

Template Week 6 – Networking

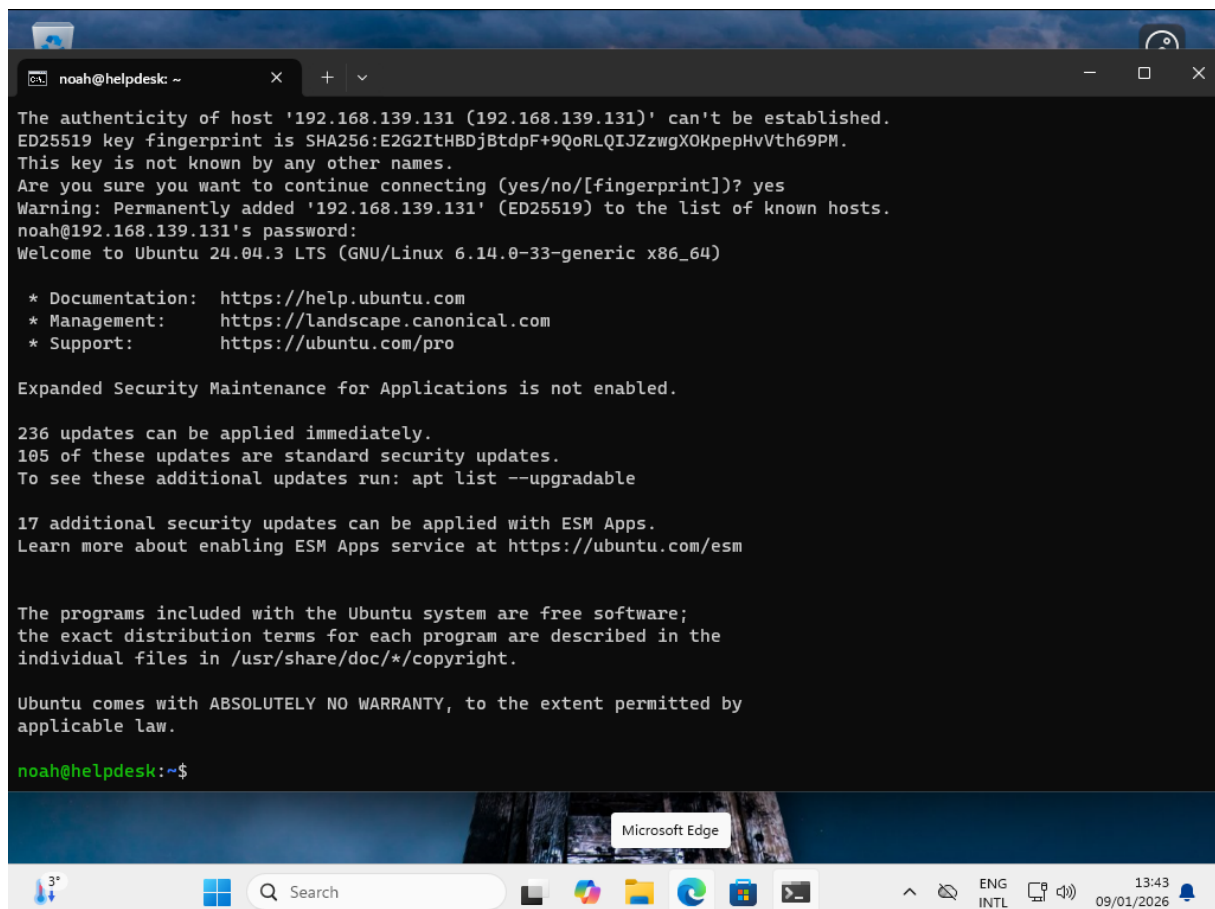
Student number: 499193

Assignment 6.1: Working from home

Screenshot installation openssh-server:

```
noah@helpdesk:~$ sudo systemctl enable 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/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.
noah@helpdesk:~$ sudo systemctl status ssh
○ ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: enabled)
   Active: inactive (dead)
 TriggeredBy: ● ssh.socket
     Docs: man:ssh(8)
           man:ssh_config(5)
lines 1-6/6 (END)
```

Screenshot successful SSH command execution:



```
noah@helpdesk: ~
The authenticity of host '192.168.139.131 (192.168.139.131)' can't be established.
ED25519 key fingerprint is SHA256:E2G2ItHBDjBtdpF+9QoRLQIJZzwgXOKpepHvVth69PM.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.139.131' (ED25519) to the list of known hosts.
noah@192.168.139.131's password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-33-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.

236 updates can be applied immediately.
105 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

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

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

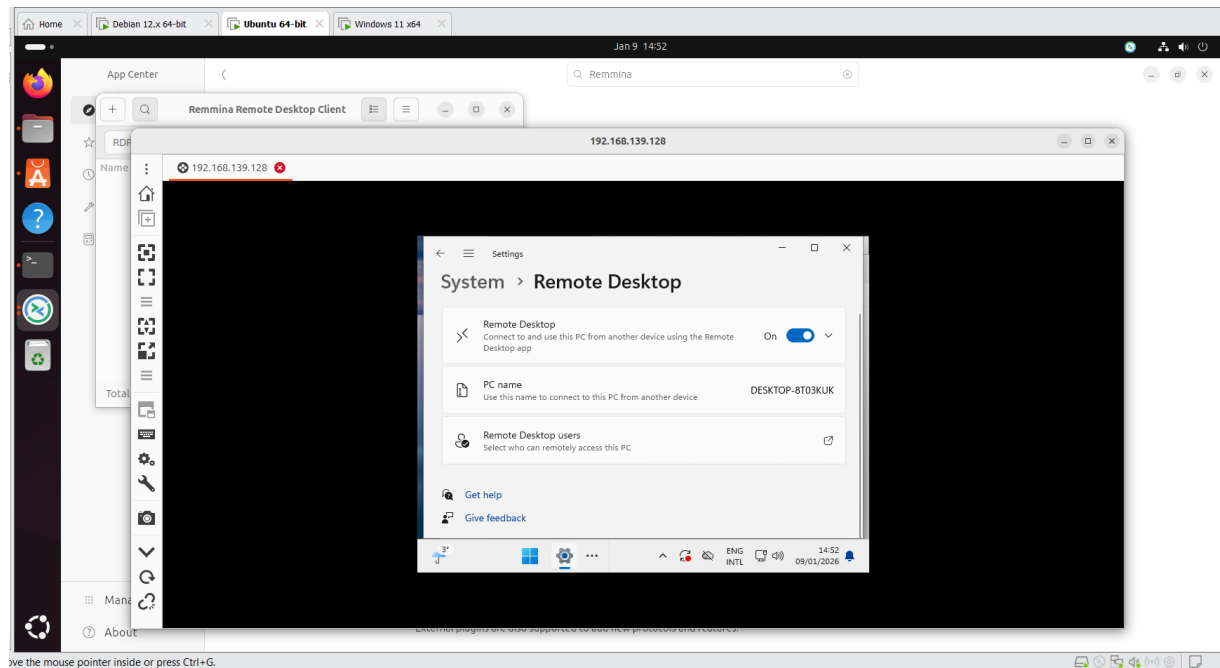
noah@helpdesk:~$
```

Screenshot successful execution SCP command:

```
C:\Users\noah>"%LOCALAPPDATA%\Programs\WinSCP\winscp.com"/command "open sftp://noah:Welkom01@192.168.139.131"
"put C:\Users\noah\Documents\bestand.txt /home/noah/" exit
Searching for host...
Connecting to host...
Authenticating...
Using username "noah".
Authenticating with pre-entered password.
Authenticated.
Starting the session...
Session started.
Active session: [1] noah@192.168.139.131
C:\...\bestand.txt | 7 B | 0,0 KB/s | binary | 100%

C:\Users\noah>
```

Screenshot remmina:



Assignment 6.2: IP addresses websites

Relevant screenshots nslookup command:

```
C:\Users\noahv>nslookup amazon.com
Server: UnKnown
Address: 10.0.0.1

Non-authoritative answer:
Name: amazon.com
Addresses: 98.87.170.74
          98.87.170.71
          98.82.161.185

C:\Users\noahv>nslookup google.com
Server: UnKnown
Address: 10.0.0.1

Non-authoritative answer:
Name: google.com
Addresses: 2a00:1450:400e:801::200e
          142.250.179.174

C:\Users\noahv>nslookup one.one.one.one
Server: UnKnown
Address: 10.0.0.1

Non-authoritative answer:
Name: one.one.one.one
Addresses: 2606:4700:4700::1001
          2606:4700:4700::1111
          1.1.1.1
          1.0.0.1

C:\Users\noahv>nslookup dns.google.com
Server: UnKnown
Address: 10.0.0.1

Non-authoritative answer:
Name: dns.google.com
Addresses: 2001:4860:4860::8888
          2001:4860:4860::8844
          8.8.8.8
          8.8.4.4

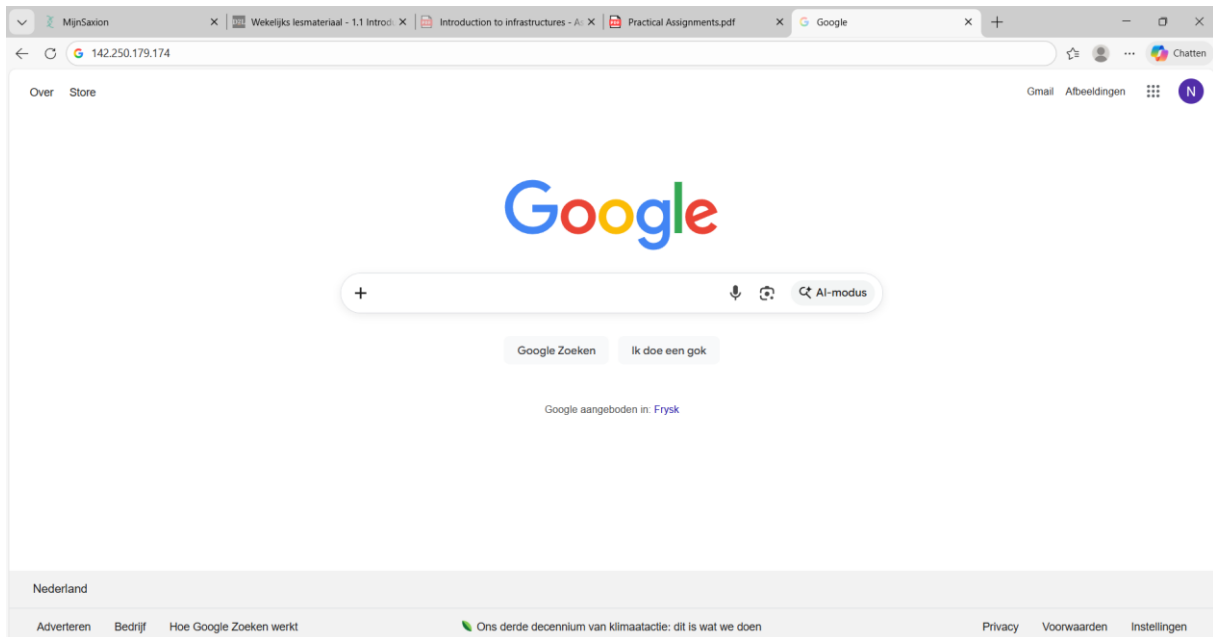
C:\Users\noahv>nslookup bol.com
Server: UnKnown
Address: 10.0.0.1

Non-authoritative answer:
Name: bol.com
Address: 79.170.100.62

C:\Users\noahv>nslookup w3schools.com
DNS request timed out.
  timeout was 2 seconds.
Server: UnKnown
Address: 10.0.0.1

DNS request timed out.
  timeout was 2 seconds.
DNS request timed out.
  timeout was 2 seconds.
Non-authoritative answer:
Name: w3schools.com
Addresses: 76.223.115.82
          13.248.240.135
```

Screenshot website visit via IP address:



Assignment 6.3: subnetting

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

$$32 - 25 = 7$$

$$2^7 = \underline{128}$$

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

192.168.110.129 – 192.168.110.254

.128 network address

.255 broadcast

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

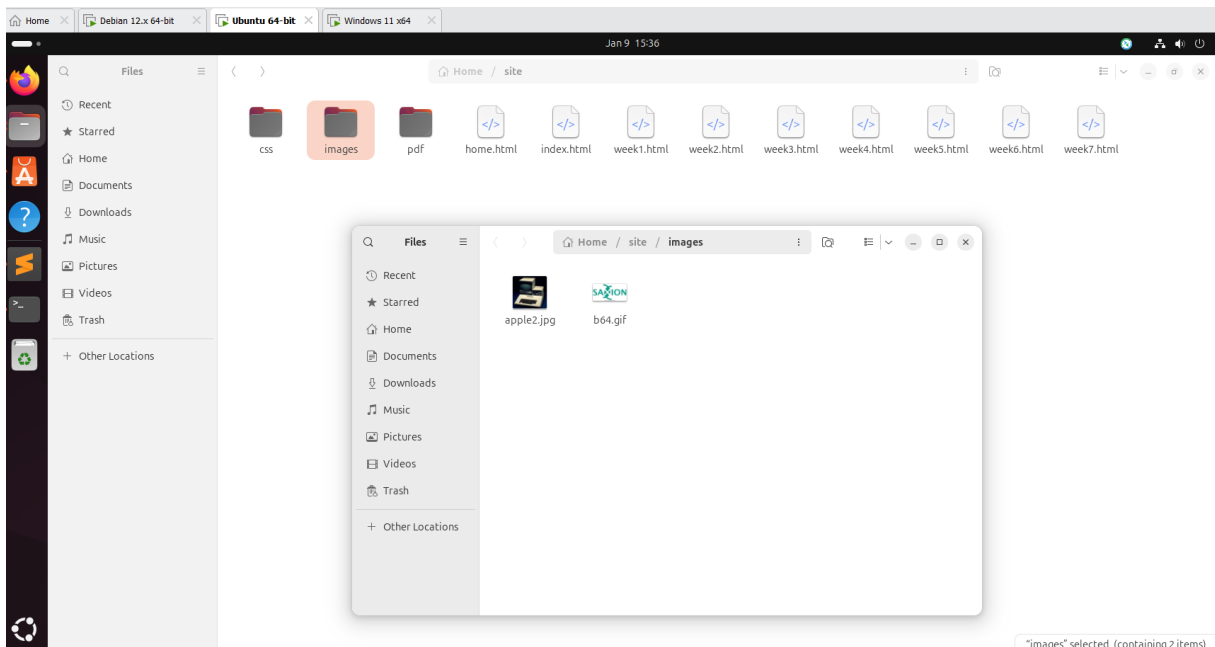
```
noah@helptest:~$ ipcalc 192.168.110.128/25
Address: 192.168.110.128      11000000.10101000.01101110.1 0000000
Netmask: 255.255.255.128 = 25 11111111.11111111.11111111.1 0000000
Wildcard: 0.0.0.127          00000000.00000000.00000000.0 1111111
=>
Network: 192.168.110.128/25  11000000.10101000.01101110.1 0000000
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
```

Assignment 6.4: HTML

Screenshot IP address Ubuntu VM:

```
noah@helpdesk:~/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 00:0c:29:45:dd:1c brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.139.131/24 brd 192.168.139.255 scope global dynamic noprefixroute ens33
        valid_lft 1392sec preferred_lft 1392sec
noah@helpdesk:~/site$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
192.168.139.128 - - [09/Jan/2026 15:32:56] "GET / HTTP/1.1" 200 -
192.168.139.128 - - [09/Jan/2026 15:32:56] "GET /css/mypdfstyle.css HTTP/1.1" 200 -
192.168.139.128 - - [09/Jan/2026 15:32:56] "GET /home.html HTTP/1.1" 200 -
192.168.139.128 - - [09/Jan/2026 15:32:56] code 404, message File not found
192.168.139.128 - - [09/Jan/2026 15:32:56] "GET /favicon.ico HTTP/1.1" 404 -
192.168.139.128 - - [09/Jan/2026 15:32:56] "GET /images/b64.gif HTTP/1.1" 200 -
192.168.139.128 - - [09/Jan/2026 15:32:56] "GET /images/apple2.jpg HTTP/1.1" 200 -
192.168.139.128 - - [09/Jan/2026 15:34:07] "GET / HTTP/1.1" 304 -
192.168.139.128 - - [09/Jan/2026 15:34:07] "GET /home.html HTTP/1.1" 200 -
```

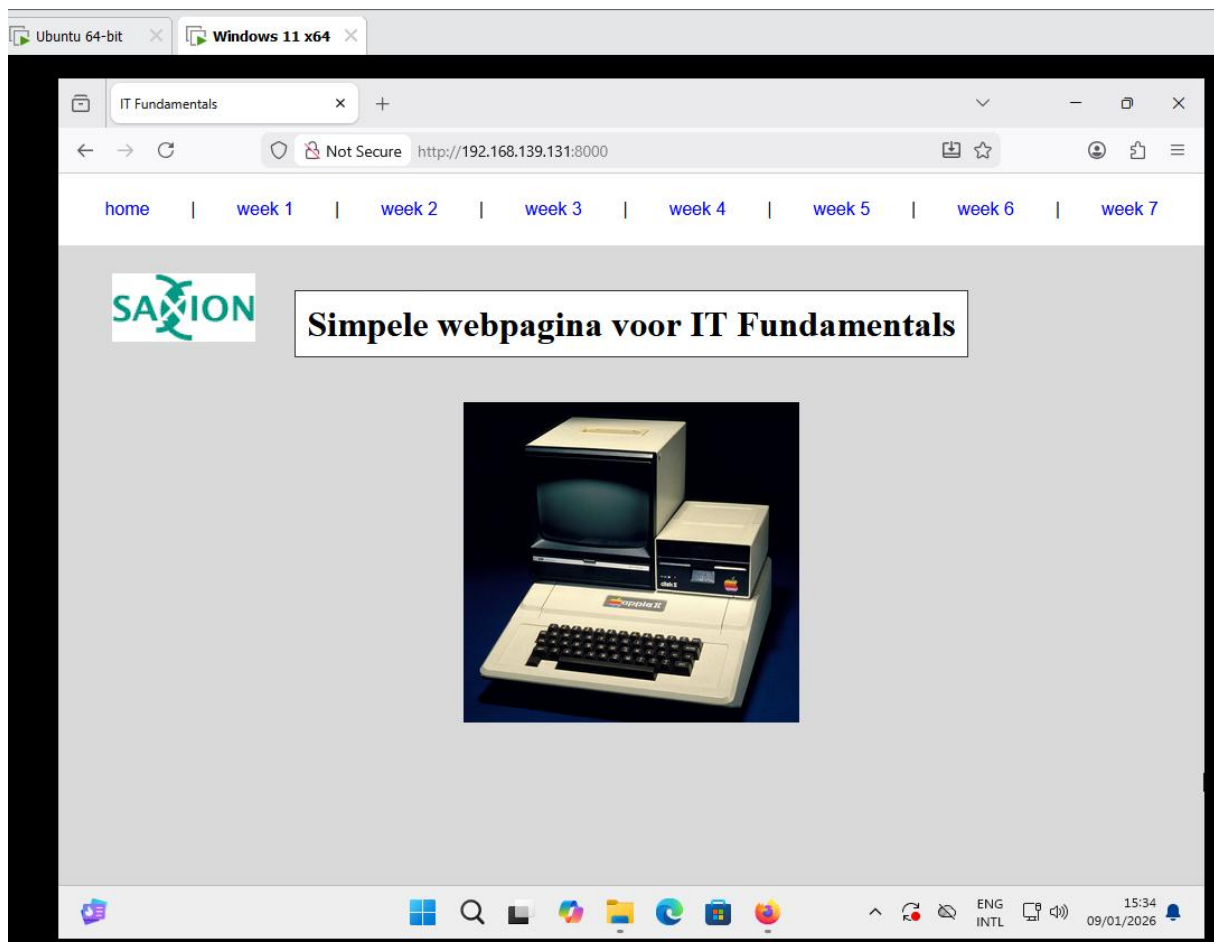
Screenshot of Site directory contents:



Screenshot python3 webserver command:

```
noah@helpdesk:~/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 00:0c:29:45:dd:1c brd ff:ff:ff:ff:ff:ff
   altname enp2s1
   inet 192.168.139.131/24 brd 192.168.139.255 scope global dynamic noprefixroute ens33
       valid_lft 1392sec preferred_lft 1392sec
noah@helpdesk:~/site$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
192.168.139.128 - - [09/Jan/2026 15:32:56] "GET / HTTP/1.1" 200 -
192.168.139.128 - - [09/Jan/2026 15:32:56] "GET /css/mypdfstyle.css HTTP/1.1" 200 -
192.168.139.128 - - [09/Jan/2026 15:32:56] "GET /home.html HTTP/1.1" 200 -
192.168.139.128 - - [09/Jan/2026 15:32:56] code 404, message File not found
192.168.139.128 - - [09/Jan/2026 15:32:56] "GET /favicon.ico HTTP/1.1" 404 -
192.168.139.128 - - [09/Jan/2026 15:32:56] "GET /images/b64.gif HTTP/1.1" 200 -
192.168.139.128 - - [09/Jan/2026 15:32:56] "GET /images/apple2.jpg HTTP/1.1" 200 -
192.168.139.128 - - [09/Jan/2026 15:34:07] "GET / HTTP/1.1" 304 -
192.168.139.128 - - [09/Jan/2026 15:34:07] "GET /home.html 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.

```
import java.util.Scanner;

public class Main {

    public static void main(String[] args) {
        Scanner userInput = new Scanner(System.in);
        boolean continueProgram = true;

        while (continueProgram) {
            System.out.println("\nPlease select an option: ");
            System.out.println("1) Bit Calculations");
            System.out.println("2) Calculate Network Segment");
            System.out.println("0) Exit");
            int choice = userInput.nextInt();
            userInput.nextLine();

            if (choice == 1) {
                bitMenu(userInput);
            } else if (choice == 2) {
                System.out.println("Enter IP address: ");
                String ip = userInput.nextLine();

                System.out.println("Enter subnet mask: ");
                String subnet = userInput.nextLine();

                calculateNetworkDetails(ip, subnet);
            }
        }
    }
}
```

```

    } else if (choice == 0) {
        continueProgram = false;
    }
}
}

// -----
// BIT CALCULATIONS MENU
// -----
public static void bitMenu(Scanner userInput) {
    System.out.println("Enter a number for bit calculations: ");
    int number = userInput.nextInt();
    userInput.nextLine();

    boolean back = false;

    while (!back) {
        System.out.println("\nBit Calculation Menu");
        System.out.println("1) Is number odd?");
        System.out.println("2) Is number a power of two?");
        System.out.println("3) Show two's complement");
        System.out.println("0) Back to main menu");

        int input = userInput.nextInt();
        userInput.nextLine();

        if (input == 1) {
            if ((number & 1) == 1)
                System.out.println("Number is odd");
            else
                System.out.println("Number is even");
        }

        } else if (input == 2) {
            if (number > 0 && (number & (number - 1)) == 0)
                System.out.println("Number is a power of 2");
            else
                System.out.println("Number isn't a power of 2");
        }

        } else if (input == 3) {
            int twocomp = ~number + 1;
            System.out.println("Two's complement: " + twocomp);
        }

        } else if (input == 0) {
            back = true;
        }
    }
}
}

```

```

// NETWORK CALCULATION
// -----
public static void calculateNetworkDetails(String ip, String subnet) {

    String[] ipParts = ip.split("\\.");
    String[] subnetParts = subnet.split("\\.");

    int[] ipOctets = new int[4];
    int[] maskOctets = new int[4];
    int[] networkOctets = new int[4];

    int cidr = 0;

    for (int i = 0; i < 4; i++) {
        ipOctets[i] = Integer.parseInt(ipParts[i]);
        maskOctets[i] = Integer.parseInt(subnetParts[i]);
        networkOctets[i] = ipOctets[i] & maskOctets[i];

        cidr += Integer.bitCount(maskOctets[i]); // count 1-bits
    }

    // Convert to binary strings
    String ipBinary = toBinary(ipOctets);
    String maskBinary = toBinary(maskOctets);
    String networkBinary = toBinary(networkOctets);

    // Subnet size
    int hostBits = 32 - cidr;
    int totalIPs = (int) Math.pow(2, hostBits);

    // Range
    int firstIP = networkOctets[3];
    int lastIP = networkOctets[3] + totalIPs - 1;

    System.out.println("\n-----");
    System.out.println("IP Address: " + ipBinary);
    System.out.println("Subnet Mask: " + maskBinary);
    System.out.println("-----");
    System.out.println("Network Addr: " + networkBinary);

    System.out.println("This gives " +
        networkOctets[0] + "." +
        networkOctets[1] + "." +
        networkOctets[2] + "." +
        networkOctets[3] +
        " in decimal as the network address.");
}

```

```
System.out.println("For a /" + cidr + " subnet, each segment has " +  
    totalIPs + " IP addresses.");
```

```
System.out.println("The range of this network segment is from " +  
    networkOctets[0] + "." + networkOctets[1] + "." +  
    networkOctets[2] + "." + firstIP +  
    " to " +  
    networkOctets[0] + "." + networkOctets[1] + "." +  
    networkOctets[2] + "." + lastIP);
```

```
}
```

```
// Convert 4 octets to binary
```

```
// -----
```

```
public static String toBinary(int[] octets) {  
    return String.format("%8s.%8s.%8s.%8s",  
        Integer.toBinaryString(octets[0]).replace(' ', '0'),  
        Integer.toBinaryString(octets[1]).replace(' ', '0'),  
        Integer.toBinaryString(octets[2]).replace(' ', '0'),  
        Integer.toBinaryString(octets[3]).replace(' ', '0')  
    ).replace(' ', '0');
```

```
}
```

```
}
```

```
C:\Users\noahv\.jdk\ms-21.0.8\bin\java.exe "-javaagent:C:\Program Files\
```

```
Please select an option:
```

- 1) Bit Calculations
- 2) Calculate Network Segment
- 0) Exit

```
2
```

```
Enter IP address:
```

```
192.168.1.100
```

```
Enter subnet mask:
```

```
255.255.255.224
```

```
-----  
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 has 32 IP addresses.
```

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

```
Please select an option:
```

- 1) Bit Calculations
- 2) Calculate Network Segment
- 0) Exit

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