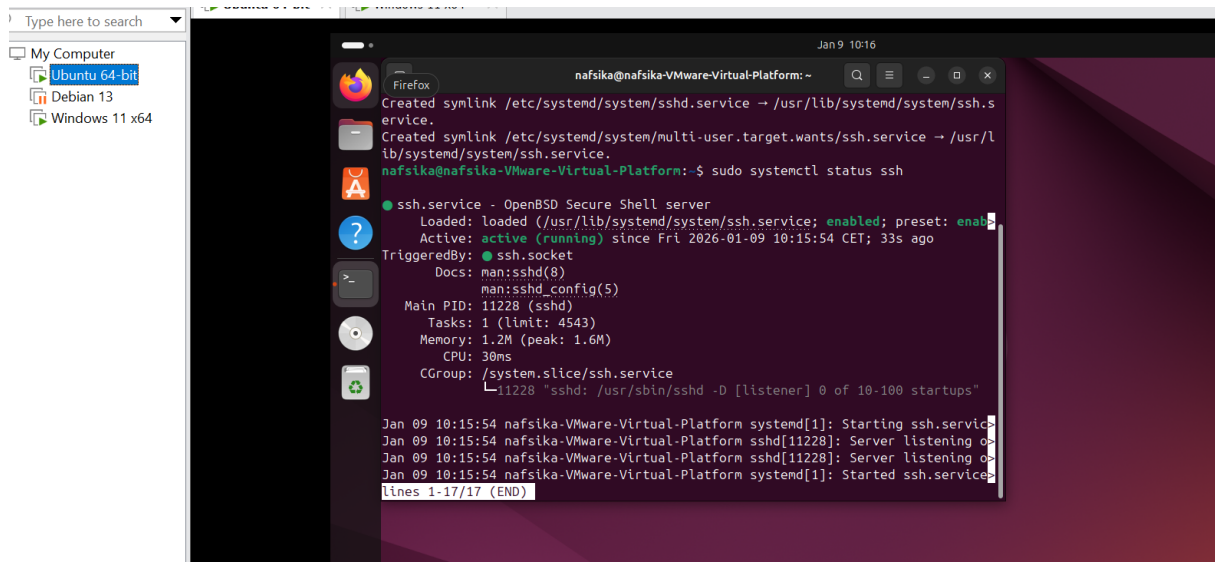


Template Week 6 – Networking

Student number: 579185 – Nafsika Pagkali

Assignment 6.1: Working from home

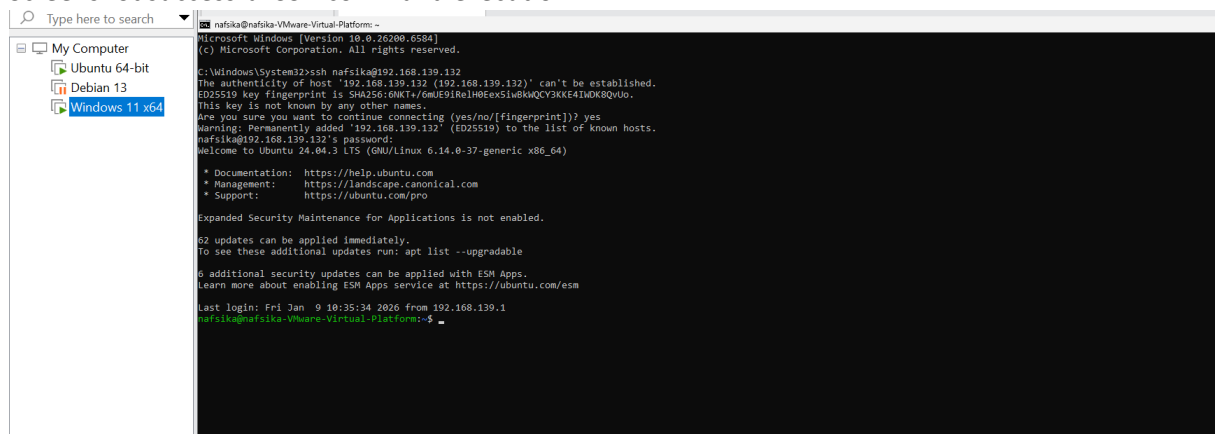
Screenshot installation openssh-server:



```
Created symlink /etc/systemd/system/ssh.service → /usr/lib/systemd/system/ssh.service.
Created symlink /etc/systemd/system/multi-user.target.wants/ssh.service → /usr/lib/systemd/system/ssh.service.
nafsika@nafsika-VMware-Virtual-Platform:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: enabled)
   Active: active (running) since Fri 2026-01-09 10:15:54 CET; 33s ago
   TriggeredBy: ● ssh.socket
     Docs: man:sshd(8)
           man:sshd_config(5)
    Main PID: 11228 (sshd)
      Tasks: 1 (limit: 4543)
     Memory: 1.2M (peak: 1.6M)
        CPU: 30ms
    CGroup: /system.slice/ssh.service
            └─11228 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Jan 09 10:15:54 nafsika-VMware-Virtual-Platform systemd[1]: Starting ssh.service:
Jan 09 10:15:54 nafsika-VMware-Virtual-Platform sshd[11228]: Server listening on
Jan 09 10:15:54 nafsika-VMware-Virtual-Platform sshd[11228]: Server listening on
Jan 09 10:15:54 nafsika-VMware-Virtual-Platform systemd[1]: Started ssh.service:
lines 1-17/17 (END)
```

Screenshot successful SSH command execution:



```
Microsoft Windows [Version 10.0.26200.6584]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>ssh nafsika@192.168.139.132
The authenticity of host '192.168.139.132 (192.168.139.132)' can't be established.
ED25519 key fingerprint is SHA256:0MKTa/6mUE91ReIH0EexSiw8kMQCY3KKE41MDK8QvUo.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.139.132' (ED25519) to the list of known hosts.
nafsika@192.168.139.132's password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-37-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

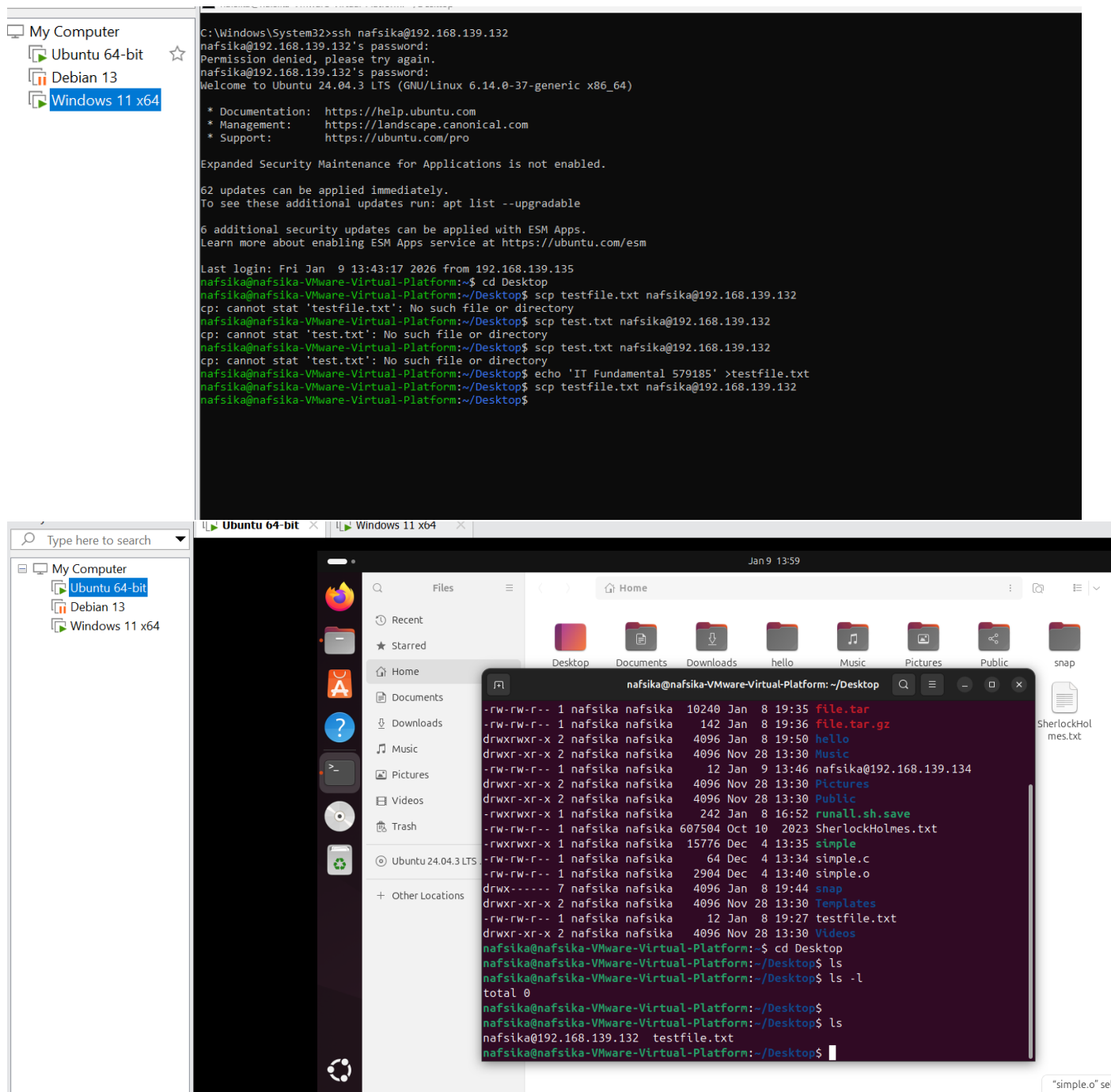
Expanded Security Maintenance for Applications is not enabled.

62 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

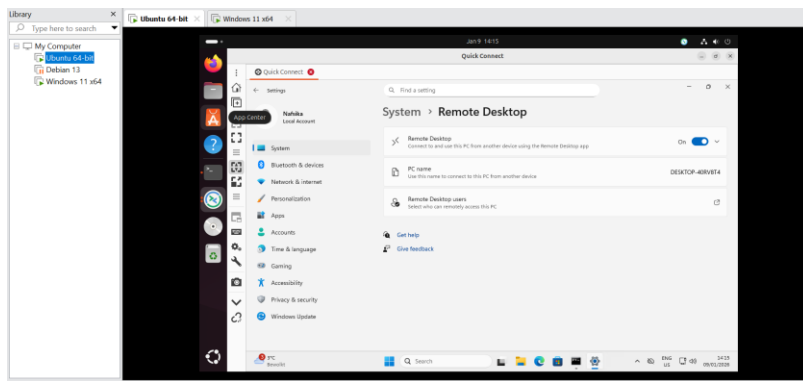
6 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

Last login: Fri Jan  9 10:35:34 2026 from 192.168.139.1
nafsika@nafsika-VMware-Virtual-Platform:~$
```

Screenshot successful execution SCP command:

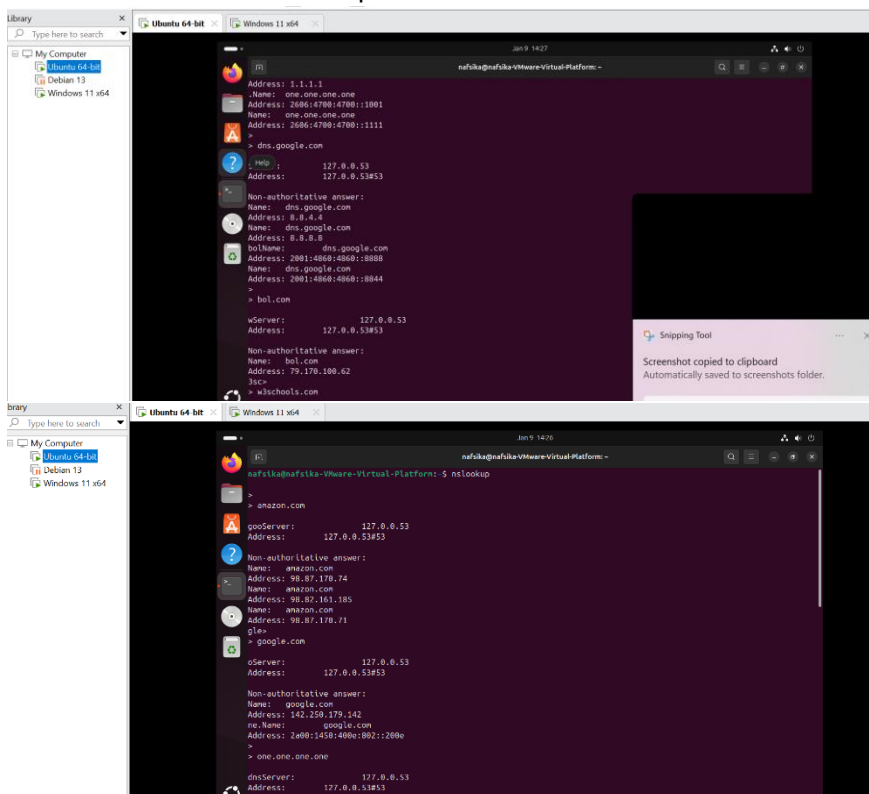


Screenshot remmina:

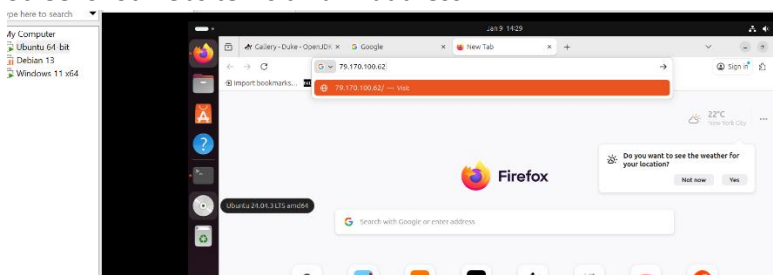


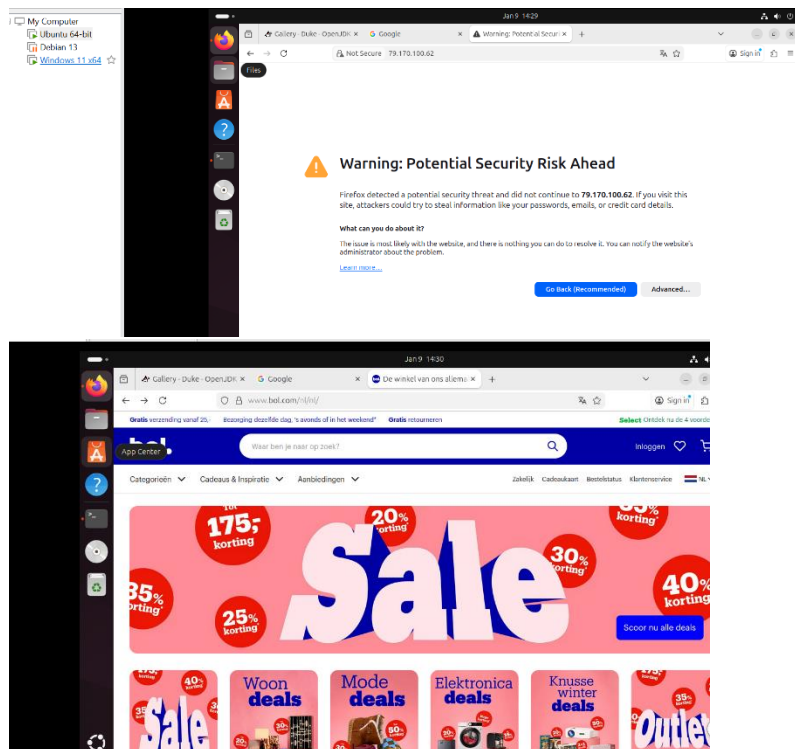
Assignment 6.2: IP addresses websites

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?

25 subnet means = $32 - 25 = 7$ bits

Number of IP addresses is 2 to the power of 7 = 128 Ips

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

192.168.110.129 – 192.168.110.254

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

```
sudo apt install ipcalc
nafsika@nafsika-VMware-Virtual-Platform:~$ sudo apt install ipcalc
[sudo] password for nafsika:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  ipcalc
0 upgraded, 1 newly installed, 0 to remove and 72 not upgraded.
Need to get 24.5 kB of archives.
After this operation, 72.7 kB of additional disk space will be used.
Get:1 http://nl.archive.ubuntu.com/ubuntu noble/universe amd64 ipcalc all 0.51-1 [24.5 kB]
Fetched 24.5 kB in 0s (217 kB/s)
Selecting previously unselected package ipcalc.
(Reading database ... 203868 files and directories currently installed.)
Preparing to unpack .../archives/ipcalc_0.51-1_all.deb ...
Unpacking ipcalc (0.51-1) ...
Setting up ipcalc (0.51-1) ...
Processing triggers for man-db (2.12.0-4build2) ...
nafsika@nafsika-VMware-Virtual-Platform:~$ ipcalc 192.168.110.128/25
Address: 192.168.110.128      11000000,10101000,01101110,1 0000000
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 0000001
HostMax: 192.168.110.254    11000000,10101000,01101110,1 1111110
Broadcast: 192.168.110.255  11000000,10101000,01101110,1 1111111
Hosts/Net: 126              Class C, Private Internet
nafsika@nafsika-VMware-Virtual-Platform:~$
```

Explain the above calculation in your own words.

A /25 subnet uses 25 bits for the network part and 7 bits for hosts.

This results in 128 total IP addresses per subnet where the first address is reserved as the network address, and the last address is reserved as the broadcast address leaving 126 available IP addresses. Here the subnet starts at 192.168.110.128 and ends at 192.168.110.255.

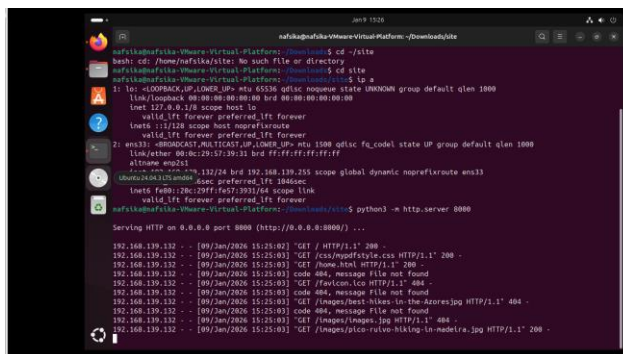
Assignment 6.4: HTML

Screenshot IP address Ubuntu VM:

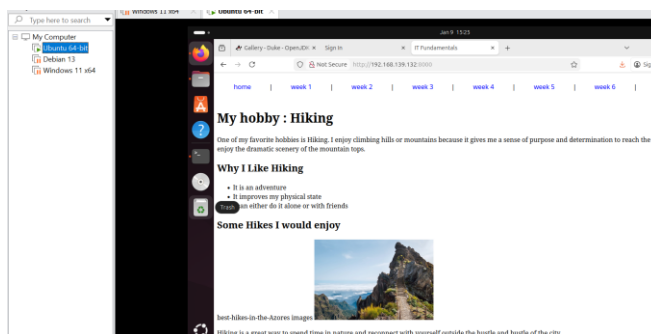
```
bash: cd: /home/nafsika/Site: No such file or directory
nafsika@nafsika-VMware-Virtual-Platform:~/Downloads$ cd ~/Site
bash: cd: /home/nafsika/Site: No such file or directory
nafsika@nafsika-VMware-Virtual-Platform:~/Downloads$ cd ~/site
bash: cd: /home/nafsika/site: No such file or directory
nafsika@nafsika-VMware-Virtual-Platform:~/Downloads$ cd site
nafsika@nafsika-VMware-Virtual-Platform:~/Downloads/site$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:00:00:00 brd ff:ff:ff:ff:ff:ff
    inet 192.168.139.132/24 brd 192.168.139.255 scope global dynamic noprefixroute ens33
        valid_lft 1046sec preferred_lft 1046sec
    inet6 fe80::20c:29ff:fe57:3931/64 scope link
        valid_lft forever preferred_lft forever
```

Screenshot of Site directory contents:

Screenshot python3 webserver command:



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.

```
import java.util.Scanner;

public class Bitwise {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a number: ");
        int number = scanner.nextInt();
        scanner.nextLine(); // consume newline

        int choice = 0;

        while (choice != 6) {
            System.out.println("\n===== BITWISE OPERATIONS MENU =====");
            System.out.println("1. Is number odd?");
            System.out.println("2. Is number a power of 2?");
            System.out.println("3. Two's complement of number");
            System.out.println("4. Change number");
            System.out.println("5. Calculate network segment");
            System.out.println("6. Exit");
            System.out.print("Enter your choice: ");

            choice = scanner.nextInt();
            scanner.nextLine(); // consume newline

            switch (choice) {

                case 1:
                    if (isOdd(number)) {
                        System.out.println(number + " is odd.");
                    } else {
                        System.out.println(number + " is even.");
                    }
                    break;

                case 2:
                    if (isPowerOfTwo(number)) {
                        System.out.println(number + " is a power of 2.");
                    } else {
                        System.out.println(number + " is NOT a power of 2.");
                    }
                    break;
```

```

case 3:
    System.out.println("Two's complement of " + number + " is: "
        + twosComplement(number));
    break;

case 4:
    System.out.print("Enter a new number: ");
    number = scanner.nextInt();
    scanner.nextLine();
    break;

case 5:
    System.out.print("Enter IP address (e.g. 192.168.1.100): ");
    String ip = scanner.nextLine();

    System.out.print("Enter subnet mask (e.g. 255.255.255.224): ");
    String subnet = scanner.nextLine();

    calculateNetworkSegment(ip, subnet);
    break;

case 6:
    System.out.println("Exiting program...");
    break;

default:
    System.out.println("Invalid choice. Try again.");
}
}

scanner.close();
}

// BITWISE METHODS

// Check if number is odd using bitwise AND
public static boolean isOdd(int num) {
    return (num & 1) == 1;
}

// Check if number is power of 2
public static boolean isPowerOfTwo(int num) {
    return num > 0 && (num & (num - 1)) == 0;
}

// Calculate two's complement
public static int twosComplement(int num) {
    return (~num) + 1;
}

```



```

}

// NETWORK SEGMENT

public static void calculateNetworkSegment(String ip, String subnet) {

    int ipInt = ipToInt(ip);
    int subnetInt = ipToInt(subnet);

    // Bitwise AND to get network address
    int networkInt = ipInt & subnetInt;

    // Calculate subnet size
    int hostBits = 32 - Integer.bitCount(subnetInt);
    int totalIPs = 1 << hostBits;

    int broadcast = networkInt + totalIPs - 1;

    System.out.println("\nNetwork Address: " + intToIp(networkInt));
    System.out.println("Subnet size: " + totalIPs + " IP addresses");
    System.out.println("Network range: " + intToIp(networkInt)
        + " - " + intToIp(broadcast));
}

// Convert dotted decimal IP to int
public static int ipToInt(String ip) {
    String[] parts = ip.split("\\.");
    int result = 0;

    for (String part : parts) {
        result = (result << 8) | Integer.parseInt(part);
    }

    return result;
}

// Convert int back to dotted decimal IP
public static String intToIp(int ip) {
    return ((ip >> 24) & 0xFF) + "." +
        ((ip >> 16) & 0xFF) + "." +
        ((ip >> 8) & 0xFF) + "." +
        (ip & 0xFF);
}
}

```

```
C:\Users\utente\jdk\ms-21.0.9\bin\java.exe --enable-preview "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2025.2.1\lib\idea_rt.jar"
Enter a number: 5

===== BITWISE OPERATIONS MENU =====
1. Is number odd?
2. Is number a power of 2?
3. Two's complement of number
4. Change number
5. Calculate network segment
6. Exit
Enter your choice: 5
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

Network Address: 192.168.1.96
Subnet size: 32 IP addresses
Network range: 192.168.1.96 - 192.168.1.127

===== BITWISE OPERATIONS MENU =====
```

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