Template Week 6 – Networking

Student number: 572750

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.
(This is the source code with the code from week 2 in there)

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

import java.util.ArrayList;

public class Main implements Runnable {
```

```
    public static void main(String[] args) {
        SaxionApp.start(new Main(), 1100, 550);
    }

    public void printmenu() {
        SaxionApp.printLine("1. Even or odd?");
        SaxionApp.printLine("2. Power of 2?");
        SaxionApp.printLine("3. 2's compliment of number");
        SaxionApp.printLine("4. Binary network address");
        SaxionApp.print("Choice: ");

    }
    public void evenodd() {
        SaxionApp.print("Pick a number: ");
        int number = SaxionApp.readInt();
        int result = (number & 1);
        if(result == 1) SaxionApp.printLine(number + " is odd");
        else SaxionApp.printLine(number + " is even");
    }
    public void power2() {
        SaxionApp.print("Pick a number: ");
        int number = SaxionApp.readInt();
        int result = (number&(number-1));
        if(result == 0) SaxionApp.printLine(number + " is a power of 2");
        else SaxionApp.printLine(number + " isn't a power of 2");
    }
    public void twos() {
        SaxionApp.print("Pick a number: ");
        int number = SaxionApp.readInt();
        int result = ~number + 1;
        SaxionApp.printLine("Number: "+result);
    }
    public void networkSegment() {
        SaxionApp.print("What is the IP address you want to use?: ");
        String IPaddr = SaxionApp.readString();
        SaxionApp.print("What is the subnet you want to use?: ");
        String Subaddr = SaxionApp.readString();

        String[] ipOctets = IPaddr.split("\\.");
        String[] subOctets = Subaddr.split("\\.");

        int networkRange = calculateIPRange(Subaddr);
        int[] networkStart = new int[4];
        int[] networkEnd = new int[4];

        for (int i = 0; i < 4; i++) {
            int ip = Integer.parseInt(ipOctets[i]);
            int sub = Integer.parseInt(subOctets[i]);
            networkStart[i] = ip & sub;
            networkEnd[i] = networkStart[i];
        }

        int remainingIPs = networkRange - 1;
        for (int i = 3; i >= 0; i--) {
            networkEnd[i] += remainingIPs;
            if (networkEnd[i] > 255) {
                remainingIPs = networkEnd[i] / 256;
                networkEnd[i] %= 256;
            } else {
                remainingIPs = 0;
```

```
            }
        }

        String networkAddressStart = String.join(".",
                String.valueOf(networkStart[0]),
                String.valueOf(networkStart[1]),
                String.valueOf(networkStart[2]),
                String.valueOf(networkStart[3])
        );
        String networkAddressEnd = String.join(".",
                String.valueOf(networkEnd[0]),
                String.valueOf(networkEnd[1]),
                String.valueOf(networkEnd[2]),
                String.valueOf(networkEnd[3])
        );

        SaxionApp.printLine("Network Address: " + networkAddressStart);
        SaxionApp.printLine("Network Segment: " + networkAddressStart + "
to " + networkAddressEnd);
    }

    public int calculateIPRange(String subnetMask) {
        String[] subnetParts = subnetMask.split("\\.");
        int prefixLength = 0;

        for (String part : subnetParts) {
            int value = Integer.parseInt(part);
            while (value > 0) {
                if ((value & 1) == 1) {
                    prefixLength++;
                }
                value >>= 1;
            }
        }

        return (int) Math.pow(2, (32 - prefixLength));
    }

    public void run() {
        printmenu();
        int choice = SaxionApp.readInt();
        //SaxionApp.clear();
        SaxionApp.printLine();
        if (choice == 1) {
            evenodd();
        }
        else if (choice == 2) {
            power2();
        }
        else if (choice == 3) {
            twos();
        }
        else if (choice == 4) {
            networkSegment();
        }
        else {
            SaxionApp.printLine("This is not an option");
        }

    }
}
```

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