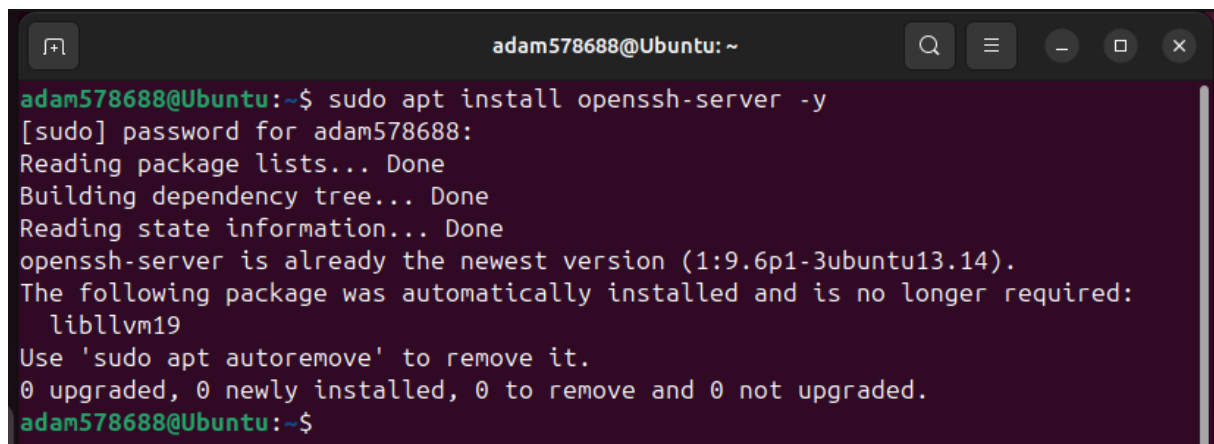


# Template Week 6 – Networking

Student number: 578688

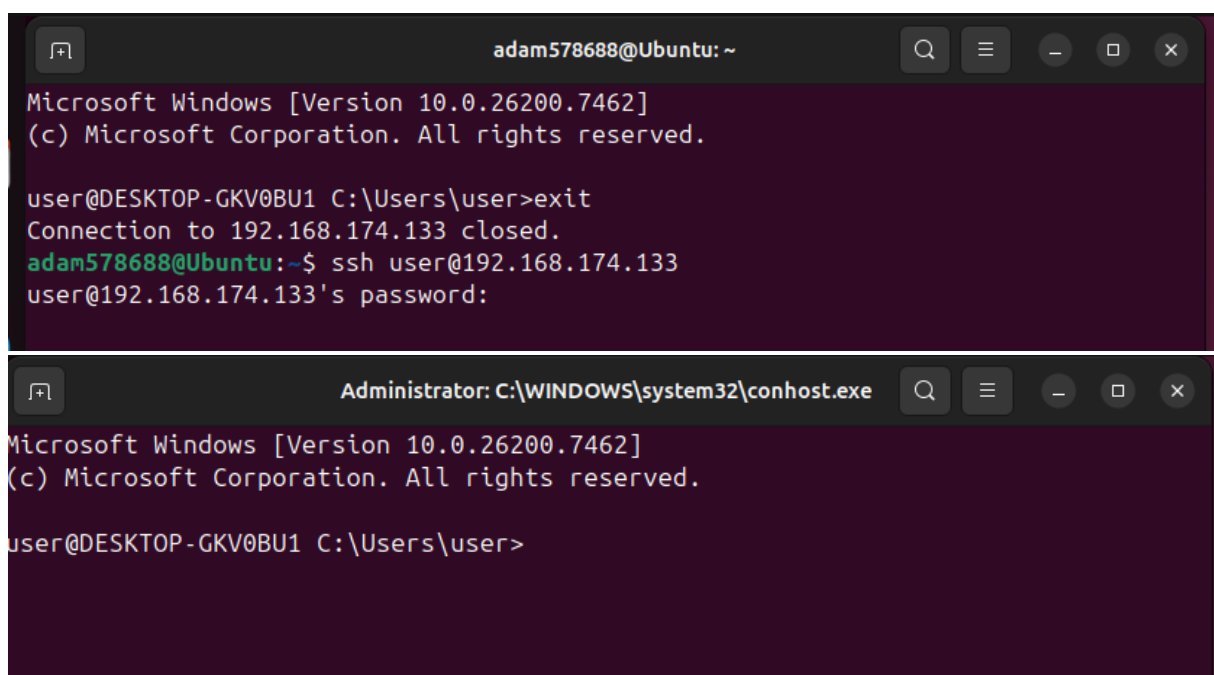
## Assignment 6.1: Working from home

Screenshot installation openssh-server:



```
adam578688@Ubuntu: ~  
adam578688@Ubuntu:~$ sudo apt install openssh-server -y  
[sudo] password for adam578688:  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
openssh-server is already the newest version (1:9.6p1-3ubuntu13.14).  
The following package was automatically installed and is no longer required:  
  libllvm19  
Use 'sudo apt autoremove' to remove it.  
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.  
adam578688@Ubuntu:~$
```

Screenshot successful SSH command execution:

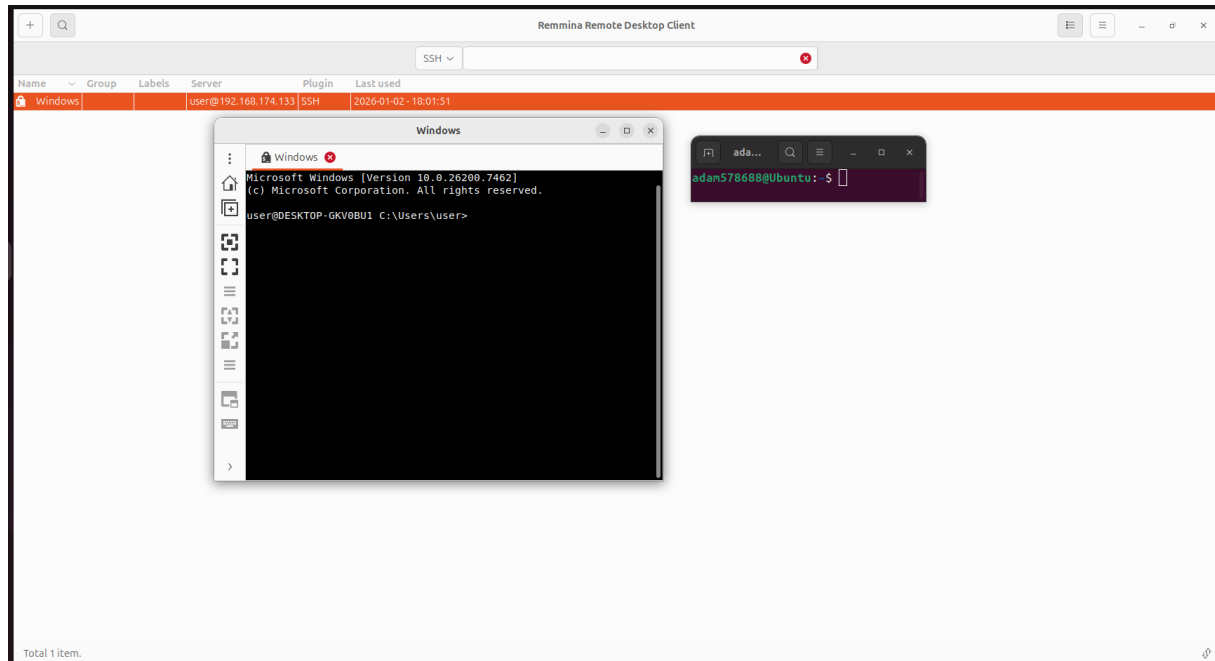


```
adam578688@Ubuntu: ~  
Microsoft Windows [Version 10.0.26200.7462]  
(c) Microsoft Corporation. All rights reserved.  
  
user@DESKTOP-GKV0BU1 C:\Users\user>exit  
Connection to 192.168.174.133 closed.  
adam578688@Ubuntu:~$ ssh user@192.168.174.133  
user@192.168.174.133's password:  
  
Administrator: C:\WINDOWS\system32\conhost.exe  
Microsoft Windows [Version 10.0.26200.7462]  
(c) Microsoft Corporation. All rights reserved.  
  
user@DESKTOP-GKV0BU1 C:\Users\user>
```

Screenshot successful execution SCP command:

```
adam578688@Ubuntu:~$ scp user@192.168.174.133:/C:/Users/user/Downloads/Tumble.png ~/Pictures
user@192.168.174.133's password:
Tumble.png                                100% 199KB  3.6MB/s   00:00
adam578688@Ubuntu:~$
```

Screenshot remmina:



## Assignment 6.2: IP addresses websites

Relevant screenshots nslookup command:

```
adam578688@Ubuntu:~$ nslookup
> amazon.com
Server:      127.0.0.53
Address:     127.0.0.53#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:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   google.com
Address: 142.250.179.142
Name:   google.com
Address: 2a00:1450:400e:806::200e
> one.one.one.one
Server:      127.0.0.53
Address:     127.0.0.53#53

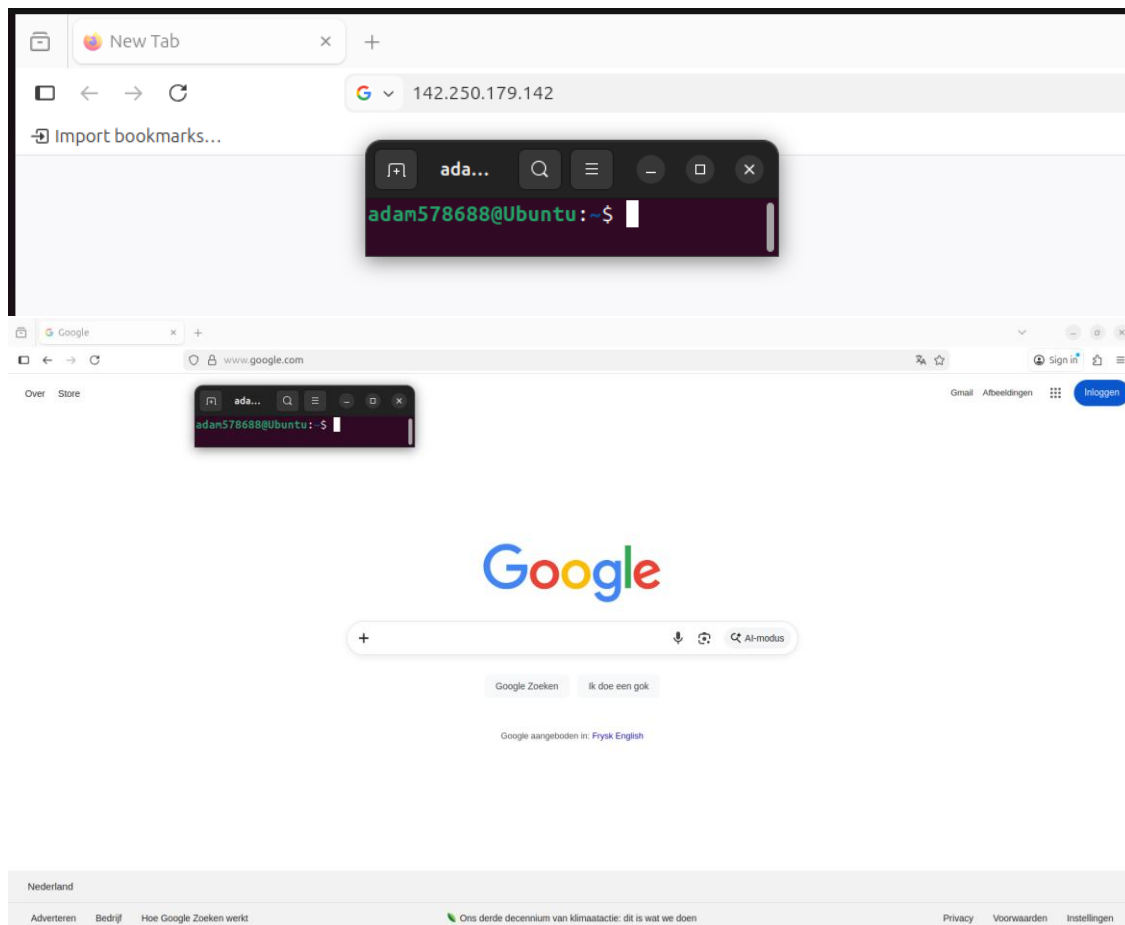
Non-authoritative answer:
Name:   one.one.one.one
Address: 1.0.0.1
Name:   one.one.one.one
Address: 1.1.1.1
Name:   one.one.one.one
Address: 2606:4700:4700::1001
Name:   one.one.one.one
Address: 2606:4700:4700::1111

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
> bol.com
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   bol.com
Address: 79.170.100.42
> w3schools.com
Server:      127.0.0.53
Address:     127.0.0.53#53

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

126

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

11000000 10101000 01101110 10000000

&

11111111 11111111 11111111 10000000

=

11000000 10101000 01101110 10000000

192.168.110.129 – 192.168.110.254

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

```
adam578688@Ubuntu: ~  
adam578688@Ubuntu:~$ ipcalc 192.168.110.128/25  
Address: 192.168.110.128      11000000.10101000.01101110.1 00000000  
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  
adam578688@Ubuntu:~$
```

Explain the above calculation in your own words.

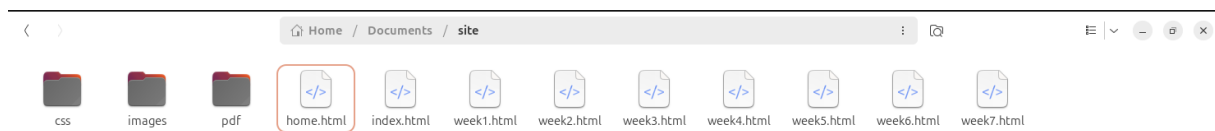
To figure out how many IP addresses we can use, you count the remaining zeros in the subnet mask which in this case is 7 since  $32 - 25$  is 7, then you do 2 raised to that number of zeroes,  $2^7$  is 128 which means that 192.168.110.128 – 192.168.110.255 are all viable, however 192.168.110.128 is reserved for the network and 192.168.110.255 is reserved for Broadcast, meaning the only IP's that can be assigned to a computer are within the following range of 192.168.110.129 to 192.168.110.254.

#### Assignment 6.4: HTML

Screenshot IP address Ubuntu VM:

```
adam578688@Ubuntu:~$ ip a  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default 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 fq_codel state UP group default qlen 1000  
    link/ether 1a:f1:be:65:57:33 brd ff:ff:ff:ff:ff:ff permaddr 00:0c:29:88:34:86  
    altname enp2s1  
    inet 192.168.174.129/24 brd 192.168.174.255 scope global dynamic noprefixroute ens33  
        valid_lft 1601sec preferred_lft 1601sec  
    inet6 fe80::f56a:671c:71a0:b68a/64 scope link noprefixroute  
        valid_lft forever preferred_lft forever  
adam578688@Ubuntu:~$
```

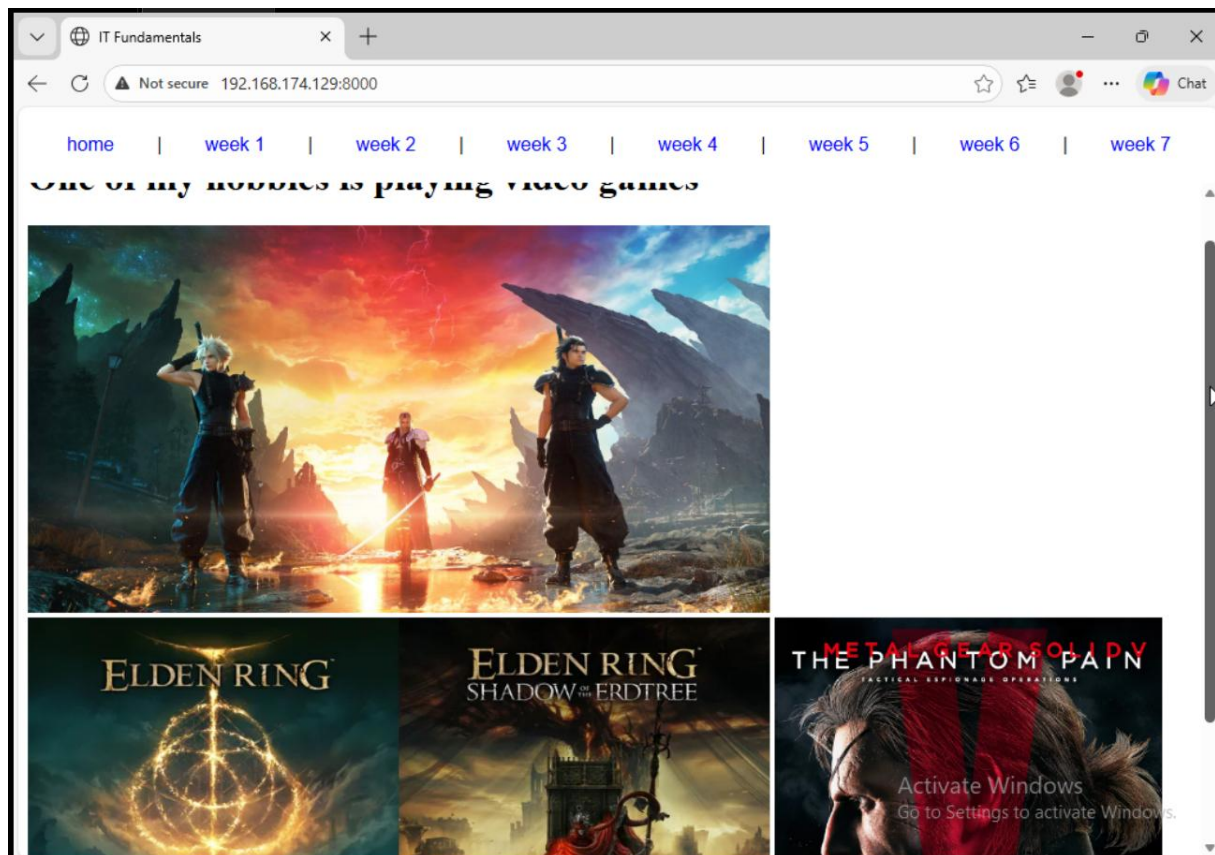
Screenshot of Site directory contents:



Screenshot python3 webserver command:

```
adam578688@Ubuntu:~$ ls
Desktop  Downloads  file.txt  Music     Public  simple.c  Templates
Documents file.tar.gz hello     Pictures  simple  snap      Videos
adam578688@Ubuntu:~$ cd Documents/site
adam578688@Ubuntu:~/Documents/site$ ls
css      images    pdf        week2.html  week4.html  week6.html
home.html index.html week1.html  week3.html  week5.html  week7.html
adam578688@Ubuntu:~/Documents/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



### 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 javax.xml.parsers.SAXParser;
import java.util.Scanner;

public class BitwiseCalculator {
    static Scanner myObj = new Scanner(System.in);
    static Scanner reader = new Scanner(System.in);

    static int customNumber = 0;

    public static void main(String[] args) {

        while (true) {

            System.out.print("1.\tIs number odd?\n" +
                "2.\tIs number a power of 2?\n" +
                "3.\tTwo's complement of number?\n" +
                "4. Network Segment?\n");
            int number = myObj.nextInt();

            if (number < 4) {
                System.out.print("Enter a Number: ");
                customNumber = myObj.nextInt();
            }
        }
    }
}
```



```

switch (number) {
    case 1:
        EvenOrOdd(customNumber);
        break;
    case 2:
        System.out.print(powerOfTwo(customNumber) + "\n\n");
        break;
    case 3:
        twosComplement(customNumber);
        break;
    case 4:
        networkSegmentCalculator();
    case 0:
        break;
}

}

}

```

```

public static void EvenOrOdd(int customNumber) {

```

```

    if ((1 & customNumber) == 1) {
        System.out.print(true + "\n\n");
    } else if ((1 & customNumber) == 0) {
        System.out.print(false + "\n\n");
    }
}

```

```

public static boolean powerOfTwo(int customNumber) {

```

```

    return (customNumber & (customNumber - 1)) == 0;
}

```

```

public static void twosComplement(int number) {

```

```

    number = ~number + 1;
    System.out.println("Number: " + number);
}

```

```

public static void networkSegmentCalculator() {

```

```

StringBuilder networkSegment = new StringBuilder();

System.out.print("Enter the IP address: ");
String IP = reader.nextLine();
System.out.print("Enter the subnet: ");
String subnet = reader.nextLine();

String[] individualIP = IP.split("\\.");
String[] individualSubnet = subnet.split("\\.");

for (int i = 0; i < 4; i++) {
    int ipPart = Integer.parseInt(individualIP[i]);
    int subnetPart = Integer.parseInt(individualSubnet[i]);

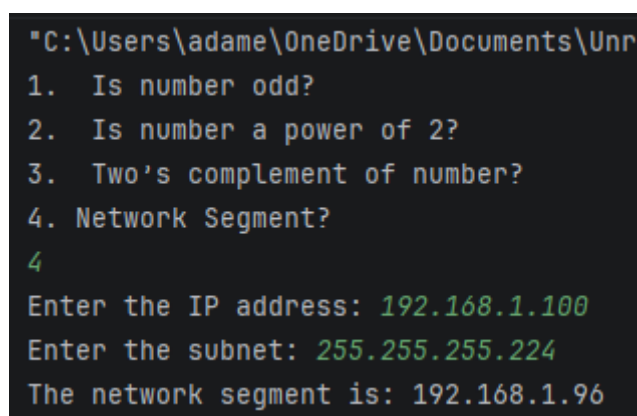
    networkSegment.append(ipPart & subnetPart);

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

System.out.println("The network segment is: " + networkSegment);
}

}

```



```

"C:\Users\adame\OneDrive\Documents\Unr
1. Is number odd?
2. Is number a power of 2?
3. Two's complement of number?
4. Network Segment?
4
Enter the IP address: 192.168.1.100
Enter the subnet: 255.255.255.224
The network segment is: 192.168.1.96

```

Ready? Save this file and export it as a pdf file with the name: [week6.pdf](#)