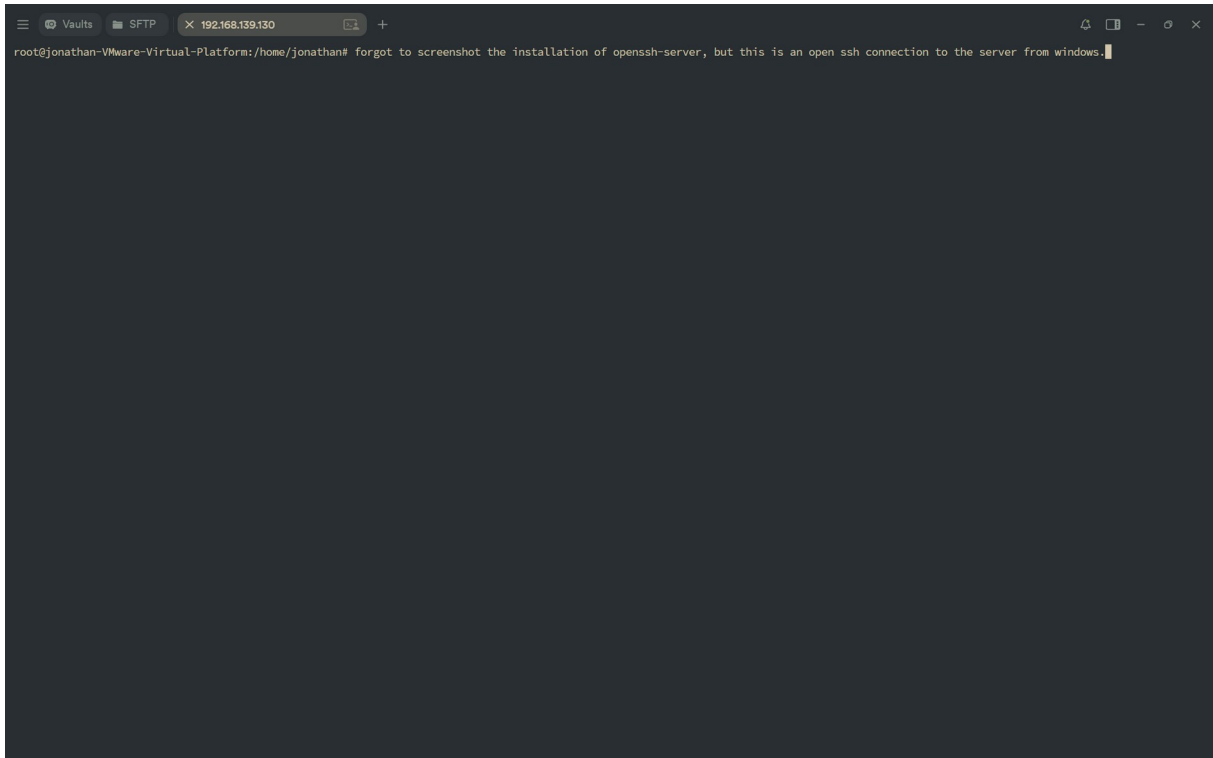


Template Week 6 – Networking

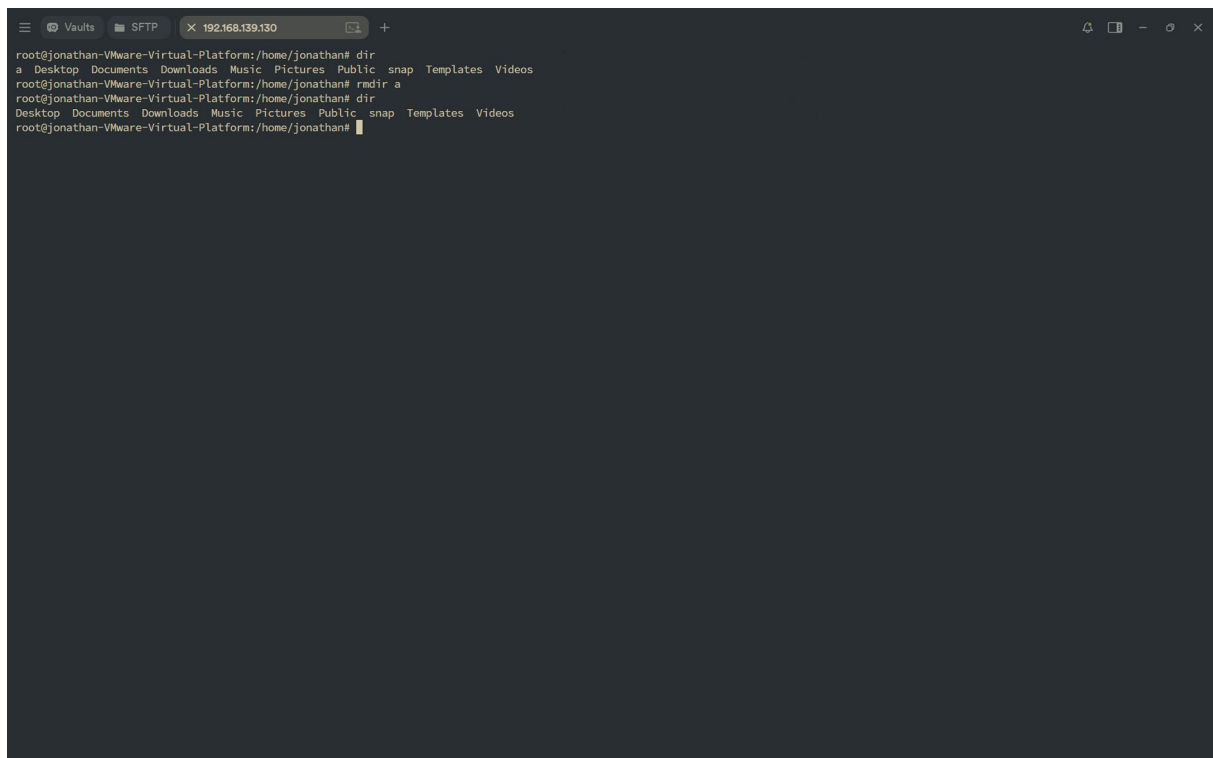
Student number: 560830

Assignment 6.1: Working from home

Screenshot installation openssh-server:



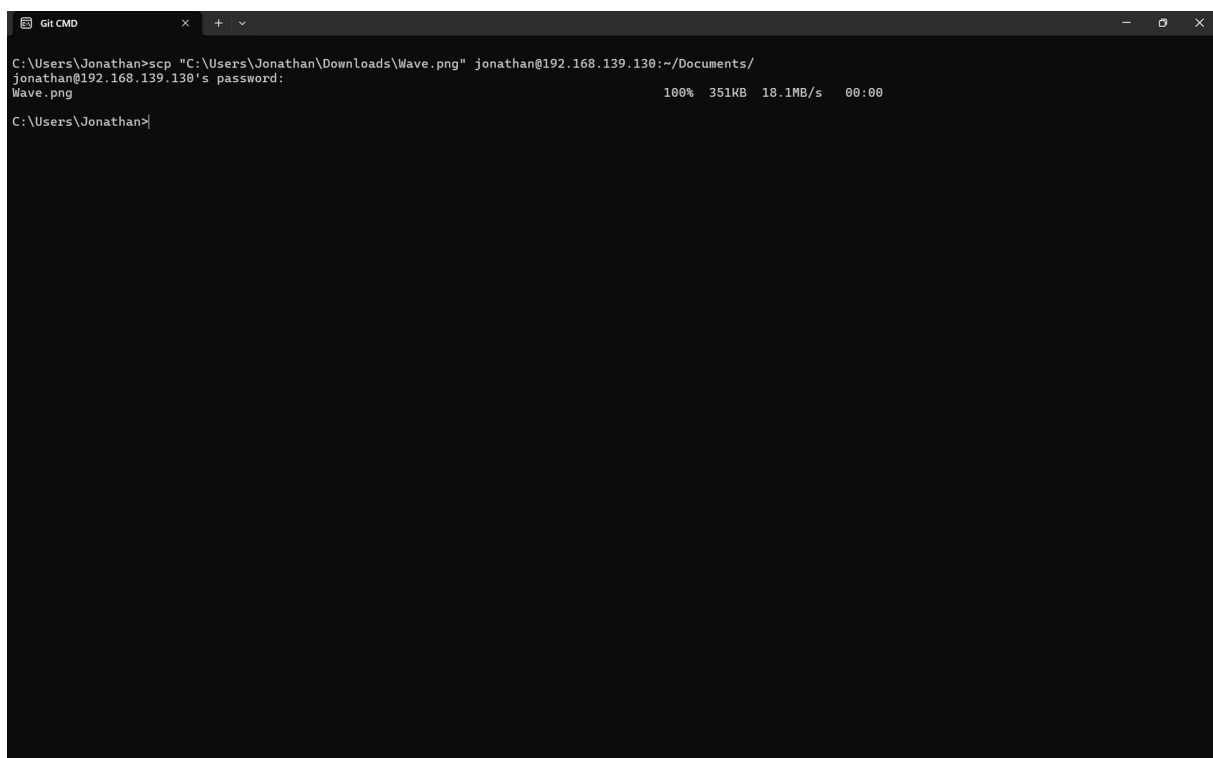
Screenshot successful SSH command execution:



```
root@jonathan-VMware-Virtual-Platform:/home/jonathan# dir
a Desktop Documents Downloads Music Pictures Public snap Templates Videos
root@jonathan-VMware-Virtual-Platform:/home/jonathan# rmdir a
root@jonathan-VMware-Virtual-Platform:/home/jonathan# dir
Desktop Documents Downloads Music Pictures Public snap Templates Videos
root@jonathan-VMware-Virtual-Platform:/home/jonathan#
```

(Successful deletion of a folder remotely)

Screenshot successful execution SCP command:

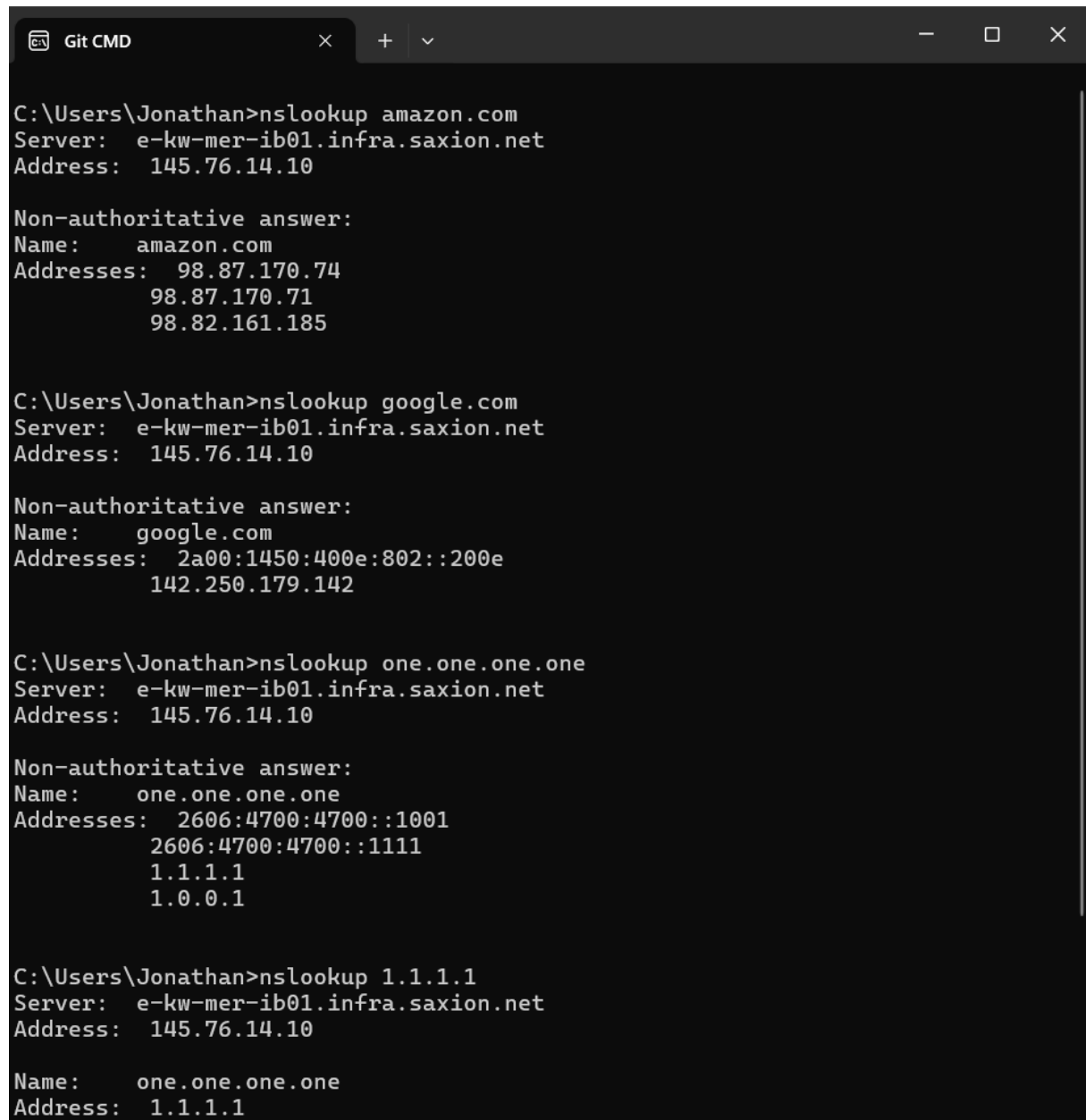


```
C:\Users\Jonathan>scp "C:\Users\Jonathan\Downloads\Wave.png" jonathan@192.168.139.130:~/Documents/
jonathan@192.168.139.130's password:
Wave.png                               100% 351KB 18.1MB/s 00:00
C:\Users\Jonathan>
```

Screenshot remmina:

Assignment 6.2: IP addresses websites

Relevant screenshots nslookup command:



```
Git CMD
C:\Users\Jonathan>nslookup amazon.com
Server: e-kw-mer-ib01.infra.saxion.net
Address: 145.76.14.10

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

C:\Users\Jonathan>nslookup google.com
Server: e-kw-mer-ib01.infra.saxion.net
Address: 145.76.14.10

Non-authoritative answer:
Name: google.com
Addresses: 2a00:1450:400e:802::200e
          142.250.179.142

C:\Users\Jonathan>nslookup one.one.one.one
Server: e-kw-mer-ib01.infra.saxion.net
Address: 145.76.14.10

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

C:\Users\Jonathan>nslookup 1.1.1.1
Server: e-kw-mer-ib01.infra.saxion.net
Address: 145.76.14.10

Name: one.one.one.one
Address: 1.1.1.1
```

```

C:\Users\Jonathan>nslookup dns.google.com
Server: e-kw-mer-ib01.infra.saxion.net
Address: 145.76.14.10

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

C:\Users\Jonathan>nslookup bol.com
Server: e-kw-mer-ib01.infra.saxion.net
Address: 145.76.14.10

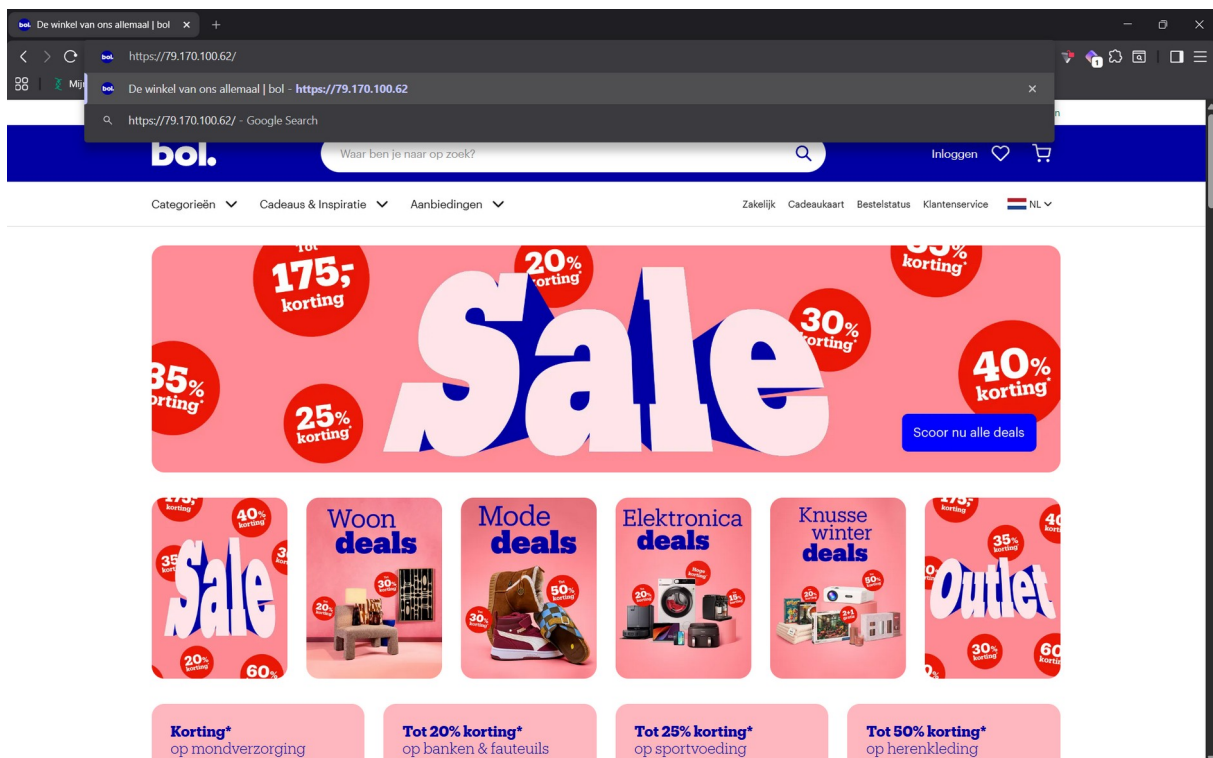
Non-authoritative answer:
Name: bol.com
Address: 79.170.100.62

C:\Users\Jonathan>nslookup w3schools.com
Server: e-kw-mer-ib01.infra.saxion.net
Address: 145.76.14.10

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

```

Screenshot website visit via IP address:



Assignment 6.3: subnetting

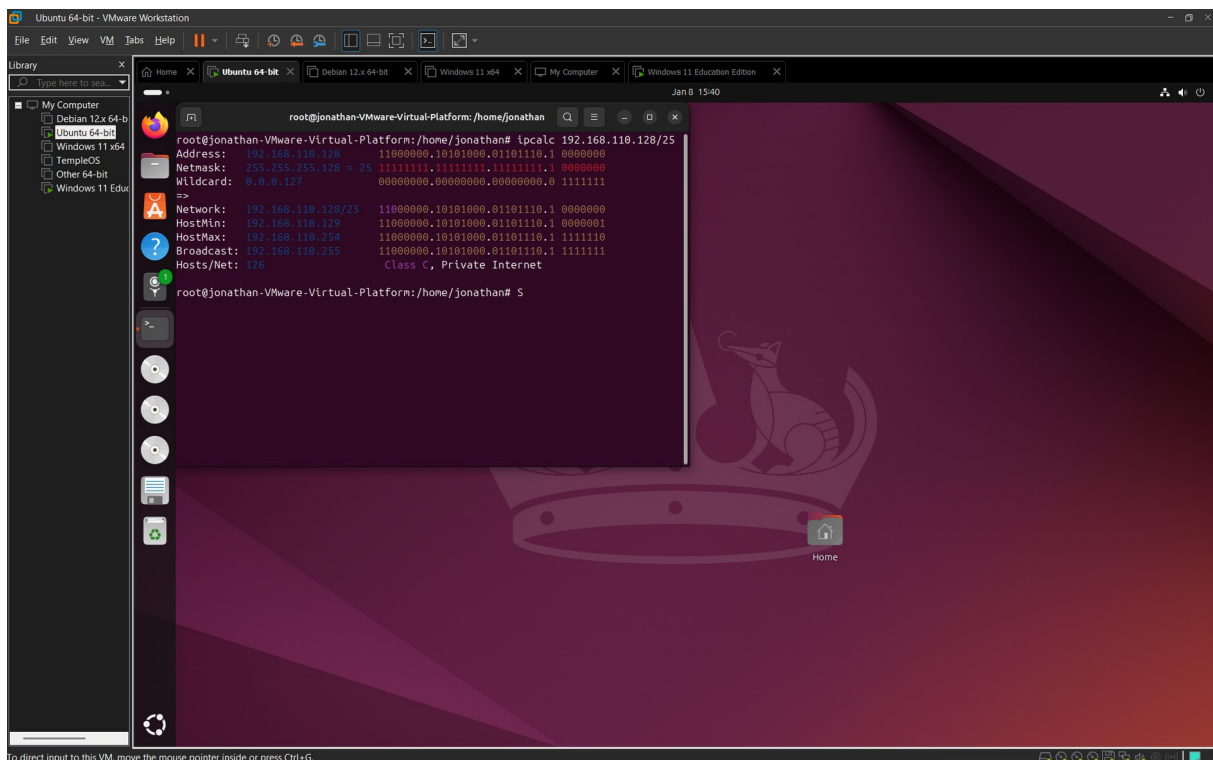
How many IP addresses are in this network configuration 192.168.110.128/25?

25 bits leaves 7 bits for host addresses, or 2^7 / 128 addresses.

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

Between 192.168.110.128 and 192.168.110.255, so the usable ips are 192.168.110.129 to 192.168.110.254

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



```
root@jonathan-Virtual-Platform:/home/jonathan# 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 00000000
Wildcard: 0.0.0.127        00000000.00000000.00000000.0 11111111
=>
Network: 192.168.110.128/25 11000000.10101000.01101110.1 00000000
HostMin: 192.168.110.129   11000000.10101000.01101110.1 00000001
HostMax: 192.168.110.254   11000000.10101000.01101110.1 11111110
Broadcast: 192.168.110.255 11000000.10101000.01101110.1 11111111
Hosts/Net: 126              Class C, Private Internet

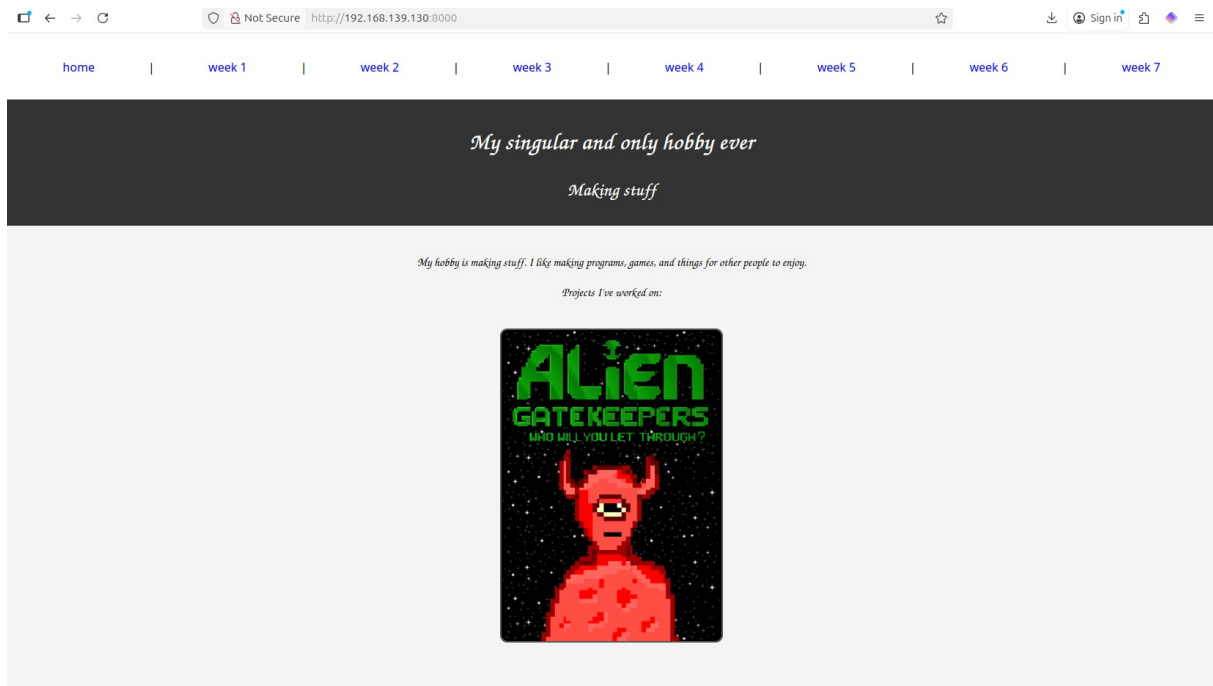
root@jonathan-Virtual-Platform:/home/jonathan#
```

Explain the above calculation in your own words.

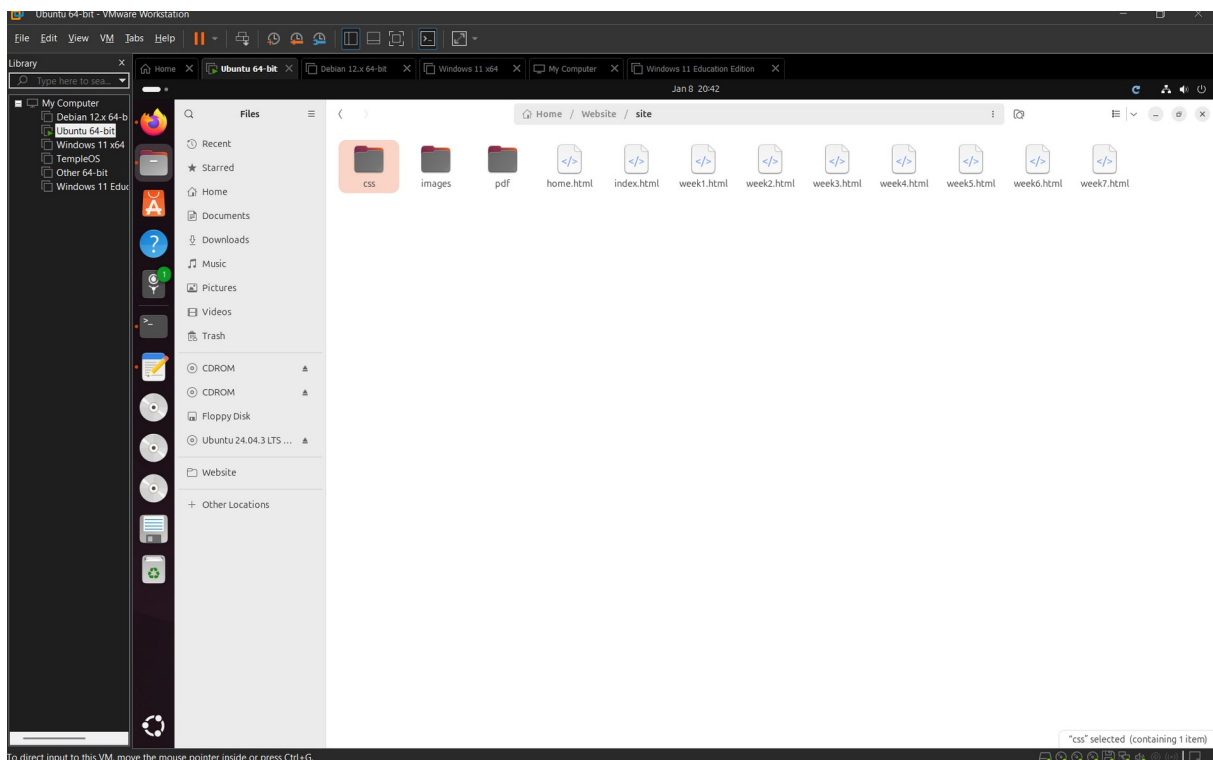
With /25 indicating the subnet, 25 bits are used for the network and 7 bits remain for hosts. This means 2^7 ip addresses with two being excluded for network and broadcast, leaving 126.

Assignment 6.4: HTML

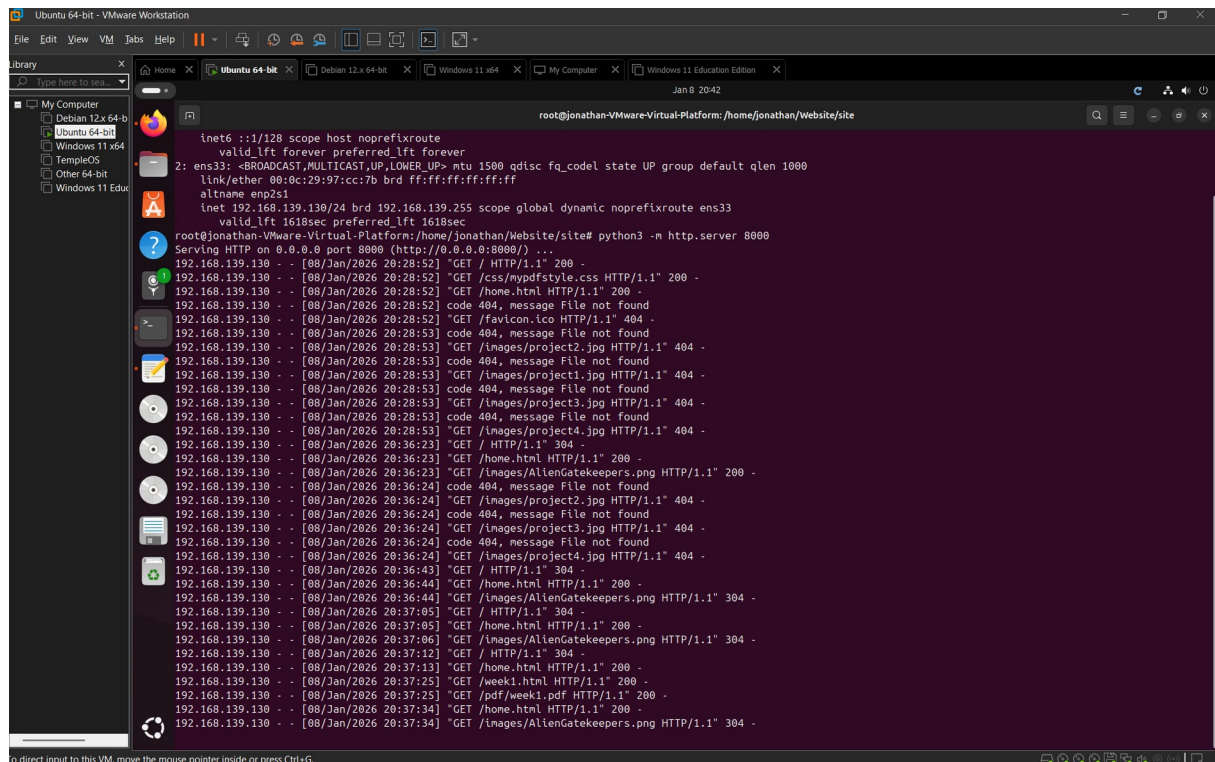
Screenshot IP address Ubuntu VM:



Screenshot of Site directory contents:

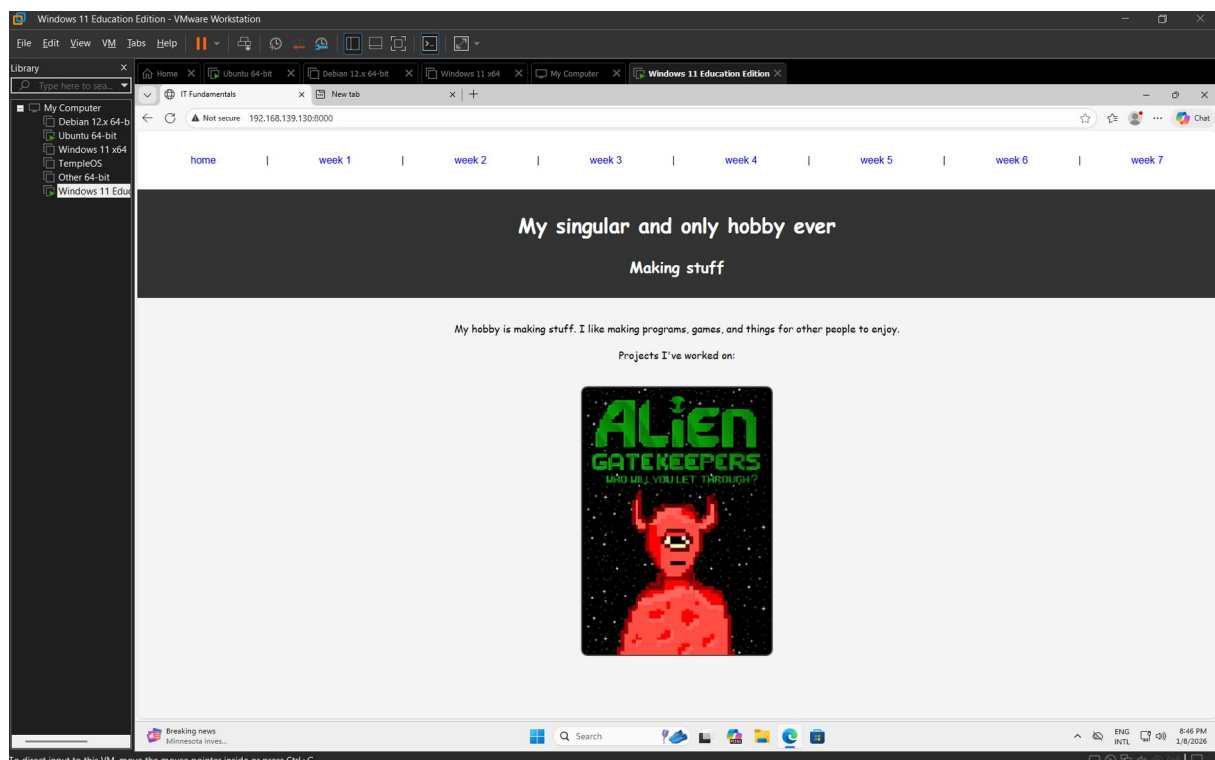


Screenshot python3 webserver command:



```
root@jonathan-Virtual-Platform:/home/jonathan/Website/site# python3 -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
192.168.139.130 - - [08/Jan/2026 20:28:52] "GET / HTTP/1.1" 200 -
192.168.139.130 - - [08/Jan/2026 20:28:52] "GET /css/mypdfstyle.css HTTP/1.1" 200 -
192.168.139.130 - - [08/Jan/2026 20:28:52] "GET /home.html HTTP/1.1" 200 -
192.168.139.130 - - [08/Jan/2026 20:28:52] "code 404, message File not found"
192.168.139.130 - - [08/Jan/2026 20:28:52] "GET /favicon.ico HTTP/1.1" 404 -
192.168.139.130 - - [08/Jan/2026 20:28:53] "code 404, message File not found"
192.168.139.130 - - [08/Jan/2026 20:28:53] "GET /images/project2.jpg HTTP/1.1" 404 -
192.168.139.130 - - [08/Jan/2026 20:28:53] "code 404, message File not found"
192.168.139.130 - - [08/Jan/2026 20:28:53] "GET /images/project1.jpg HTTP/1.1" 404 -
192.168.139.130 - - [08/Jan/2026 20:28:53] "code 404, message File not found"
192.168.139.130 - - [08/Jan/2026 20:28:53] "GET /images/project3.jpg HTTP/1.1" 404 -
192.168.139.130 - - [08/Jan/2026 20:28:53] "code 404, message File not found"
192.168.139.130 - - [08/Jan/2026 20:28:53] "GET /images/project4.jpg HTTP/1.1" 404 -
192.168.139.130 - - [08/Jan/2026 20:28:53] "code 404, message File not found"
192.168.139.130 - - [08/Jan/2026 20:28:53] "GET / HTTP/1.1" 304 -
192.168.139.130 - - [08/Jan/2026 20:36:23] "GET /home.html HTTP/1.1" 200 -
192.168.139.130 - - [08/Jan/2026 20:36:23] "GET /images/AlienGatekeepers.png HTTP/1.1" 200 -
192.168.139.130 - - [08/Jan/2026 20:36:24] "code 404, message File not found"
192.168.139.130 - - [08/Jan/2026 20:36:24] "GET /images/project2.jpg HTTP/1.1" 404 -
192.168.139.130 - - [08/Jan/2026 20:36:24] "code 404, message File not found"
192.168.139.130 - - [08/Jan/2026 20:36:24] "GET /images/project3.jpg HTTP/1.1" 404 -
192.168.139.130 - - [08/Jan/2026 20:36:24] "code 404, message File not found"
192.168.139.130 - - [08/Jan/2026 20:36:24] "GET /images/project4.jpg HTTP/1.1" 404 -
192.168.139.130 - - [08/Jan/2026 20:36:24] "code 404, message File not found"
192.168.139.130 - - [08/Jan/2026 20:36:43] "GET / HTTP/1.1" 304 -
192.168.139.130 - - [08/Jan/2026 20:36:44] "GET /home.html HTTP/1.1" 200 -
192.168.139.130 - - [08/Jan/2026 20:36:44] "GET /images/AlienGatekeepers.png HTTP/1.1" 304 -
192.168.139.130 - - [08/Jan/2026 20:37:05] "GET / HTTP/1.1" 304 -
192.168.139.130 - - [08/Jan/2026 20:37:05] "GET /home.html HTTP/1.1" 200 -
192.168.139.130 - - [08/Jan/2026 20:37:06] "GET /images/AlienGatekeepers.png HTTP/1.1" 304 -
192.168.139.130 - - [08/Jan/2026 20:37:12] "GET / HTTP/1.1" 304 -
192.168.139.130 - - [08/Jan/2026 20:37:13] "GET /home.html HTTP/1.1" 200 -
192.168.139.130 - - [08/Jan/2026 20:37:25] "GET /week1.html HTTP/1.1" 200 -
192.168.139.130 - - [08/Jan/2026 20:37:25] "GET /pdf/week1.pdf HTTP/1.1" 200 -
192.168.139.130 - - [08/Jan/2026 20:37:34] "GET /home.html HTTP/1.1" 200 -
192.168.139.130 - - [08/Jan/2026 20:37:34] "GET /images/AlienGatekeepers.png HTTP/1.1" 304 -
```

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 nl.saxion.app.SaxionApp;

public class Application implements Runnable {

    public void run() {

        while (true) {

            SaxionApp.clear();

            SaxionApp.println("This program has four modes:");

            SaxionApp.println("1. Check if a number is odd");

            SaxionApp.println("2. Check if a number is a power of two");

            SaxionApp.println("3. Check the two's complement of a number");

            SaxionApp.println("4. Calculate a network segment (IP address and subnet mask)");

            SaxionApp.print("\nWhich option do you pick: ");

            int pickedMode = SaxionApp.readInt();

            SaxionApp.clear();

            // Mode 1: Check if a number is odd

            if (pickedMode == 1) {

                SaxionApp.print("Input a number to check: ");

                int pickedNumber = SaxionApp.readInt();
```



```

        SaxionApp.clear();

        SaxionApp.print("The number \"" + pickedNumber + "\" is ");

        if (checkIfNumberIsOdd(pickedNumber)) {
            SaxionApp.println("odd.");
        } else {
            SaxionApp.println("even.");
        }

        SaxionApp.pause();
    }

    // Mode 2: Check if a number is a power of two
    else if (pickedMode == 2) {
        SaxionApp.print("Input a number to check: ");

        int pickedNumber = SaxionApp.readInt();

        SaxionApp.clear();
        SaxionApp.print("The number \"" + pickedNumber + "\" is ");

        if (checkIfNumberIsAPowerOfTwo(pickedNumber)) {
            SaxionApp.println("a power of two.");
        } else {
            SaxionApp.println("not a power of two.");
        }

        SaxionApp.pause();
    }

    // Mode 3: Two's complement
    else if (pickedMode == 3) {
        SaxionApp.print("Input a number to calculate the two's complement
of: ");

        int pickedNumber = SaxionApp.readInt();
    }

```

```

        SaxionApp.clear();

        SaxionApp.print("The two's complement of the number \"" +
pickedNumber + "\" is ");

        SaxionApp.println(~pickedNumber + 1);

        SaxionApp.print("Which, when flipped back, is: ");

        SaxionApp.println(~(~pickedNumber + 1) + 1);

        SaxionApp.pause();

    }

    // Mode 4: Calculate the network segment
    else if (pickedMode == 4) {

        SaxionApp.print("Input an IP address (e.g., 192.168.1.10): ");

        String ipAddress = SaxionApp.readString();

        SaxionApp.print("Input a subnet mask (e.g., 255.255.255.0): ");

        String subnetMask = SaxionApp.readString();

        SaxionApp.clear();

        String networkSegment = "";

        try {

            networkSegment = calculateNetworkSegment(ipAddress, subnetMask);

            SaxionApp.print("The network segment for IP address \"" +
ipAddress + "\" and subnet mask \"" + subnetMask + "\" is ");

            SaxionApp.println(networkSegment);

        } catch (IllegalArgumentException e) {

            SaxionApp.println("Error: Invalid input format. Please ensure
both inputs are valid IP addresses.");

        }

        SaxionApp.pause();

    }

}

}

public static void main(String[] args) {

    SaxionApp.start(new Application(), 1000, 300);

}

```

```

public static boolean checkIfNumberIsAPowerOfTwo(int numberToCheck) {
    return (numberToCheck > 0 && (numberToCheck & (numberToCheck - 1)) == 0);
}

public static boolean checkIfNumberIsOdd(int numberToCheck) {
    return (numberToCheck & 1) == 1;
}

public static String calculateNetworkSegment(String ipAddress, String
subnetMask) {
    String[] ipParts = ipAddress.split("\\.");
    String[] subnetParts = subnetMask.split("\\.");

    if (ipParts.length != 4 || subnetParts.length != 4) {
        throw new IllegalArgumentException("Invalid IP address or subnet mask
format.");
    }

    StringBuilder networkSegment = new StringBuilder();

    for (int i = 0; i < 4; i++) {
        int ipPart = Integer.parseInt(ipParts[i]);
        int subnetPart = Integer.parseInt(subnetParts[i]);
        networkSegment.append(ipPart & subnetPart);

        if (i < 3) {
            networkSegment.append(".");
        }
    }

    return networkSegment.toString();
}
}

```

```
Saxion Drawingboard
The network segment for IP address "192.168.1.100" and subnet mask "255.255.255.224" is 192.168.1.96

PRESS ANY KEY TO CONTINUE
```

The screenshot shows a terminal window with a dark background. The title bar at the top reads "Saxion Drawingboard". The main text area displays a message: "The network segment for IP address "192.168.1.100" and subnet mask "255.255.255.224" is 192.168.1.96". Below this message, there is a grey bar with the text "PRESS ANY KEY TO CONTINUE". On the right side of the terminal, there is a vertical scrollbar and some faint, partially visible text from the adjacent window, including "p", "(", "h", "c", and "h".