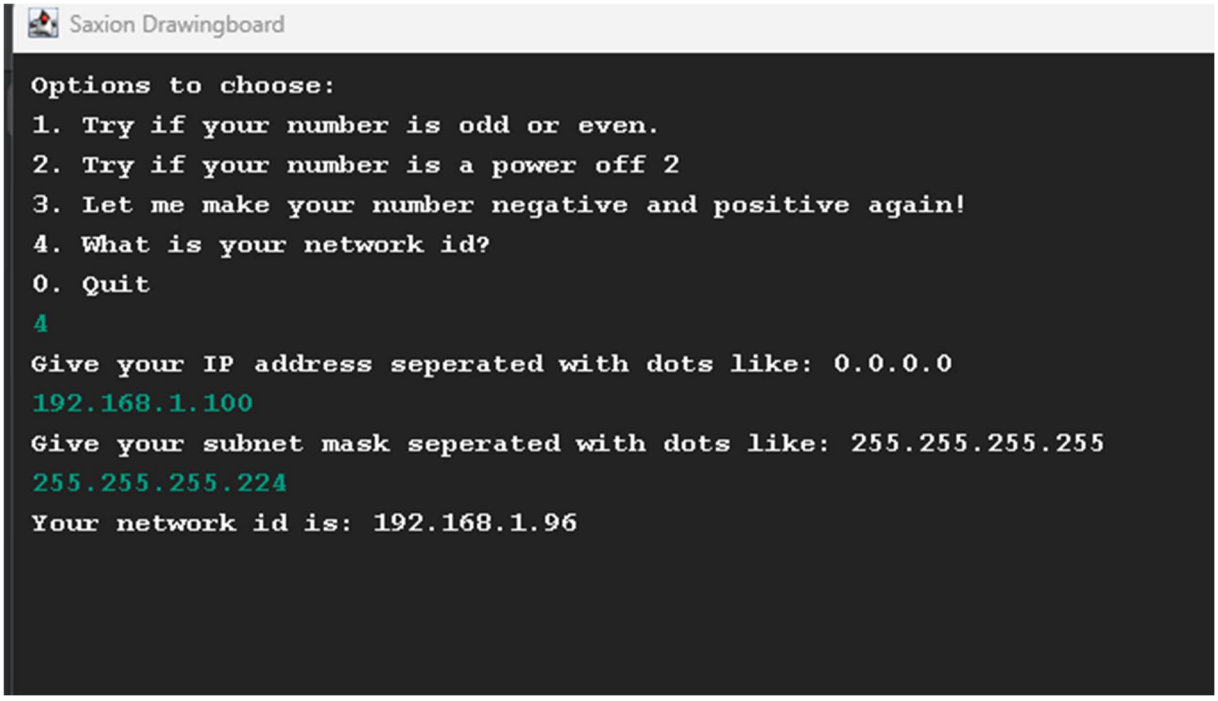
Bonus point assignment – week 6

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.

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

public class Main implements Runnable {
    public static String network_id = "";
    public static String ip_address;
    public static String subnet_mask;


    public static void main(String[] args) {
        SaxionApp.start(new Main(), 1200, 600);


    }

    @Override
    public void run() {
        menu();
    }

    public void menu() {
        int option = -1;
        while(option != 0) {
            SaxionApp.printLine("Options to choose:");
            SaxionApp.printLine("1. Try if your number is odd or even.");
            SaxionApp.printLine("2. Try if your number is a power off 2");
            SaxionApp.printLine("3. Let me make your number negative and
positive again!");
            SaxionApp.printLine("4. What is your network id?");
            SaxionApp.printLine("0. Quit");
            option = SaxionApp.readInt();
            switch (option) {
                case 1:
                    oddOrEven();
                    break;
                case 2:
                    powerOff2();
                    break;
                case 3:
                    twoComplement();
                    break;
                case 4:
                    requestIPSubnet();
                    break;

            }
            SaxionApp.pause();
            SaxionApp.clear();
        }
    }
```

```
    public static void oddOrEven() {
        SaxionApp.printLine("Which number do you want to check?");
        int number = SaxionApp.readInt();

        if ((number & 1) == 1) {
            SaxionApp.printLine("Number is odd");
        } else {
            SaxionApp.printLine("Number is even");
        }
    }

    public static void powerOff2() {
        SaxionApp.printLine("Which number do you want to check?");
        int number = SaxionApp.readInt();

        if ((number & (number - 1)) == 0) {
            SaxionApp.printLine("Number is a power of 2");
        } else {
            SaxionApp.printLine("Number isn't a power of 2");
        }
    }

    public static void twoComplement() {
        SaxionApp.printLine("Which number do you want to check?");
        int number = SaxionApp.readInt();

        int x = ~number + 1;
        SaxionApp.printLine("Two complement of five (to negative): " + x);

        int y = ~x + 1;
        SaxionApp.printLine("Again two complement of the negative five (to
positive): " + y);
    }

    public static void requestIPSubnet() {
        SaxionApp.printLine("Give your IP address seperated with dots like:
0.0.0.0");
        ip_address = SaxionApp.readString();
        SaxionApp.printLine("Give your subnet mask seperated with dots
like: 255.255.255.255");
        subnet_mask = SaxionApp.readString();
        giveNetworkID(ip_address, subnet_mask);
    }
```

```
    public static void giveNetworkID(String ip_address, String subnet_mask)
{
        String[] ipAddressesSplits = ip_address.split("\\.");
        String[] subnetMaskSplits = subnet_mask.split("\\.");

        int[] ipAddressSplitIntegers = new int[4];
        int[] subnetSplitIntegers = new int[4];
        int[] networkPart = new int[4];

        String networkIDWithDot = "";

        for (int i = 0; i < ipAddressSplitIntegers.length; i++) {
            ipAddressSplitIntegers[i] =
Integer.parseInt(ipAddressesSplits[i]);
        }

        for (int i = 0; i < subnetMaskSplits.length; i++) {
            subnetSplitIntegers[i] = Integer.parseInt(subnetMaskSplits[i]);
        }

        for (int i = 0; i < ipAddressesSplits.length; i++) {
            networkPart[i] = ipAddressSplitIntegers[i] &
subnetSplitIntegers[i];
        }

        for (int networkParts:
                networkPart) {
            networkIDWithDot = networkIDWithDot + networkParts + ".";
        }

        network_id = networkIDWithDot.substring(0,networkIDWithDot.length()
- 1);
        SaxionApp.printLine("Your network id is: " + network_id);
    }
}
```