Template Week 6 – Networking

Student number: 568670

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

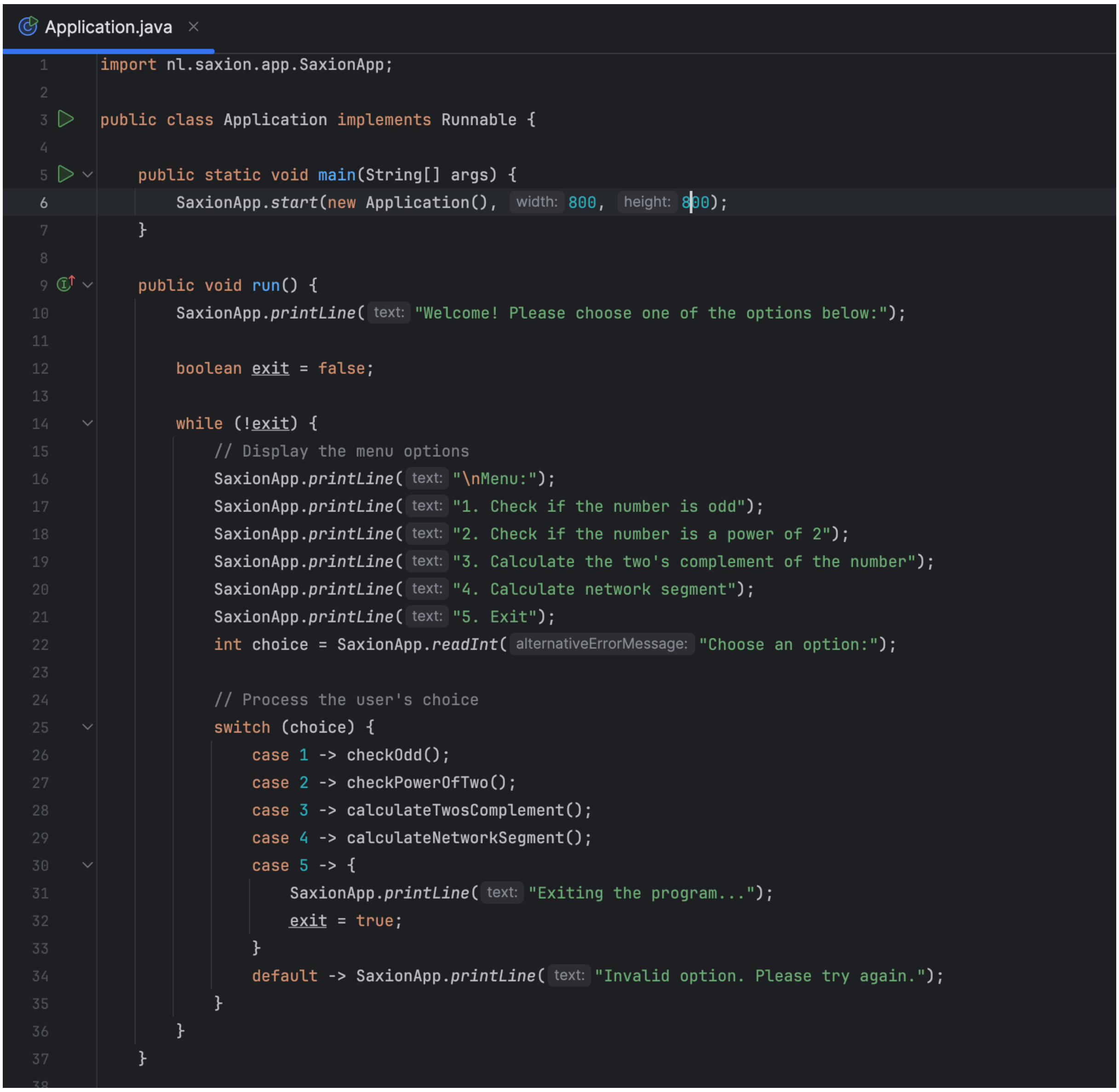
Bonus point assignment – week 6

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.

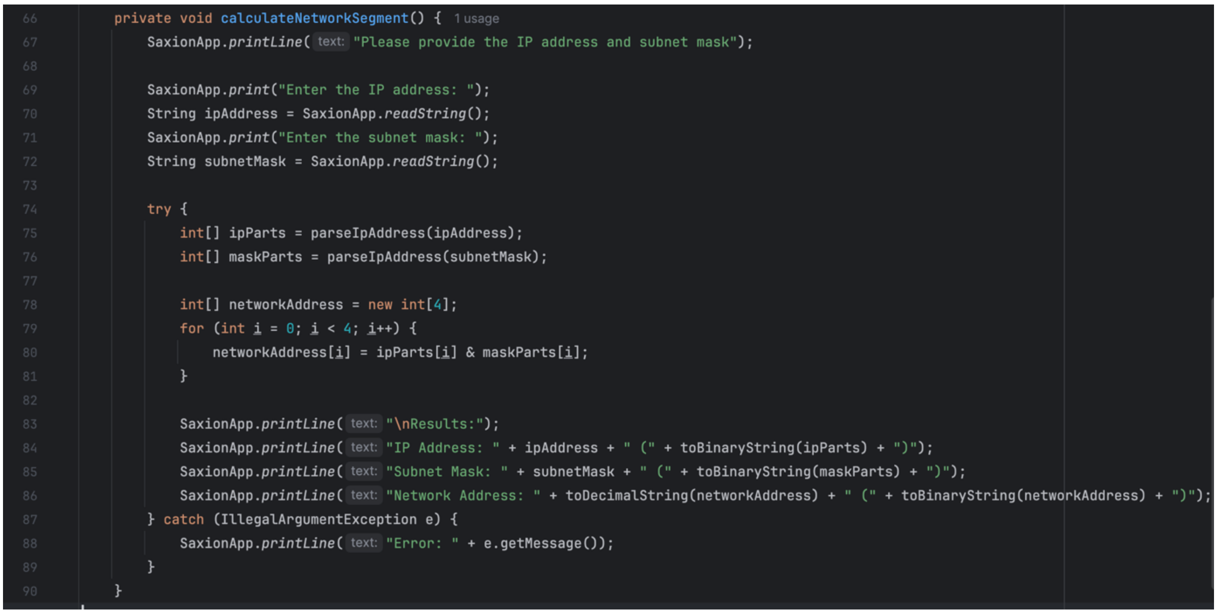
IP: 192.168.1.100 and subnet: 255.255.255.224 for /27

```
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.
```

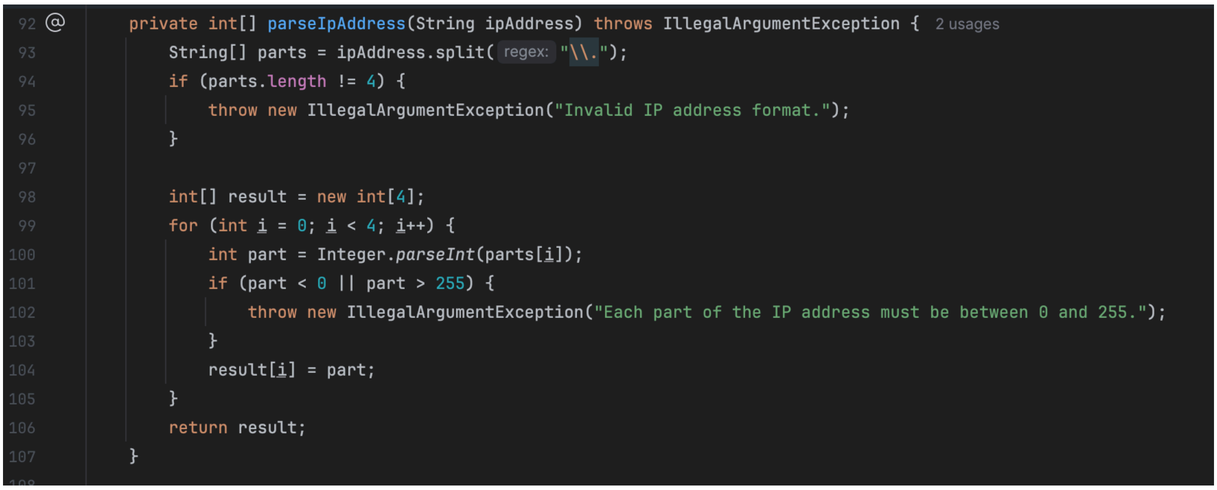
Paste source code here, with a screenshot of a working application.



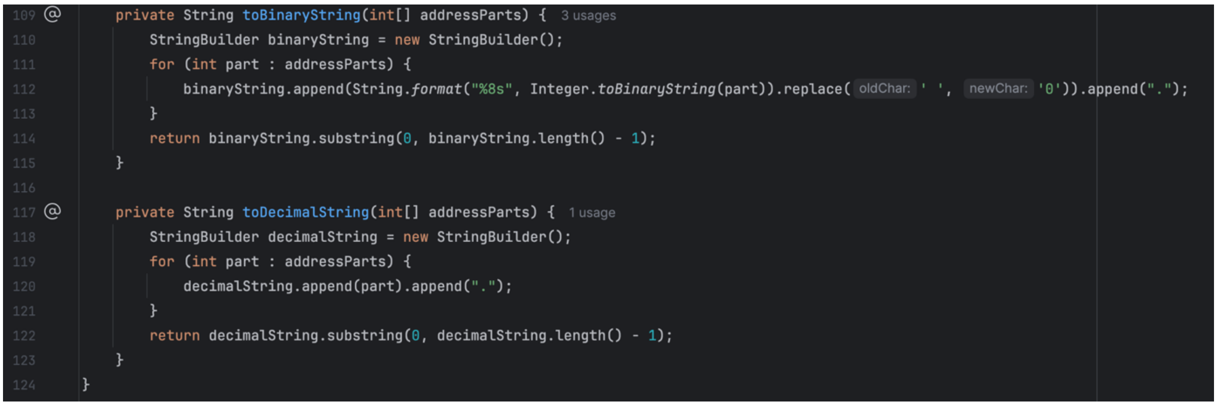
```java
import nl.saxion.app.SaxionApp;

public class Application implements Runnable {

    public static void main(String[] args) {
        SaxionApp.start(new Application(),  width: 800,  height: 800);
    }

    public void run() {
        SaxionApp.printLine( text: "Welcome! Please choose one of the options below:");

        boolean exit = false;

        while (!exit) {
            // Display the menu options
            SaxionApp.printLine( text: "\nMenu:");
            SaxionApp.printLine( text: "1. Check if the number is odd");
            SaxionApp.printLine( text: "2. Check if the number is a power of 2");
            SaxionApp.printLine( text: "3. Calculate the two's complement of the number");
            SaxionApp.printLine( text: "4. Calculate network segment");
            SaxionApp.printLine( text: "5. Exit");
            int choice = SaxionApp.readInt( alternativeErrorMessage: "Choose an option:");

            // Process the user's choice
            switch (choice) {
                case 1 -> checkOdd();
                case 2 -> checkPowerOfTwo();
                case 3 -> calculateTwosComplement();
                case 4 -> calculateNetworkSegment();
                case 5 -> {
                    SaxionApp.printLine( text: "Exiting the program...");
                    exit = true;
                }
                default -> SaxionApp.printLine( text: "Invalid option. Please try again.");
            }
        }
    }
```



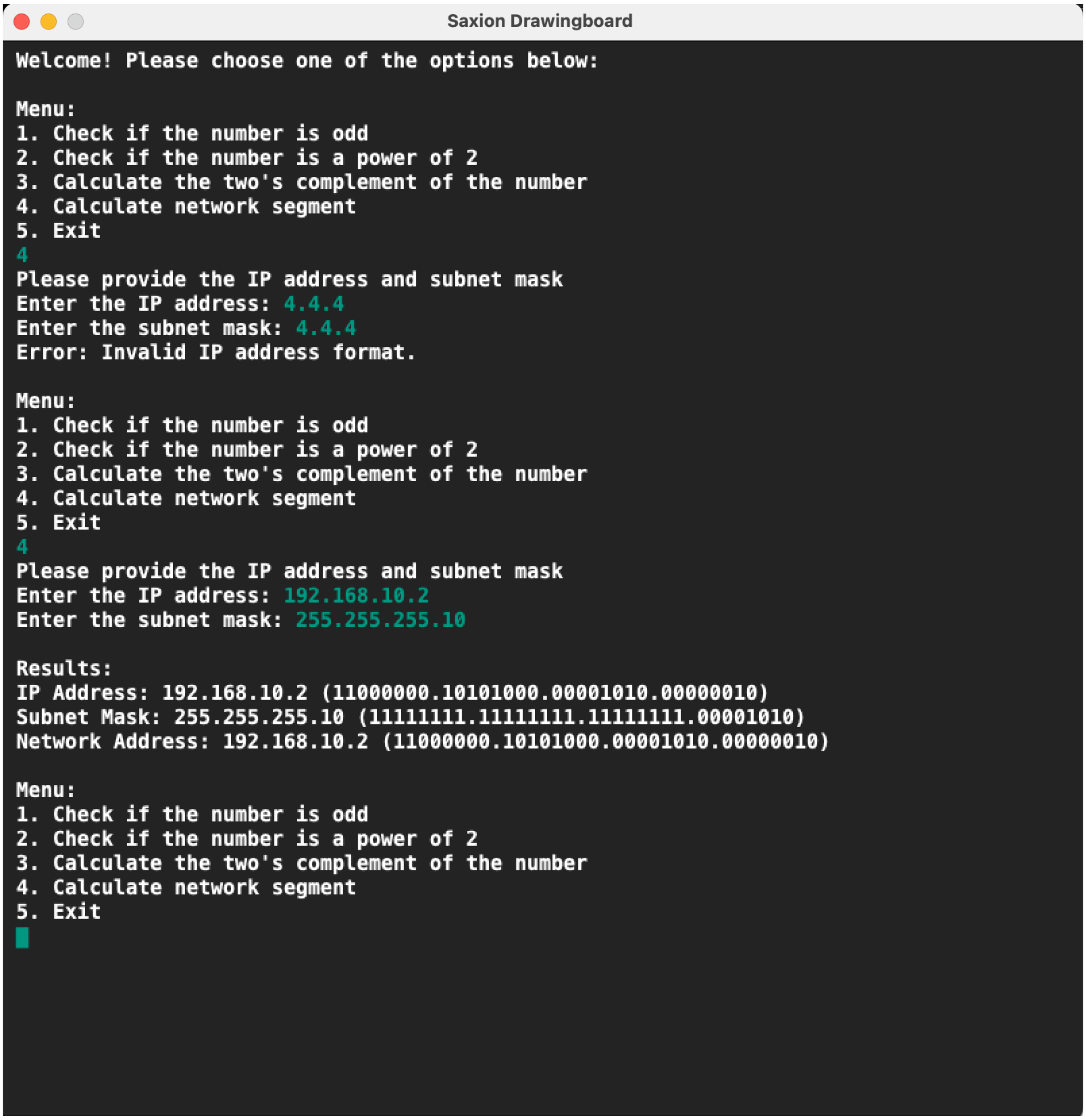
```java
    private void calculateNetworkSegment() {  1 usage
        SaxionApp.printLine( text: "Please provide the IP address and subnet mask");

        SaxionApp.print("Enter the IP address: ");
        String ipAddress = SaxionApp.readString();
        SaxionApp.print("Enter the subnet mask: ");
        String subnetMask = SaxionApp.readString();

        try {
            int[] ipParts = parseIpAddress(ipAddress);
            int[] maskParts = parseIpAddress(subnetMask);

            int[] networkAddress = new int[4];
            for (int i = 0; i < 4; i++) {
                networkAddress[i] = ipParts[i] & maskParts[i];
            }

            SaxionApp.printLine( text: "\nResults:");
            SaxionApp.printLine( text: "IP Address: " + ipAddress + " (" + toBinaryString(ipParts) + ")");
            SaxionApp.printLine( text: "Subnet Mask: " + subnetMask + " (" + toBinaryString(maskParts) + ")");
            SaxionApp.printLine( text: "Network Address: " + toDecimalString(networkAddress) + " (" + toBinaryString(networkAddress) + ")");
        } catch (IllegalArgumentException e) {
            SaxionApp.printLine( text: "Error: " + e.getMessage());
        }
    }
```



```java
    private int[] parseIpAddress(String ipAddress) throws IllegalArgumentException {  2 usages
        String[] parts = ipAddress.split( regex: "\\.");
        if (parts.length != 4) {
            throw new IllegalArgumentException("Invalid IP address format.");
        }

        int[] result = new int[4];
        for (int i = 0; i < 4; i++) {
            int part = Integer.parseInt(parts[i]);
            if (part < 0 || part > 255) {
                throw new IllegalArgumentException("Each part of the IP address must be between 0 and 255.");
            }
            result[i] = part;
        }
        return result;
    }
```



```java
    private String toBinaryString(int[] addressParts) {  3 usages
        StringBuilder binaryString = new StringBuilder();
        for (int part : addressParts) {
            binaryString.append(String.format("%8s", Integer.toBinaryString(part)).replace( oldChar: ' ',  newChar: '0')).append(".");
        }
        return binaryString.substring(0, binaryString.length() - 1);
    }

    private String toDecimalString(int[] addressParts) {  1 usage
        StringBuilder decimalString = new StringBuilder();
        for (int part : addressParts) {
            decimalString.append(part).append(".");
        }
        return decimalString.substring(0, decimalString.length() - 1);
    }
}
```



```
Welcome! Please choose one of the options below:

Menu:
1. Check if the number is odd
2. Check if the number is a power of 2
3. Calculate the two's complement of the number
4. Calculate network segment
5. Exit
4
Please provide the IP address and subnet mask
Enter the IP address: 4.4.4
Enter the subnet mask: 4.4.4
Error: Invalid IP address format.

Menu:
1. Check if the number is odd
2. Check if the number is a power of 2
3. Calculate the two's complement of the number
4. Calculate network segment
5. Exit
4
Please provide the IP address and subnet mask
Enter the IP address: 192.168.10.2
Enter the subnet mask: 255.255.255.10

Results:
IP Address: 192.168.10.2 (11000000.10101000.00001010.00000010)
Subnet Mask: 255.255.255.10 (11111111.11111111.11111111.00001010)
Network Address: 192.168.10.2 (11000000.10101000.00001010.00000010)

Menu:
1. Check if the number is odd
2. Check if the number is a power of 2
3. Calculate the two's complement of the number
4. Calculate network segment
5. Exit
```

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