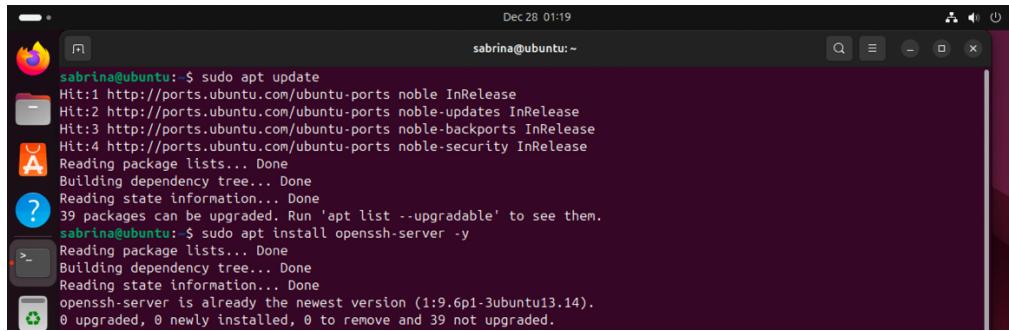


# Template Week 6 – Networking

Student number: 585303

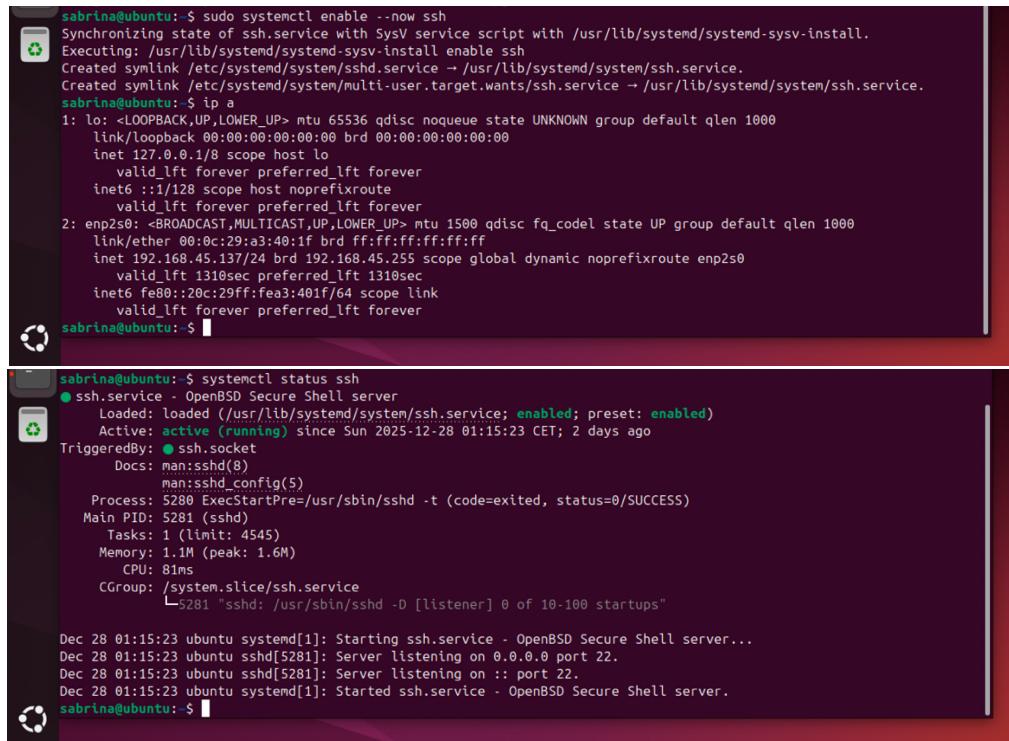
## Assignment 6.1: Working from home

Screenshot installation openssh-server:



```
sabrina@ubuntu:~$ sudo apt update
Hit:1 http://ports.ubuntu.com/ubuntu-ports noble InRelease
Hit:2 http://ports.ubuntu.com/ubuntu-ports noble-updates InRelease
Hit:3 http://ports.ubuntu.com/ubuntu-ports noble-backports InRelease
Hit:4 http://ports.ubuntu.com/ubuntu-ports noble-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
39 packages can be upgraded. Run 'apt list --upgradable' to see them.
sabrina@ubuntu:~$ 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).
0 upgraded, 0 newly installed, 0 to remove and 39 not upgraded.
```

Screenshot successful SSH command execution:



```
sabrina@ubuntu:~$ 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/ssh.service.
Created symlink /etc/systemd/system/multi-user.target.wants/ssh.service → /usr/lib/systemd/system/ssh.service.

sabrina@ubuntu:~$ 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: enp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:a3:40:1f brd ff:ff:ff:ff:ff:ff
    inet 192.168.45.137/24 brd 192.168.45.255 scope global dynamic noprefixroute enp2s0
        valid_lft 1310sec preferred_lft 1310sec
    inet6 fe80::20c:29ff:fea3:401f/64 scope link
        valid_lft forever preferred_lft forever

sabrina@ubuntu:~$ 

sabrina@ubuntu:~$ systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: enabled)
   Active: active (running) since Sun 2025-12-28 01:15:23 CET; 2 days ago
     TriggeredBy: ● ssh.socket
      Docs: man:sshd(8)
              man:sshd_config(5)
    Process: 5280 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 5281 (sshd)
      Tasks: 1 (limit: 4545)
     Memory: 1.1M (peak: 1.6M)
        CPU: 81ms
       CGroup: /system.slice/ssh.service
               └─5281 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Dec 28 01:15:23 ubuntu systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
Dec 28 01:15:23 ubuntu sshd[5281]: Server listening on 0.0.0.0 port 22.
Dec 28 01:15:23 ubuntu sshd[5281]: Server listening on :: port 22.
Dec 28 01:15:23 ubuntu systemd[1]: Started ssh.service - OpenBSD Secure Shell server.

sabrina@ubuntu:~$
```

```

Jan 8 13:17
sabrina@sabrina-VMware20-1:~ ssh sabrina@localhost
Hint: Some lines were ellipsized, use -l to show in full.
sabrina@sabrina-VMware20-1:~ $ ssh sabrina@localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ED25519 key fingerprint is SHA256:llCNPbh9Iwibbf9WljNLL+IQraSBqqt5DbGHNJF99Y.
This key is not known by any other names.
A Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'localhost' (ED25519) to the list of known hosts.
sabrina@localhost's password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-37-generic aarch64)

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

Expanded Security Maintenance for Applications is not enabled.

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

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

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.

sabrina@sabrina-VMware20-1:~ whoami
hostname

```

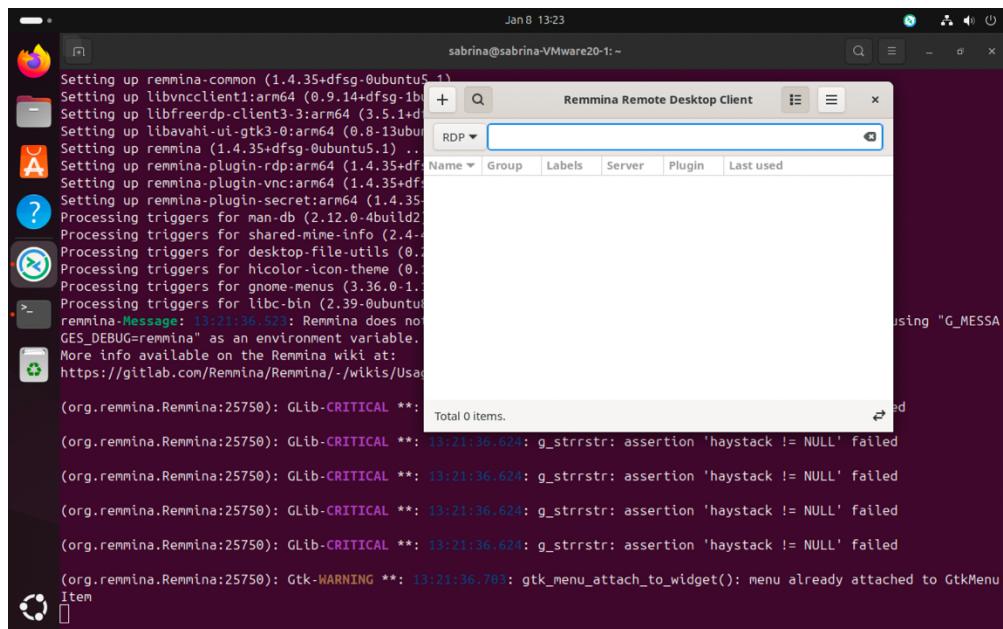
Screenshot successful execution SCP command:

```

sabrina@sabrina-VMware20-1:~ whoami
hostname
sabrina
sabrina-VMware20-1
sabrina@sabrina-VMware20-1:~ echo "SCP working Sabrina" > scp_test.txt
sabrina@sabrina-VMware20-1:~ ls -l scp_test.txt
-rw-rw-r-- 1 sabrina sabrina 20 Jan  8 13:19 scp_test.txt
sabrina@sabrina-VMware20-1:~ $ scp scp_test.txt sabrina@localhost:/home/sabrina/
sabrina@localhost's password:
scp_test.txt                                              100%   20     4.7KB/s   00:00
sabrina@sabrina-VMware20-1:~ ls -l /home/sabrina/scp_test.txt
-rw-rw-r-- 1 sabrina sabrina 20 Jan  8 13:19 /home/sabrina/scp_test.txt
sabrina@sabrina-VMware20-1:~ $ cat /home/sabrina/scp_test.txt
SCP working Sabrina
sabrina@sabrina-VMware20-1:~ 

```

Screenshot remmina:



## Assignment 6.2: IP addresses websites

Relevant screenshots nslookup command:

```
Last login: Tue Jan  6 16:06:01 on console
[unistuff@macbookair ~ % nslookup google.com
Server:      145.2.14.10
Address:     145.2.14.10#53

Non-authoritative answer:
Name:   google.com
Address: 142.250.179.142

[unistuff@macbookair ~ % nslookup amazon.com
Server:      145.2.14.10
Address:     145.2.14.10#53

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

[unistuff@macbookair ~ % nslookup one.one.one.one
Server:      145.2.14.10
Address:     145.2.14.10#53

Non-authoritative answer:
Name:   one.one.one.one
Address: 1.1.1.1
Name:   one.one.one.one
Address: 1.0.0.1

[unistuff@macbookair ~ % exit
Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.
Deleting expired sessions...      5 completed.

[Process completed]
```

```
[unistuff@macbookair ~ % nslookup dns.google.com
Server:      145.2.14.10
Address:     145.2.14.10#53

Non-authoritative answer:
Name:   dns.google.com
Address: 8.8.4.4
Name:   dns.google.com
Address: 8.8.8.8

[unistuff@macbookair ~ % nslookup bol.com
Server:      145.2.14.10
Address:     145.2.14.10#53

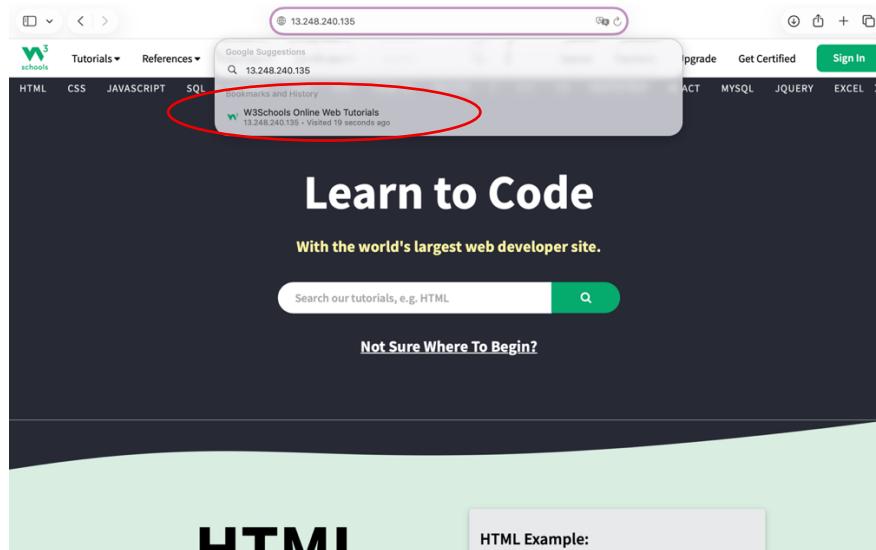
Non-authoritative answer:
Name:   bol.com
Address: 79.170.100.62

[unistuff@macbookair ~ % nslookup w3schools.com
Server:      145.2.14.10
Address:     145.2.14.10#53

Non-authoritative answer:
Name:   w3schools.com
Address: 13.248.240.135
Name:   w3schools.com
Address: 76.223.115.82

unistuff@macbookair ~ % sabrina
```

Screenshot website visit via IP address:



### Assignment 6.3: subnetting

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

128

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**

Explain the above calculation in your own words.

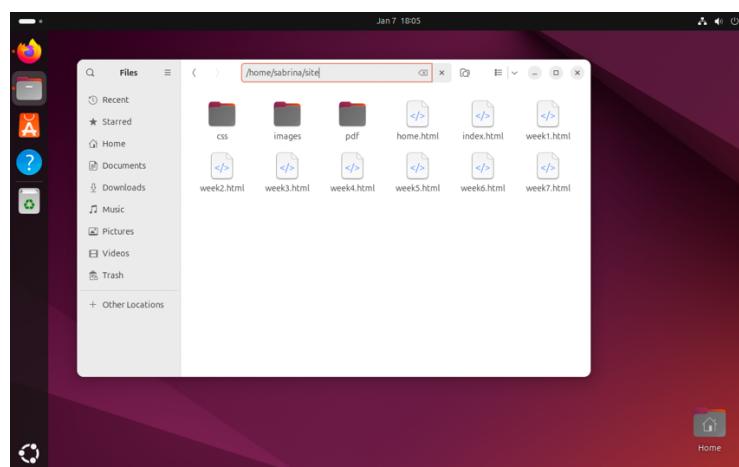
A /25 network uses 25 bits for the network and leaves 7 bits for hosts, which gives 128 total IP addresses. Two addresses are reserved for special use (the network address and the broadcast address), so 126 addresses can be used by computers. This means the usable IP range is from 192.168.110.129 to 192.168.110.254, which matches the result shown by the ipcalc command.

### Assignment 6.4: HTML

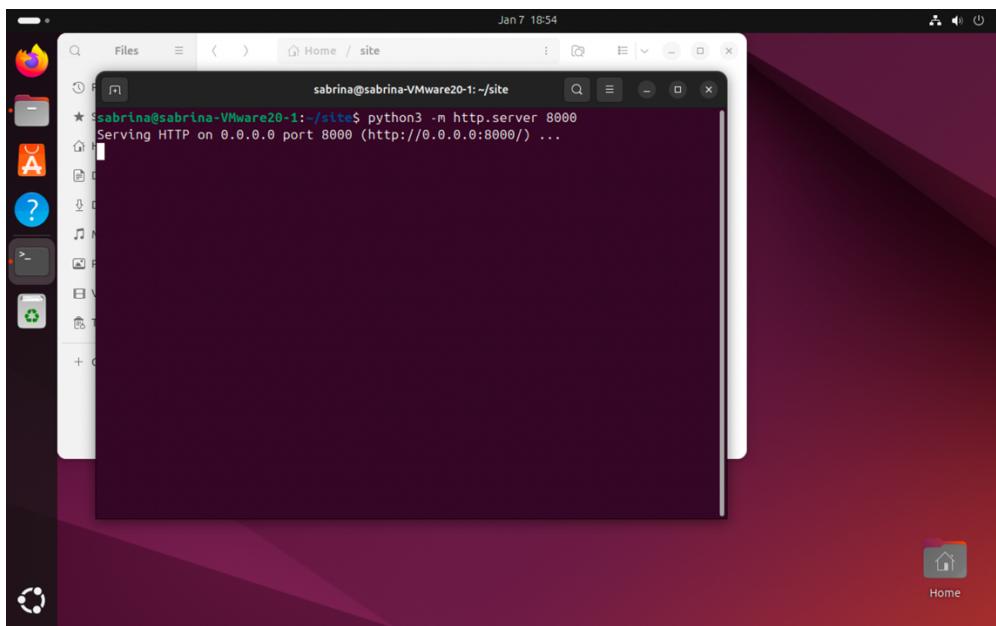
Screenshot IP address Ubuntu VM:

```
sabrina@sabrina-VMware20-1:~/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: enp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:ce:32:d1 brd ff:ff:ff:ff:ff:ff
        inet 192.168.45.141/24 brd 192.168.45.255 scope global dynamic noprefixroute
            valid_lft 1239sec preferred_lft 1239sec
        inet6 fe80::20c:29ff:fece:32d1/64 scope link
            valid_lft forever preferred_lft forever
sabrina@sabrina-VMware20-1:~/site$
```

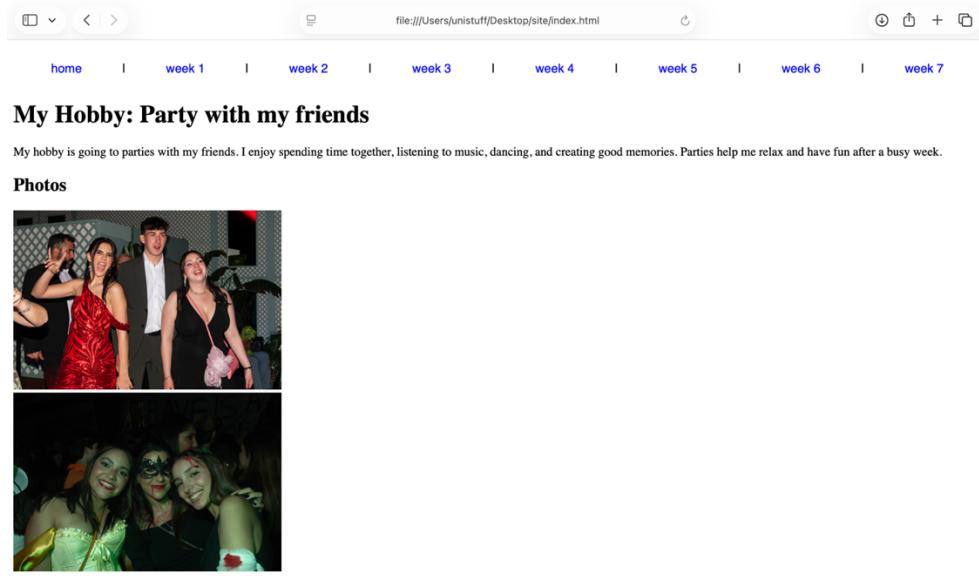
Screenshot of Site directory contents:



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 nl.saxion.app.SaxionApp;

public class Application implements Runnable {

    public static void main(String[] args) {
        SaxionApp.start(new Application(), 400, 400);
    }

    public void run() {
        boolean run = true;

        while (run) {
            SaxionApp.printLine("1. Check if a number is odd or even");
            SaxionApp.printLine("2. Check if a number is a power of 2");
            SaxionApp.printLine("3. Find the two's complement of a number");
            SaxionApp.printLine("4. Calculate network segment");
            SaxionApp.printLine("0. Exit");
            SaxionApp.print("Option: ");

            int option = SaxionApp.readInt();

            if (option == 1) {
                isOdd();
            } else if (option == 2) {
                isPower2();
            } else if (option == 3) {
                twosComp();
            } else if (option == 4) {
                networkSegment();
            } else if (option == 0) {
                run = false;
            } else {
                SaxionApp.printLine("Invalid option!");
            }

            if (run) {
                SaxionApp.pause();
                SaxionApp.clear();
            }
        }
    }

    private void isOdd() {
        SaxionApp.print("Enter a number: ");
        int number = SaxionApp.readInt();

        if ((number & 1) == 1) {
            SaxionApp.printLine("The number " + number + " is odd.");
        } else {
            SaxionApp.printLine("The number " + number + " is even.");
        }
    }

    private void isPower2() {
        SaxionApp.print("Enter a number: ");
        int number = SaxionApp.readInt();

        if (number > 0 && (number & (number - 1)) == 0) {
            SaxionApp.printLine("The number " + number + " is a power of 2.");
        } else {
            SaxionApp.printLine("The number " + number + " is not a power of 2.");
        }
    }

    private void twosComp() {
        SaxionApp.print("Enter a number: ");
        int n = SaxionApp.readInt();

        int result = (~n) + 1;
        SaxionApp.printLine("The two's complement of " + n + " is " + result + ".");
    }

    //6.5 part//

    private void networkSegment() {

```

```

SaxionApp.print("Input IP address (e.g. 192.168.1.100): ");
String ipStr = SaxionApp.readString();

SaxionApp.print("Input subnet mask (e.g. 255.255.255.224): ");
String maskStr = SaxionApp.readString();

int[] ip = splitIPv4(ipStr);
int[] mask = splitIPv4(maskStr);

int[] network = new int[4];
for (int i = 0; i < 4; i++) {
    network[i] = ip[i] & mask[i];
}

int cidr = 27;
int segmentSize = 32;
int[] broadcast = new int[4];
broadcast[0] = network[0];
broadcast[1] = network[1];
broadcast[2] = network[2];
broadcast[3] = network[3] + 31;

SaxionApp.printLine("");
SaxionApp.printLine(ipStr + "/" + cidr);
SaxionApp.printLine("Calculate the network segment");
SaxionApp.printLine("IP Address: " + toBinaryDotted(ip));
SaxionApp.printLine("Subnet Mask: " + toBinaryDotted(mask));
SaxionApp.printLine("-----");
SaxionApp.printLine("Network Addr: " + toBinaryDotted(network));
SaxionApp.printLine("");

SaxionApp.printLine("This gives " + toDecimalDotted(network) + " in decimal as the network address.");
SaxionApp.printLine("For a /" + cidr + " subnet, each segment (or subnet) has " + segmentSize + " IP addresses (2^5).");
SaxionApp.printLine("The range of this network segment is from " +
    toDecimalDotted(network) + " to " + toDecimalDotted(broadcast) + ".");
}

private int[] splitIPv4(String s) {
    String[] parts = s.trim().split("\\.");
    int[] out = new int[4];
    for (int i = 0; i < 4; i++) {
        out[i] = Integer.parseInt(parts[i]);
    }
    return out;
}

private String toBinaryDotted(int[] a) {
    return bin8(a[0]) + "." + bin8(a[1]) + "." + bin8(a[2]) + "." + bin8(a[3]);
}

private String bin8(int n) {
    String b = Integer.toBinaryString(n);
    while (b.length() < 8) {
        b = "0" + b;
    }
    return b;
}

private String toDecimalDotted(int[] a) {
    return a[0] + "." + a[1] + "." + a[2] + "." + a[3];
}
}

```

