Template Week 6 – Networking

Student number:571334

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

    System.out.print("Enter IP address with CIDR notation: ");
    String userInput = reader.next();

    int cidrSegment = Integer.parseInt(userInput.split( regex: "/")[1]);
    String[] ipSegment = userInput.split( regex: "/")[0].split( regex: "\\.");

    String binaryIp = ipToBinary(ipSegment);
    String binarySubnet = cidrToBinarySubnet(cidrSegment);

    System.out.printf("IP Address:  %s\n", binaryIp);
    System.out.printf("Subnet Mask: %s\n", binarySubnet);
    System.out.println("-".repeat( count: 48));
    System.out.printf("Network addr: %s", calculateNetworkSegment(binaryIp, binarySubnet));
}

private static String cidrToBinarySubnet(int cidr) {  1 usage
    StringBuilder binarySubnet = new StringBuilder();
    final int ipLength = 32;

    binarySubnet.append("1".repeat(cidr))
            .append("0".repeat( count: ipLength - cidr));

    return format(binarySubnet);
}
```

```java
private static String ipToBinary(String[] ip) {  1 usage
    StringBuilder binaryIp = new StringBuilder();

    for (String i : ip) {
        String binary = Integer.toBinaryString(Integer.parseInt(i));

        if (binary.length() != 8) {
            binary = "0".repeat( count: 8 - binary.length()) + binary;
        }

        binaryIp.append(binary);
    }

    return format(binaryIp);
}

private static String calculateNetworkSegment(String binaryIp, String binarySubnet) {  1 usage
    StringBuilder networkAddr = new StringBuilder();

    for (int i = 0; i < binaryIp.length(); i++) {
        if (binarySubnet.charAt(i) == '.') {
            networkAddr.append(".");
        } else {
            int ipBit = binaryIp.charAt(i) - '0';
            int subnetBit = binarySubnet.charAt(i) - '0';
            networkAddr.append(ipBit & subnetBit);
        }
    }

    return networkAddr.toString();
}

private static String format(StringBuilder sb) {  2 usages
    return sb.insert( offset: 8,  str: ".")
            .insert( offset: 17,  str: ".")
            .insert( offset: 26,  str: ".").toString();
}
```

```
C:\Users\mmast\.jdks\corretto-21.0.4\bin\java.exe "-jav
Enter IP address with CIDR notation: 192.168.1.100/27
IP Address:  11000000.10101000.00000001.01100100
Subnet Mask: 11111111.11111111.11111111.11100000
-----------------------------------------------
Network addr: 11000000.10101000.00000001.01100000
Process finished with exit code 0
```

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