

Template Week 6 – Networking

Student number: 550498

Assignment 6.1: Working from home

Screenshot installation openssh-server:

Screenshot successful SSH command execution:

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?

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

Check your two previous answers with this calculator:

<https://www.calculator.net/ip-subnet-calculator.html>

Explain the above calculation in your own words.

Assignment 6.4: HTML

Screenshot IP address Ubuntu VM:

Screenshot of Site directory contents:

Screenshot python3 webserver command:

Screenshot web browser visits your site

Bonus point assignment – week 6

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;
```

```
import java.awt.*;
```

```
public class Application implements Runnable {
```

```
public static void main(String[] args) {
    SaxionApp.start(new Application(), 1000, 1000); // Adjust window size as needed
}
```

```
@Override
```

```
public void run() {
    SaxionApp.println("Voer een IP-adres in (voorbeeld, 192.168.1.100):" );
    String ipAddress = SaxionApp.readString();
    SaxionApp.println("Voer een Subnet Mask in(voorbeeld, 255.255.255.224):");
    String subnetMask = SaxionApp.readString();
```

```
int ipoct1 = 0, ipoct2 = 0, ipoct3 = 0, ipoct4 = 0;
int submaskoct1 = 0, submaskoct2 = 0, submaskoct3 = 0, submaskoct4 = 0;
```

```
if (ipAddress.matches("\\d+\\.\\d+\\.\\d+\\.\\d+")) {
    String temp = "";
    int dotCount = 0;
    for (int i = 0; i < ipAddress.length(); i++) {
        char ch = ipAddress.charAt(i);
        if (ch == '.') {
            dotCount++;
            int value = Integer.parseInt(temp);
            temp = "";
            if (dotCount == 1) ipoct1 = value;
            else if (dotCount == 2) ipoct2 = value;
            else if (dotCount == 3) ipoct3 = value;
        } else {
            temp += ch;
        }
    }
    ipoct4 = Integer.parseInt(temp);
}
```

```
if (subnetMask.matches("\\d+\\.\\d+\\.\\d+\\.\\d+")) {
    String temp = "";
    int dotCount = 0;
    for (int i = 0; i < subnetMask.length(); i++) {
        char ch = subnetMask.charAt(i);
        if (ch == '.') {
            dotCount++;
            int value = Integer.parseInt(temp);
            temp = "";
            if (dotCount == 1) submaskoct1 = value;
            else if (dotCount == 2) submaskoct2 = value;
            else if (dotCount == 3) submaskoct3 = value;
        } else {
            temp += ch;
        }
    }
}
```

```

    }
}
submaskoct4 = Integer.parseInt(temp);
}

int net1 = ipoct1 & submaskoct1;
int net2 = ipoct2 & submaskoct2;
int net3 = ipoct3 & submaskoct3;
int net4 = ipoct4 & submaskoct4;

String binaryIP = toBinary(ipoct1) + "." + toBinary(ipoct2) + "." + toBinary(ipoct3) + "." +
toBinary(ipoct4);
String binarySubnet = toBinary(submaskoct1) + "." + toBinary(submaskoct2) + "." +
toBinary(submaskoct3) + "." + toBinary(submaskoct4);
String binaryNetwork = toBinary(net1) + "." + toBinary(net2) + "." + toBinary(net3) + "." +
toBinary(net4);

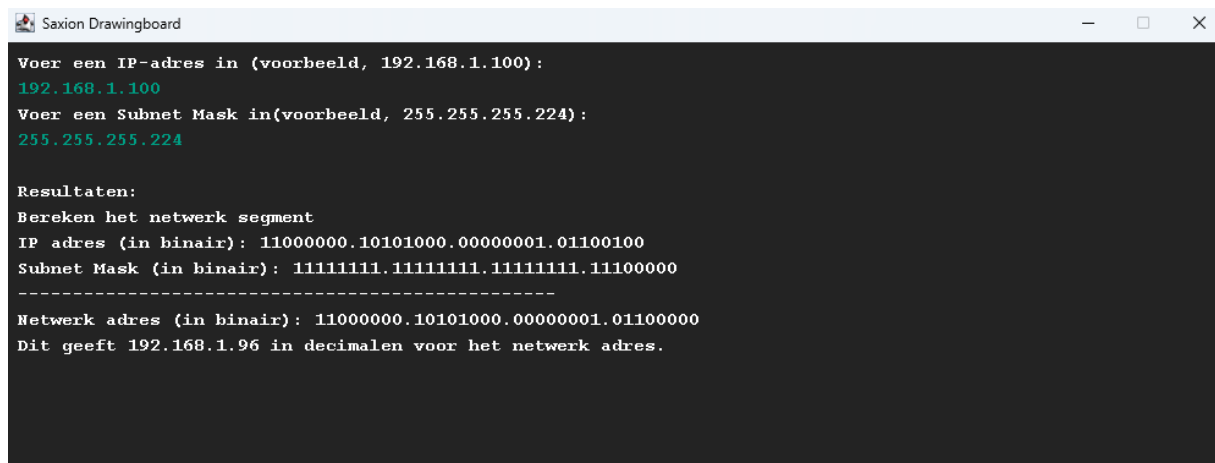
SaxionApp.printLine();
SaxionApp.printLine("Resultaten:");
SaxionApp.printLine("Bereken het netwerk segment");
SaxionApp.printLine("IP adres (in binair): " + binaryIP);
SaxionApp.printLine("Subnet Mask (in binair): " + binarySubnet);

SaxionApp.printLine("-----");
SaxionApp.printLine("Netwerk adres (in binair): " + binaryNetwork);
SaxionApp.printLine("Dit geeft " + net1 + "." + net2 + "." + net3 + "." + net4 + " in decimalen voor
het netwerk adres.");

}

private String toBinary(int number) {
    String binary = Integer.toBinaryString(number);
    while (binary.length() < 8) {
        binary = "0" + binary;
    }
    return binary;
}
}

```



```
Saxion Drawingboard
Voer een IP-adres in (voorbeeld, 192.168.1.100):
192.168.1.100
Voer een Subnet Mask in(voorbeeld, 255.255.255.224):
255.255.255.224

Resultaten:
Bereken het netwerk segment
IP adres (in binair): 11000000.10101000.00000001.01100100
Subnet Mask (in binair): 11111111.11111111.11111111.11100000
-----
Netwerk adres (in binair): 11000000.10101000.00000001.01100000
Dit geeft 192.168.1.96 in decimalen voor het netwerk adres.
```

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