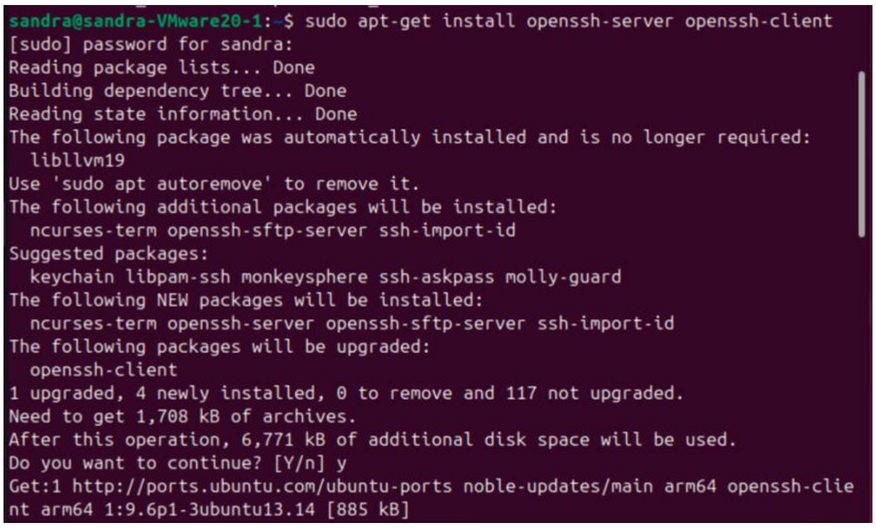
Template Week 6 – Networking

Student number: 580606

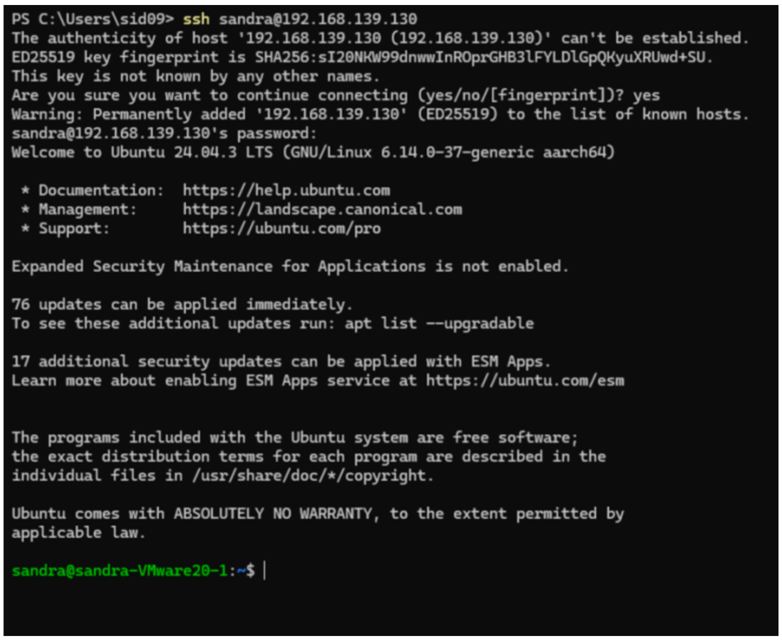
Assignment 6.1: Working from home

Screenshot installation openssh-server:



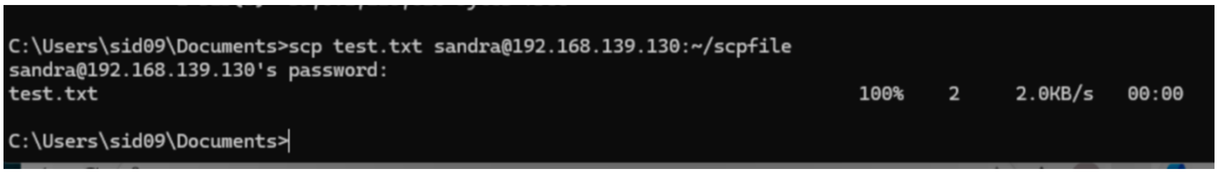
```
sandra@sandra-VMware20-1:~$ sudo apt-get install openssh-server openssh-client
[sudo] password for sandra:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  libllvm19
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  ncurses-term openssh-sftp-server ssh-import-id
Suggested packages:
  keychain libpam-ssh monkeysphere ssh-askpass molly-guard
The following NEW packages will be installed:
  ncurses-term openssh-server openssh-sftp-server ssh-import-id
The following packages will be upgraded:
  openssh-client
1 upgraded, 4 newly installed, 0 to remove and 117 not upgraded.
Need to get 1,708 kB of archives.
After this operation, 6,771 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ports.ubuntu.com/ubuntu-ports noble-updates/main arm64 openssh-clie
nt arm64 1:9.6p1-3ubuntu13.14 [885 kB]
```

Screenshot successful SSH command execution:



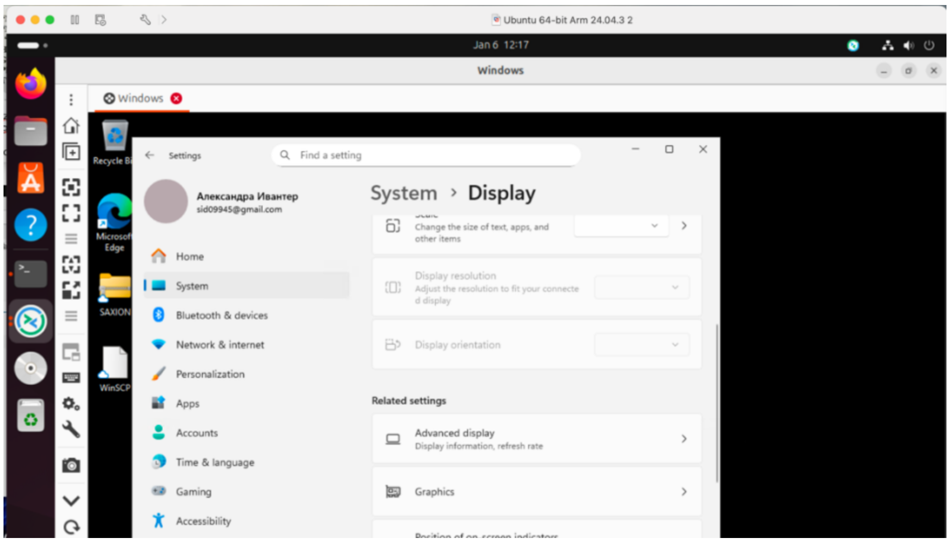
```
PS C:\Users\sid09> ssh sandra@192.168.139.130
The authenticity of host '192.168.139.130 (192.168.139.130)' can't be established.
ED25519 key fingerprint is SHA256:sI20NKW99dnwwInROprGHB3lFYLDlGpQKyuXRUwd+SU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.139.130' (ED25519) to the list of known hosts.
sandra@192.168.139.130's password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-37-generic aarch64)

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

Expanded Security Maintenance for Applications is not enabled.

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

17 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm


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.

sandra@sandra-VMware20-1:~$ |
```

Screenshot successful execution SCP command:



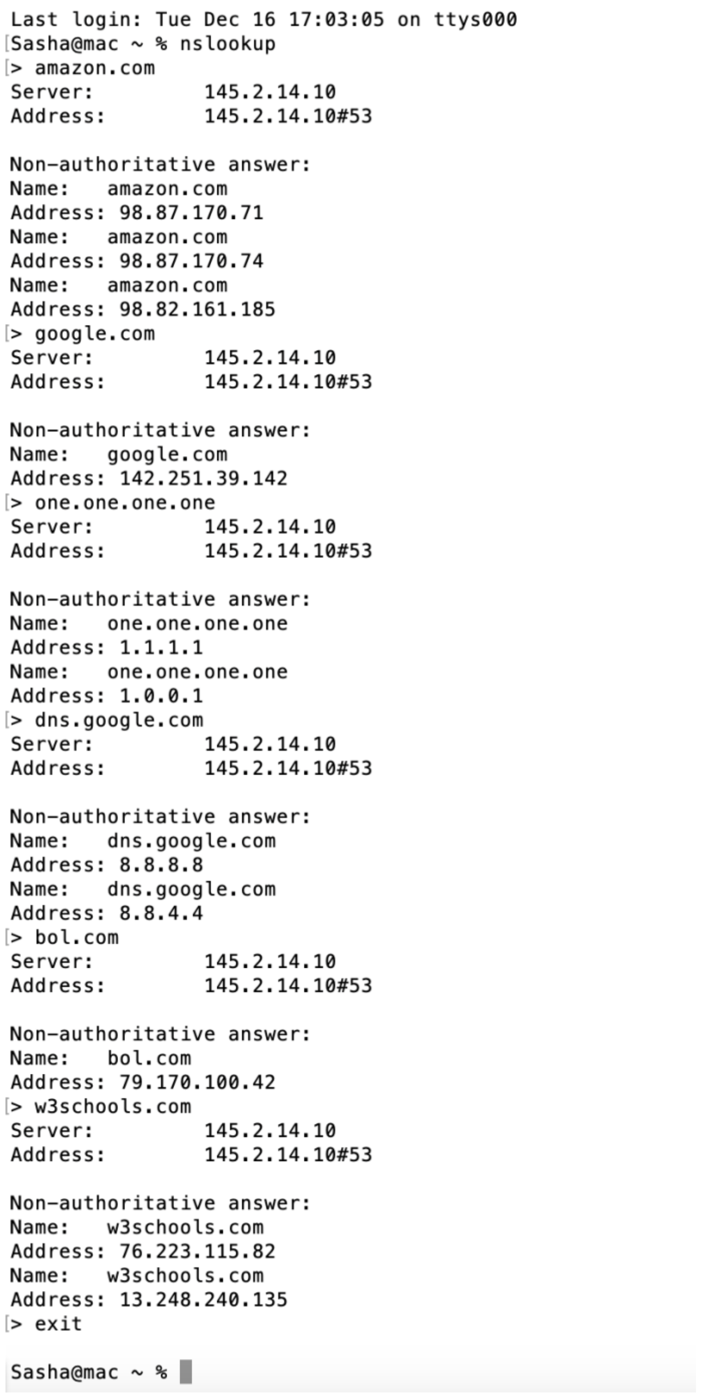
```
C:\Users\sid09\Documents>scp test.txt sandra@192.168.139.130:~/scpfile
sandra@192.168.139.130's password:
test.txt                                                  100%    2     2.0KB/s   00:00

C:\Users\sid09\Documents>
```

Screenshot remmina:



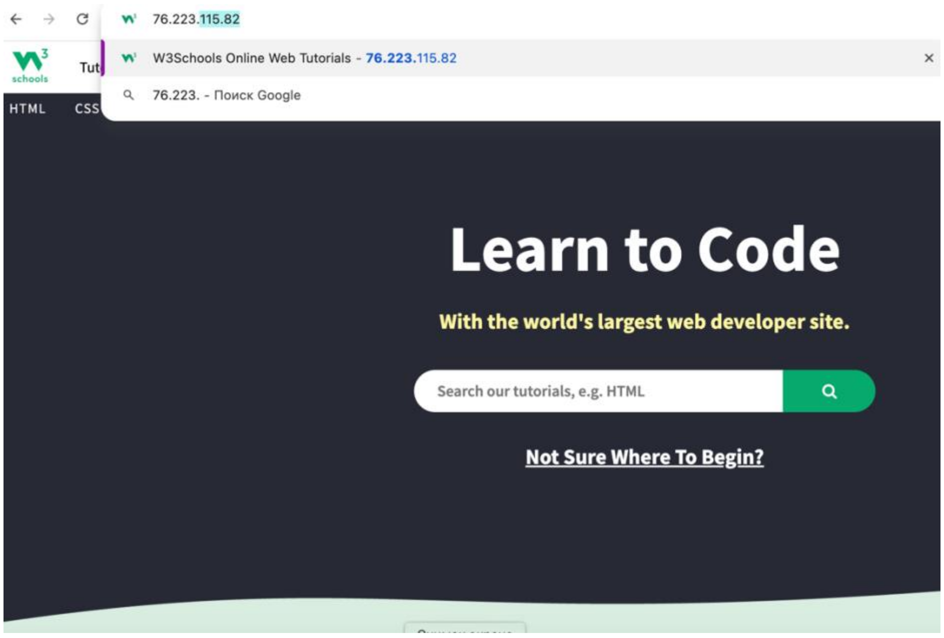
Assignment 6.2: IP addresses websites

Relevant screenshots nslookup command:



```
Last login: Tue Dec 16 17:03:05 on ttys000
Sasha@mac ~ % nslookup
> amazon.com
Server:         145.2.14.10
Address:        145.2.14.10#53

Non-authoritative answer:
Name:   amazon.com
Address: 98.87.170.71
Name:   amazon.com
Address: 98.87.170.74
Name:   amazon.com
Address: 98.82.161.185
> google.com
Server:         145.2.14.10
Address:        145.2.14.10#53

Non-authoritative answer:
Name:   google.com
Address: 142.251.39.142
> one.one.one.one
Server:         145.2.14.10
Address:        145.2.14.10#53

Non-authoritative answer:
Name:   one.one.one.one
Address: 1.1.1.1
Name:   one.one.one.one
Address: 1.0.0.1
> dns.google.com
Server:         145.2.14.10
Address:        145.2.14.10#53

Non-authoritative answer:
Name:   dns.google.com
Address: 8.8.8.8
Name:   dns.google.com
Address: 8.8.4.4
> bol.com
Server:         145.2.14.10
Address:        145.2.14.10#53

Non-authoritative answer:
Name:   bol.com
Address: 79.170.100.42
> w3schools.com
Server:         145.2.14.10
Address:        145.2.14.10#53

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

Sasha@mac ~ %
```

Screenshot website visit via IP address:


76.223.115.82
W3Schools Online Web Tutorials - 76.223.115.82
76.223. - Поиск Google
Learn to Code
With the world's largest web developer site.
Search our tutorials, e.g. HTML
Not Sure Where To Begin?

Assignment 6.3: subnetting

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

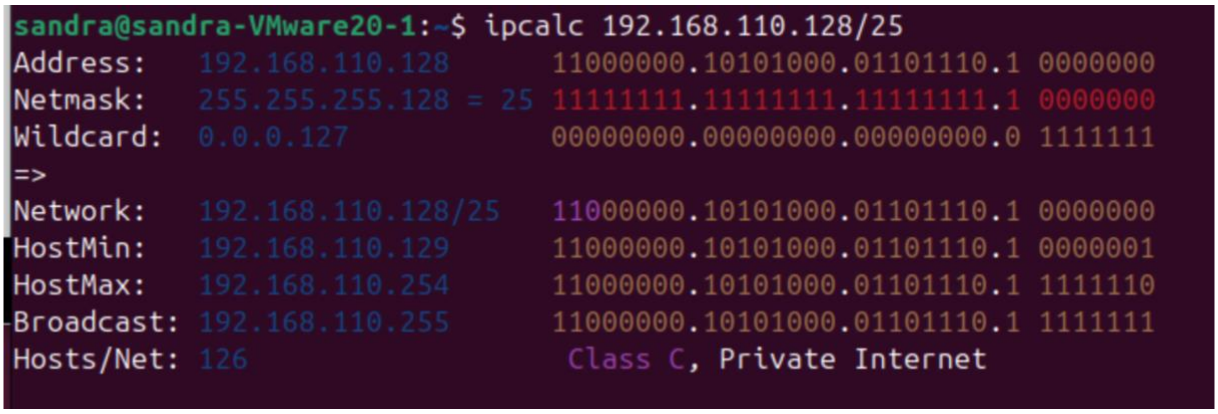
(32 – 25) = 7 (bits)

2 ** 7 = 128 – addresses in the network

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

The first and last IP addresses are allocated to Network Addresses, so the usable range is 1-126

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



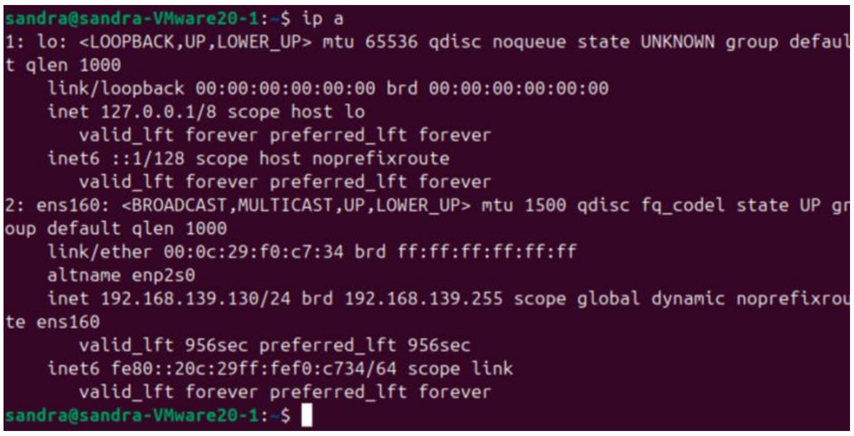
```
sandra@sandra-VMware20-1:~$ ipcalc 192.168.110.128/25
Address:   192.168.110.128        11000000.10101000.01101110.1 0000000
Netmask:   255.255.255.128 = 25   11111111.11111111.11111111.1 0000000
Wildcard:  0.0.0.127              00000000.00000000.00000000.0 1111111
=>
Network:   192.168.110.128/25     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.

The first three lines determine the bits allocated to computer addresses in the network. The fourth and fifth lines show the min and max addresses that can be allocated to computers.

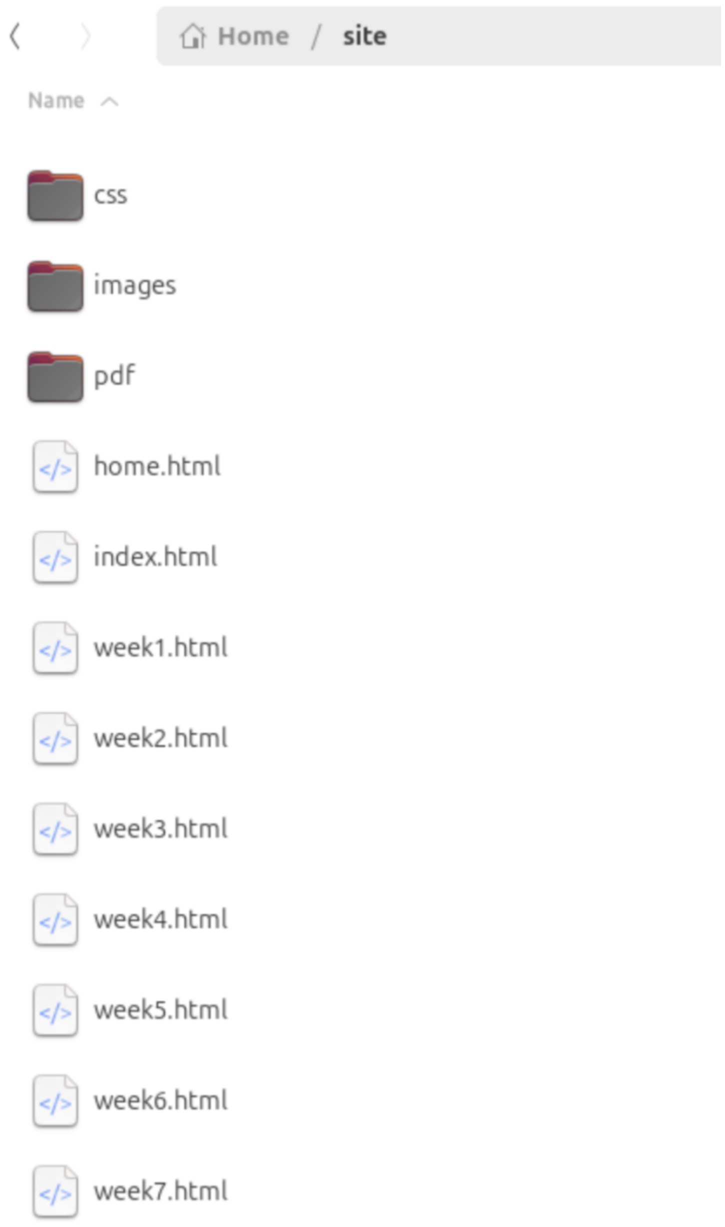
Assignment 6.4: HTML

Screenshot IP address Ubuntu VM:



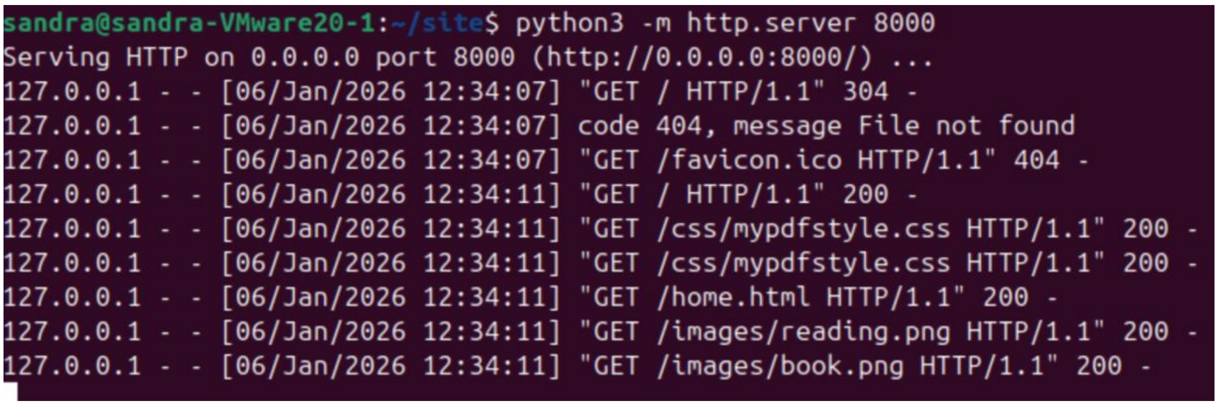
```
sandra@sandra-VMware20-1:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gr
oup default qlen 1000
    link/ether 00:0c:29:f0:c7:34 brd ff:ff:ff:ff:ff:ff
    altname enp2s0
    inet 192.168.139.130/24 brd 192.168.139.255 scope global dynamic noprefixrou
te ens160
       valid_lft 956sec preferred_lft 956sec
    inet6 fe80::20c:29ff:fef0:c734/64 scope link
       valid_lft forever preferred_lft forever
sandra@sandra-VMware20-1:~$
```

Screenshot of Site directory contents:



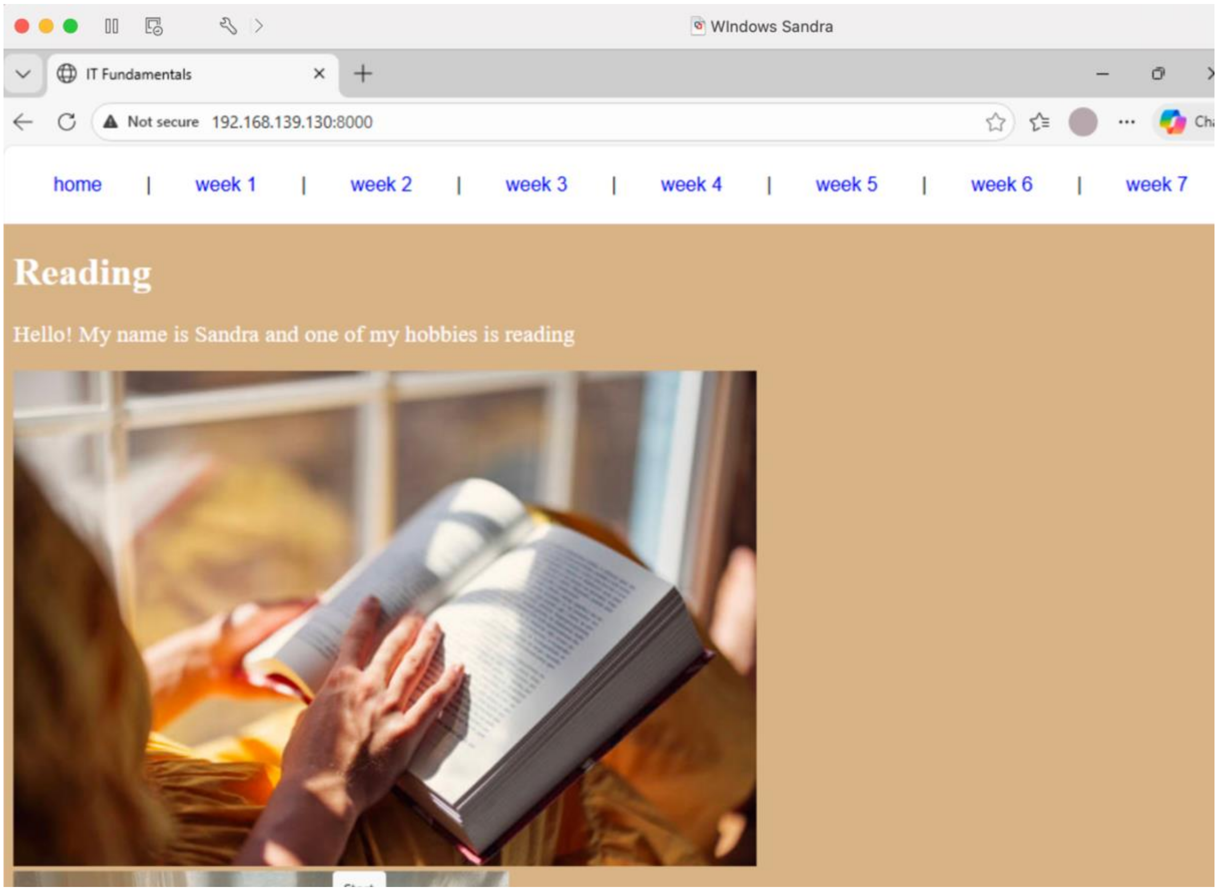
css
images
pdf
home.html
index.html
week1.html
week2.html
week3.html
week4.html
week5.html
week6.html
week7.html

Screenshot python3 webserver command:



```
sandra@sandra-VMware20-1:~/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 - - [06/Jan/2026 12:34:07] "GET / HTTP/1.1" 304 -
127.0.0.1 - - [06/Jan/2026 12:34:07] code 404, message File not found
127.0.0.1 - - [06/Jan/2026 12:34:07] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [06/Jan/2026 12:34:11] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [06/Jan/2026 12:34:11] "GET /css/mypdfstyle.css HTTP/1.1" 200 -
127.0.0.1 - - [06/Jan/2026 12:34:11] "GET /css/mypdfstyle.css HTTP/1.1" 200 -
127.0.0.1 - - [06/Jan/2026 12:34:11] "GET /home.html HTTP/1.1" 200 -
127.0.0.1 - - [06/Jan/2026 12:34:11] "GET /images/reading.png HTTP/1.1" 200 -
127.0.0.1 - - [06/Jan/2026 12:34:11] "GET /images/book.png HTTP/1.1" 200 -
```

Screenshot web browser visits your site



Windows Sandra
IT Fundamentals
Not secure 192.168.139.130:8000
home | week 1 | week 2 | week 3 | week 4 | week 5 | week 6 | week 7

Reading

Hello! My name is Sandra and one of my hobbies is reading

Assignment 6.5: Network segment

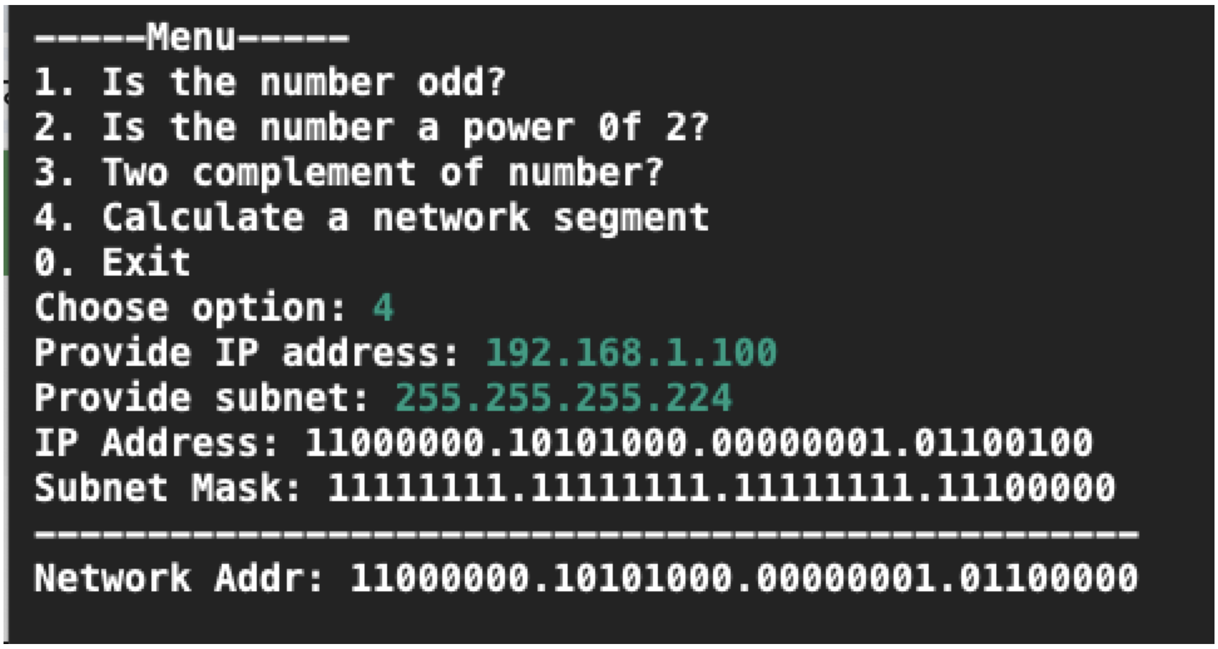
Remember that bitwise java application you've made in week 2? Expand that application so that you can also calculate a network segment as explained in the PowerPoint slides of week 6. Use the bitwise & AND operator. You need to be able to input two Strings. An IP address and a subnet.

IP: 192.168.1.100 and subnet: 255.255.255.224 for /27

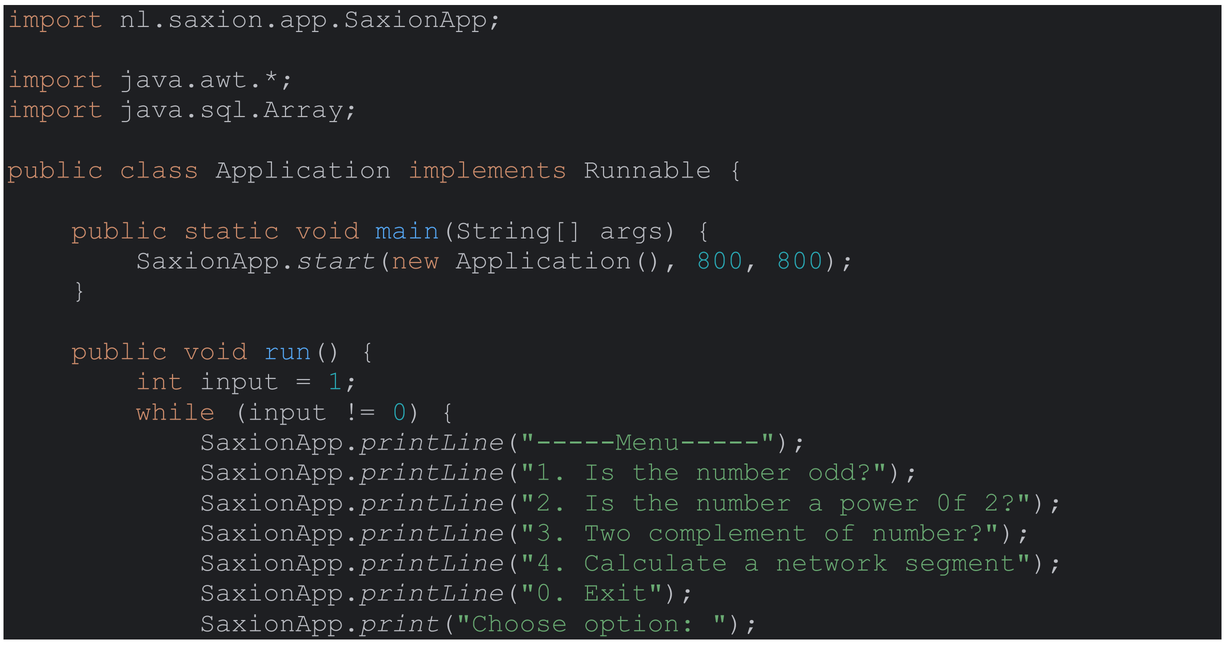
```
Example: 192.168.1.100/27
Calculate the network segment
IP Address:   11000000.10101000.00000001.01100100
Subnet Mask:  11111111.11111111.11111111.11100000
-------------------------------------------------
Network Addr: 11000000.10101000.00000001.01100000
```

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.



```
-----Menu-----
1. Is the number odd?
2. Is the number a power 0f 2?
3. Two complement of number?
4. Calculate a network segment
0. Exit
Choose option: 4
Provide IP address: 192.168.1.100
Provide subnet: 255.255.255.224
IP Address: 11000000.10101000.00000001.01100100
Subnet Mask: 11111111.11111111.11111111.11100000
-------------------------------------------------
Network Addr: 11000000.10101000.00000001.01100000
```



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

import java.awt.*;
import java.sql.Array;

public class Application implements Runnable {

    public static void main(String[] args) {
        SaxionApp.start(new Application(), 800, 800);
    }

    public void run() {
        int input = 1;
        while (input != 0) {
            SaxionApp.printLine("-----Menu-----");
            SaxionApp.printLine("1. Is the number odd?");
            SaxionApp.printLine("2. Is the number a power 0f 2?");
            SaxionApp.printLine("3. Two complement of number?");
            SaxionApp.printLine("4. Calculate a network segment");
            SaxionApp.printLine("0. Exit");
            SaxionApp.print("Choose option: ");
            input = SaxionApp.readInt();
            int number;
            switch (input) {
                case 1:
                    SaxionApp.print("Provide input number: ");
                    number = SaxionApp.readInt();
                    if (isOdd(number)) {
                        SaxionApp.printLine("Number " + number + " is
odd.");
                    } else {
                        SaxionApp.printLine("Number " + number + " is
even.");
                    }
                    break;
                case 2:
                    SaxionApp.print("Provide input number: ");
                    number = SaxionApp.readInt();
                    if (isAPowerOfTwo(number)) {
                        SaxionApp.printLine("Number " + number + " is a
power of 2.");
                    } else {
                        SaxionApp.printLine("Number " + number + " is not a
power of 2.");
                    }
                    break;
                case 3:
                    SaxionApp.print("Provide input number: ");
                    number = SaxionApp.readInt();
                    SaxionApp.printLine("Two compliment of " + number + "
is " + twoCompliment(number));
                    break;
                case 4:
                    SaxionApp.print("Provide IP address: ");
                    String ip = SaxionApp.readString();
                    SaxionApp.print("Provide subnet: ");
                    String subnet = SaxionApp.readString();
                    String[] IPArray = ip.split("\\.");
                    String[] subnetArray = subnet.split("\\.");
                    String bitIP = "";
                    String bitSubnet = "";
                    String bitSegment = "";
                    for (int i = 0; i < 4; i++) {

System.out.println(Integer.parseInt(subnetArray[i]));
                        System.out.println(subnetArray[i]);
                    }

                    for (int i = 0; i < 4; i++) {
                        bitIP += String.format("%08d",
Integer.valueOf(Integer.toBinaryString(Integer.parseInt(IPArray[i])))) +
".";
                        bitSubnet += String.format("%08d",
Integer.valueOf(Integer.toBinaryString(Integer.parseInt(subnetArray[i]))))
+ ".";
                        bitSegment += String.format("%08d",
Integer.valueOf(Integer.toBinaryString(Integer.parseInt(subnetArray[i]) &
Integer.parseInt(IPArray[i])))) + ".";
                    }
                    SaxionApp.printLine("IP Address: " + bitIP.substring(0,
bitIP.length() - 1));
                    SaxionApp.printLine("Subnet Mask: " +
```

```java
bitSubnet.substring(0, bitSubnet.length() - 1));
                    SaxionApp.printLine("-----------------------------------
---------------");
                    SaxionApp.printLine("Network Addr: " +
bitSegment.substring(0, bitSegment.length() - 1));
                    break;
                case 0:
                    SaxionApp.quit();
                    break;
                default:
                    SaxionApp.printLine("Invalid input, try again",
Color.RED);
            }
            SaxionApp.pause();
            SaxionApp.clear();

        }


    }

    public boolean isOdd(int number) {
        return (number & 1) == 1;
    }

    public boolean isAPowerOfTwo(int number) {
        return (number & (number - 1)) == 0;
    }

    public int twoCompliment(int number) {
        return ~number + 1;
    }
}
```