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

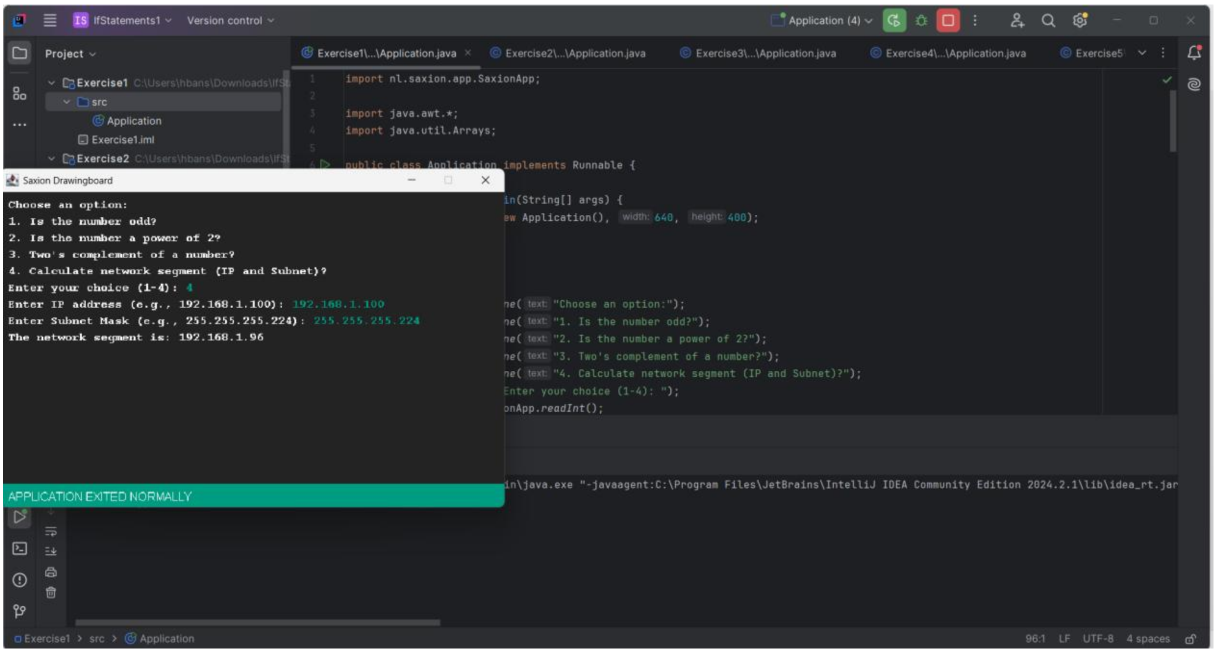
Remember that bitwise java application you've made in week 2? Expand that application so that you can also calculate a network segment as explained in the PowerPoint slides of week 6. Use the bitwise & AND operator. You need to be able to input two Strings. An IP address and a subnet.

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 (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.


IfStatements1
Version control
Project
Exercise1 C:\Users\hbans\Downloads\IfSt
src
Application
Exercise1.iml
Exercise2 C:\Users\hbans\Downloads\IfSt
Application (4)
Exercise1\...\Application.java
Exercise2\...\Application.java
Exercise3\...\Application.java
Exercise4\...\Application.java
Exercise5
Saxion Drawingboard
Exercise1 > src > Application
96:1 LF UTF-8 4 spaces

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

import java.awt.*;
import java.util.Arrays;

public class Application implements Runnable {
```

```
Choose an option:
1. Is the number odd?
2. Is the number a power of 2?
3. Two's complement of a number?
4. Calculate network segment (IP and Subnet)?
Enter your choice (1-4): 4
Enter IP address (e.g., 192.168.1.100): 192.168.1.100
Enter Subnet Mask (e.g., 255.255.255.224): 255.255.255.224
The network segment is: 192.168.1.96
```

APPLICATION EXITED NORMALLY

```
in\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.2.1\lib\idea_rt.jar
```

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

import java.awt.*;
import java.util.Arrays;

public class Application implements Runnable {

    public static void main(String[] args) {
        SaxionApp.start(new Application(), 640, 400);
    }

    public void run() {
        SaxionApp.printLine("Choose an option:");
        SaxionApp.printLine("1. Is the number odd?");
        SaxionApp.printLine("2. Is the number a power of 2?");
        SaxionApp.printLine("3. Two's complement of a number?");
        SaxionApp.printLine("4. Calculate network segment (IP and Subnet)?");
        SaxionApp.print("Enter your choice (1-4): ");
        int choice = SaxionApp.readInt();

        switch (choice) {
            case 1 -> {
                SaxionApp.print("Enter a number: ");
                int number = SaxionApp.readInt();
                boolean isOdd = isOdd(number);
                SaxionApp.printLine("The number " + number + " is " + (isOdd ? "odd." : "even."));
            }
            case 2 -> {
                SaxionApp.print("Enter a number: ");
                int number = SaxionApp.readInt();
                boolean isPowerOfTwo = isPowerOfTwo(number);
                SaxionApp.printLine("The number " + number + " is " + (isPowerOfTwo ? "" : "not ") + "a
power of 2.");
            }
            case 3 -> {
                SaxionApp.print("Enter a number: ");
                int number = SaxionApp.readInt();
                int twosComplement = getTwosComplement(number);
                SaxionApp.printLine("The two's complement of " + number + " is " + twosComplement + ".");
            }
            case 4 -> {
                SaxionApp.print("Enter IP address (e.g., 192.168.1.100): ");
                String ipAddress = SaxionApp.readString();
                SaxionApp.print("Enter Subnet Mask (e.g., 255.255.255.224): ");
                String subnetMask = SaxionApp.readString();

                String networkAddress = calculateNetworkSegment(ipAddress, subnetMask);
                SaxionApp.printLine("The network segment is: " + networkAddress);
            }
            default -> SaxionApp.printLine("Invalid choice!", Color.RED);
        }
    }

    private boolean isOdd(int number) {
        return (number & 1) == 1;
    }

    private boolean isPowerOfTwo(int number) {
        return number > 0 && (number & (number - 1)) == 0;
    }

    private int getTwosComplement(int number) {
        return ~number + 1;
    }

    private String calculateNetworkSegment(String ipAddress, String subnetMask) {
        int[] ip = parseIpAddress(ipAddress);
        int[] subnet = parseIpAddress(subnetMask);
        int[] networkAddress = new int[4];
        for (int i = 0; i < 4; i++) {
            networkAddress[i] = ip[i] & subnet[i];
        }
        return String.format("%d.%d.%d.%d", networkAddress[0], networkAddress[1],
networkAddress[2], networkAddress[3]);
    }

    private int[] parseIpAddress(String address) {
        return Arrays.stream(address.split("\\."))
                .mapToInt(Integer::parseInt)
                .toArray();
    }
}
```

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