Template Week 6 – Networking

Student number: 544483

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

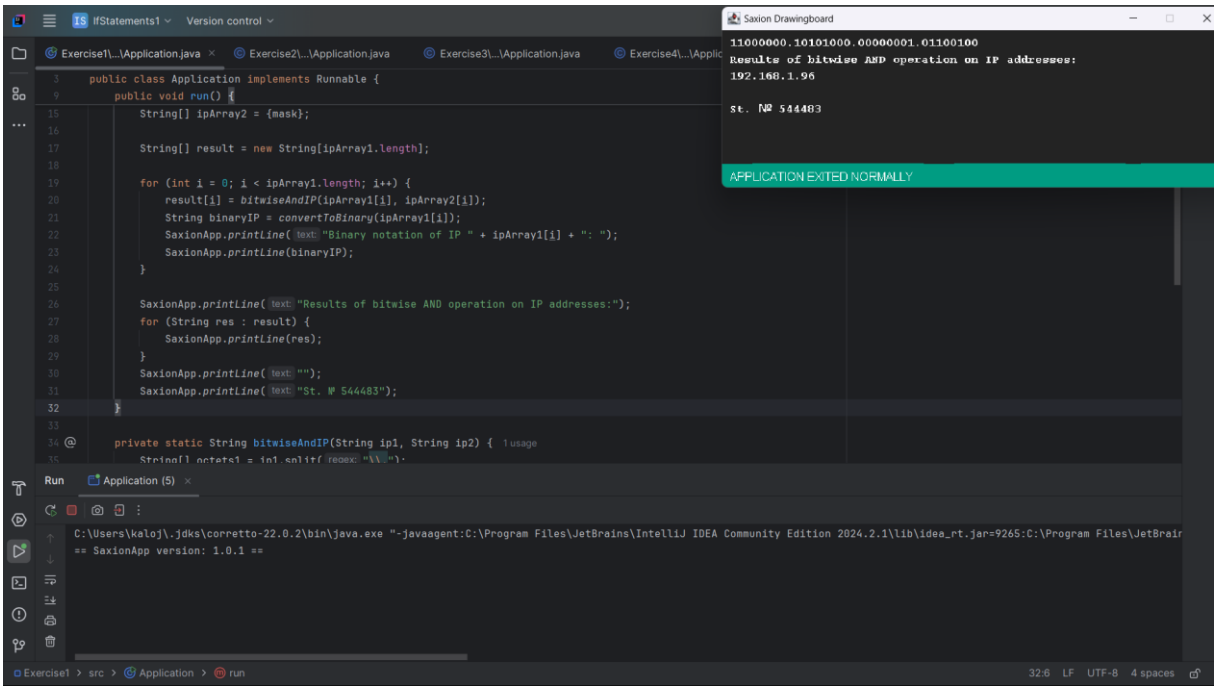
```
Example: 192.168.1.100/27
Calculate the network segment
IP Address:   11000000.10101000.00000001.01100100
Subnet Mask:  11111111.11111111.11111111.11100000
-----------------------------------------------
Network Addr: 11000000.10101000.00000001.01100000

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.

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

public class Application implements Runnable {
```

```
public static void main(String[] args) {
    SaxionApp.start(new Application(), 640, 200);
}

public void run() {
    SaxionApp.print("Please enter an IP address: ");
    String ip = SaxionApp.readString();
    SaxionApp.print("Please enter a Subnet Mask: ");
    String mask = SaxionApp.readString();
    String[] ipArray1 = {ip};
    String[] ipArray2 = {mask};

    String[] result = new String[ipArray1.length];

    for (int i = 0; i < ipArray1.length; i++) {
        result[i] = bitwiseAndIP(ipArray1[i], ipArray2[i]);
        String binaryIP = convertToBinary(ipArray1[i]);
        SaxionApp.printLine("Binary notation of IP " + ipArray1[i] + ": ");
        SaxionApp.printLine(binaryIP);
    }

    SaxionApp.printLine("Results of bitwise AND operation on IP addresses:");
    for (String res : result) {
        SaxionApp.printLine(res);
    }
    SaxionApp.printLine("");
    SaxionApp.printLine("St. № 544483");
}

private static String bitwiseAndIP(String ip1, String ip2) {
```

```
        String[] octets1 = ip1.split("\\.");
        String[] octets2 = ip2.split("\\.");

        int[] resultOctets = new int[4];
        for (int i = 0; i < 4; i++) {
            resultOctets[i] = Integer.parseInt(octets1[i]) & Integer.parseInt(octets2[i]);
        }

        return resultOctets[0] + "." + resultOctets[1] + "." + resultOctets[2] + "." + resultOctets[3];
    }

    private static String convertToBinary(String ip) {
        StringBuilder binaryIP = new StringBuilder();
        String[] octets = ip.split("\\.");

        for (int i = 0; i < octets.length; i++) {
            int octet = Integer.parseInt(octets[i]);
            String binaryOctet = String.format("%08d", Integer.parseInt(Integer.toBinaryString(octet)));
            binaryIP.append(binaryOctet);
            if (i < octets.length - 1) {
                binaryIP.append(".");
            }
        }

        return binaryIP.toString();
    }
}
```



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