

Student number: 571894

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 java.util.Scanner;
```

```
public class Main {
```

```
    // Convert decimal IP address to a 32-bit binary string
```

```
    public static String decimalToBinary(int num) {
```

```
        String binaryString = Integer.toBinaryString(num);
```

```
        // Ensure the binary string is 32 bits long by padding with leading zeros
```

```
        while (binaryString.length() < 32) {
```

```
            binaryString = "0" + binaryString;
```

```
        }
```

```
        return binaryString;
```

```
    }
```

```
    // Convert a string IP address (e.g. "192.168.1.100") to a decimal integer
```

```
    public static int ipToDecimal(String ip) {
```

```

String[] parts = ip.split("\\.");
int decimalIp = 0;
for (int i = 0; i < 4; i++) {
    decimalIp |= Integer.parseInt(parts[i]) << (24 - (i * 8));
}
return decimalIp;
}

// Convert a decimal IP to its string format (e.g. 3232235876 to "192.168.1.100")
public static String decimalToIp(int decimal) {
    return ((decimal >> 24) & 0xFF) + "." +
        ((decimal >> 16) & 0xFF) + "." +
        ((decimal >> 8) & 0xFF) + "." +
        (decimal & 0xFF);
}

// Calculate the network address (bitwise AND between IP and subnet mask)
public static String calculateNetworkSegment(String ip, String subnetMask) {
    // Convert IP and subnet mask to decimal format
    int ipDecimal = ipToDecimal(ip);
    int subnetDecimal = ipToDecimal(subnetMask);

    // Perform bitwise AND
    int networkDecimal = ipDecimal & subnetDecimal;

    // Convert the result back to dotted decimal format
    return decimalToIp(networkDecimal);
}

// Calculate the range of IP addresses in the network segment
public static String calculateIpRange(String networkAddress, String subnetMask) {

```

```

int subnetDecimal = ipToDecimal(subnetMask);

int networkDecimal = ipToDecimal(networkAddress);


// Calculate the number of IPs in the subnet (based on the subnet mask)
int hostBits = 32 - Integer.bitCount(subnetDecimal);
int totalIps = 1 << hostBits; // 2^number_of_host_bits


// The range of IPs goes from the network address to (network + totalIps - 1)
int startRange = networkDecimal;
int endRange = startRange + totalIps - 1;


return decimalToIp(startRange) + " to " + decimalToIp(endRange);
}


// Method to display the menu
public static void displayMenu() {
    System.out.println("Please select an option:");
    System.out.println("1. Is number odd?");
    System.out.println("2. Is number a power of 2?");
    System.out.println("3. Two's complement of the number");
    System.out.println("4. Calculate network segment");
    System.out.println("5. Exit");
}


public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    int keuze = 0;
    int nummer = 0;


    while (true) {
        displayMenu();

```

```

keuze = scanner.nextInt();

if (keuze == 5) {
    System.out.println("Exiting the program. Goodbye!");
    break;
}

// Prompt the user for a number
if (keuze == 4) {
    scanner.nextLine(); // Consume newline
    System.out.print("Enter an IP address (e.g. 192.168.1.100): ");
    String ip = scanner.nextLine();
    System.out.print("Enter a subnet mask (e.g. 255.255.255.224): ");
    String subnetMask = scanner.nextLine();

    // Calculate the network segment
    String networkSegment = calculateNetworkSegment(ip, subnetMask);
    System.out.println("Network Address: " + networkSegment);

    // Calculate the range of IP addresses
    String ipRange = calculateIpRange(networkSegment, subnetMask);
    System.out.println("IP Range: " + ipRange);
    continue;
}

System.out.print("Enter a number: ");
nummer = scanner.nextInt();
switch (keuze) {
    case 1:
        if (isOdd(nummer)) {
            System.out.println(nummer + " is odd.");

```

```

    } else {
        System.out.println(nummer + " is even.");
    }
    break;

case 2:
    if (isPowerOfTwo(nummer)) {
        System.out.println(nummer + " is a power of 2.");
    } else {
        System.out.println(nummer + " is NOT a power of 2.");
    }
    break;

case 3:
    int complement = twoComplement(nummer);
    System.out.println("The two's complement of " + nummer + " is " + complement);
    break;

default:
    System.out.println("Invalid choice. Please select a valid option.");
    break;
}
}
scanner.close();
}

// Method to check if a number is odd
public static boolean isOdd(int num) {
    return (num & 1) == 1;
}

```

```
// Method to check if a number is a power of two
public static boolean isPowerOfTwo(int num) {
    return (num > 0) && ((num & (num - 1)) == 0);
}

// Method to calculate the two's complement
public static int twoComplement(int num) {
    return ~num + 1;
}
}
```

```
Enter an IP address (e.g. 192.168.1.100): 192.168.1.100
Enter a subnet mask (e.g. 255.255.255.224): 255.255.255.224
Network Address: 192.168.1.96
IP Range: 192.168.1.96 to 192.168.1.127
Please select an option:
1. Is number odd?
2. Is number a power of 2?
ion Control Alt+9 ment of the number
```

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