Week 6 – Networking

Student number: 566780

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(), 1024, 768);
  }

  public void run() {
    SaxionApp.printLine("Enter your IP address : ");
    String ipAddress = SaxionApp.readString();
    SaxionApp.printLine("Enter your subnet : ");
    String subNetAddress = SaxionApp.readString();

    String[] ipString = ipAddress.split("\\.");
    int[] ipInteger = new int[ipString.length];

    String[] subnetString = subNetAddress.split("\\.");
    int[] subnetInteger = new int[subnetString.length];

    for (int x = 0; x < ipString.length; x++) {
      ipInteger[x] = Integer.parseInt(ipString[x]);
      subnetInteger[x] = Integer.parseInt(subnetString[x]);
```

```
        }

        int[] networkAddress = new int[ipInteger.length];

        for (int x = 0; x < ipString.length; x++) {
            networkAddress[x] = ipInteger[x] & subnetInteger[x];
        }

        for (int i = 0; i < networkAddress.length; i++) {
            SaxionApp.print(convertToBin(networkAddress[i]));
            if (i < networkAddress.length - 1) {
                SaxionApp.print(".");
            }
        }
    }

    String convertToBin(int address) {
        String binaryString = Integer.toBinaryString(address);
        while (binaryString.length() < 8) {
            binaryString = "0" + binaryString;
        }
        return binaryString;
    }
}
```