The idea for this homework is to collect a pcap file and analyse it, to find interesting parts of it.

On the undergrad environment, start the tinycore virtual machine:

`/usr/local/tinycore/tinycore -f [savefile] -c [packetcap] &`

You can reuse your old savefile or start a new one. The name after -c option will store the packet capture. In tinycore, open firefox and go to a website. The TAs will do an example with `wikipedia.com`.

After you load the website, shut down the tinycore and copy the packet capture for local analysis.

Find the following pieces of information from the packet capture:

- IP address of DHCP offerer
- IP address that gets assigned to tinycore
- The DNS server that is used to query the domain you chose (e.g., `www.wikipedia.com`)
- The IP address of the domain you chose.
- The HTTP GET for "/" on that domain.
- The reply to that GET. Is it a 301 to another site?
- The sequence number for the TCP packet in the HTTP GET. Note that wireshark will likely give relative sequence numbers, so look at the raw hex of the packet and compute its value. Note that these are in network order; you can check your math by finding a later packet's sequence number and seeing if the relative to real values add up.

If TLS is not used, go to an HTTPS website and capture that traffic; otherwise just continue using your capture. Then find the following pieces of information from the packet capture

- The version of TLS that is used
- The negotiated cipher suite
- The CA for the certificate
- Whether OSCP or OSCP-stapling is used. OSCP stapling will look like a Certificate Status message from server, while OSCP will be its own extra protocol to somewhere else.
- If OSCP or OSCP-stapling is used, when was the certificate checked and how long will it remain valid.

Wireshark allows packet to be filtered by the fields. Once you find the IP for the server, you can enter `ip.src == W.X.Y.Z || ip.dst = W.X.Y.Z` as the filter to only look at packets of that form. You can filter for `http` with `tcp.port == 80` and for `dns` with `udp.port == 53`.