Department of Computer Science
Computer Networks
Due: Thursday 26th Sept (23.59)

Your name:
TA Name:
Estimated Time: 20 hours

This is pair assignment, you may work either on your own or with a partner of your choice.

This assignment should be completed using C++ 11, the hand-in format is up to you as long as the program compiles with the make command from the source folder.

Please use zip to bundle your source code and program submission. Do **NOT** include any hidden files (.git, .DS_Store .vscode) files in your submission. All code used to complete the assignment should be submitted, with a Makefile than can be used to compile your code and a README text file explaining how to compile and run your program(s).

This assignment requires that you use your laptop to create a port scanning/knocking program that interacts with a server on 130.208.246.249.

Question:	1	2	3	4	5	Total
Points:	35	35	20	10	10	110
Score:						

Speak easy to the port, and perhaps it will let you in.

In this assignment you will be introduced to the delights of packet crafting, bit twiddling and UDP subterfuge.

Somewhere on the TSAM server (130.208.246.249), a server is listening to some ports in the range 4000-4100. Find the ports, send them the right packets, and use the secret knock to gain access to the secret information!

During the first week the ports are less likely to drop packets.

1 . . . . *35 points*

Write a UDP port scanner, that takes in as arguments the IP address of the machine, and a range of ports to scan between. The scanner should be run with the command:

```
./scanner <IP address> <low port> <high port>
```

Use it to scan between ports 4000-4100 on 130.208.246.249 and print out the open ports that you find in this range.

This requires to send some UDP datagram to each of the ports and wait some limited time for a response.

Do not rely on the ports always being the same. Also, note that UDP is an unreliable protocol. Some packets may be dropped randomly.

2 . . . . *35 points*

You should have discovered 4 open ports in part 1. The ports you discovered are puzzle ports, safeguarding information about two additional ports which are not showing up on your scan. Your task is to write ***a separate program*** to solve the puzzle ports, in order to reveal the two hidden ports and the secret phrase. Each port will send you instructions on how to reveal its secret port if you send it a UDP message.

The program should be run with the command:

```
./puzzlesolver <IP address> <port1> <port2> <port3> <port4>
```

The program should interact with the ports discovered in part 1 by sending them a UDP message following the instructions provided by the puzzle port.

The puzzle ports will change over time, so do not hard code the ports, but rather supply them as command line arguments to your program.

3 . . . . *20 points*

When the oracle receives a comma-separated message containing the hidden ports it will reply with a message telling you the order and no. of knocks to use. For the final part of this assignment, you should modify your program from part 2 to knock on the hidden ports in the correct order, and print out the message from the final hidden port.

Each knock must contain the message ”knock”, except for the last knock, which should contain the secret phrase from part 2.

4 . . . . *10 points*

Points will be awarded for code quality, commenting and submission as follows:

(a) (3 points) Code compiles using the supplied Makefile

(b) (2 points) Code follows command line invocations described above.

(c) (5 points) Code is well commented, and modular

5 . . . . *10 points*

For 10 bonus points. After completing the port-knocking, you were sent a secret message, follow the instructions in the secret message for 10 bonus points.