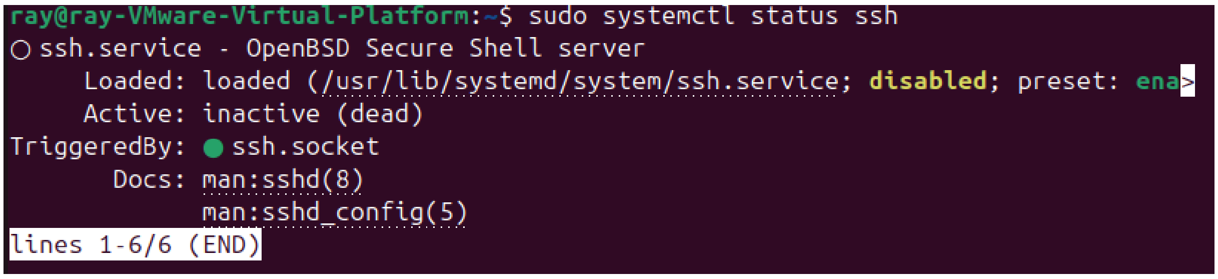
Template Week 6 – Networking

Student number:

Assignment 6.1: Working from home

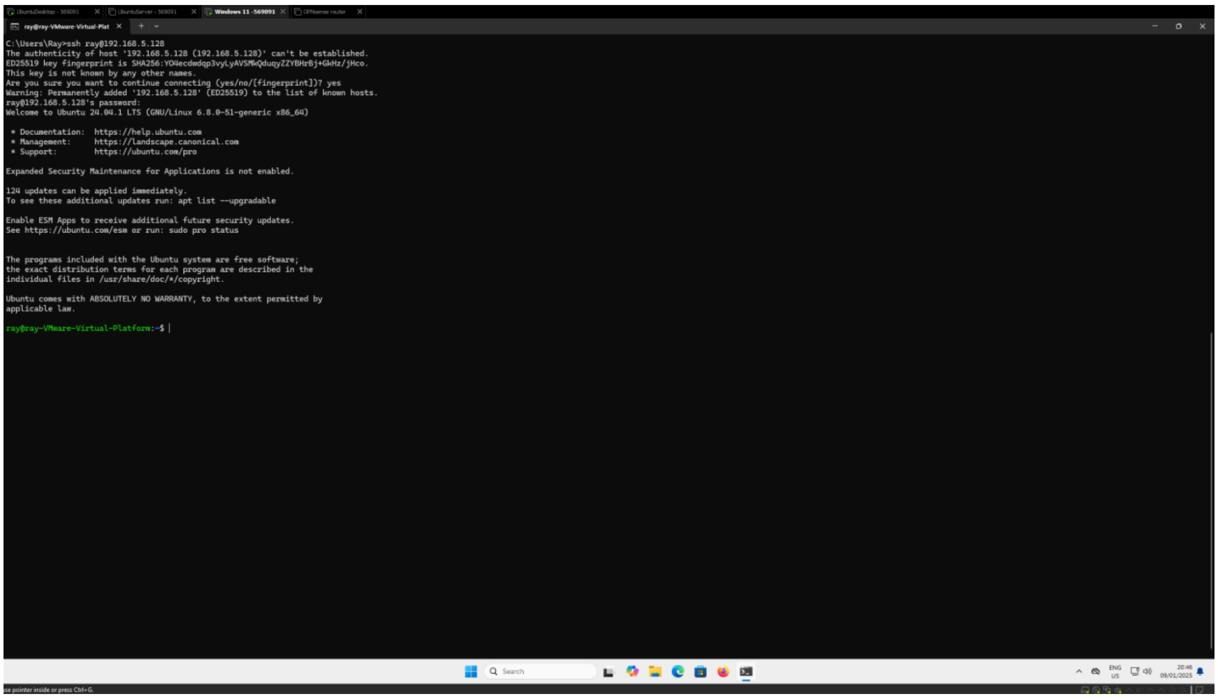
Screenshot installation openssh-server:

sudo apt install openssh-server



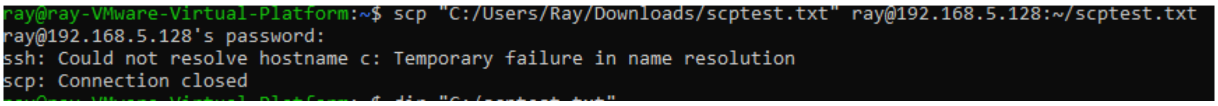
```
ray@ray-VMware-Virtual-Platform:~$ sudo systemctl status ssh
○ ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: ena>
     Active: inactive (dead)
TriggeredBy: ● ssh.socket
       Docs: man:sshd(8)
             man:sshd_config(5)
lines 1-6/6 (END)
```

Screenshot successful SSH command execution:



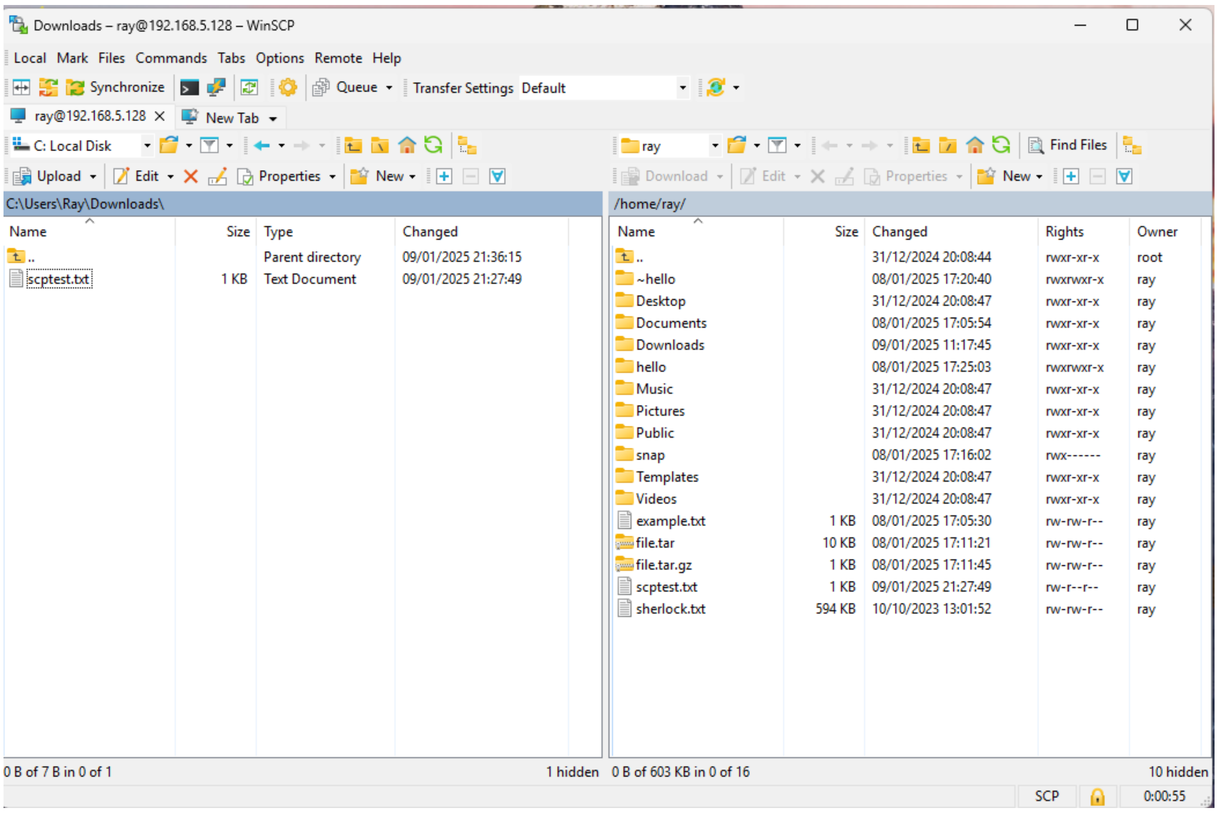
```
C:\Users\Ray>ssh ray@192.168.5.128
The authenticity of host '192.168.5.128 (192.168.5.128)' can't be established.
ED25519 key fingerprint is SHA256:YO4ecdwdqp3vyLyAVSMkQduqyZZYBHrBj+GkHz/jHco.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.5.128' (ED25519) to the list of known hosts.
ray@192.168.5.128's password:
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-51-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

124 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


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

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

ray@ray-VMware-Virtual-Platform:~$ |
```

Screenshot successful execution SCP command:



```
ray@ray-VMware-Virtual-Platform:~$ scp "C:/Users/Ray/Downloads/scptest.txt" ray@192.168.5.128:~/scptest.txt
ray@192.168.5.128's password:
ssh: Could not resolve hostname c: Temporary failure in name resolution
scp: Connection closed
```

Kept getting these errors for hours on end. Ended up using the application instead just do demonstrate that it works.


Downloads – ray@192.168.5.128 – WinSCP
Local Mark Files Commands Tabs Options Remote Help
Synchronize
Queue
Transfer Settings Default
ray@192.168.5.128
New Tab
C: Local Disk
Upload
Edit
Properties
New
ray
Download
Find Files

C:\Users\Ray\Downloads\

Name	Size	Type	Changed
..		Parent directory	09/01/2025 21:36:15
scptest.txt	1 KB	Text Document	09/01/2025 21:27:49

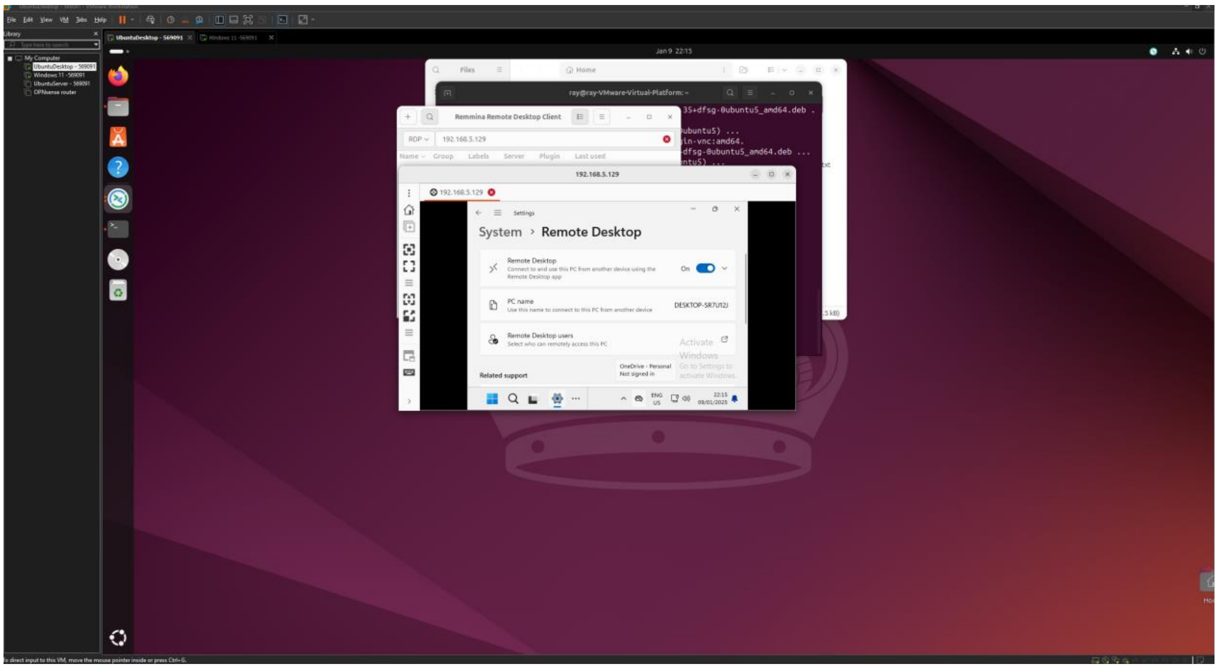
/home/ray/

Name	Size	Changed	Rights	Owner
..		31/12/2024 20:08:44	rwxr-xr-x	root
~hello		08/01/2025 17:20:40	rwxrwxr-x	ray
Desktop		31/12/2024 20:08:47	rwxr-xr-x	ray
Documents		08/01/2025 17:05:54	rwxr-xr-x	ray
Downloads		09/01/2025 11:17:45	rwxr-xr-x	ray
hello		08/01/2025 17:25:03	rwxrwxr-x	ray
Music		31/12/2024 20:08:47	rwxr-xr-x	ray
Pictures		31/12/2024 20:08:47	rwxr-xr-x	ray
Public		31/12/2024 20:08:47	rwxr-xr-x	ray
snap		08/01/2025 17:16:02	rwx------	ray
Templates		31/12/2024 20:08:47	rwxr-xr-x	ray
Videos		31/12/2024 20:08:47	rwxr-xr-x	ray
example.txt	1 KB	08/01/2025 17:05:30	rw-rw-r--	ray
file.tar	10 KB	08/01/2025 17:11:21	rw-rw-r--	ray
file.tar.gz	1 KB	08/01/2025 17:11:45	rw-rw-r--	ray
scptest.txt	1 KB	09/01/2025 21:27:49	rw-r--r--	ray
sherlock.txt	594 KB	10/10/2023 13:01:52	rw-rw-r--	ray

0 B of 7 B in 0 of 1 1 hidden 0 B of 603 KB in 0 of 16 10 hidden

SCP 0:00:55

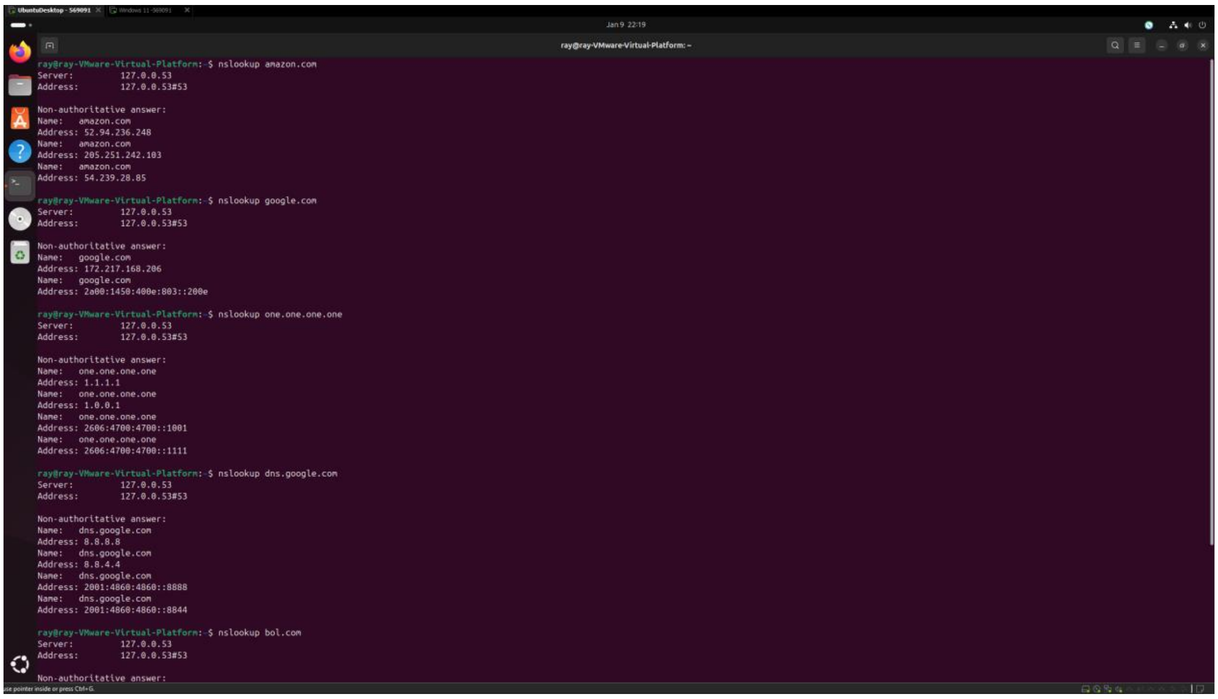
Screenshot remmina:


Jan 9 22:15
Files
Home
ray@ray-VMware-Virtual-Platform: ~
Remmina Remote Desktop Client
RDP
192.168.5.129
Name Group Labels Server Plugin Last used
192.168.5.129
Settings
System › Remote Desktop
Remote Desktop
Connect to and use this PC from another device using the Remote Desktop app
On
PC name
Use this name to connect to this PC from another device
DESKTOP-SR7U12J
Remote Desktop users
Select who can remotely access this PC
Related support
OneDrive - Personal
Not signed in
Activate Windows
Go to Settings to activate Windows.
ENG US
22:15
09/01/2025

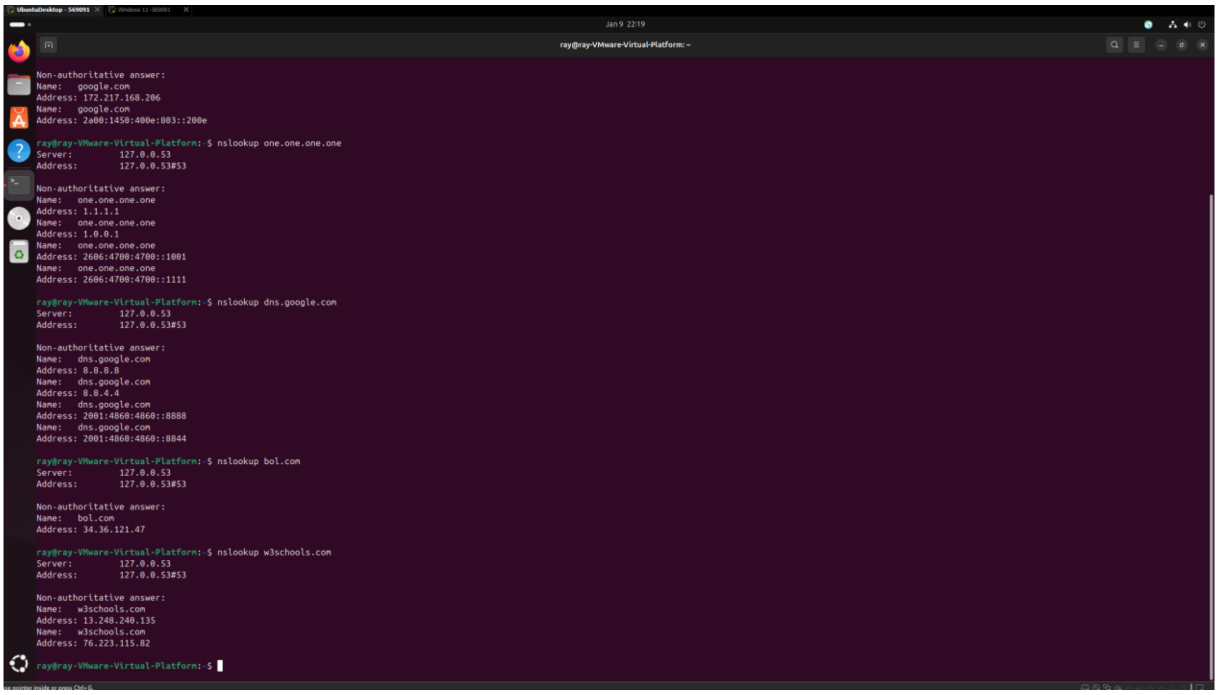
```
35+dfsg-0ubuntu5_amd64.deb .
ubuntu5) ...
in-vnc:amd64.
-dfsg-0ubuntu5_amd64.deb ...
ntu5) ...
```

Assignment 6.2: IP addresses websites

Relevant screenshots nslookup command:



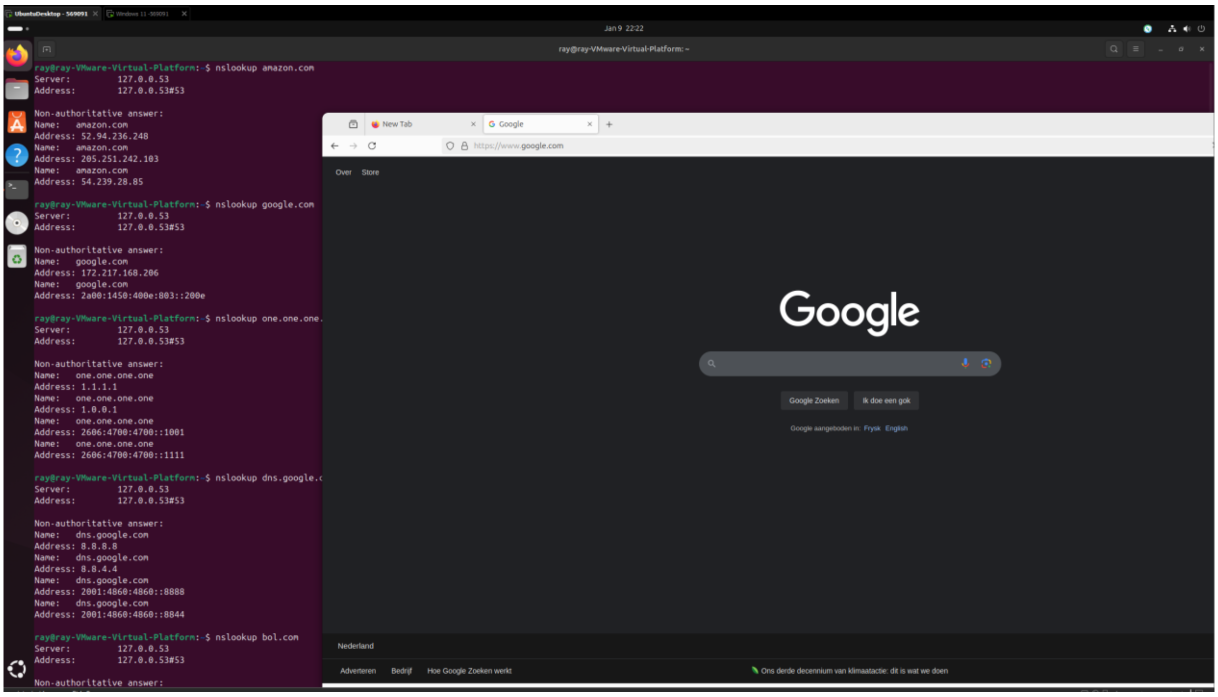
```
ray@ray-VMware-Virtual-Platform:~$ nslookup amazon.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   amazon.com
Address: 52.94.236.248
Name:   amazon.com
Address: 205.251.242.103
Name:   amazon.com
Address: 54.239.28.85

ray@ray-VMware-Virtual-Platform:~$ nslookup google.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   google.com
Address: 172.217.168.206
Name:   google.com
Address: 2a00:1450:400e:803::200e

ray@ray-VMware-Virtual-Platform:~$ nslookup one.one.one.one
Server:         127.0.0.53
Address:        127.0.0.53#53

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

ray@ray-VMware-Virtual-Platform:~$ nslookup dns.google.com
Server:         127.0.0.53
Address:        127.0.0.53#53

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

ray@ray-VMware-Virtual-Platform:~$ nslookup bol.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
```



```
Non-authoritative answer:
Name:   google.com
Address: 172.217.168.206
Name:   google.com
Address: 2a00:1450:400e:803::200e

ray@ray-VMware-Virtual-Platform:~$ nslookup one.one.one.one
Server:         127.0.0.53
Address:        127.0.0.53#53

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

ray@ray-VMware-Virtual-Platform:~$ nslookup dns.google.com
Server:         127.0.0.53
Address:        127.0.0.53#53

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

ray@ray-VMware-Virtual-Platform:~$ nslookup bol.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   bol.com
Address: 34.36.121.47

ray@ray-VMware-Virtual-Platform:~$ nslookup w3schools.com
Server:         127.0.0.53
Address:        127.0.0.53#53

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

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

Screenshot website visit via IP address:

Used the google IP address 172.217.168.206



```
ray@ray-VMware-Virtual-Platform:~$ nslookup amazon.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   amazon.com
Address: 52.94.236.248
Name:   amazon.com
Address: 205.251.242.103
Name:   amazon.com
Address: 54.239.28.85

ray@ray-VMware-Virtual-Platform:~$ nslookup google.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   google.com
Address: 172.217.168.206
Name:   google.com
Address: 2a00:1450:400e:803::200e

ray@ray-VMware-Virtual-Platform:~$ nslookup one.one.one.
Server:         127.0.0.53
Address:        127.0.0.53#53

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

ray@ray-VMware-Virtual-Platform:~$ nslookup dns.google.c
Server:         127.0.0.53
Address:        127.0.0.53#53

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

ray@ray-VMware-Virtual-Platform:~$ nslookup bol.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
```

Assignment 6.3: subnetting

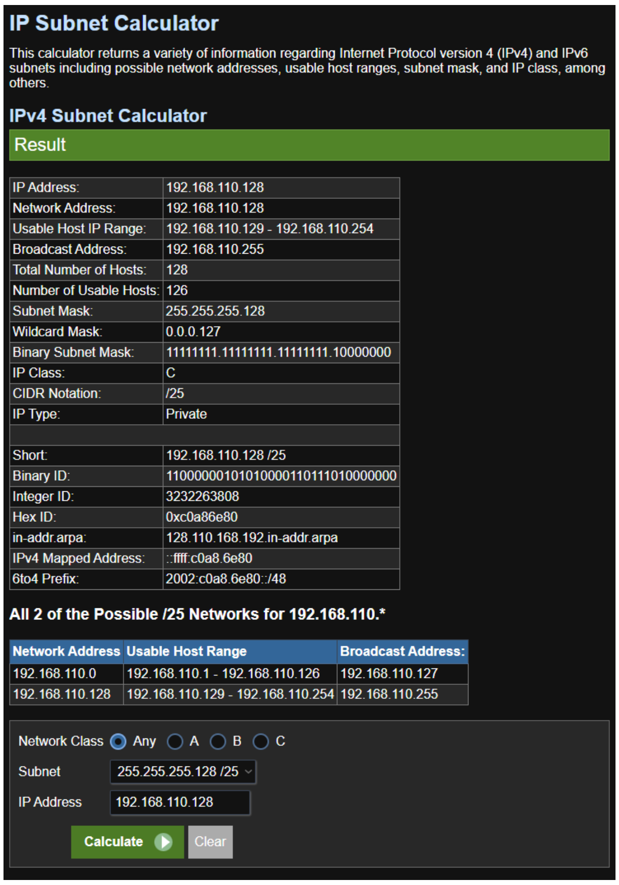
How many IP addresses are in this network configuration 192.168.110.128/25?

A /25 subnet has 128 IP addresses.

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

The usable range for the 192.168.110.128/25 subnet is from 192.168.110.129 to 192.168.110.254.

Check your two previous answers with this calculator:
https://www.calculator.net/ip-subnet-calculator.html



IP Subnet Calculator

This calculator returns a variety of information regarding Internet Protocol version 4 (IPv4) and IPv6 subnets including possible network addresses, usable host ranges, subnet mask, and IP class, among others.

IPv4 Subnet Calculator

Result

IP Address:	192.168.110.128
Network Address:	192.168.110.128
Usable Host IP Range:	192.168.110.129 - 192.168.110.254
Broadcast Address:	192.168.110.255
Total Number of Hosts:	128
Number of Usable Hosts:	126
Subnet Mask:	255.255.255.128
Wildcard Mask:	0.0.0.127
Binary Subnet Mask:	11111111.11111111.11111111.10000000
IP Class:	C
CIDR Notation:	/25
IP Type:	Private
Short:	192.168.110.128 /25
Binary ID:	11000000101010000110111010000000
Integer ID:	3232263808
Hex ID:	0xc0a86e80
in-addr.arpa:	128.110.168.192.in-addr.arpa
IPv4 Mapped Address:	::ffff:c0a8.6e80
6to4 Prefix:	2002:c0a8.6e80::/48

All 2 of the Possible /25 Networks for 192.168.110.*

Network Address	Usable Host Range	Broadcast Address:
192.168.110.0	192.168.110.1 - 192.168.110.126	192.168.110.127
192.168.110.128	192.168.110.129 - 192.168.110.254	192.168.110.255

Network Class [✓ Any] A B C

Subnet: 255.255.255.128 /25

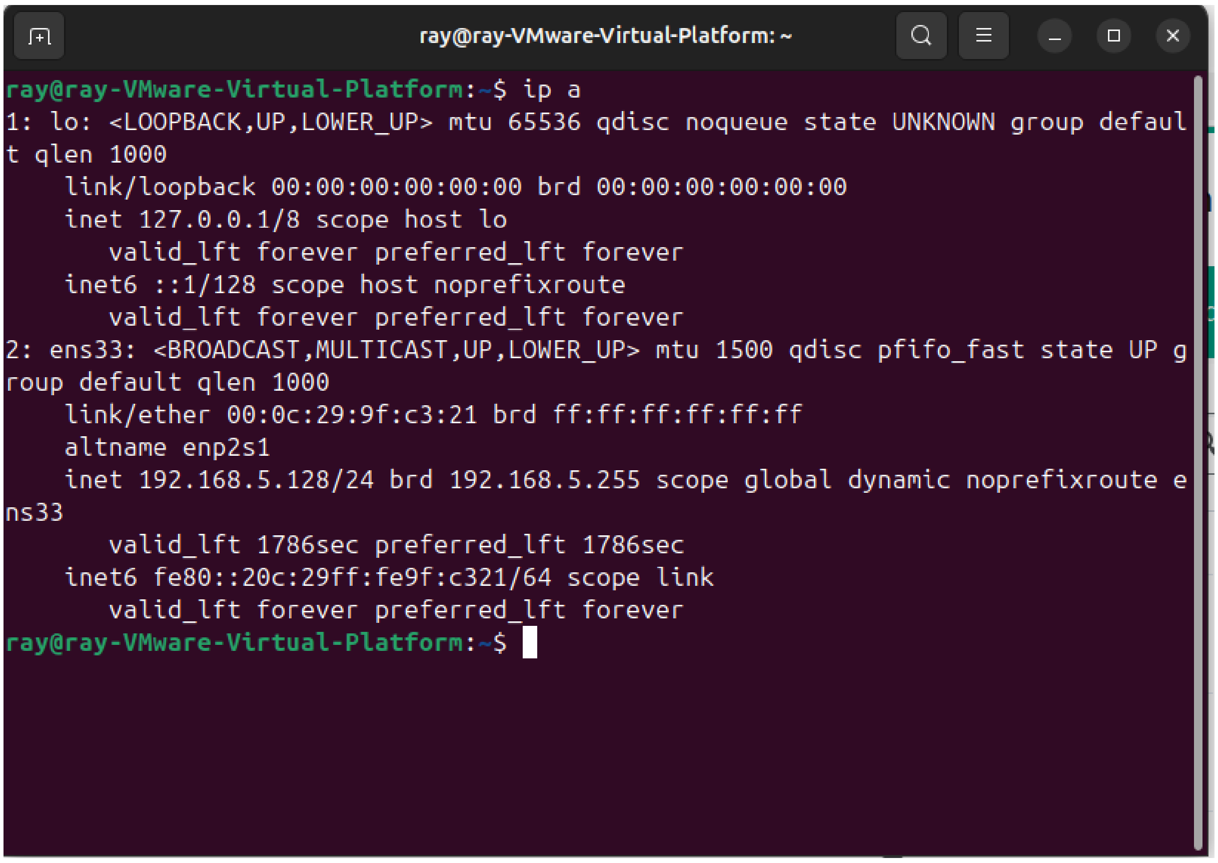
IP Address: 192.168.110.128

Explain the above calculation in your own words.

A /25 subnet mask splits the IP range into two parts: the network and the usable IPs. The first 128 IPs belong to the network (192.168.110.0 to 192.168.110.127), and the remaining 128 IPs are usable by devices (192.168.110.128 to 192.168.110.254).

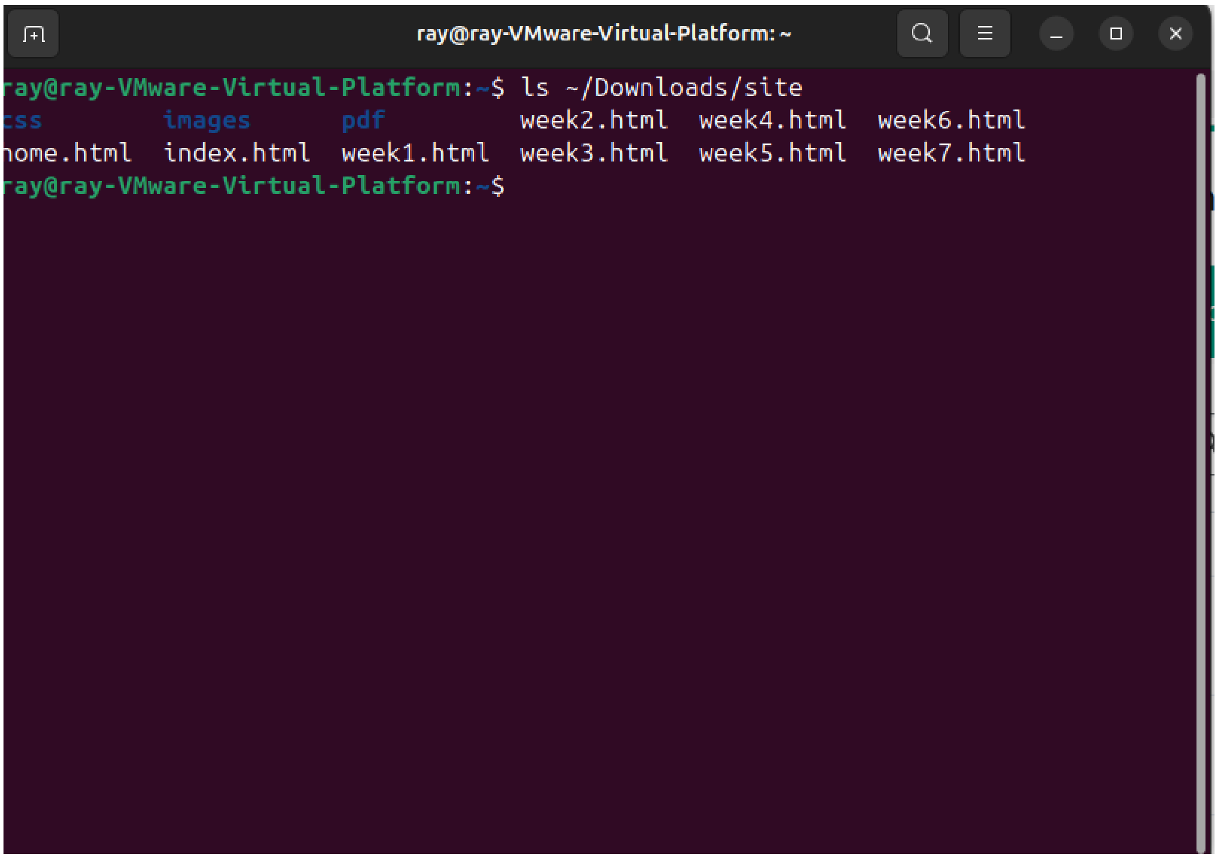
Assignment 6.4: HTML

Screenshot IP address Ubuntu VM:



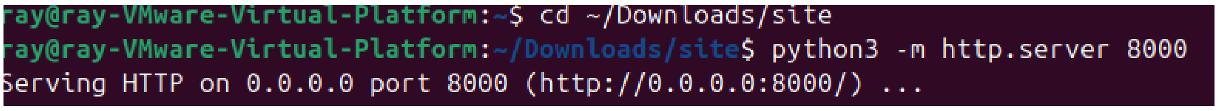
```
ray@ray-VMware-Virtual-Platform:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP g
roup default qlen 1000
    link/ether 00:0c:29:9f:c3:21 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.5.128/24 brd 192.168.5.255 scope global dynamic noprefixroute e
ns33
       valid_lft 1786sec preferred_lft 1786sec
    inet6 fe80::20c:29ff:fe9f:c321/64 scope link
       valid_lft forever preferred_lft forever
ray@ray-VMware-Virtual-Platform:~$
```

Screenshot of Site directory contents:



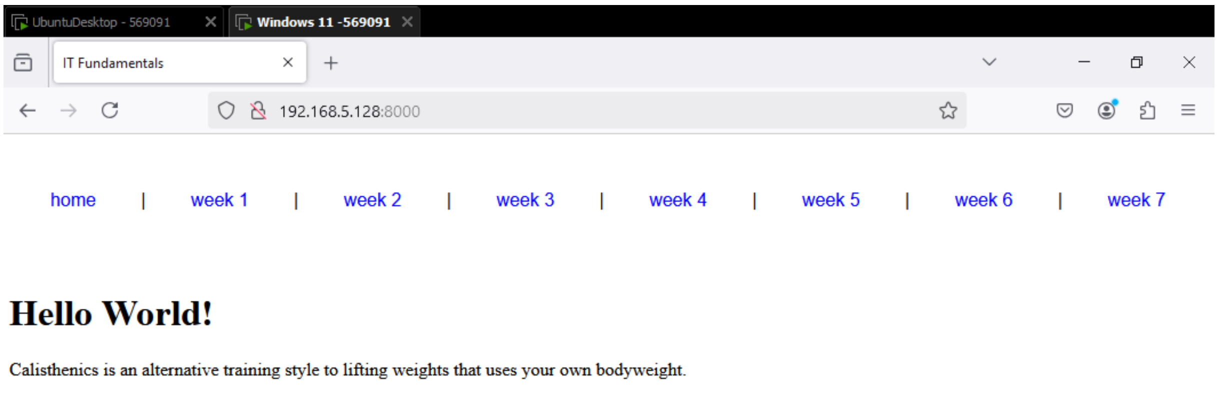
```
ray@ray-VMware-Virtual-Platform:~$ ls ~/Downloads/site
css         images      pdf         week2.html  week4.html  week6.html
home.html   index.html  week1.html  week3.html  week5.html  week7.html
ray@ray-VMware-Virtual-Platform:~$
```

Screenshot python3 webserver command:



```
ray@ray-VMware-Virtual-Platform:~$ cd ~/Downloads/site
ray@ray-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/) ...
```

Screenshot web browser visits your site



home | week 1 | week 2 | week 3 | week 4 | week 5 | week 6 | week 7

Hello World!

Calisthenics is an alternative training style to lifting weights that uses your own bodyweight.

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.

Code provided in lesson:

```
public class Main {

  public static void main(String[] args) {

    String ip = "192.168.1.100";

    String[] octets = ip.split("\\.");


    for (int i = 0; i < octets.length; i++) {

      System.out.print(octets[i]);

      if (i < 3) {

        System.out.print(".");

      }

    }

  }

}
```

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 (2⁵).
The range of this network segment is from 192.168.1.96 to 192.168.1.127.
```

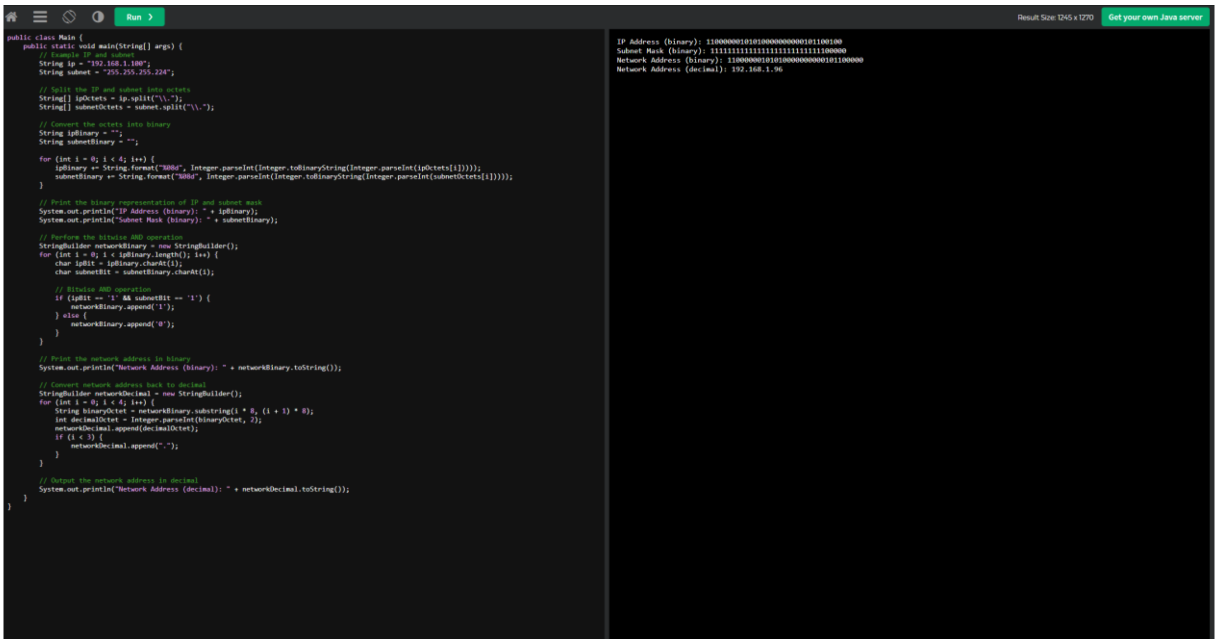
Paste source code here, with a screenshot of a working application.

Code provided in lesson:

```
public class Main {
 public static void main(String[] args) {
   String ip = "192.168.1.100";
   String[] octets = ip.split("\\.");

   for (int i = 0; i < octets.length; i++) {
     System.out.print(octets[i]);
     if (i < 3) {
       System.out.print(".");
     }
   }
 }
}
```

Finished code:

```
public class Main {
    public static void main(String[] args) {
        // Example IP and subnet
        String ip = "192.168.1.100";
        String subnet = "255.255.255.224";

        // Split the IP and subnet into octets
        String[] ipOctets = ip.split("\\.");
        String[] subnetOctets = subnet.split("\\.");

        // Convert the octets into binary
        String ipBinary = "";
        String subnetBinary = "";

        for (int i = 0; i < 4; i++) {
            ipBinary += String.format("%08d",
Integer.parseInt(Integer.toBinaryString(Integer.parseInt(ipOctets[i]))));
            subnetBinary += String.format("%08d",
Integer.parseInt(Integer.toBinaryString(Integer.parseInt(subnetOctets[i]))));
        }

        // Print the binary representation of IP and subnet mask
        System.out.println("IP Address (binary): " + ipBinary);
        System.out.println("Subnet Mask (binary): " + subnetBinary);

        // Perform the bitwise AND operation
        StringBuilder networkBinary = new StringBuilder();
        for (int i = 0; i < ipBinary.length(); i++) {
            char ipBit = ipBinary.charAt(i);
            char subnetBit = subnetBinary.charAt(i);

            // Bitwise AND operation
            if (ipBit == '1' && subnetBit == '1') {
                networkBinary.append('1');
            } else {
                networkBinary.append('0');
            }
        }

        // Print the network address in binary
        System.out.println("Network Address (binary): " + networkBinary.toString());

        // Convert network address back to decimal
        StringBuilder networkDecimal = new StringBuilder();
        for (int i = 0; i < 4; i++) {
            String binaryOctet = networkBinary.substring(i * 8, (i + 1) * 8);
            int decimalOctet = Integer.parseInt(binaryOctet, 2);
            networkDecimal.append(decimalOctet);
            if (i < 3) {
                networkDecimal.append(".");
            }
        }

        // Output the network address in decimal
        System.out.println("Network Address (decimal): " + networkDecimal.toString());
    }
}
```



```
public class Main {
    public static void main(String[] args) {
        // Example IP and subnet
        String ip = "192.168.1.100";
        String subnet = "255.255.255.224";

        // Split the IP and subnet into octets
        String[] ipOctets = ip.split("\\.");
        String[] subnetOctets = subnet.split("\\.");

        // Convert the octets into binary
        String ipBinary = "";
        String subnetBinary = "";

        for (int i = 0; i < 4; i++) {
            ipBinary += String.format("%08d", Integer.parseInt(Integer.toBinaryString(Integer.parseInt(ipOctets[i]))));
            subnetBinary += String.format("%08d", Integer.parseInt(Integer.toBinaryString(Integer.parseInt(subnetOctets[i]))));
        }

        // Print the binary representation of IP and subnet mask
        System.out.println("IP Address (binary): " + ipBinary);
        System.out.println("Subnet Mask (binary): " + subnetBinary);

        // Perform the bitwise AND operation
        StringBuilder networkBinary = new StringBuilder();
        for (int i = 0; i < ipBinary.length(); i++) {
            char ipBit = ipBinary.charAt(i);
            char subnetBit = subnetBinary.charAt(i);

            // Bitwise AND operation
            if (ipBit == '1' && subnetBit == '1') {
                networkBinary.append('1');
            } else {
                networkBinary.append('0');
            }
        }

        // Print the network address in binary
        System.out.println("Network Address (binary): " + networkBinary.toString());

        // Convert network address back to decimal
        StringBuilder networkDecimal = new StringBuilder();
        for (int i = 0; i < 4; i++) {
            String binaryOctet = networkBinary.substring(i * 8, (i + 1) * 8);
            int decimalOctet = Integer.parseInt(binaryOctet, 2);
            networkDecimal.append(decimalOctet);
            if (i < 3) {
                networkDecimal.append(".");
            }
        }

        // Output the network address in decimal
        System.out.println("Network Address (decimal): " + networkDecimal.toString());
    }
}
```

```
IP Address (binary): 11000000101010000000000101100100
Subnet Mask (binary): 11111111111111111111111111100000
Network Address (binary): 11000000101010000000000101100000
Network Address (decimal): 192.168.1.96
```

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