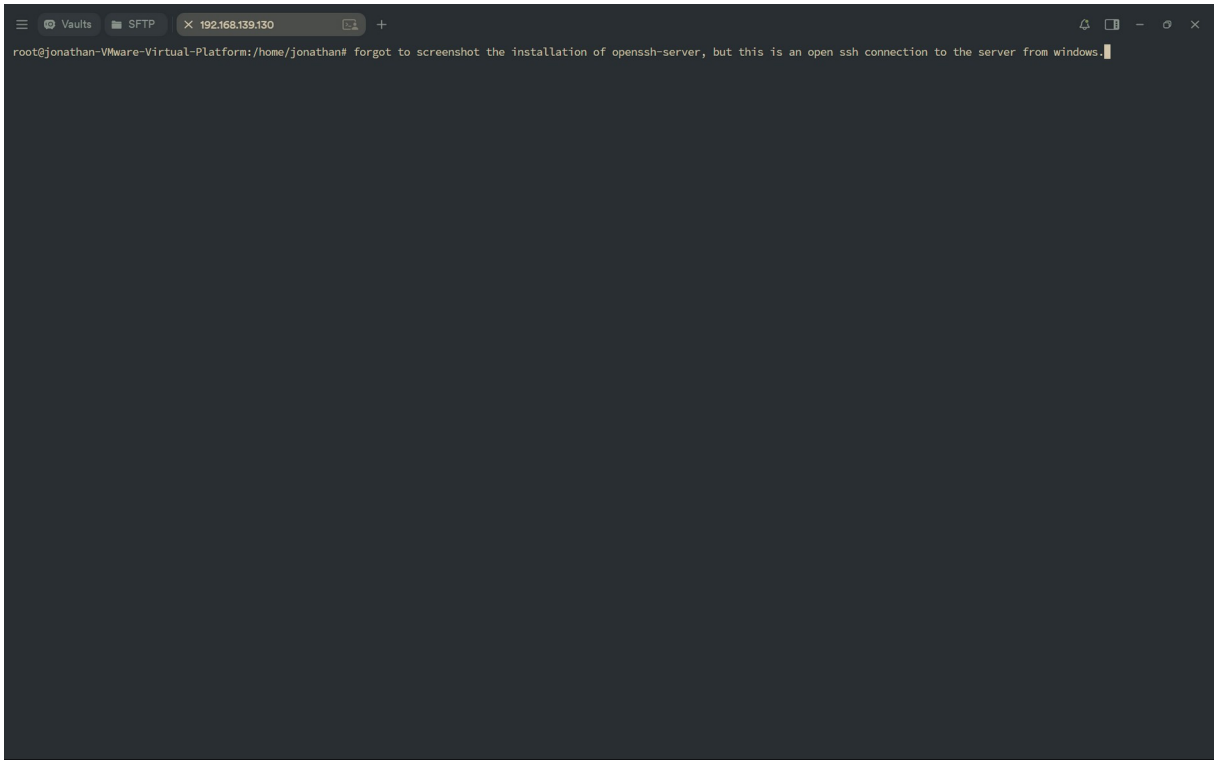
Template Week 6 – Networking

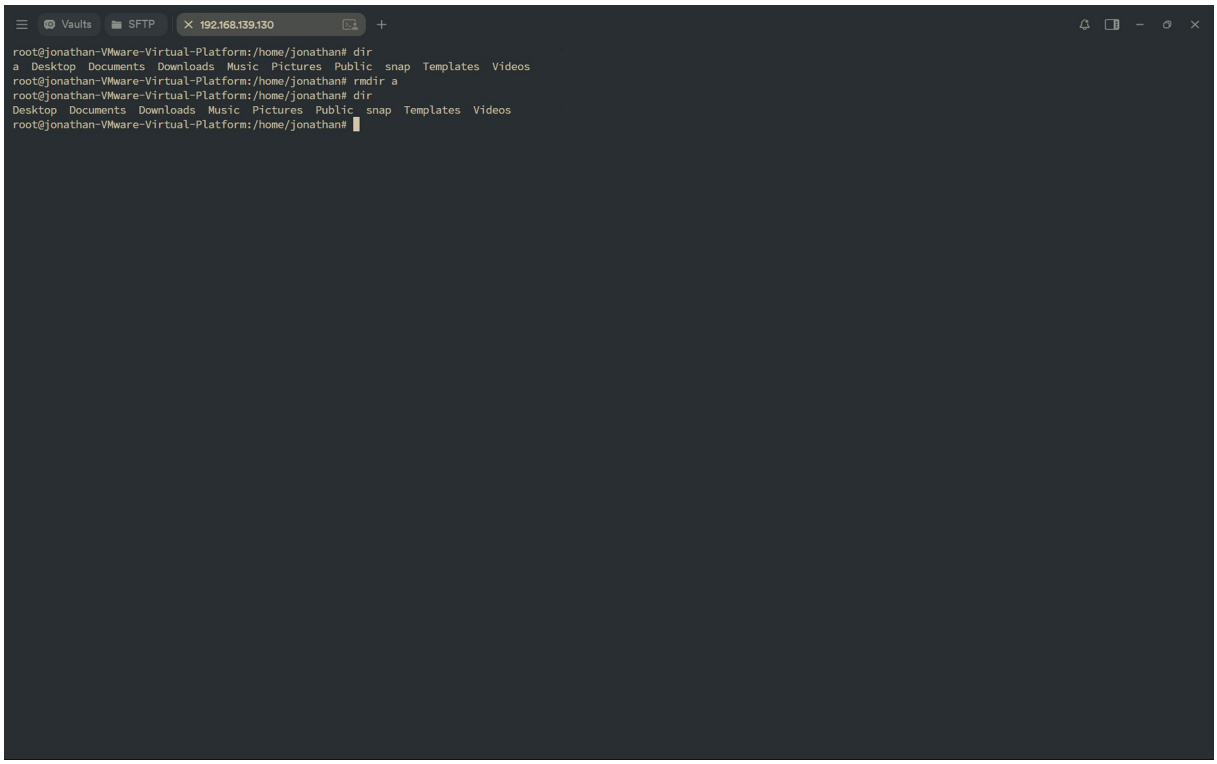
Student number: 560830

Assignment 6.1: Working from home

Screenshot installation openssh-server:

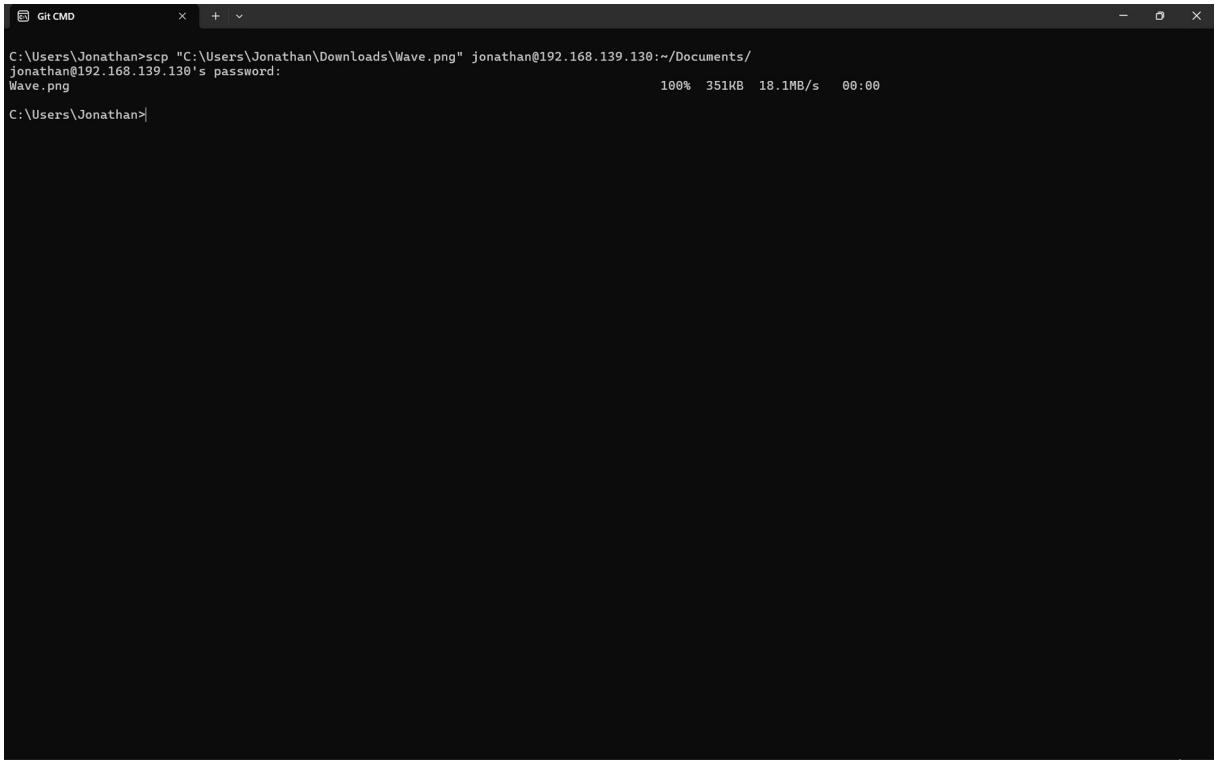


Screenshot successful SSH command execution:



(Succesful deletion of a folder remotely)

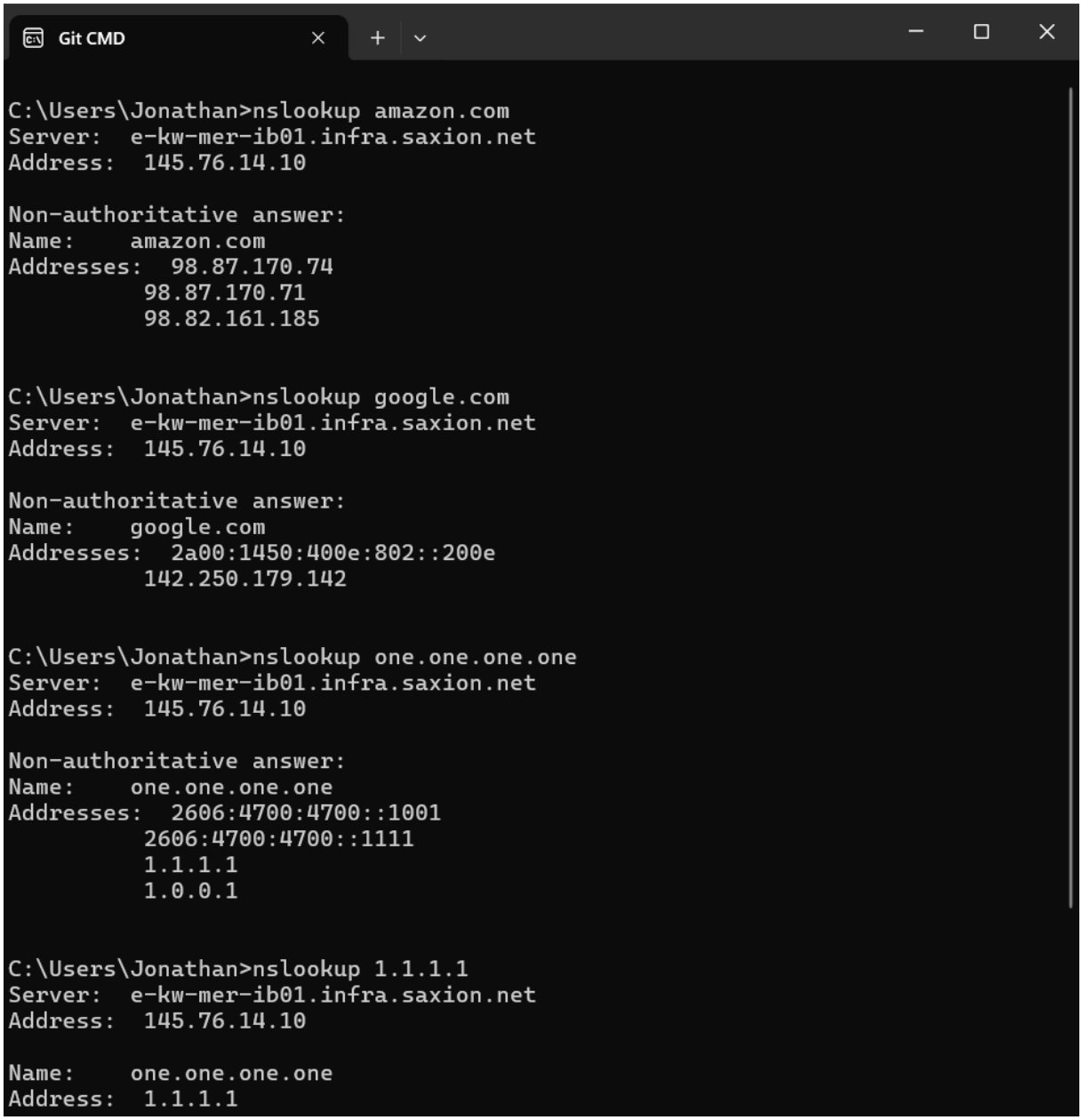
Screenshot successful execution SCP command:



Screenshot remmina:

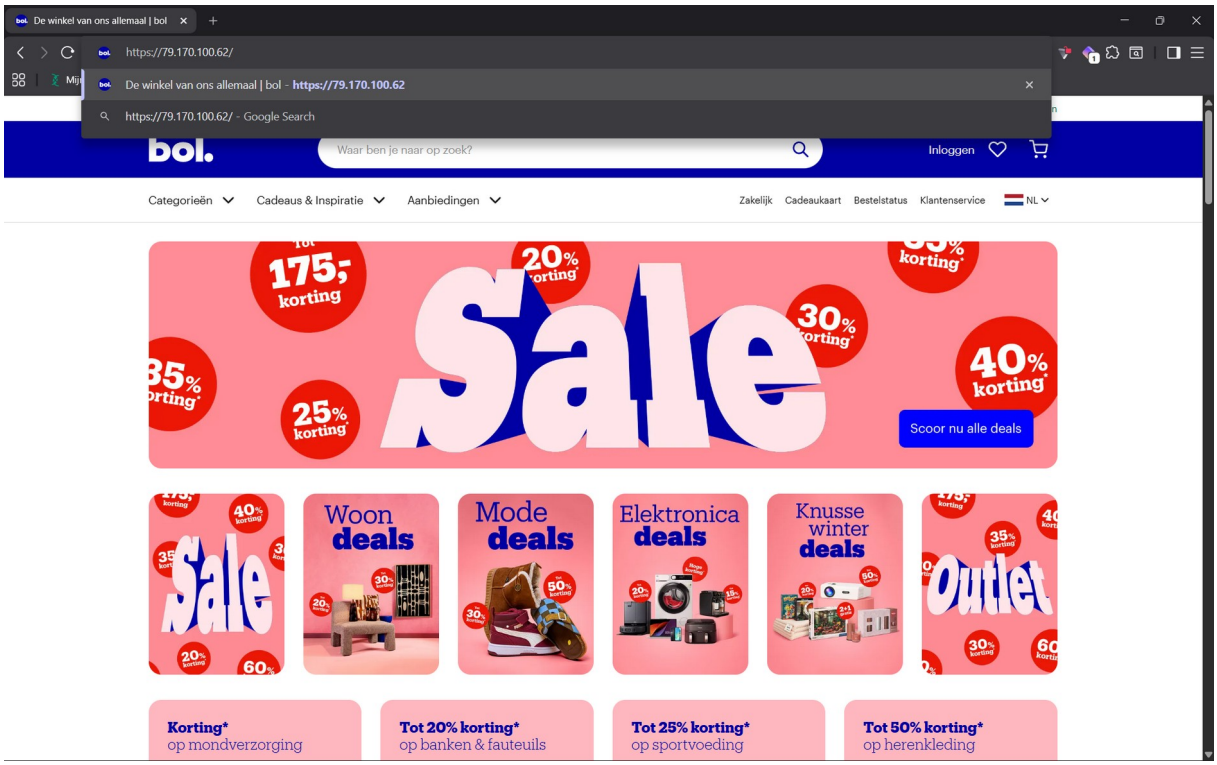
Assignment 6.2: IP addresses websites

Relevant screenshots nslookup command:



```
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

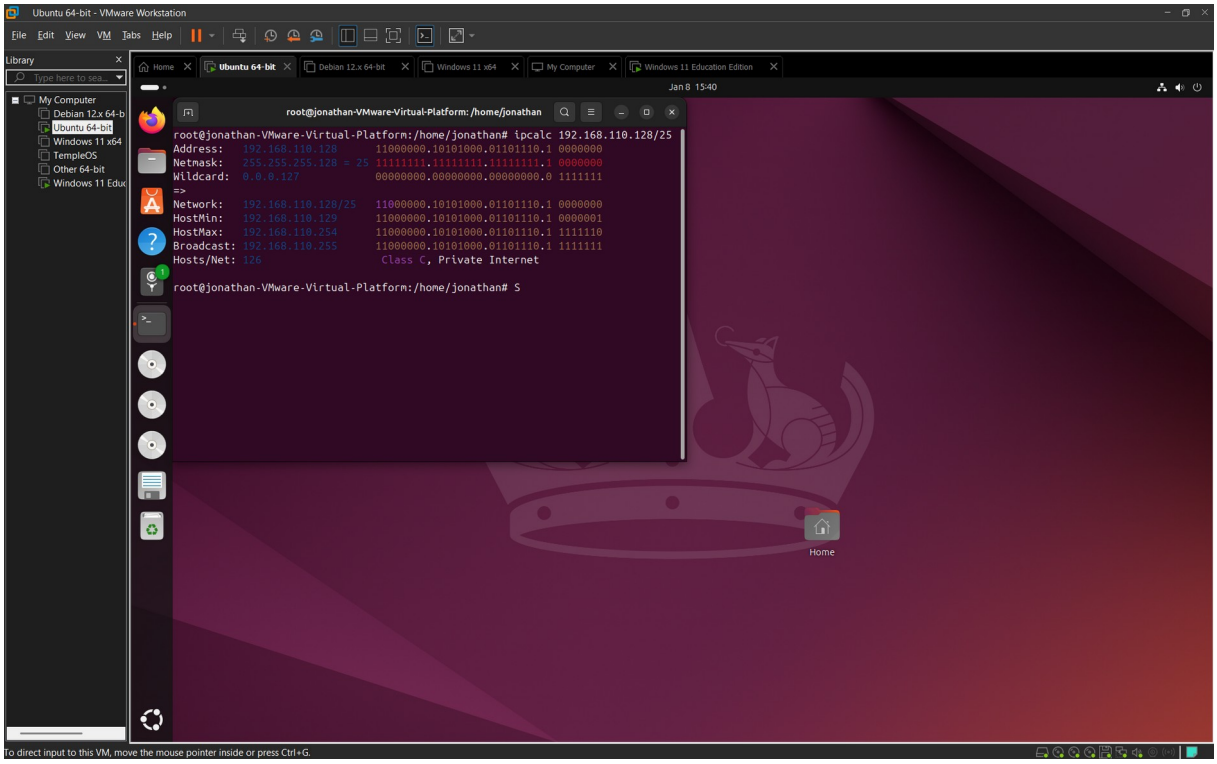
How many IP addresses are in this network configuration 192.168.110.128/25?

25 bits leaves 7 bits for host addresses, or 7^2 / 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 192.168.110.12**9** to 192.168.110.25**4**

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

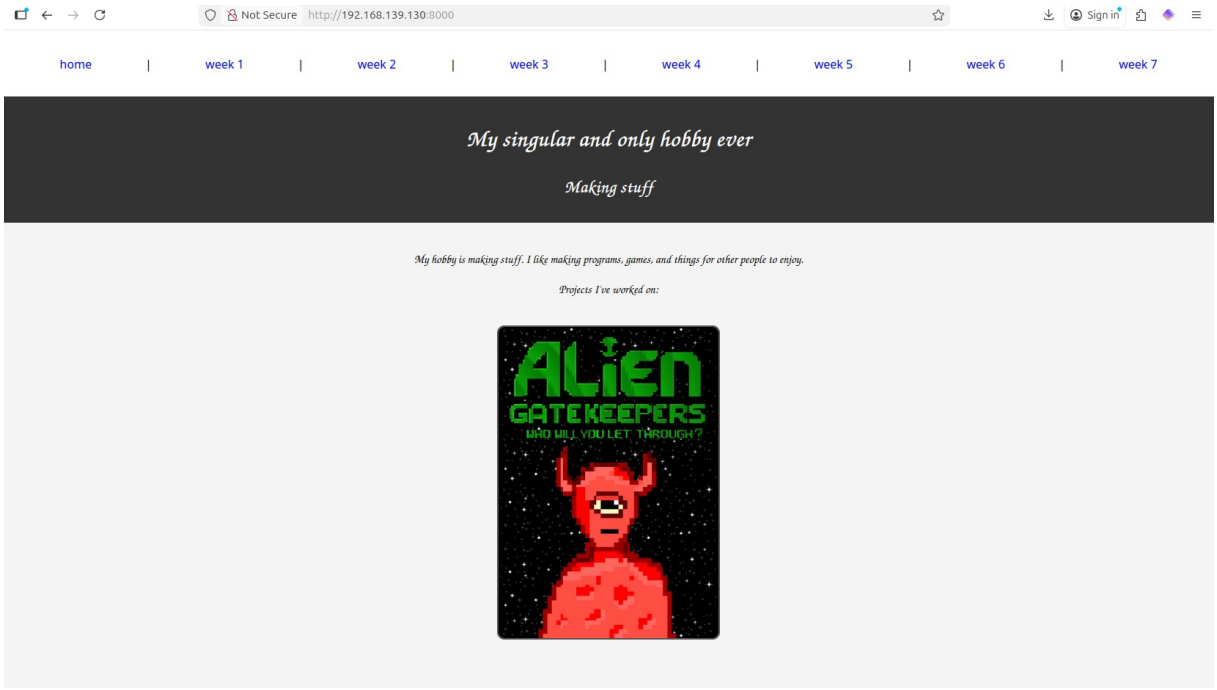


Explain the above calculation in your own words.

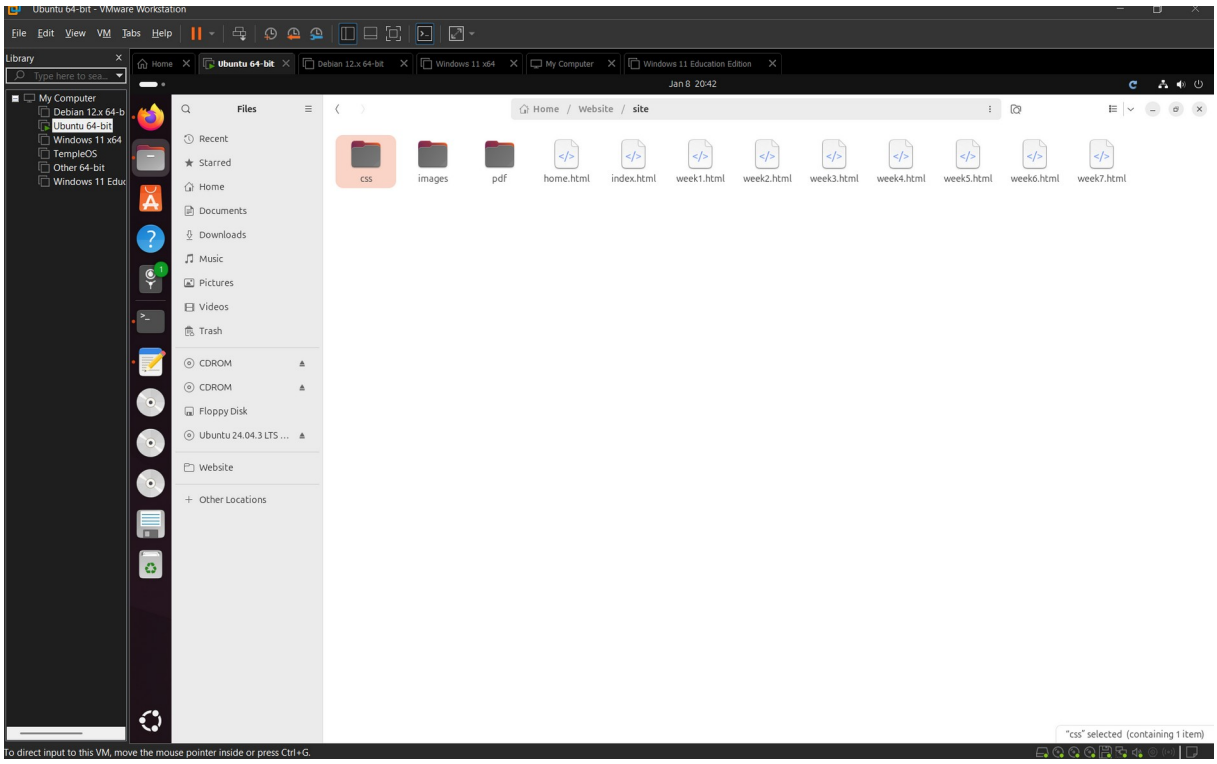
With /25 indicating the subnet, 25 bits are used for the network and 7 bits remain for hosts. This means 2^7 ip addresses with two being excluded for network and broadcast, leaving 126.

Assignment 6.4: HTML

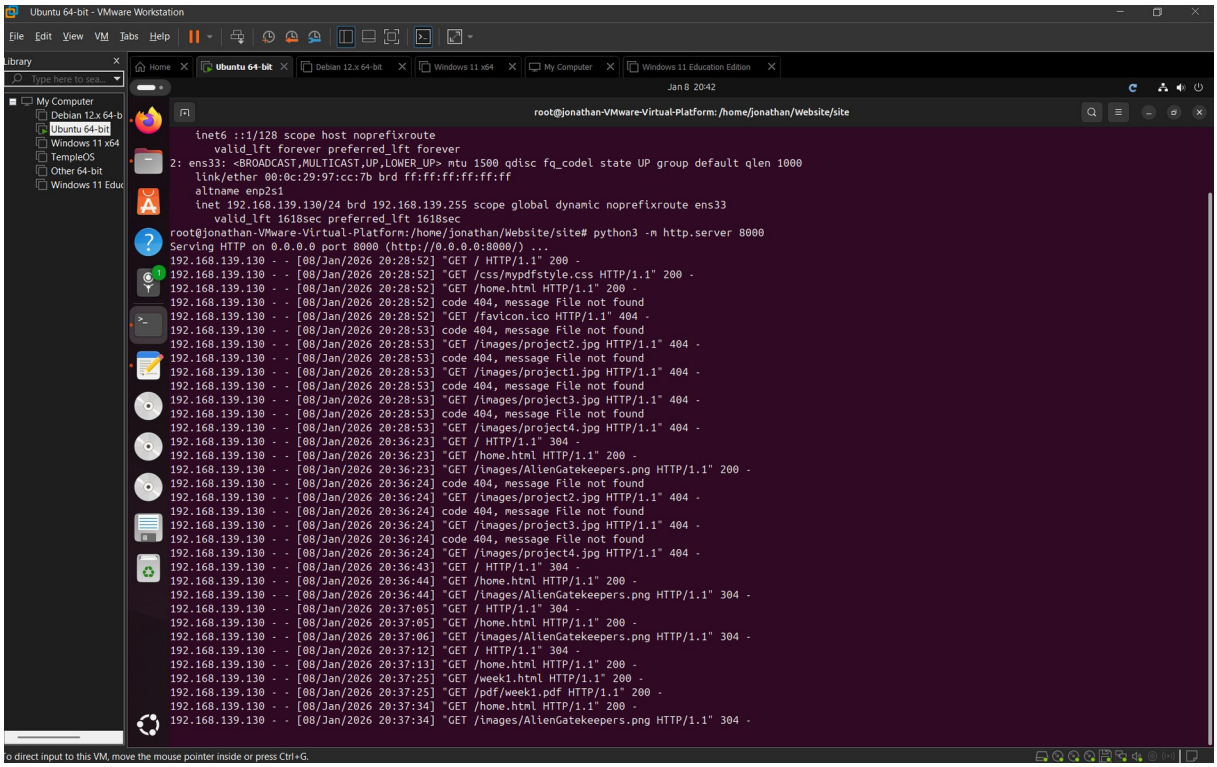
Screenshot IP address Ubuntu VM:



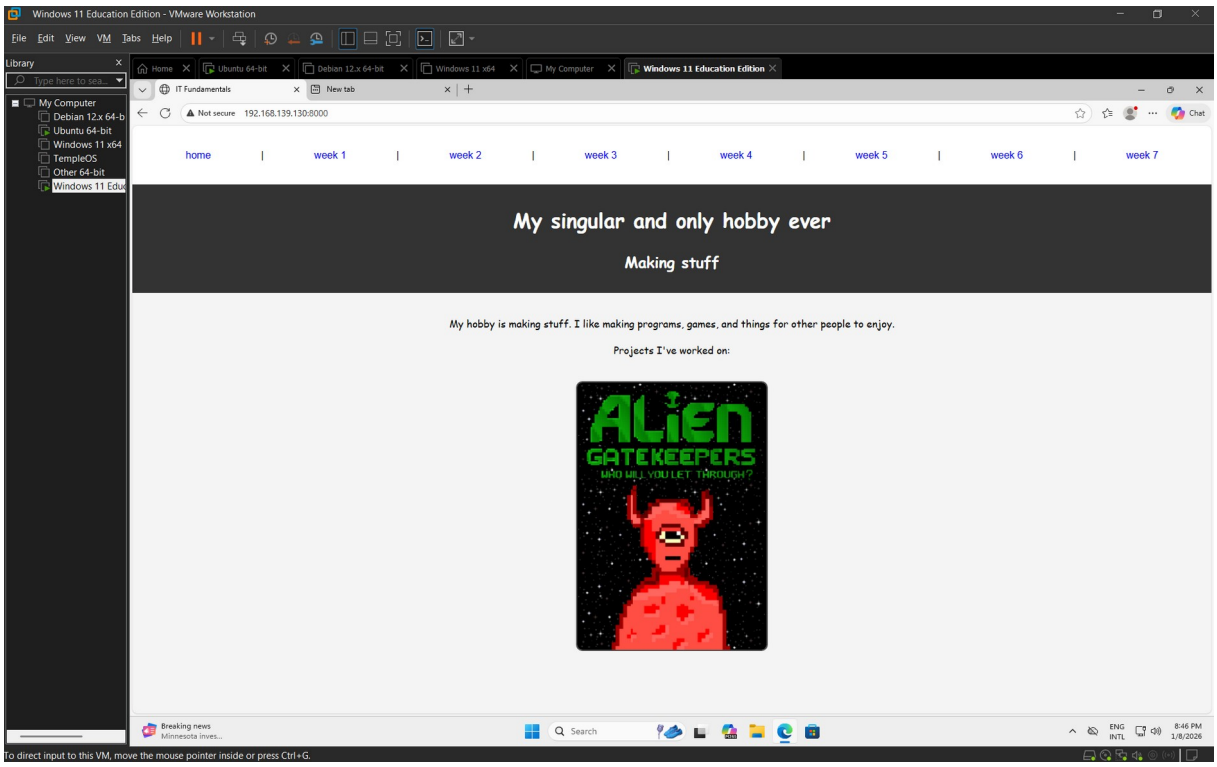
Screenshot of Site directory contents:



Screenshot python3 webserver command:



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.
```

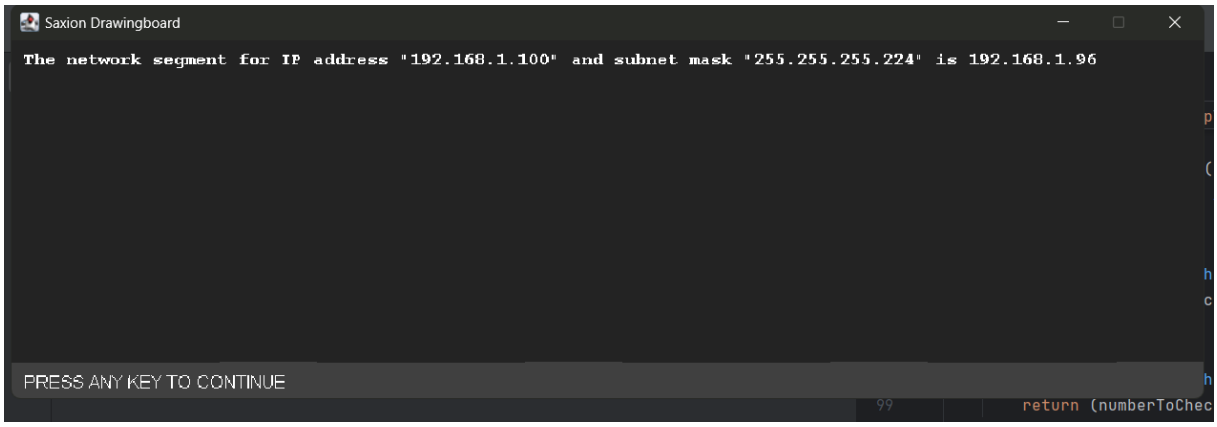
Paste source code here, with a screenshot of a working application.

```
import nl.saxion.app.SaxionApp;


public class Application implements Runnable {

   public void run() {

       while (true) {

           SaxionApp.clear();

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

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

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

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

           SaxionApp.printLine("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.printLine("odd.");
                } else {
                    SaxionApp.printLine("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.printLine("a power of two.");
                } else {
                    SaxionApp.printLine("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.printLine(~pickedNumber + 1);

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

                SaxionApp.printLine(~(~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.printLine(networkSegment);

                } catch (IllegalArgumentException e) {

                    SaxionApp.printLine("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
99
return (numberToChec