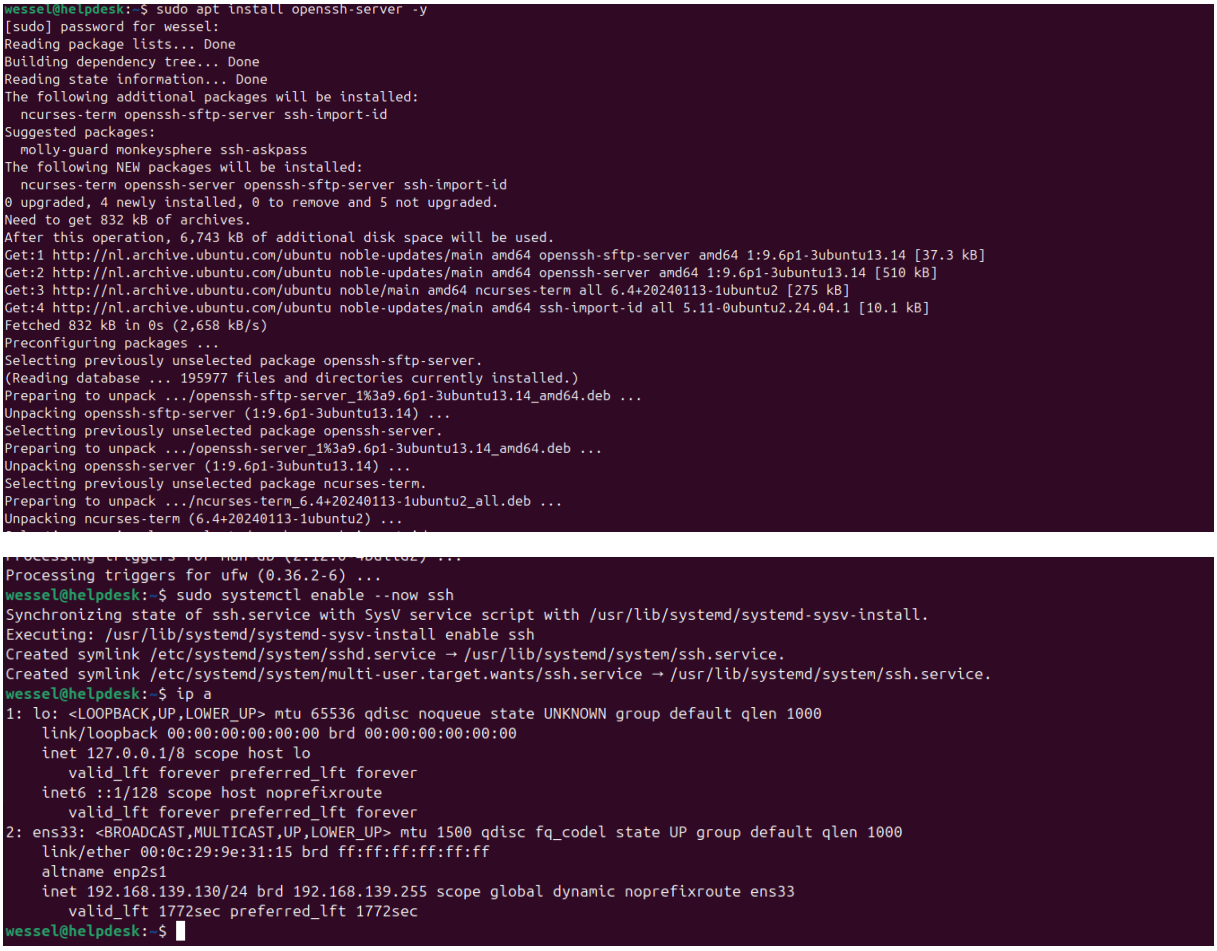
Template Week 6 – Networking

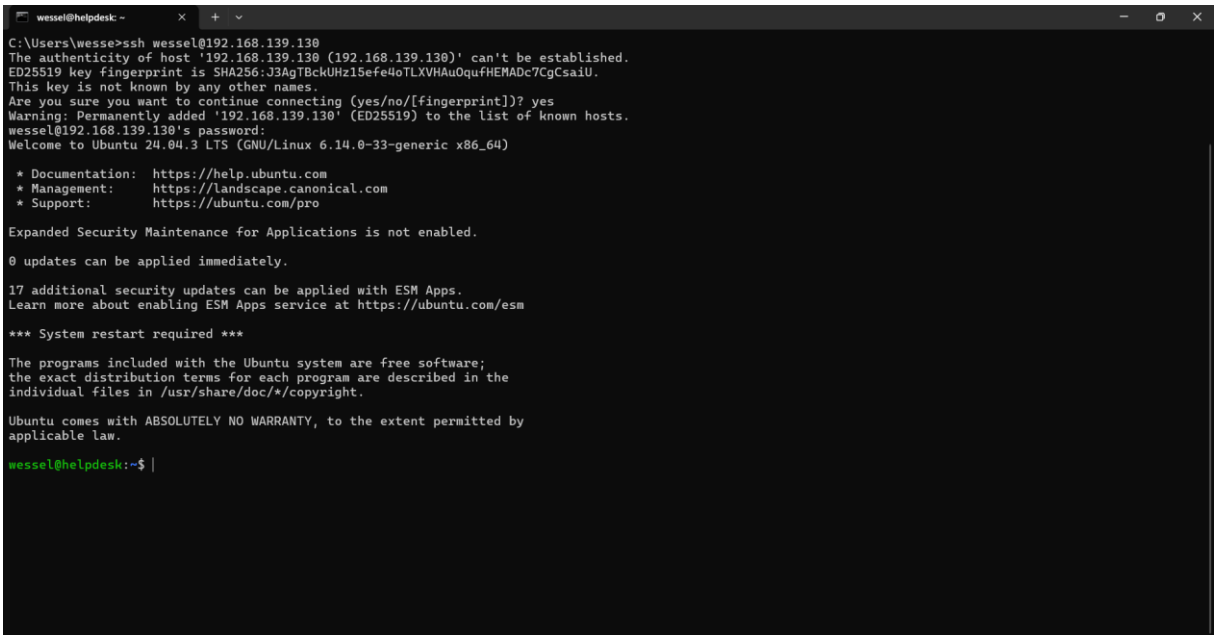
Student number:

Assignment 6.1: Working from home

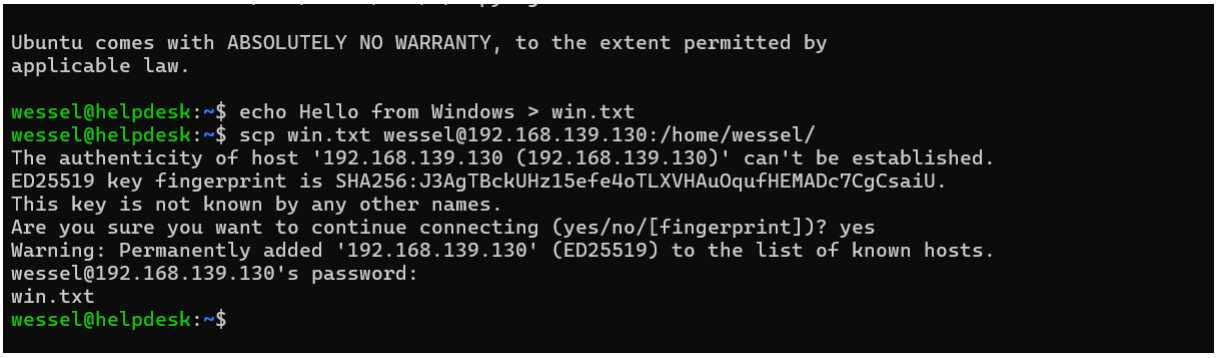
Screenshot installation openssh-server:

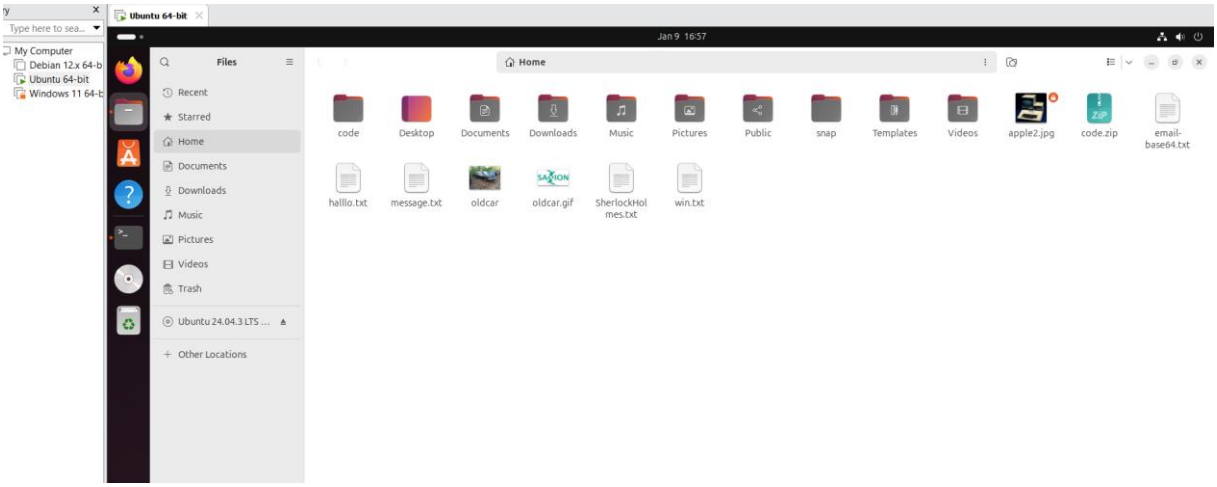


Screenshot successful SSH command execution:



Screenshot successful execution SCP command:





Screenshot remmina:

Hij kom mijn ip niet vinden

Assignment 6.2: IP addresses websites

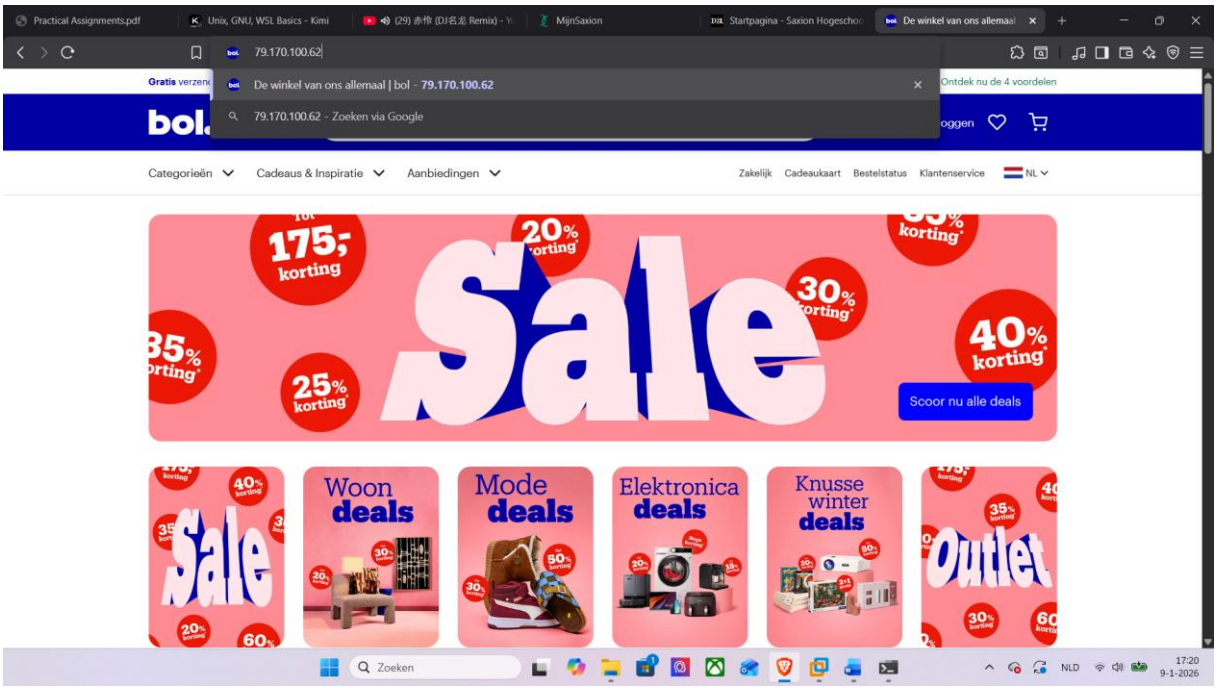
Relevant screenshots nslookup command:

```
wessel@helpdesk:~$ exit
logout
Connection to 192.168.139.130 closed.

C:\Users\wesse>nslookup amazon.com
Server:  mijnmodem.kpn
Address:  2a02:a46f:3878:0:46fb:5aff:fecb:f5d3

Non-authoritative answer:
Name:    amazon.com
Addresses:  98.87.170.74
          98.87.170.71
          98.82.161.185


C:\Users\wesse>nslookup google.com
Server:  mijnmodem.kpn
Address:  2a02:a46f:3878:0:46fb:5aff:fecb:f5d3

Non-authoritative answer:
Name:    google.com
Addresses:  2a00:1450:400e:806::200e
          142.251.142.206


C:\Users\wesse>
```

```
C:\Users\wesse>nslookup one.one.one.one
Server:  mijnmodem.kpn
Address:  2a02:a46f:3878:0:46fb:5aff:fecb:f5d3

Non-authoritative answer:
Name:    one.one.one.one
Addresses:  2606:4700:4700::1111
          2606:4700:4700::1001
          1.0.0.1
          1.1.1.1


C:\Users\wesse>dns.google.com
'dns.google.com' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\wesse>nslookup dns.google.com
Server:  mijnmodem.kpn
Address:  2a02:a46f:3878:0:46fb:5aff:fecb:f5d3

Non-authoritative answer:
Name:    dns.google.com
Addresses:  2001:4860:4860::8888
          2001:4860:4860::8844
          8.8.4.4
          8.8.8.8
```

```
C:\Users\wesse>nslookup bol.com
Server:  mijnmodem.kpn
Address:  2a02:a46f:3878:0:46fb:5aff:fecb:f5d3

Non-authoritative answer:
Name:    bol.com
Address:  79.170.100.62


C:\Users\wesse>

C:\Users\wesse>

C:\Users\wesse>nslookup w3schools.com
Server:  mijnmodem.kpn
Address:  2a02:a46f:3878:0:46fb:5aff:fecb:f5d3

Non-authoritative answer:
Name:    w3schools.com
Addresses:  13.248.240.135
          76.223.115.82
```

Screenshot website visit via IP address:



Assignment 6.3: subnetting

How many IP addresses are in this network configuration 192.168.110.128/25?

128

What is the usable IP range to hand out to the connected computers?

192.168.110.129 t/m 192.168.110.254 (126 bruikbare adressen)

Check your two previous answers with this Linux command: `ipcalc 192.168.110.128/25`

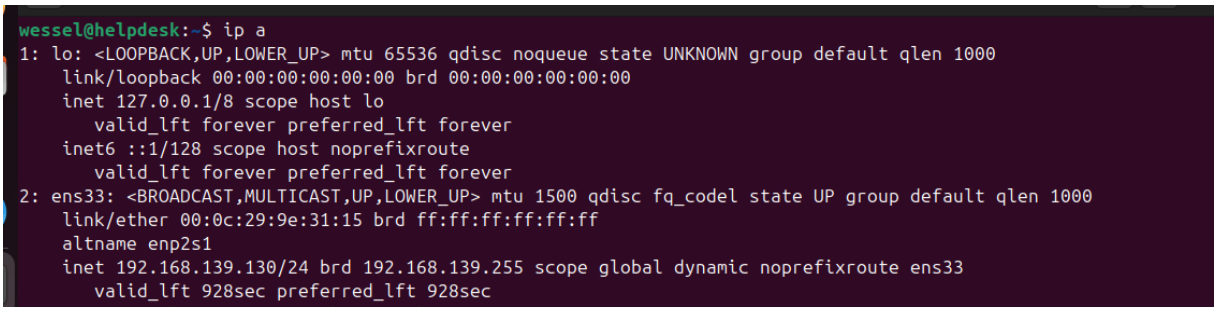
```
wessel@helpdesk:~$ ipcalc 192.168.110.128/25
Address:   192.168.110.128      11000000.10101000.01101110.1 0000000
Netmask:   255.255.255.128 = 25 11111111.11111111.11111111.1 0000000
Wildcard:  0.0.0.127            00000000.00000000.00000000.0 1111111
=>
Network:   192.168.110.128/25   11000000.10101000.01101110.1 0000000
HostMin:   192.168.110.129      11000000.10101000.01101110.1 0000001
HostMax:   192.168.110.254      11000000.10101000.01101110.1 1111110
Broadcast: 192.168.110.255      11000000.10101000.01101110.1 1111111
Hosts/Net: 126                   Class C, Private Internet
```

Explain the above calculation in your own words.

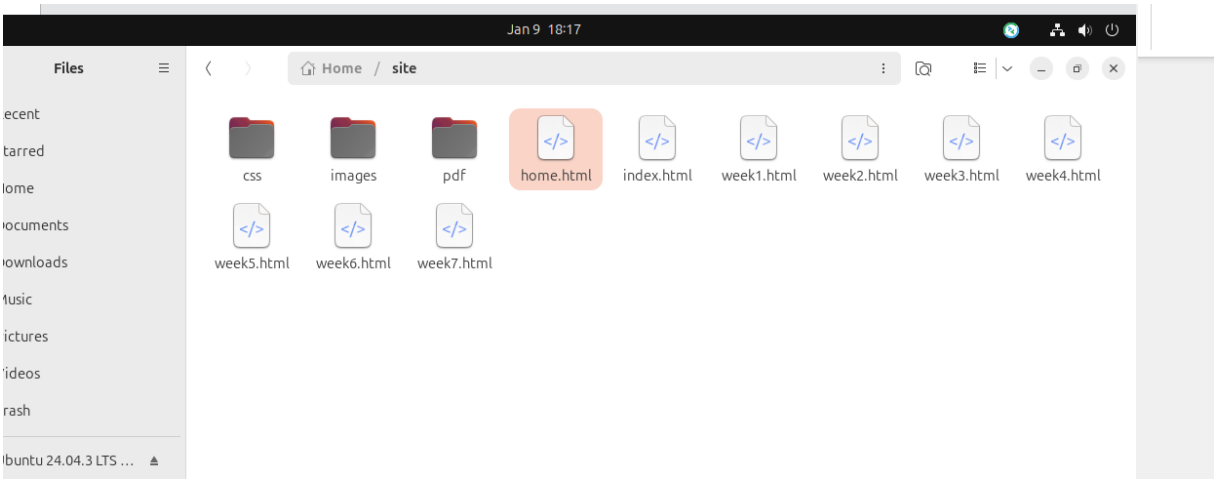
het subnet begint op 128, de eerste 129 is de gateway/printer enz., 254 is de laatste pc; 128 én 255 zijn ‘gereserveerd’ dus die mag je niet uitdelen

Assignment 6.4: HTML

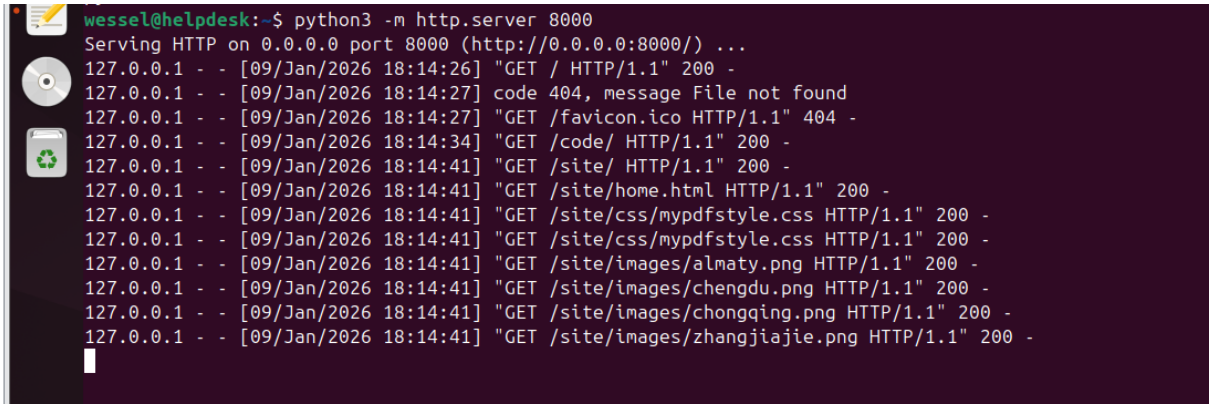
Screenshot IP address Ubuntu VM:



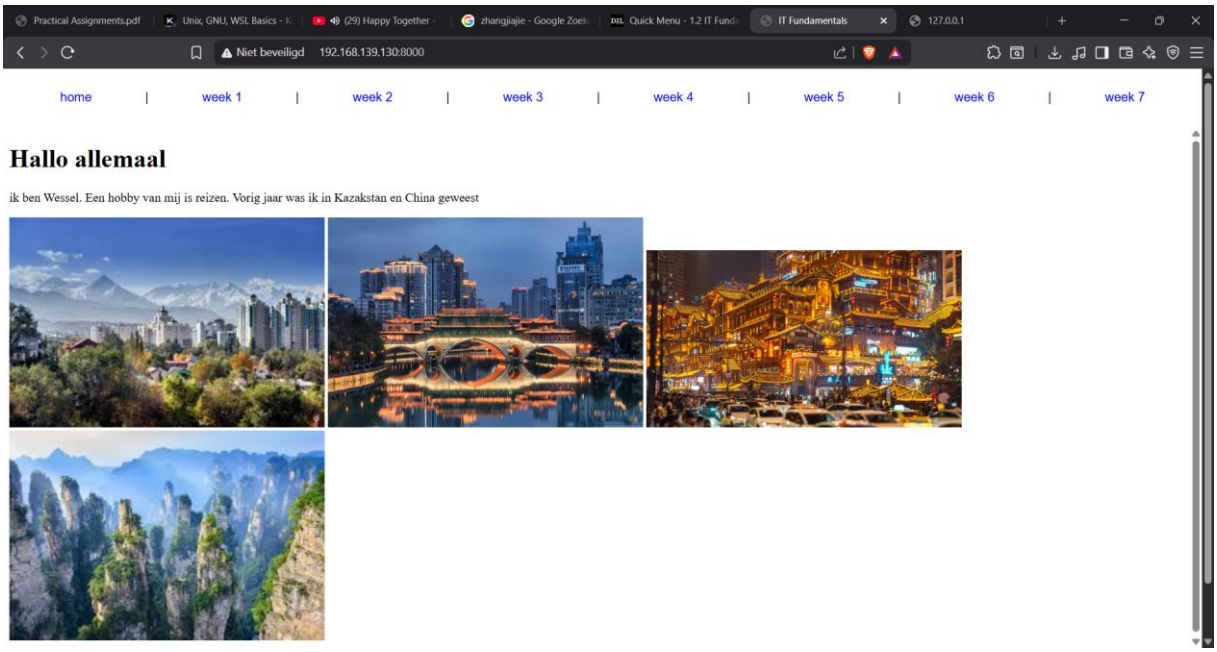
Screenshot of Site directory contents:



Screenshot python3 webserver command:



Screenshot web browser visits your site



Assignment 6.5: Network segment

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

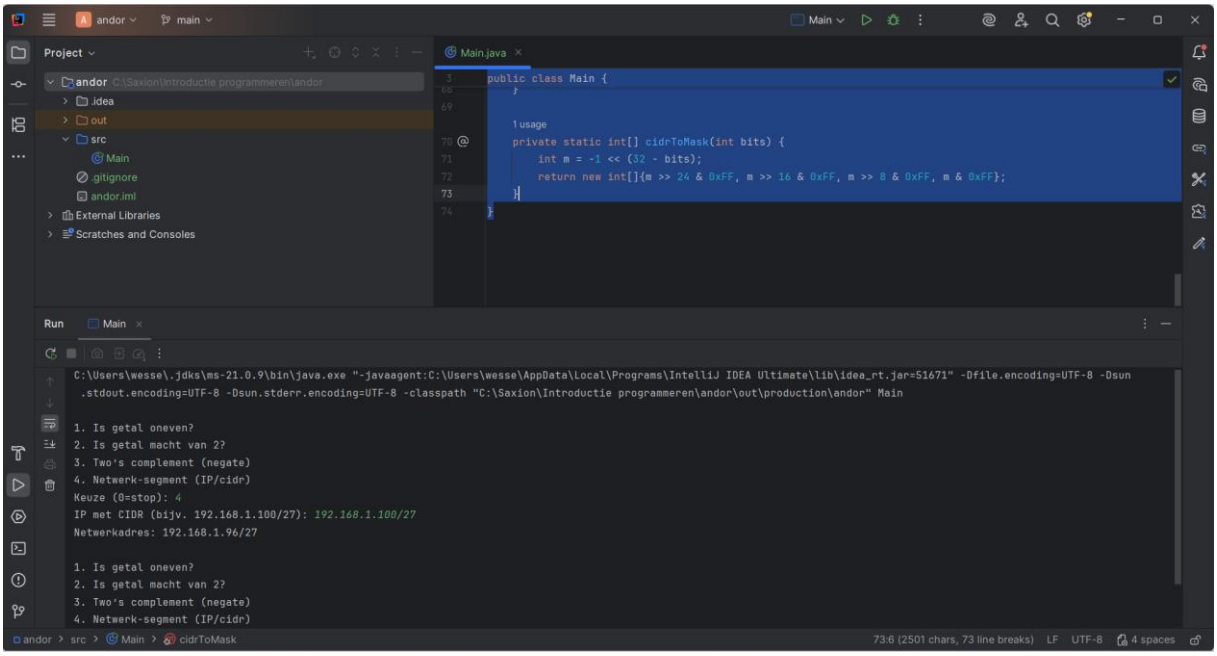
```
Example: 192.168.1.100/27
Calculate the network segment
IP Address:   11000000.10101000.00000001.01100100
Subnet Mask:  11111111.11111111.11111111.11100000
-----------------------------------------------
Network Addr: 11000000.10101000.00000001.01100000

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 {
   private static final Scanner sc = new Scanner(System.in);

   public static void main(String[] args) {
      while (true) {
```

```
        System.out.println("\n1. Is number odd?");
        System.out.println("2. Is number a power of 2?");
        System.out.println("3. Two's complement of number");
        System.out.println("4. Netwerk-segment ");
        System.out.print("Keuze: ");
        switch (sc.nextLine()) {
            case "1" -> oddDemo();
            case "2" -> powerOf2Demo();
            case "3" -> twosComplementDemo();
            case "4" -> netSegmentDemo();
            case "0" -> { return; }
            default  -> System.out.println("Ongeldige keuze");
        }
    }
}

private static void oddDemo() {
    System.out.print("number: ");
    int n = Integer.parseInt(sc.nextLine());
    System.out.println((n & 1) == 1 ? "oneven" : "even");
}

private static void powerOf2Demo() {
    System.out.print("number : ");
    int n = Integer.parseInt(sc.nextLine());
    boolean ok = n > 0 && (n & (n - 1)) == 0;
    System.out.println(ok ? "macht van 2" : "geen macht van 2");
}

private static void twosComplementDemo() {
    System.out.print("Number: ");
    int n = Integer.parseInt(sc.nextLine());
    System.out.println("Two's complement: " + (~n + 1));
}

private static void netSegmentDemo() {
    System.out.print("IP : ");
    String input = sc.nextLine();
    String[] parts = input.split("/");
    String ip = parts[0];
    int cidr = Integer.parseInt(parts[1]);

    int[] net = calcNet(ip, cidr);
    System.out.printf("Netwerkadres: %d.%d.%d.%d/%d%n",
            net[0], net[1], net[2], net[3], cidr);
}

private static int[] calcNet(String ip, int cidr) {
```

```
        int[] ipOct = toInts(ip);
        int[] maskOct = cidrToMask(cidr);
        int[] netOct = new int[4];
        for (int i = 0; i < 4; i++) netOct[i] = ipOct[i] & maskOct[i];
        return netOct;
    }

    private static int[] toInts(String s) {
        String[] p = s.split("\\.");
        int[] a = new int[4];
        for (int i = 0; i < 4; i++) a[i] = Integer.parseInt(p[i]);
        return a;
    }

    private static int[] cidrToMask(int bits) {
        int m = -1 << (32 - bits);
        return new int[]{m >> 24 & 0xFF, m >> 16 & 0xFF, m >> 8 & 0xFF, m & 0xFF};
    }
}
```



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