Week 6 – Networking

Student number:

570215

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

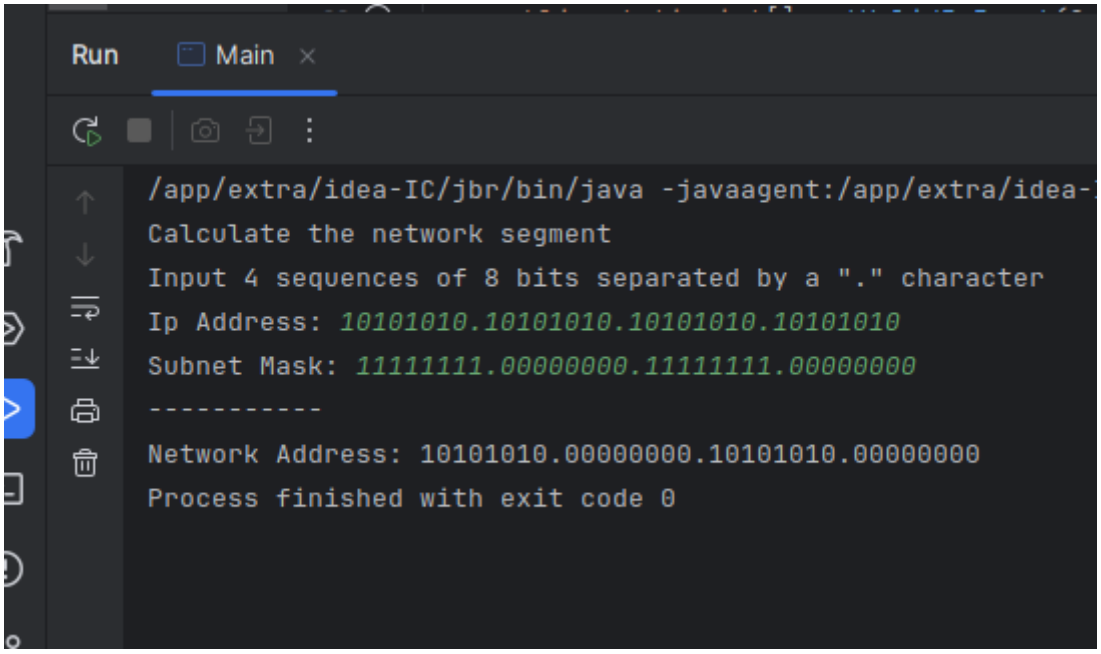
```
Example: 192.168.1.100/27
Calculate the network segment
IP Address:   11000000.10101000.00000001.01100100
Subnet Mask:  11111111.11111111.11111111.11100000
-----------------------------------------------
Network Addr: 11000000.10101000.00000001.01100000

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 java.util.Arrays;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        System.out.println("Calculate the network segment");
        System.out.println("Input 4 sequences of 8 bits separated by a \".\" character");
        Scanner scanner = new Scanner(System.in);

        System.out.print("Ip Address: ");
        int[] ipInput = getValidIpInput(scanner);

        System.out.print("Subnet Mask: ");
        int[] subnetInput = getValidIpInput(scanner);

        System.out.println("-----------");
```

```
        int[] networkAddress = calculateNetworkAddress(ipInput, subnetInput);
        System.out.print("Network Address: ");
        for (int i = 0; i < networkAddress.length; i++) {
            if (i > 0)
                System.out.print(".");
            // Convert to binary string. Pad to 8 bits if neccesary.
            System.out.print(String.format("%8s",
Integer.toBinaryString(networkAddress[i])).replace(' ', '0'));
        }

    }

    public static int[] getValidIpInput(Scanner scanner)
    {
        while (true) {
            boolean validInput = true;
            // Read user input
            String ipInput = scanner.nextLine();
            // Check if string doesn't contain invalid characters
            for (char character : ipInput.toCharArray()) {
                if (character != '0' && character != '1' && character != '.') {
                    System.out.println("Invalid Input, Try again");
                    validInput = false;
                }
            }
            // Split string on '.'
            String[] ipInputSplit = ipInput.split("\\.");
            // Check if string consists of 4 segments
            if (ipInputSplit.length != 4) {
                System.out.println("Input is not 4 sequences of binary numbers. Try again");
                validInput = false;
            }
            // Parse string array into decimal int array
            int[] ipInputIntArray = new int[ipInputSplit.length];
            for (int i = 0; i < ipInputSplit.length; i++) {
                if (ipInputSplit[i].length() != 8) {
                    System.out.println("Part of this input is not exactly 8 bits. Try again");
                    validInput = false;
                }
                // Parse string to int, also calculate the binary to decimal.
                ipInputIntArray[i] = Integer.parseInt(ipInputSplit[i], 2);
            }
            if (validInput)
                return ipInputIntArray;
        }
    }



    public static int[] calculateNetworkAddress(int[] ipAddress, int[] subnetMask) {
        int[] networkAddress = new int[ipAddress.length];
        for (int i = 0; i < ipAddress.length; i++) {
            networkAddress[i] = ipAddress[i] & subnetMask[i];
        }
        return networkAddress;
```

```
    }
}
```



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