Template Week 6 – Networking

Student number:570350

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

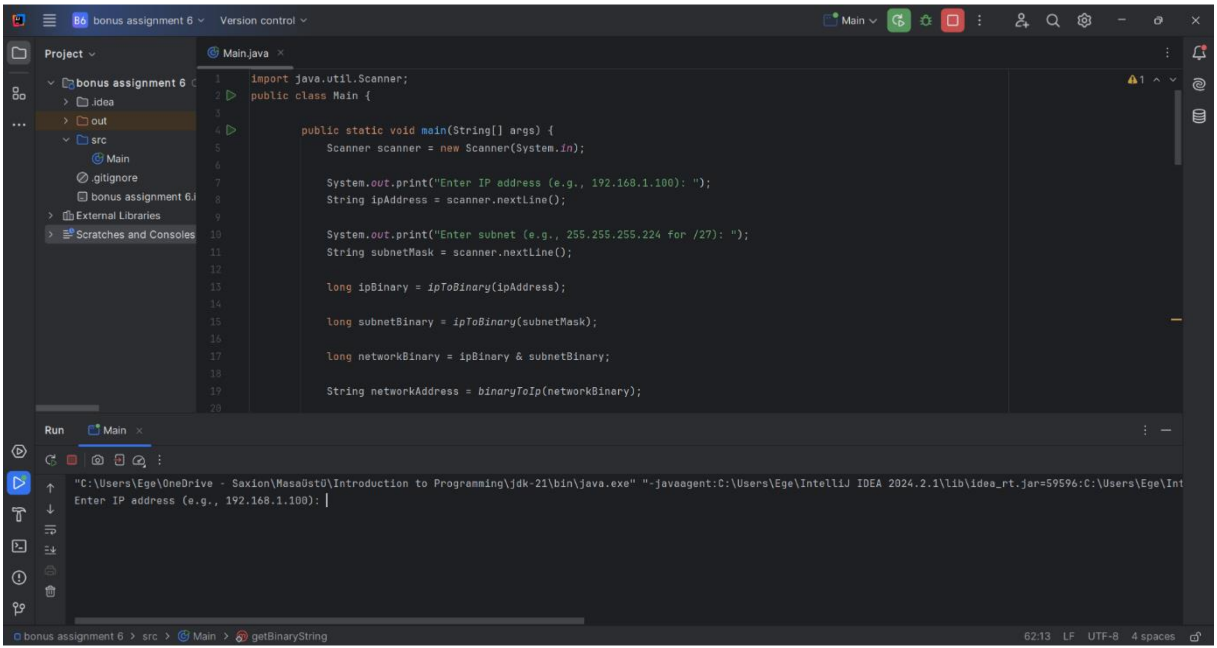
Remember that bitwise java application you've made in week 2? Expand that application so that you can also calculate a network segment as explained in the PowerPoint slides of week 6. Use the bitwise & AND operator. You need to be able to input two Strings. An IP address and a subnet.

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.
```

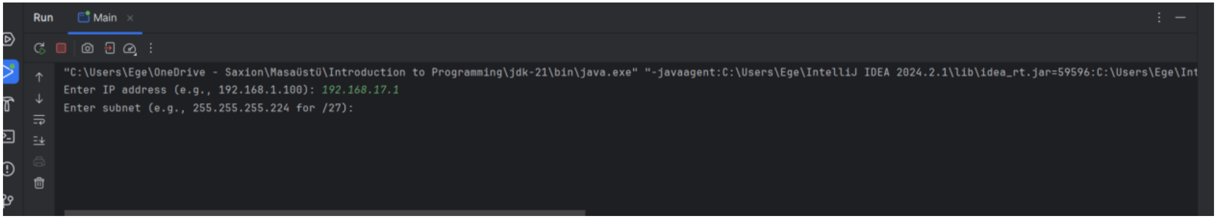
Paste source code here, with a screenshot of a working application.

```
import java.util.Scanner;
public class Main {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter IP address (e.g., 192.168.1.100): ");
        String ipAddress = scanner.nextLine();

        System.out.print("Enter subnet (e.g., 255.255.255.224 for /27): ");
        String subnetMask = scanner.nextLine();

        long ipBinary = ipToBinary(ipAddress);

        long subnetBinary = ipToBinary(subnetMask);

        long networkBinary = ipBinary & subnetBinary;

        String networkAddress = binaryToIp(networkBinary);

        System.out.println("\nIP Address:     " + getBinaryString(ipBinary));
        System.out.println("Subnet Mask:    " + getBinaryString(subnetBinary));
        System.out.println("Network Addr:   " + getBinaryString(networkBinary));

        System.out.println("\nNetwork Address in decimal: " + networkAddress);

        int hostBits = 32 - countLeadingOnes(subnetBinary);
        int hosts = (1 << hostBits) - 1;

        long lastAddressBinary = networkBinary | hosts;
        String lastAddress = binaryToIp(lastAddressBinary);

        System.out.println("Address Range: " + networkAddress + " - " + lastAddress);
        System.out.println("Number of usable hosts: " + (hosts - 1));

        scanner.close();
    }

    private static long ipToBinary(String ip) {
        String[] parts = ip.split("\\.");
        long result = 0;
        for (String part : parts) {
            result = (result << 8) | Integer.parseInt(part);
        }
        return result;
    }

    private static String binaryToIp(long binary) {
        return String.format("%d.%d.%d.%d",
                (binary >> 24) & 0xFF,
                (binary >> 16) & 0xFF,
                (binary >> 8) & 0xFF,
                binary & 0xFF);
    }

    private static String getBinaryString(long number) {
        String binary = Long.toBinaryString(number);

        while (binary.length() < 32) {
            binary = "0" + binary;
        }

        return binary.replaceAll("(.{8})", "$1.");
    }

    private static int countLeadingOnes(long number) {
        int count = 0;
        for (int i = 31; i >= 0; i--) {
            if ((number & (1L << i)) != 0) {
                count++;
            } else {
                break;
            }
        }
        return count;
    }
}
```

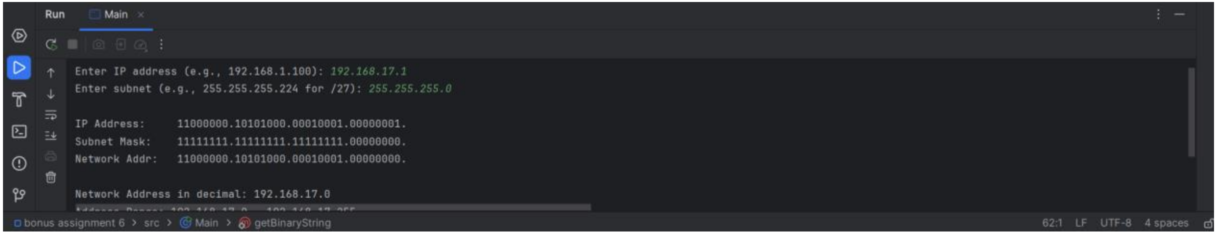


```java
import java.util.Scanner;
public class Main {

        public static void main(String[] args) {
            Scanner scanner = new Scanner(System.in);

            System.out.print("Enter IP address (e.g., 192.168.1.100): ");
            String ipAddress = scanner.nextLine();

            System.out.print("Enter subnet (e.g., 255.255.255.224 for /27): ");
            String subnetMask = scanner.nextLine();

            long ipBinary = ipToBinary(ipAddress);

            long subnetBinary = ipToBinary(subnetMask);

            long networkBinary = ipBinary & subnetBinary;

            String networkAddress = binaryToIp(networkBinary);
```

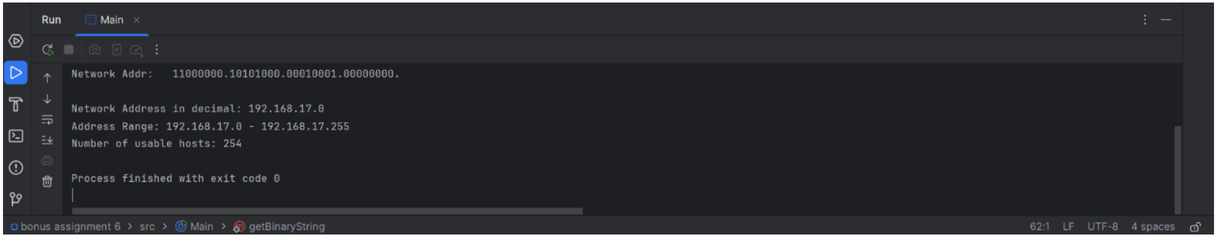
```
"C:\Users\Ege\OneDrive - Saxion\MasaÜstÜ\Introduction to Programming\jdk-21\bin\java.exe" "-javaagent:C:\Users\Ege\IntelliJ IDEA 2024.2.1\lib\idea_rt.jar=59596:C:\Users\Ege\Int
Enter IP address (e.g., 192.168.1.100): 
```



```
"C:\Users\Ege\OneDrive - Saxion\MasaÜstÜ\Introduction to Programming\jdk-21\bin\java.exe" "-javaagent:C:\Users\Ege\IntelliJ IDEA 2024.2.1\lib\idea_rt.jar=59596:C:\Users\Ege\Int
Enter IP address (e.g., 192.168.1.100): 192.168.17.1
Enter subnet (e.g., 255.255.255.224 for /27): 
```



```
Enter IP address (e.g., 192.168.1.100): 192.168.17.1
Enter subnet (e.g., 255.255.255.224 for /27): 255.255.255.0

IP Address:     11000000.10101000.00010001.00000001.
Subnet Mask:    11111111.11111111.11111111.00000000.
Network Addr:   11000000.10101000.00010001.00000000.

Network Address in decimal: 192.168.17.0
```



```
Network Addr:   11000000.10101000.00010001.00000000.

Network Address in decimal: 192.168.17.0
Address Range: 192.168.17.0 - 192.168.17.255
Number of usable hosts: 254

Process finished with exit code 0
```

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