1. **Subnets**

   (a) Discuss the difference between a classful network and CIDR.

   (b) When is the subnet mask actually needed?

2. **IPv4 vs. IPv6**

   (a) Discuss the differences of the Internet Protocol version 4 and 6.

   (b) How long does the address space last in case of IPv4/IPv6, when an address is assigned every pico-second.

   (c) How can the two versions of IP coexist?

3. **IPv6 Tunnel**
   There are different tunnel brokers that provide your IPv6 capable system with an IPv6 address such that you can access the internet over IPv6.

   (a) How can this be achieved technically, even if the rest of your infrastructure at home or your ISP only understands IPv4?

   (b) Go to `www.tunnelbroker.net` and register for an account.

   (c) Create a regular tunnel on the website by providing your current IPv4 address.

   (d) Configure your machine as explained in the *Example Configurations* section.

   (e) Check that your configuration works, i.e. visit `www.wieistmeineip.de/ipv6-test` or make a call to traceroute6: *traceroute6 google.com*

   (f) Capture some IPv6 traffic using Wireshark.

4. **RIOT Networking**
   RIOT already provides a lightweight IPv6 network stack called 6LoWPAN. This stack can be used on different devices including a native Linux system by simulating link layer devices (network tap). To point a native RIOT process to a tap interface, you can simply append the numbered tap interface to the program call, i.e. *./bin/chatter.elf tap0*.

   (a) Create two tap network devices on your computer. A script in your local RIOT directory *RIOT/cpu/native/tapsetup.sh* helps you with that.

   (b) Take a look at the *sixlowpan* sample project. Create a new native project called *chatter*, which listens for IP packets and displays their content on the terminal.

   (c) Up to now, nobody sends you any messages. Expand your implementation, such that a chatter instance reads a line from the standard input, and sends the message to the other device.

   (d) Why does RIOT use 6LoWPAN and what is the difference to the IPv6 network stack on your computer?

   Note 1) The current available version of the *sixlowpan* example has two bugs in the *init()* function. To fix them, take a look at the *strtok* documentation and use TRANSCEIVER_NATIVE instead of TRANSCEIVER_CC1100.

   Note 2) The two nodes have to be configured properly to allow communication over IP. The easiest solution is to initialize both nodes as adhoc routers (*sixlowpan_lowpan_adhoc_init*).