

Template Week 6 – Networking

Student number: 573029 Eren Köngül

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 nl.saxion.app.SaxionApp;

public class Application implements Runnable {

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

    public void run() {
        SaxionApp.println("Eren Köngül 573029");
        SaxionApp.println("1. Even or odd?");
        SaxionApp.println("2. Power of 2?");
        SaxionApp.println("3. Two's complement");
        SaxionApp.println("4. Calculate network segment");
        int enteredvalue = SaxionApp.readInt();

        if (enteredvalue == 1) {
            OddEven();
        } else if (enteredvalue == 2) {
            powerof2();
        } else if (enteredvalue == 3) {
            twoscomplement();
        } else if (enteredvalue == 4) {
```

```

        Network();
    }

}

public void OddEven() {
    SaxionApp.println("Enter number");
    int number = SaxionApp.readInt();

    if (Odd(number)) {
        SaxionApp.println(number + " is odd");
    } else {
        SaxionApp.println(number + " is even");
    }
}

public boolean Odd(int number) {
    return (number & 1) == 1;
}

public void powerof2() {
    SaxionApp.println("Enter number");
    int number = SaxionApp.readInt();

    if (power2(number)) {
        SaxionApp.println(number + " is power of 2");
    } else {
        SaxionApp.println(number + " is not power of 2");
    }
}

public boolean power2(int number) {
    return number > 0 && (number & (number - 1)) == 0;
}

public void twoscomplement() {
    SaxionApp.println("Enter a number");
    int number = SaxionApp.readInt();

    int negativeNumber = toTwosComplement(number);
    SaxionApp.println("Two's complement of " + number + " is: " + negativeNumber + " " +
Integer.toBinaryString(negativeNumber));
}

```

```

        int originalNumber = toTwosComplement(negativeNumber);
        SaxionApp.println("Converting back: " + negativeNumber + " becomes: " + originalNumber + " "
+ Integer.toBinaryString(originalNumber));
    }

```

```

public static int toTwosComplement(int number) {
    return ~number + 1;
}

```

```

} public void Network() {

```

```

    SaxionApp.print("IPoctet1: ");
    int IPoctet1 = SaxionApp.readInt();
    SaxionApp.print("IPoctet2: ");
    int IPoctet2 = SaxionApp.readInt();
    SaxionApp.print("IPoctet3: ");
    int IPoctet3 = SaxionApp.readInt();
    SaxionApp.print("IPoctet4: ");
    int IPoctet4 = SaxionApp.readInt();

```

```

    SaxionApp.print("SMoctet1: ");
    int SMoctet1 = SaxionApp.readInt();
    SaxionApp.print("SMoctet2: ");
    int SMoctet2 = SaxionApp.readInt();
    SaxionApp.print("SMoctet3: ");
    int SMoctet3 = SaxionApp.readInt();
    SaxionApp.print("SMoctet4: ");
    int SMoctet4 = SaxionApp.readInt();

```

```

    int network1 = IPoctet1 & SMoctet1;
    int network2 = IPoctet2 & SMoctet2;
    int network3 = IPoctet3 & SMoctet3;
    int network4 = IPoctet4 & SMoctet4;

```

```

    String IPbinary1 = toBinary(IPoctet1);
    String IPbinary2 = toBinary(IPoctet2);
    String IPbinary3 = toBinary(IPoctet3);
    String IPbinary4 = toBinary(IPoctet4);

```

```

    String SMbinary1 = toBinary(SMoctet1);
    String SMbinary2 = toBinary(SMoctet2);
    String SMbinary3 = toBinary(SMoctet3);
    String SMbinary4 = toBinary(SMoctet4);

```

```

    String networkBinary1 = toBinary(network1);

```

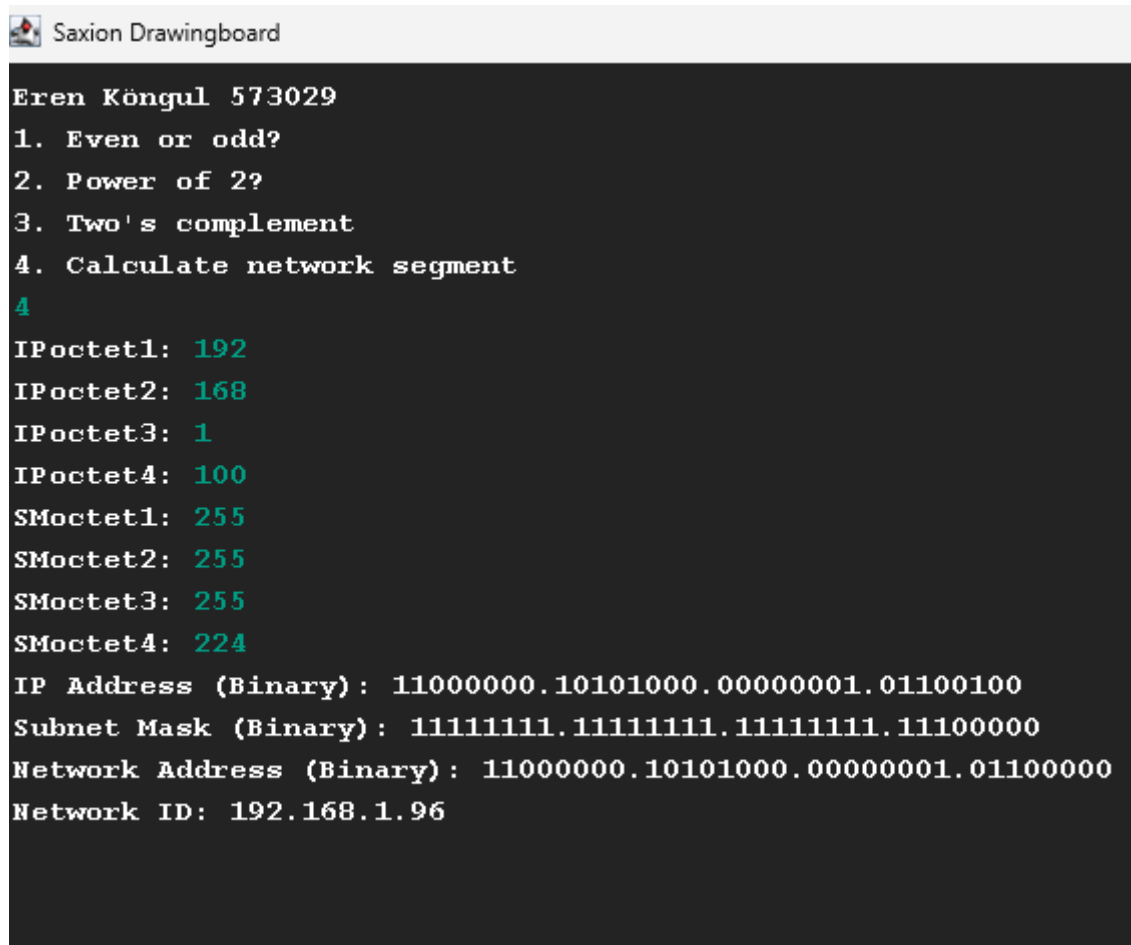
```

String networkBinary2 = toBinary(network2);
String networkBinary3 = toBinary(network3);
String networkBinary4 = toBinary(network4);

SaxionApp.println("IP Address (Binary): " + IPbinary1 + "." + IPbinary2 + "." + IPbinary3 + "." +
IPbinary4);
SaxionApp.println("Subnet Mask (Binary): " + SMbinary1 + "." + SMbinary2 + "." + SMbinary3 +
"." + SMbinary4);
SaxionApp.println("Network Address (Binary): " + networkBinary1 + "." + networkBinary2 + "."
+ networkBinary3 + "." + networkBinary4);
SaxionApp.println("Network ID: " + network1 + "." + network2 + "." + network3 + "." +
network4);
}

// Methode om een integer naar een 8-bits binaire string te converteren
public String toBinary(int value) {
    return String.format("%8s", Integer.toBinaryString(value)).replace(' ', '0');
}
}

```



Saxion Drawingboard

```

Eren Köngül 573029
1. Even or odd?
2. Power of 2?
3. Two's complement
4. Calculate network segment
4
IPoctet1: 192
IPoctet2: 168
IPoctet3: 1
IPoctet4: 100
SMoctet1: 255
SMoctet2: 255
SMoctet3: 255
SMoctet4: 224
IP Address (Binary): 11000000.10101000.00000001.01100100
Subnet Mask (Binary): 11111111.11111111.11111111.11100000
Network Address (Binary): 11000000.10101000.00000001.01100000
Network ID: 192.168.1.96

```

Ready? Save this file and export it as a pdf file with the name: [week6.pdf](#)

