Template Week 6 – Networking

Student number:

580693

Assignment 6.1: Working from home

Screenshot installation openssh-server:

```
kliment@kliment-VMware-Virtual-Platform:~$ sudo apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openssh-server is already the newest version (1:9.6p1-3ubuntu13.14).
0 upgraded, 0 newly installed, 0 to remove and 183 not upgraded.
```

Screenshot successful SSH command execution:

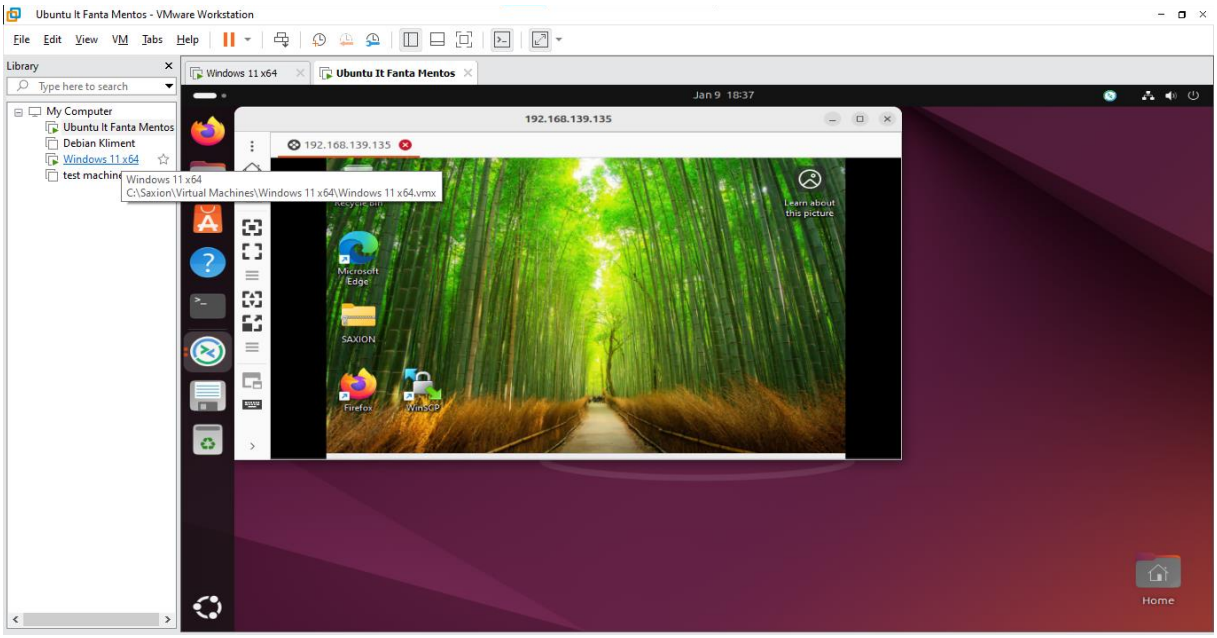
```
kliment@kliment-VMware-Virtual-Platform:~$ ssh kliment@192.168.139.134
The authenticity of host '192.168.139.134 (192.168.139.134)' can't be establishe
d.
ED25519 key fingerprint is SHA256:Klu3htZ9JjLCs2FvgHN/IPPTR7ELBCXmx7BJMmN9mo0.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.139.134' (ED25519) to the list of known host
s.
kliment@192.168.139.134's password:
Linux kliment 6.12.57+deb13-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.12.57-1 (2025-
11-05) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Jan  9 14:12:58 2026 from 192.168.139.1
kliment@kliment:~$
```

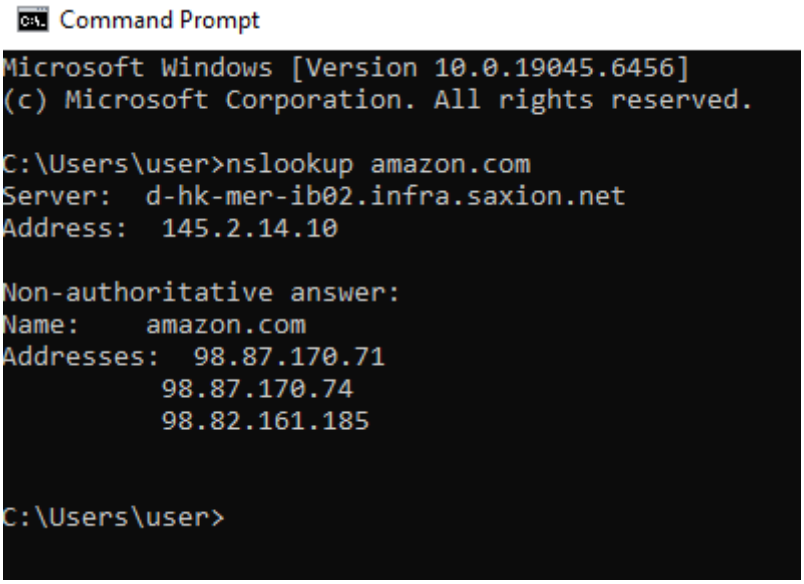
Screenshot successful execution SCP command:

Screenshot remmina:



Assignment 6.2: IP addresses websites

Relevant screenshots nslookup command:



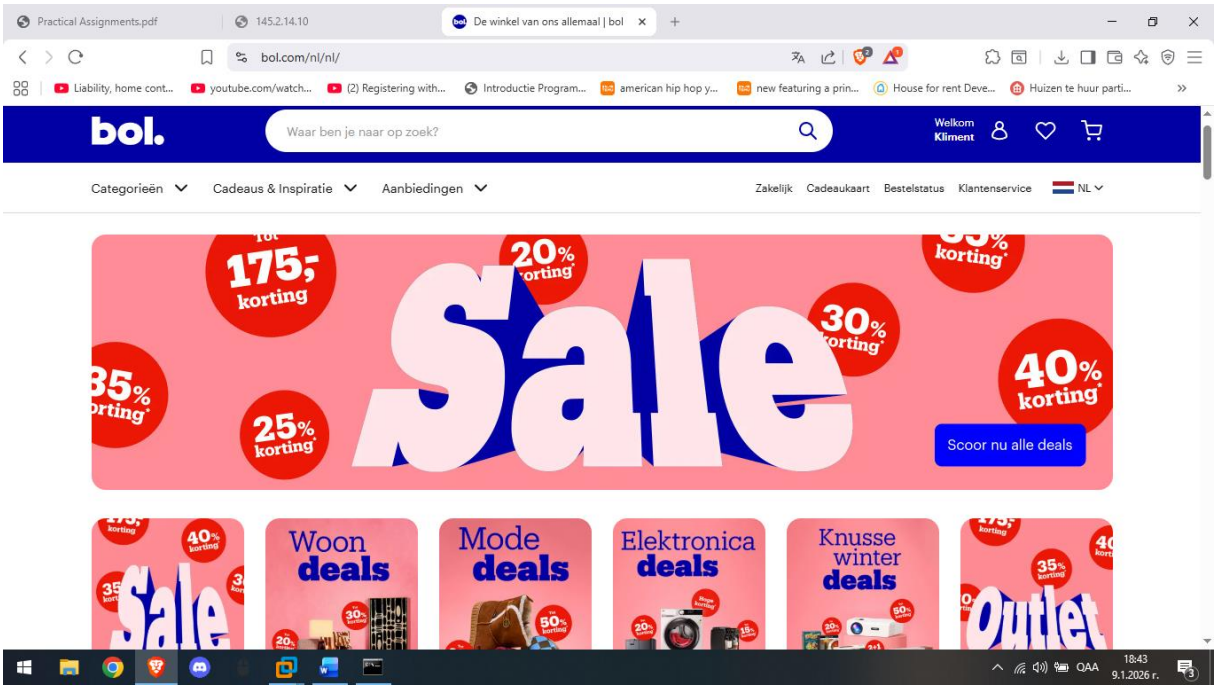
```
C:\Users\user>nslookup bol.com
Server:  d-hk-mer-ib02.infra.saxion.net
Address:  145.2.14.10

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


C:\Users\user>nslookup w3school.com
Server:  d-hk-mer-ib02.infra.saxion.net
Address:  145.2.14.10

Non-authoritative answer:
Name:    w3school.com
Addresses:  2a02:4780:13:1284:0:1b75:f1f9:3
          93.127.191.6
```

Screenshot website visit via IP address:



Assignment 6.3: subnetting

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

There are 128 IP addresses, because 32 – 25 = 7 bits = 27 , but only 126 of them are usable, because the other 2 are for reserved for specific purposes (.128 and .255).

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

The usable IP range is 192.168.110.129 - 192.168.110.254

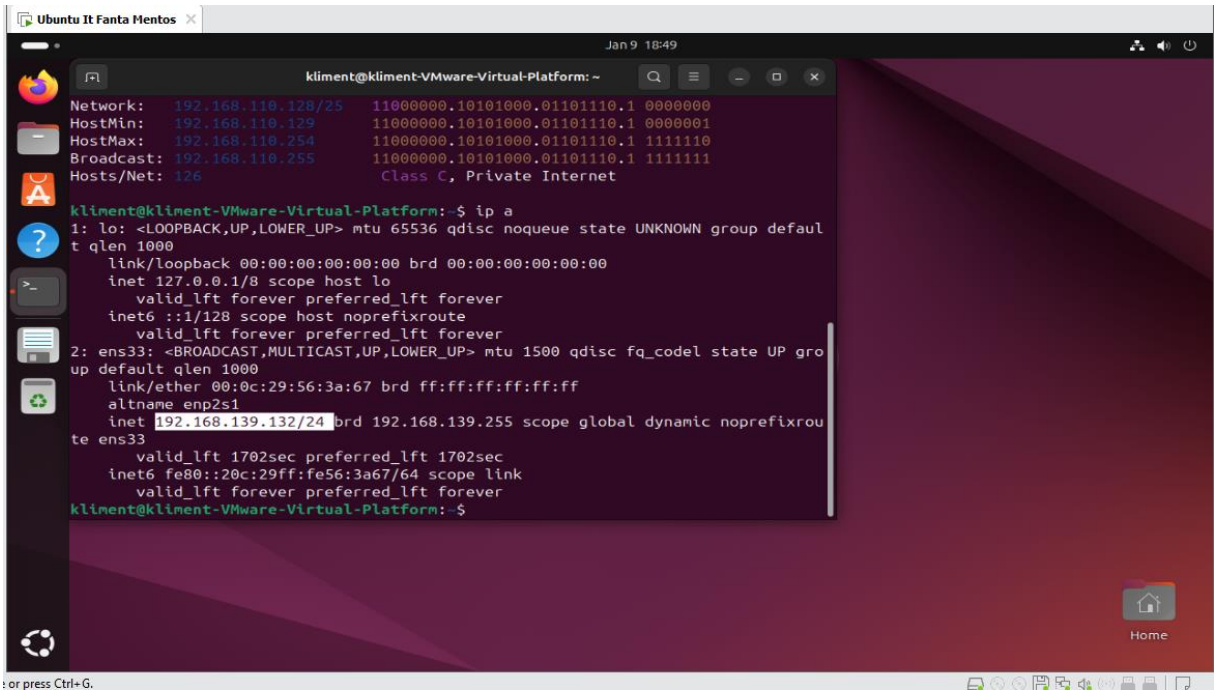
Check your two previous answers with this Linux command: `ipcalc 192.168.110.128/25`

```
kliment@kliment-VMware-Virtual-Platform:~$ 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
```

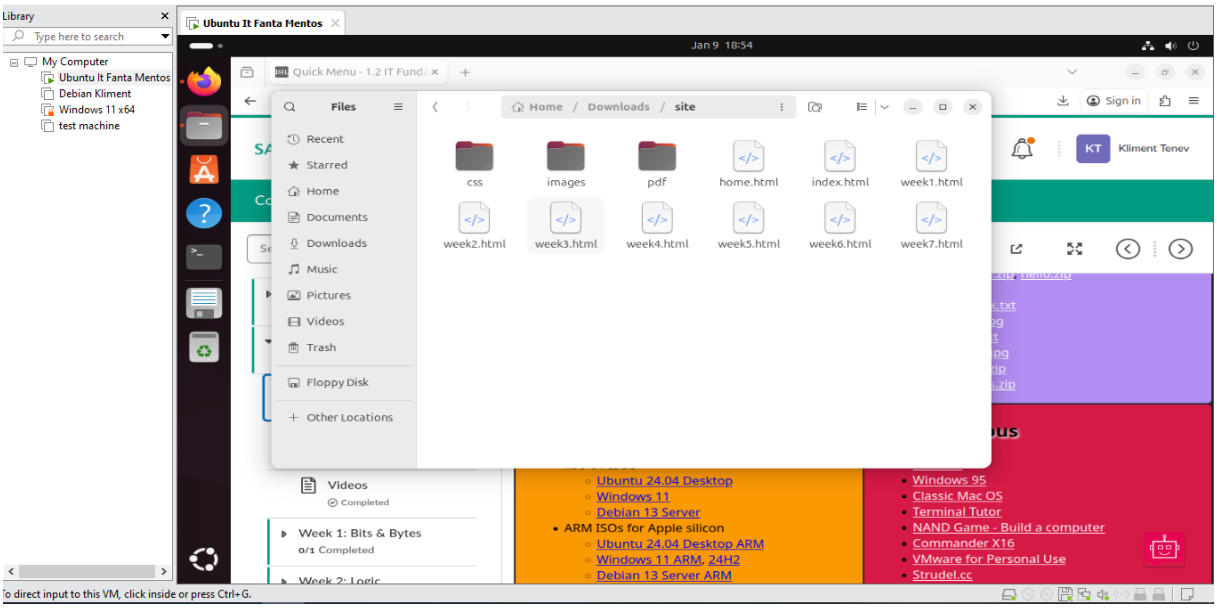
Explain the above calculation in your own words.

Assignment 6.4: HTML

Screenshot IP address Ubuntu VM:



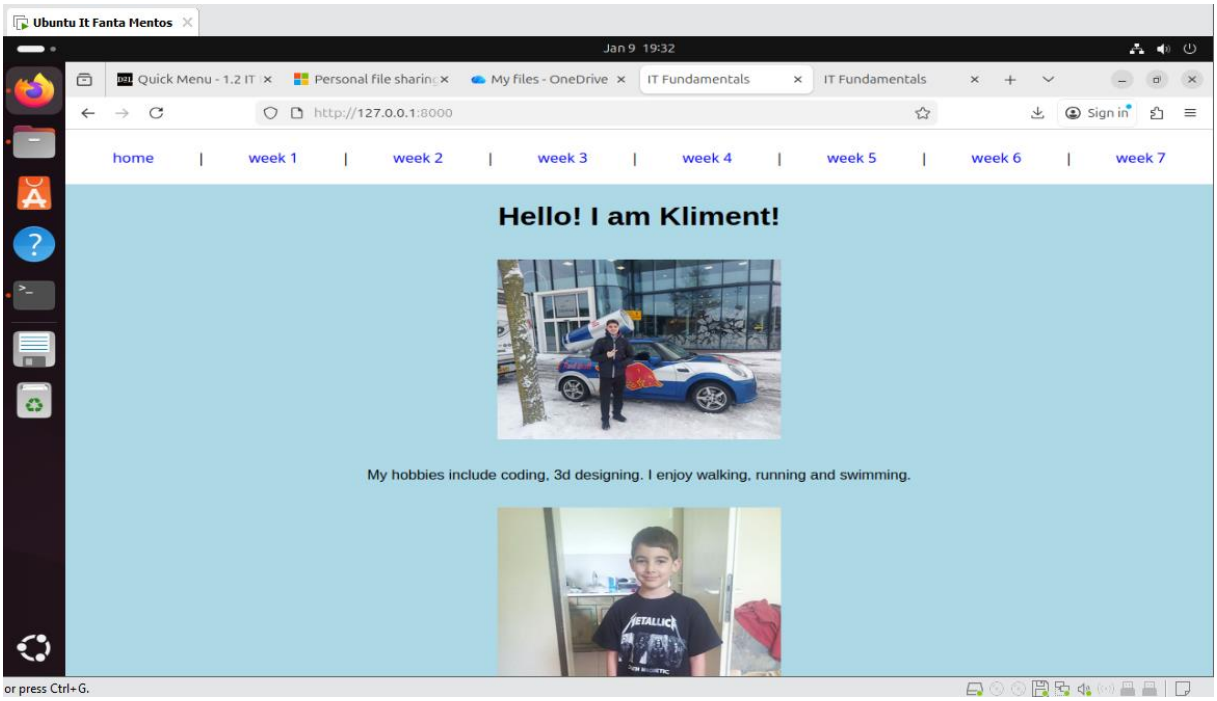
Screenshot of Site directory contents:



Screenshot python3 webserver command:

```
kliment@kliment-VMware-Virtual-Platform:~/Downloads/site$ python3 -m http.server
 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
127.0.0.1 - - [09/Jan/2026 18:59:28] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [09/Jan/2026 18:59:28] "GET /css/mypdfstyle.css HTTP/1.1" 200 -
127.0.0.1 - - [09/Jan/2026 18:59:28] "GET /home.html HTTP/1.1" 200 -
```

Screenshot web browser visits your site



Assignment 6.5: Network segment

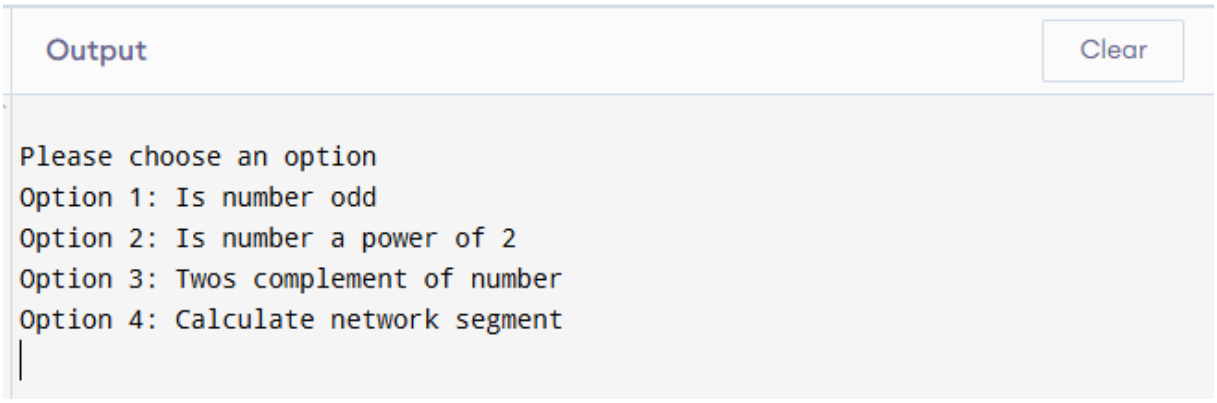
Remember that bitwise java application you've made in week 2? Expand that application so that you can also calculate a network segment as explained in the PowerPoint slides of week 6. Use the bitwise & AND operator. You need to be able to input two Strings. An IP address and a subnet.

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 {


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

```
private static void showMenu() {
    Scanner sc = new Scanner(System.in);

    System.out.println("\nPlease choose an option");
    System.out.println("Option 1: Is number odd");
    System.out.println("Option 2: Is number a power of 2");
    System.out.println("Option 3: Twos complement of number");
    System.out.println("Option 4: Calculate network segment");

    int menuChoice = sc.nextInt();
    sc.nextLine(); // clear buffer

    int number;

    switch (menuChoice) {
        case 1:
            System.out.println("Enter number");
            number = sc.nextInt();
            isNumberOdd(number);
            break;

        case 2:
            System.out.println("Enter number");
            number = sc.nextInt();
            isPowerOfTwo(number);
            break;

        case 3:
            System.out.println("Enter number");
            number = sc.nextInt();
```

```
            twoComplement(number);
            break;

        case 4:
            System.out.println("Enter IP address (e.g. 192.168.1.100):");
            String ipAddress = sc.nextLine();

            System.out.println("Enter subnet mask (e.g. 255.255.255.224):");
            String subnetMask = sc.nextLine();

            calculateNetworkSegment(ipAddress, subnetMask);
            break;

        default:
            System.out.println("Invalid option");
    }
}

public static void twoComplement(int num) {
    num = ~num;
    System.out.println("This is the negative version of the number: " + (num + 1));
}

public static void isNumberOdd(int num) {
    if ((num & 1) == 1) {
        System.out.println("Number is odd");
    } else {
        System.out.println("Number is even");
    }
}
```

```
public static void isPowerOfTwo(int num) {
    if (num > 0 && (num & (num - 1)) == 0) {
        System.out.println("The number is a power of 2");
    } else {
        System.out.println("The number is not a power of 2");
    }
}

// ===== OPTION 4 METHODS =====

public static void calculateNetworkSegment(String ipAddress, String subnetMask) {

    int[] ip = parseAddress(ipAddress);
    int[] subnet = parseAddress(subnetMask);
    int[] network = new int[4];

    System.out.println("\nIP Address:     " + toBinaryString(ip));
    System.out.println("Subnet Mask:    " + toBinaryString(subnet));
    System.out.println("-----------------------------------------");

    for (int i = 0; i < 4; i++) {
        network[i] = ip[i] & subnet[i];
    }

    System.out.println("Network Addr:   " + toBinaryString(network));
    System.out.println("Network Address (Decimal): " + toDecimalString(network));

    int subnetSize = 256 - subnet[3];
    int start = network[3];
    int end = start + subnetSize - 1;
```

```
        System.out.println("Network Range:");
        System.out.println("From " + network[0] + "." + network[1] + "." + network[2] + "." + start);
        System.out.println("To   " + network[0] + "." + network[1] + "." + network[2] + "." + end);
    }

    private static int[] parseAddress(String address) {
        String[] parts = address.split("\\.");
        int[] result = new int[4];

        for (int i = 0; i < 4; i++) {
            result[i] = Integer.parseInt(parts[i]);
        }
        return result;
    }

    private static String toBinaryString(int[] address) {
        StringBuilder sb = new StringBuilder();
        for (int i = 0; i < 4; i++) {
            sb.append(String.format("%8s",
                    Integer.toBinaryString(address[i])).replace(' ', '0'));
            if (i < 3) sb.append(".");
        }
        return sb.toString();
    }

    private static String toDecimalString(int[] address) {
        return address[0] + "." + address[1] + "." + address[2] + "." + address[3];
    }
}
```

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