Template Week 6 – Networking

Student number:

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 (2⁵).
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.

```
package part2;

import nl.saxion.app.SaxionApp;
```

```
import java.util.ArrayList;

public class Application2 implements Runnable {
    public static void main(String[] args) {
        SaxionApp.start(new Application2(), 500, 500);
    }

    public void run() {
        SaxionApp.printLine("Enter IP address (e.g., 192.168.1.100");
        SaxionApp.printLine("Enter subnet mask (e.g., 255.255.255.224");
        String ipAddress = SaxionApp.readString();
        String subnetMask = SaxionApp.readString();

        int[] ip = convertToBinaryArray(ipAddress);
        int[] subnet = convertToBinaryArray(subnetMask);

        if (ip == null || subnet == null) {
            SaxionApp.printLine("Invalid input. Please ensure IP address and subnet mask are in the
correct format.");
            return;
        }

        int[] network = calculateNetworkAddress(ip, subnet);

        SaxionApp.printLine("\nResults:");
        SaxionApp.printLine("IP Address:   " + formatBinaryArray(ip));
        SaxionApp.printLine("Subnet Mask:  " + formatBinaryArray(subnet));
        SaxionApp.printLine("Network Addr: " + formatBinaryArray(network));

        String networkAddressDecimal = convertToDecimal(network);
        SaxionApp.printLine("Network Address in Decimal: " + networkAddressDecimal);

        calculateAndDisplayRange(network, subnet);
    }

    private static int[] convertToBinaryArray(String dottedDecimal) {
        String[] parts = dottedDecimal.split("\\.");
        if (parts.length != 4) return null;

        int[] binaryArray = new int[32];
        for (int i = 0; i < 4; i++) {
            int octet;
            try {
                octet = Integer.parseInt(parts[i]);
            } catch (NumberFormatException e) {
                return null;
            }
```

```
            if (octet < 0 || octet > 255) return null;

            for (int j = 7; j >= 0; j--) {
                binaryArray[i * 8 + j] = (octet & 1);
                octet >>= 1;
            }
        }
        return binaryArray;
    }

    private static int[] calculateNetworkAddress(int[] ip, int[] subnet) {
        int[] network = new int[32];
        for (int i = 0; i < 32; i++) {
            network[i] = ip[i] & subnet[i];
        }
        return network;
    }

    private static String formatBinaryArray(int[] binaryArray) {
        StringBuilder formatted = new StringBuilder();
        for (int i = 0; i < binaryArray.length; i++) {
            formatted.append(binaryArray[i]);
            if ((i + 1) % 8 == 0 && i != binaryArray.length - 1) {
                formatted.append(".");
            }
        }
        return formatted.toString();
    }

    private static String convertToDecimal(int[] binaryArray) {
        StringBuilder decimal = new StringBuilder();
        for (int i = 0; i < 4; i++) {
            int value = 0;
            for (int j = 0; j < 8; j++) {
                value = (value << 1) | binaryArray[i * 8 + j];
            }
            decimal.append(value);
            if (i != 3) {
                decimal.append(".");
            }
        }
        return decimal.toString();
    }

    private static void calculateAndDisplayRange(int[] network, int[] subnet) {
        int hostBits = 0;
        for (int bit : subnet) {
            if (bit == 0) hostBits++;
```

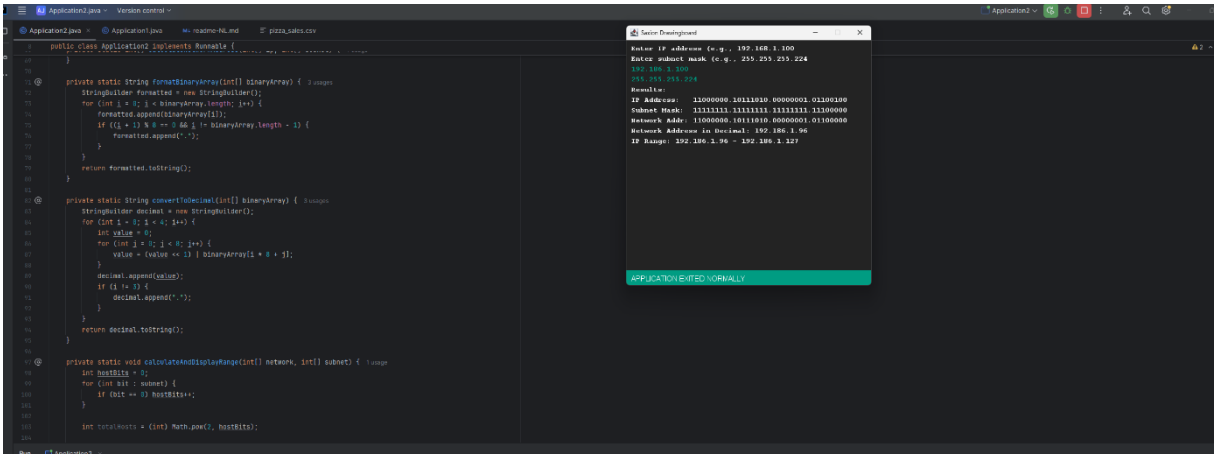
```
        }

        int totalHosts = (int) Math.pow(2, hostBits);

        int[] broadcast = network.clone();
        for (int i = 31; i >= 32 - hostBits; i--) {
            broadcast[i] = 1;
        }

        String networkAddress = convertToDecimal(network);
        String broadcastAddress = convertToDecimal(broadcast);

        SaxionApp.printLine("IP Range: " + networkAddress + " - " + broadcastAddress);
    }
}
```



Ready? Save this file and export it as a pdf file with the name: **week6.pdf**