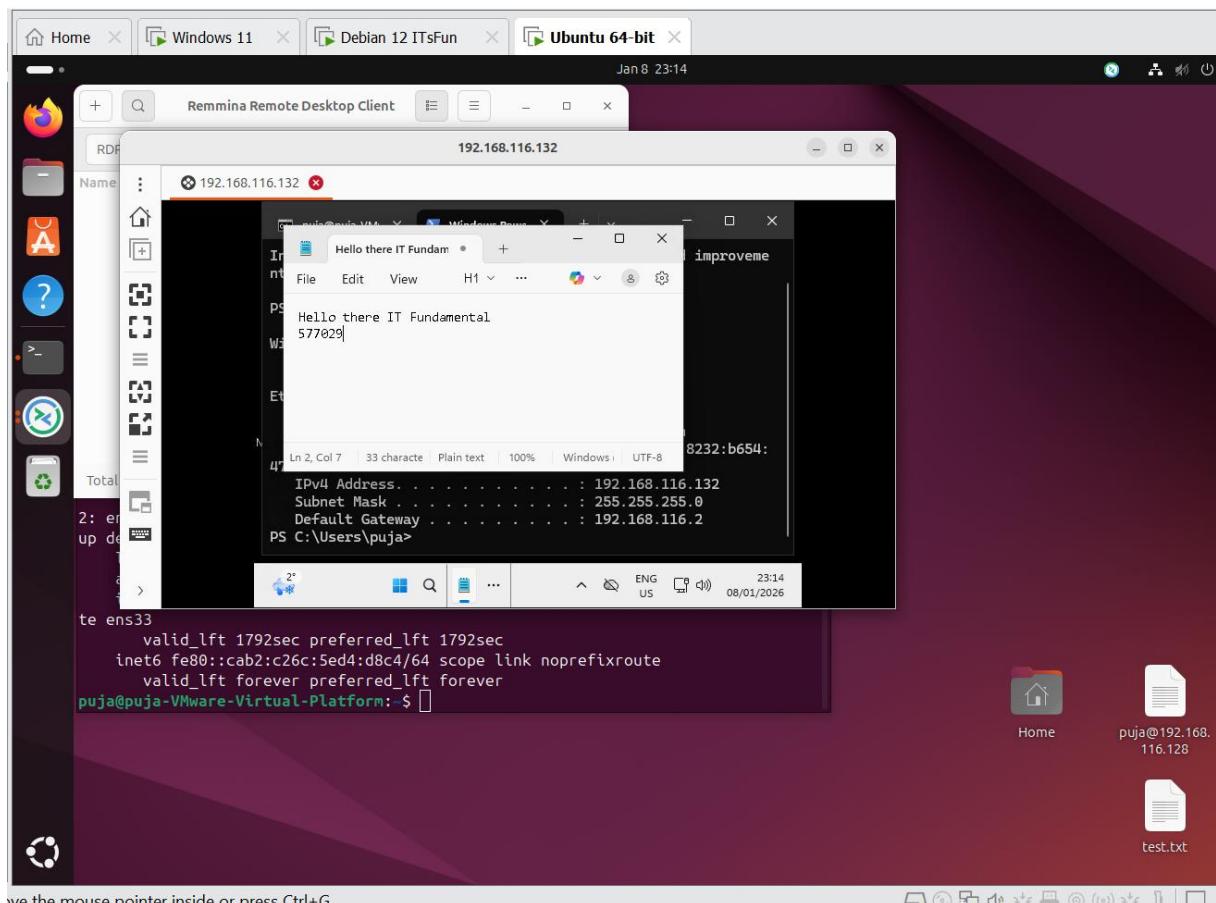


```
puja@puja-VMware-Virtual-P ~$ echo 'IT Fun 5770290' >test.txt
puja@puja-VMware-Virtual-P ~$ ls
test.txt
puja@puja-VMware-Virtual-P ~$ scp test.txt puja@192.168.116.128
puja@puja-VMware-Virtual-P ~$
```

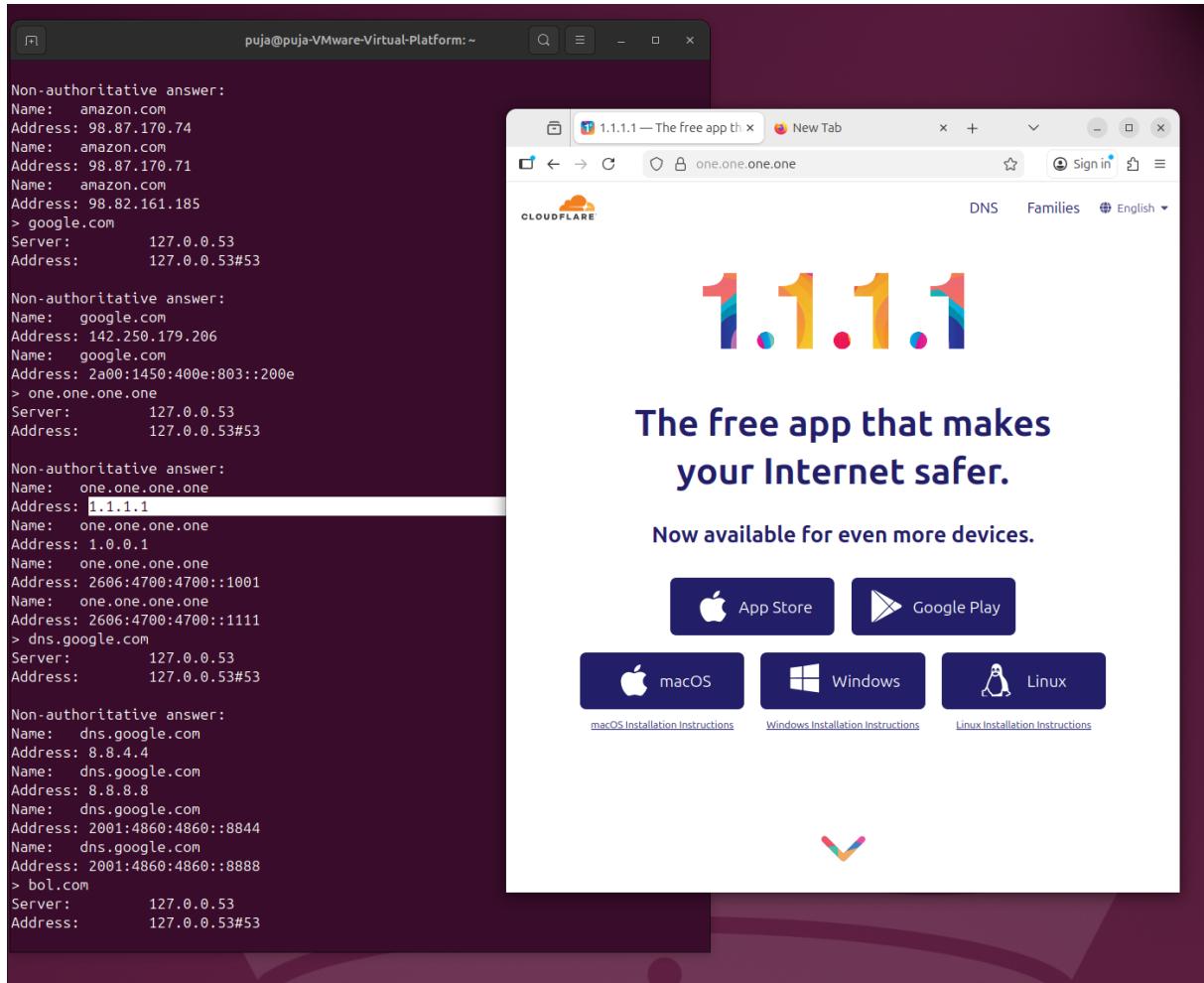
```
puja@puja-VMware-Virtual-Platform:~/Desktop$ cd Desktop
puja@puja-VMware-Virtual-Platform:~/Desktop$ ls
puja@192.168.116.128 test.txt
puja@puja-VMware-Virtual-Platform:~/Desktop$
```

Screenshot remmina:

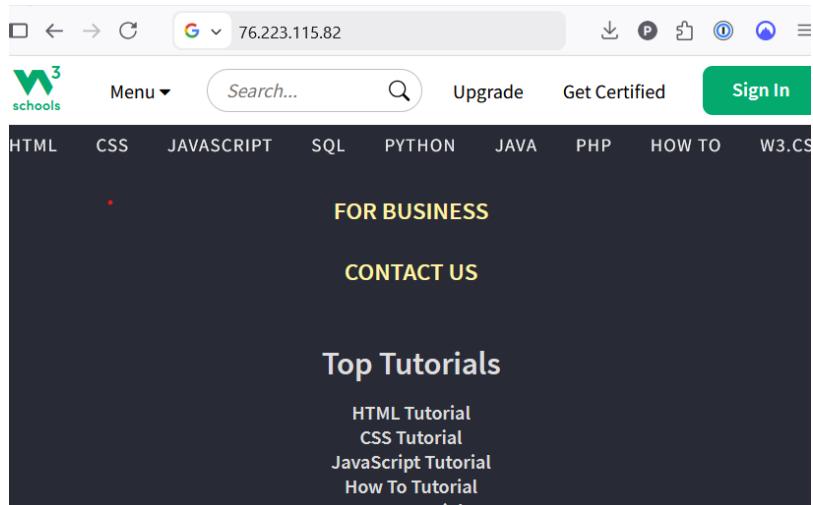


Assignment 6.2: IP addresses websites

Relevant screenshots nslookup command:



Screenshot website visit via IP address:



Assignment 6.3: subnetting

How many IP addresses are in this network configuration 192.168.110.128/25?

There are 128 IP addresses

What is the usable IP range to hand out to the connected computers?

192.168.110.129 to 192.168.110.254

Check your two previous answers with this Linux command: `ipcalc 192.168.110.128/25`

```
puja@puja-Virtual-Platform:~$ ipcalc 192.168.110.128/25
Address: 192.168.110.128      11000000.10101000.01101110.1 00000000
Netmask: 255.255.255.128 = 25 11111111.11111111.11111111.1 00000000
Wildcard: 0.0.0.127          00000000.00000000.00000000.0 11111111
=>
Network: 192.168.110.128/25  11000000.10101000.01101110.1 00000000
HostMin: 192.168.110.129    11000000.10101000.01101110.1 00000001
HostMax: 192.168.110.254    11000000.10101000.01101110.1 11111110
Broadcast: 192.168.110.255  11000000.10101000.01101110.1 11111111
Hosts/Net: 126               Class C, Private Internet
```

Explain the above calculation in your own words.

The /25 notation is CIDR netmask is telling us 25 bits are locked for the network ID. While the IP address is 32-bit container, it's leaving us with only 7 remaining bits for the device. In binary, 7 bits allow for 128 possible combinations, which will be the total of pool of addresses. However, we must always discard the very first address and very last address, so subtract those 2 leaving us with 126 usable addresses.

Assignment 6.4: HTML

Screenshot IP address Ubuntu VM:

```
inet 192.168.116.128/24 brd 192.168.116.255 scope global dynamic noprefixroute ens33
```

Screenshot of Site directory contents:

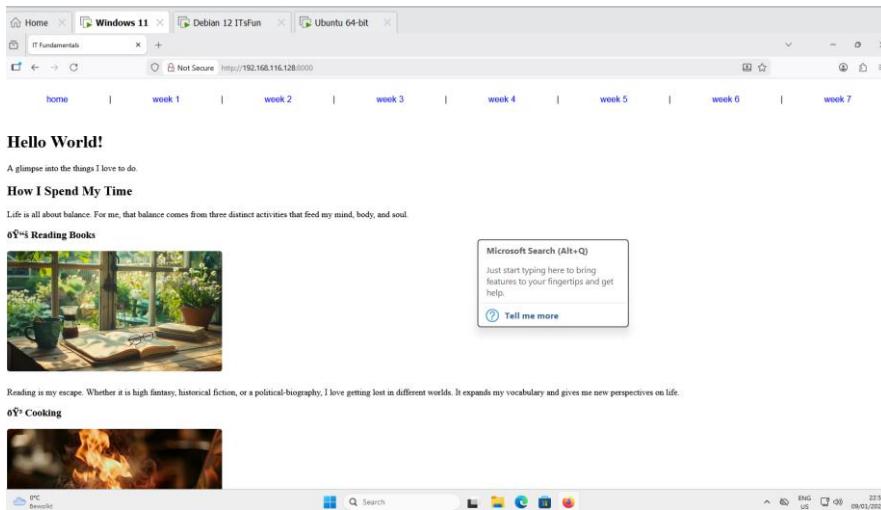
```
puja@puja-VMware-Virtual-Platform:~/site$ pwd
/home/puja/site
puja@puja-VMware-Virtual-Platform:~/site$ tree
.
├── css
│   └── mypdfstyle.css
├── home.html
└── images
    ├── cooking.jpg
    ├── journaling.jpg
    └── reading.jpg
├── index.html
└── pdf
    ├── week1.pdf
    ├── week2.pdf
    ├── week3.pdf
    ├── week4.pdf
    ├── week5.pdf
    ├── week6.pdf
    └── week7.pdf
    ├── week1.html
    ├── week2.html
    ├── week3.html
    ├── week4.html
    ├── week5.html
    ├── week6.html
    └── week7.html

4 directories, 20 files
puja@puja-VMware-Virtual-Platform:~/site$
```

Screenshot python3 webserver command:

```
puja@puja-VMware-Virtual-Platform:~/site$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
192.168.116.132 - - [09/Jan/2026 22:52:17] "GET / HTTP/1.1" 200 -
192.168.116.132 - - [09/Jan/2026 22:52:17] "GET /home.html HTTP/1.1" 200 -
192.168.116.132 - - [09/Jan/2026 22:52:17] "GET /css/mypdfstyle.css HTTP/1.1" 200 -
192.168.116.132 - - [09/Jan/2026 22:52:17] "GET /images/reading.jpg HTTP/1.1" 200 -
192.168.116.132 - - [09/Jan/2026 22:52:17] "GET /images/cooking.jpg HTTP/1.1" 200 -
192.168.116.132 - - [09/Jan/2026 22:52:17] "GET /images/journaling.jpg HTTP/1.1" 200 -
```

Screenshot web browser visits your site



Assignment 6.5: Network segment

Remember that bitwise java application you've made in week 2? Expand that application so that you can also calculate a network segment as explained in the PowerPoint slides of week 6. Use the bitwise & AND operator. You need to be able to input two Strings. An IP address and a subnet.

IP: 192.168.1.100 and subnet: 255.255.255.224 for /27

Example: 192.168.1.100/27

Calculate the network segment

IP Address: 11000000.10101000.00000001.01100100

Subnet Mask: 11111111.11111111.11111111.11100000

Network Addr: 11000000.10101000.00000001.01100000

This gives 192.168.1.96 in decimal as the network address.

For a /27 subnet, each segment (or subnet) has 32 IP addresses (2^5).

The range of this network segment is from 192.168.1.96 to 192.168.1.127.

Paste source code here, with a screenshot of a working application.

```
>Main.java ×

1 import java.util.Scanner;
2
3 public class Main {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6
7         System.out.println("--- Bitwise Operations Menu ---");
8         System.out.println("1. Is number odd?");
9         System.out.println("2. Is number a power of 2?");
10        System.out.println("3. Two's complement of number?");
11        System.out.println("4. Calculate Network Address (IP & Subnet)");
12        System.out.print("Choose an option (1-4): ");
13
14        int choice = scanner.nextInt();
15
16        // Consume the leftover newline character to prevent skipping input later
17        scanner.nextLine();
18
19        switch (choice) {
20            case 1:
21                System.out.print("Enter an integer: ");
22                int number1 = scanner.nextInt();
23                if (isOdd(number1)) {
24                    System.out.println(number1 + " is an ODD number.");
25                } else {
26                    System.out.println(number1 + " is an EVEN number.");
27                }
28                break;
29
30            case 2:
31                System.out.print("Enter an integer: ");
32                int number2 = scanner.nextInt();
33                if (isPowerOfTwo(number2)) {
34                    System.out.println(number2 + " IS a power of 2.");
35                } else {
36                    System.out.println(number2 + " is NOT a power of 2.");
37                }
38                break;
39
40            case 3:
41                System.out.print("Enter an integer: ");
42                int number3 = scanner.nextInt();
43                int result = getTwosComplement(number3);
44                System.out.println("The Two's Complement of " + number3 + " is: " + result);
45                break;
46        }
47    }
48}
```

```

47     case 4:
48         // New logic for IP and Subnet
49         System.out.print("Enter IP Address (e.g., 192.168.1.100): ");
50         String ipStr = scanner.nextLine();
51
52         System.out.print("Enter Subnet Mask (e.g., 255.255.255.224): ");
53         String subStr = scanner.nextLine();
54
55         String networkAddress = calculateNetworkAddress(ipStr, subStr);
56
57         System.out.println("\n--- Result ---");
58         System.out.println("IP Address:      " + ipStr);
59         System.out.println("Subnet Mask:     " + subStr);
60         System.out.println("Network Address: " + networkAddress);
61         break;
62
63     default:
64         System.out.println("Invalid option selected.");
65     }
66
67     scanner.close();
68 }
69
70 public static boolean isOdd(int n) { 1 usage
71     return (n & 1) == 1;
72 }
73
74 public static boolean isPowerOfTwo(int n) { 1 usage
75     return (n > 0) && ((n & (n - 1)) == 0);
76 }
77
78 public static int getTwosComplement(int n) { 1 usage
79     return ~n + 1;
80 }
81
82 @
83 public static String calculateNetworkAddress(String ip, String subnet) { 1 usage
84     String[] ipParts = ip.split( regex: "\\\\" );
85     String[] subParts = subnet.split( regex: "\\\\" );
86
87     // Validation to ensure both are standard IPv4 format
88     if (ipParts.length != 4 || subParts.length != 4) {
89         return "Error: Invalid IP or Subnet format.";
90     }

```

```

91     StringBuilder networkBuilder = new StringBuilder();
92
93     for (int i = 0; i < 4; i++) {
94         int ipOctet = Integer.parseInt(ipParts[i]);
95         int subOctet = Integer.parseInt(subParts[i]);
96
97         int networkOctet = ipOctet & subOctet;
98
99         networkBuilder.append(networkOctet);
100
101        if (i < 3) {
102            networkBuilder.append(".");
103        }
104    }
105
106
107    return networkBuilder.toString();
108}
109

```

```

--- Bitwise Operations Menu ---
1. Is number odd?
2. Is number a power of 2?
3. Two's complement of number?
4. Calculate Network Address (IP & Subnet)
Choose an option (1-4): 4
Enter IP Address (e.g., 192.168.1.100): 192.168.1.100
Enter Subnet Mask (e.g., 255.255.255.224): 255.255.255.224

--- Result ---
IP Address:      192.168.1.100
Subnet Mask:     255.255.255.224
Network Address: 192.168.1.96

Process finished with exit code 0

```

Ready? Save this file and export it as a pdf file with the name: [week6.pdf](#)