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

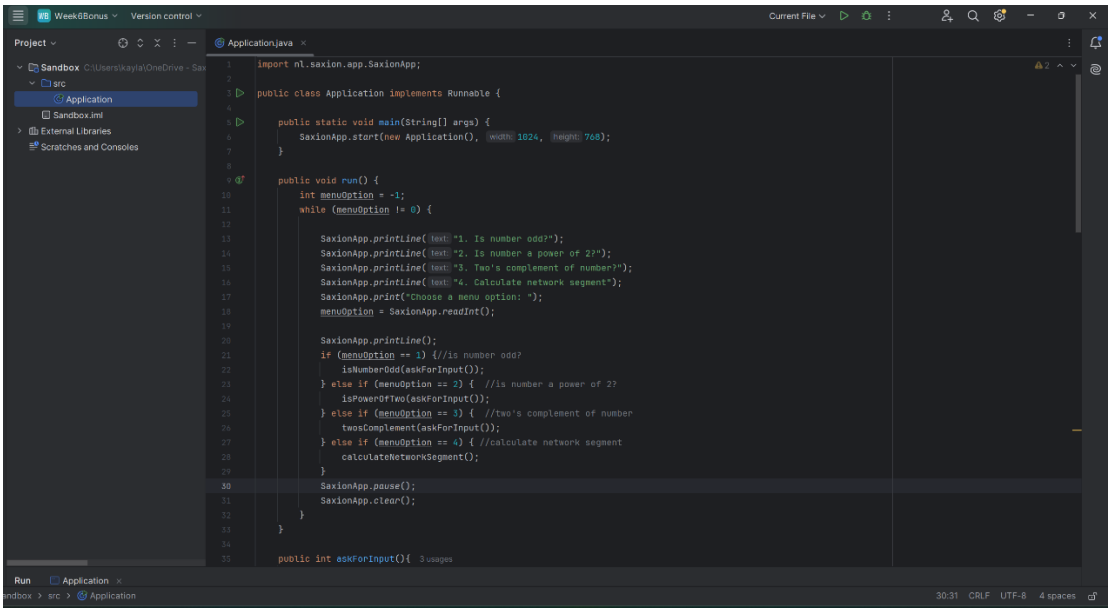
Bonus point assignment – week 6

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.

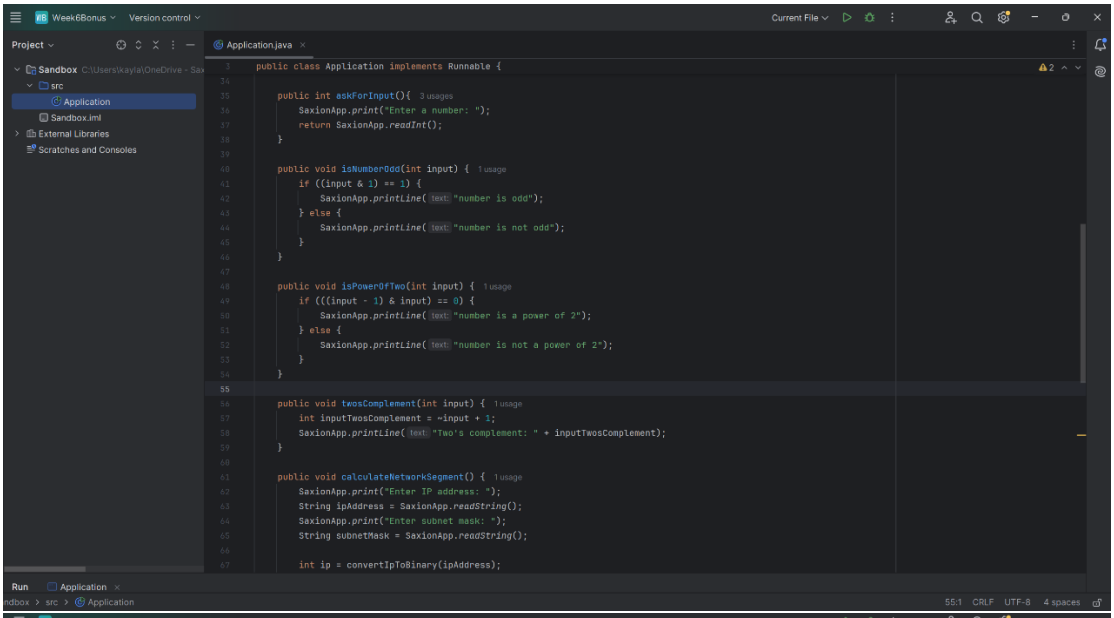
IP: 192.168.1.100 and subnet: 255.255.255.224 for /27

```
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.
```

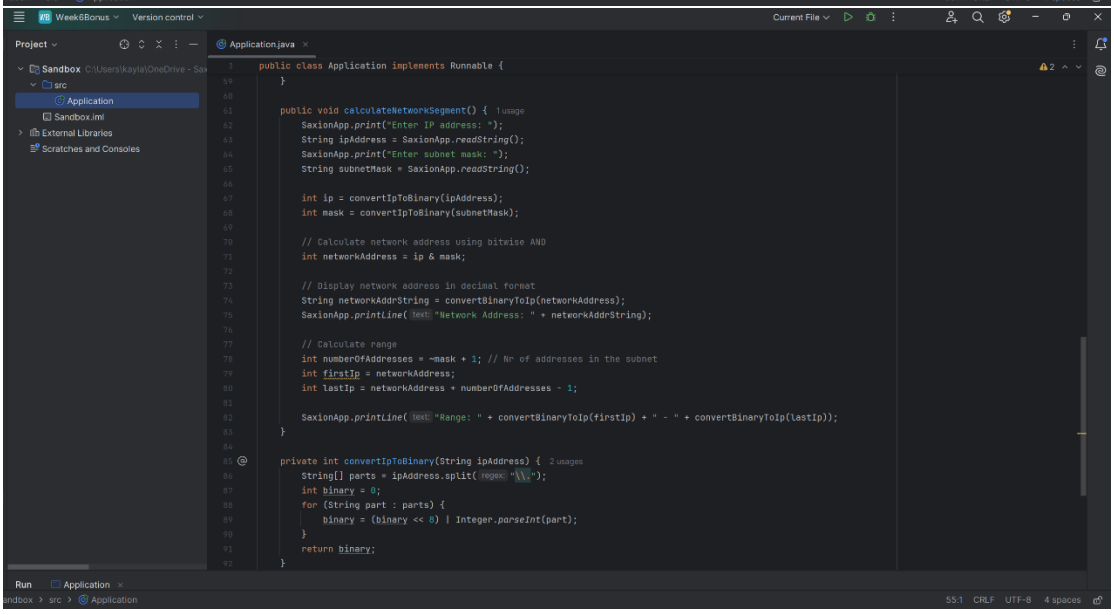
Paste source code here, with a screenshot of a working application.



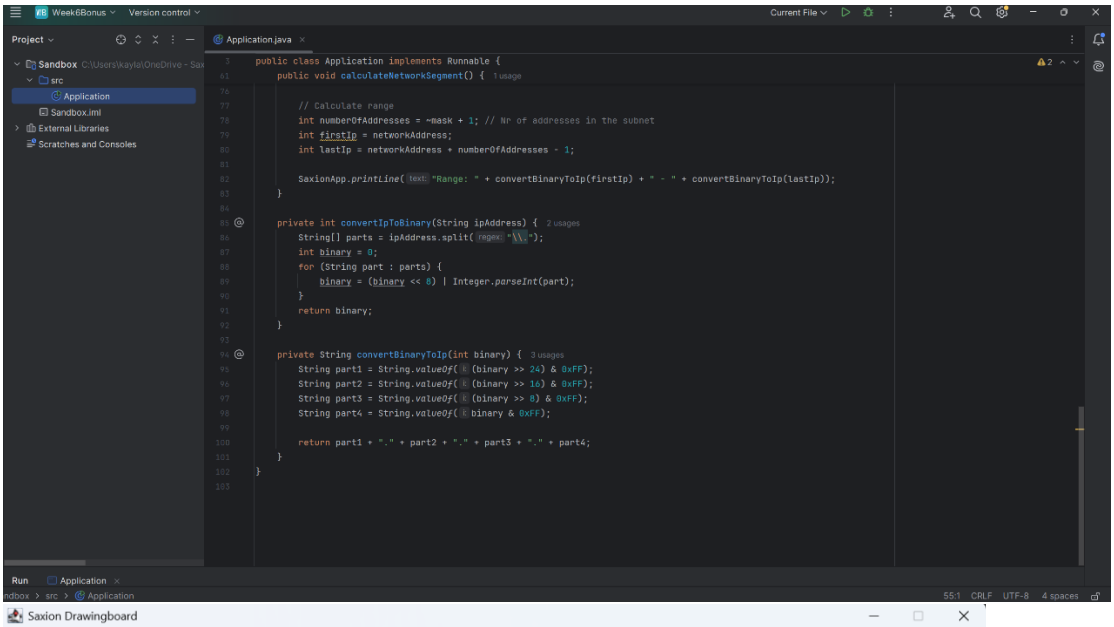
```
import nl.saxion.app.SaxionApp;

public class Application implements Runnable {

    public static void main(String[] args) {
        SaxionApp.start(new Application(), width: 1024, height: 768);
    }

    public void run() {
        int menuOption = -1;
        while (menuOption != 0) {

            SaxionApp.printLine( text: "1. Is number odd?");
            SaxionApp.printLine( text: "2. Is number a power of 2?");
            SaxionApp.printLine( text: "3. Two's complement of number?");
            SaxionApp.printLine( text: "4. Calculate network segment");
            SaxionApp.print("Choose a menu option: ");
            menuOption = SaxionApp.readInt();

            SaxionApp.printLine();
            if (menuOption == 1) {//is number odd?
                isNumberOdd(askForInput());
            } else if (menuOption == 2) {  //is number a power of 2?
                isPowerOfTwo(askForInput());
            } else if (menuOption == 3) {  //two's complement of number
                twosComplement(askForInput());
            } else if (menuOption == 4) { //calculate network segment
                calculateNetworkSegment();
            }
            SaxionApp.pause();
            SaxionApp.clear();
        }
    }

    public int askForInput(){  3 usages
```

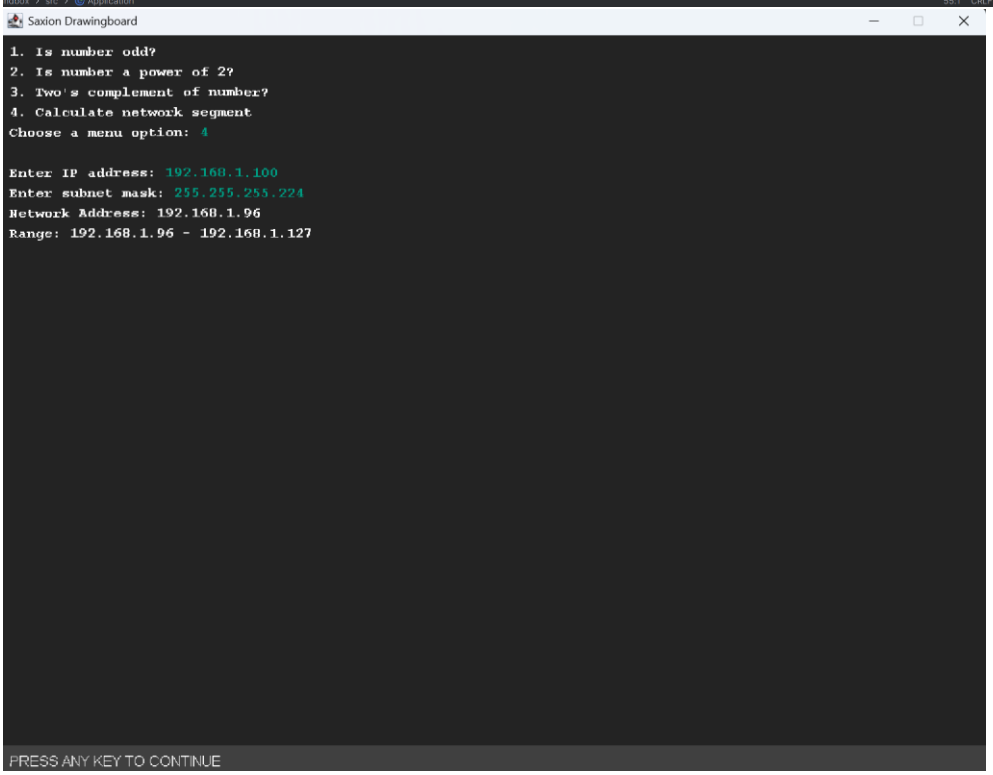


```
public class Application implements Runnable {

    public int askForInput(){  3 usages
        SaxionApp.print("Enter a number: ");
        return SaxionApp.readInt();
    }

    public void isNumberOdd(int input) {  1 usage
        if ((input & 1) == 1) {
            SaxionApp.printLine( text: "number is odd");
        } else {
            SaxionApp.printLine( text: "number is not odd");
        }
    }

    public void isPowerOfTwo(int input) {  1 usage
        if (((input - 1) & input) == 0) {
            SaxionApp.printLine( text: "number is a power of 2");
        } else {
            SaxionApp.printLine( text: "number is not a power of 2");
        }
    }

    public void twosComplement(int input) {  1 usage
        int inputTwosComplement = ~input + 1;
        SaxionApp.printLine( text: "Two's complement: " + inputTwosComplement);
    }

    public void calculateNetworkSegment() {  1 usage
        SaxionApp.print("Enter IP address: ");
        String ipAddress = SaxionApp.readString();
        SaxionApp.print("Enter subnet mask: ");
        String subnetMask = SaxionApp.readString();

        int ip = convertIpToBinary(ipAddress);
```



```
}

    public void calculateNetworkSegment() {  1 usage
        SaxionApp.print("Enter IP address: ");
        String ipAddress = SaxionApp.readString();
        SaxionApp.print("Enter subnet mask: ");
        String subnetMask = SaxionApp.readString();

        int ip = convertIpToBinary(ipAddress);
        int mask = convertIpToBinary(subnetMask);

        // Calculate network address using bitwise AND
        int networkAddress = ip & mask;

        // Display network address in decimal format
        String networkAddrString = convertBinaryToIp(networkAddress);
        SaxionApp.printLine( text: "Network Address: " + networkAddrString);

        // Calculate range
        int numberOfAddresses = ~mask + 1; // Nr of addresses in the subnet
        int firstIp = networkAddress;
        int lastIp = networkAddress + numberOfAddresses - 1;

        SaxionApp.printLine( text: "Range: " + convertBinaryToIp(firstIp) + " - " + convertBinaryToIp(lastIp));
    }

    private int convertIpToBinary(String ipAddress) {  2 usages
        String[] parts = ipAddress.split( regex: "\\.");
        int binary = 0;
        for (String part : parts) {
            binary = (binary << 8) | Integer.parseInt(part);
        }
        return binary;
    }
```





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