All exercise numbers from the textbook refer to the **second edition**.

1. Suppose that Alice and Bob are using NTRU with $N = 251$, $q = 131$, $p = 3$, and $d = 6$. How many bits are needed to represent the plaintext? How many bits are needed to represent the ciphertext? What is the message expansion ratio?
2. Exercise 7.23, parts (a),(b),(c).
3. Exercise 7.24 (note that this exercise explains why $\mathbf{f}$ is chosen from $\mathcal{T}(d+1, d)$ in NTRU, rather than from $\mathcal{T}(d, d)$ like $\mathbf{r}$ and $\mathbf{g}$).
4. Exercise 7.25.
5. Exercise 7.30.
6. Exercise 7.35 (the second part should be labeled part (b)).
7. Exercise 7.45.
8. In some implementations of NTRU, rather than fixing one public parameter d, one chooses three different parameters d_1, d_2, d_3, and stipulates that Alice chooses $\mathbf{f}$ from $\mathcal{T}(d_1+1, d_1)$ and $\mathbf{g}$ from $\mathcal{T}(d_2, d_2)$, and Bob chooses $\mathbf{r}$ from $\mathcal{T}(d_3, d_3)$. Assuming that $d_1 \geq d_2 \geq d_3$, determine an inequality of the form $q > \cdots$ to replace the inequality $q > (6d+1)p$ in table 7.3, serving the same purpose in this more general formulation (your inequality should specialize to $q > (6d+1)p$ in the case $d_1 = d_2 = d_3 = d$).

   Programming problems

   Full formulation and submission: `https://www.hackerrank.com/m158-2016-pset-11`

9. Analyze a sequence of proposed "transactions" in a basic cryptocurrency, based on ECDSA signatures with the elliptic curve secp256k1. See the online statement for a complete description of the rules underlying the system and the format of the transactions.
10. Implement functions to add or multiply two polynomials, or to centerlift or compute the inverse of a single element, in the ring R_q, where element are represented as lists of length N consisting of integers from 0 to $q-1$ inclusive.
11. Decipher an NTRU ciphertext e, given the parameters N, q, p, d, and the private elements f, g (notation as in table 7.4).
12. Cryptanalyze the congruential cryptosystem (prototype for NTRU) from section 7.1 (described in table 7.1). You will be given the public parameter q, a *public* key h (but not the private key), and a ciphertext e; you must determine the plaintext m. The recommended method is to implement Gauss's lattice basis reduction algorithm from section 7.13.1 of the textbook, which we will discuss in class on Friday. Half the test cases will also have small enough q to be susceptible to a more brute-force attack.

Note. I have decided not to offer extra credit for this, but if you are interested in learning about attacks on the NTRU system (after final exams are over), you will want to read about the LLL algorithm in section 7.13. If you implement it and want to test it on some small-N cases of NTRU, you may do so at this demonstration problem:

`https://www.hackerrank.com/contests/m158-2016-demos/challenges/m158-2016-ntru-cryptanalysis`