Please write your hackerrank username somewhere on your problem set.
Written problems are due to the Math 158 mailbox in Kassar house by **5pm** (note change) on Friday, December 11. Programming problems must be electronically submitted by **5pm** according to the instructions on problem set 1. The submission page for the programming questions is the following.

`https://www.hackerrank.com/math158-problem-set-10/`

Written problems

1. Textbook exercise 7.1.
2. Suppose that the congruential cryptosystem on page 375 (which I was calling "1Tru" in class) is modified as follows.
   - Four parameters F, G, M, R are chosen at the beginning, in addition to the modulus q.
   - When Alice makes her private and public keys, she chooses f, g so that $0 \leq f < F$ and $M \leq g < G$. She then computes her public key h as on page 375.
   - When Bob sends a message, he must choose his message m so that $0 \leq m < M$, and he must choose his random number r so that $0 \leq r < R$. Then he computes e as on page 375.

   (a) What are the values of F, G, M, R (in terms of q) used in the system described on page 375?
   (b) Suppose Alice computes b from e as on page 375. Write an inequality in terms of F, G, M, R, and q that will guarantee that the number b she computes is definitely equal to m.
   (c) Explain why decryption could fail if the requirement $M \leq g$ were dropped in Alice's procedure for generating her public key.
3. Suppose that Alice has a plaintext the form of an integer D. Describe a procedure by which Alice can convert this message into an element $\mathbf{m}$ that she can encrypt using NTru (in the notation of page 419). Such a procedure is called an *encoding scheme.*
4. Textbook exercise 7.22.
5. Textbook exercise 7.23.
6. Textbook exercise 7.25.
7. Textbook exercise 7.35 (the second part should be labeled part (b)).
8. A network of users set up a basic cryptocurrency as follows. They begin by agreeing on a digital signature scheme and a hash function. There are always exactly 1000 "coins" in existence. Each coin is said to "belong" to a *different* public key (in the beginning, some initial 1000 public keys are agreed upon as the initial owners). There is a public "transaction list," to which anyone can post anonymously at any time. The list records the order that posts are received, and it is impossible to erase a post. The users observe the following rule: a post is considered a "valid transaction" if and only if all of the following hold.
   - It has the format "k_A^{pub} k_B^{pub} S", where k_A^{pub} and k_B^{pub} are public keys, and S is a digital signature.
   - The signature is a valid signature, when k_A^{pub} is used as the verification key, for a document equal to the hash value of "k_A^{pub} k_B^{pub}".
   - k_A^{pub} is currently an owner, and k_B^{pub} is currently not an owner.

When a valid transaction is posted, ownership of the coin transfers from k_A^{pub} to k_B^{pub}.

Alice controls all of the coins for which she knows the private key for the "owner" public key. If Alice wishes to pay Bob one coin, Bob should first *create a new key-pair,* safely store the private key k_B^{priv}, and tell Alice the public key k_B^{pub}. Alice then computes a signature S as in the second bullet point (taking as k_A^{pub} *any* of the public keys for coins she currently controls), and sends it to Bob. The signature S is called Alice's "payment;" knowledge of S enables Bob to post a transaction and assume control of the coin in question.

(a) Suppose that Bob accidentally tells Alice the wrong value of k_B^{pub} (telling her a random number instead), and proceeds to receive "payment" and post it to the transaction list. Explain why the coin is now essentially lost from circulation forever.

(b) Suppose that Bob wishes to pay Carol a coin, but he will not own any coins until Alice pays him a coin next week. Describe a procedure Bob can use to send Carol "payment" now, such that the payment will become valid (and allow Carol to assume ownership of a coin) immediately after Alice pays Bob next week.

(c) Suppose Alice owns a coin using a public key k_A^{pub} (i.e. this public key is an "owner," and Alice knows the corresponding private key). At some time, Alice transfers this coin to Bob. Later, Carol wishes to pay Alice a coin. To save effort, Alice doesn't bother creating a new public key, and sends Carol the same key k_A^{pub} that she used before. Explain how Bob can now steal this coin from Alice.

(d) Suppose that a charity creates and publicizes a public key k_C^{pub} (storing the private key in a safe place), and both Alice and Bob submit "payment" (i.e. valid signatures to transfer coins to k_C^{pub}). Describe a procedure the charity can use to assume ownership of both Alice's coin and Bob's coin, despite the fact that each public key can only own one coin at a time.

Remark. This system is a cartoon of some of the basic aspects of Bitcoin, and has several serious flaws as it is described. I have also left out the most innovative aspect of the Bitcoin design, namely the blockchain and the proof-of-work system. The original Bitcoin paper (under the pseudonym Satoshi Nakamoto) is mostly nontechnical and explains these features. You might also be interested to read about B-Money (`http://www.weidai.com/bmoney.txt`), a precursor to Bitcoin that is somewhat simpler.

Programming problems

9. Write a program to encrypt an NTru message using a given random element $\mathbf{r}$, as well as a public key and parameters.

10. Write a program to decrypt an NTru ciphertext $\mathbf{e}$, given both the public and private key. The input will include the inverses $\mathbf{F}_p$ and $\mathbf{F}_q$ as part of the private key, so you do not need to write an algorithm to compute these inverses.

11. Write a program that is given the list of initial "owners" in the currency from problem 8 and a sequence of posts of the form "$k_A^{\text{pub}} k_B^{\text{pub}} S$" to the transaction list, either accepts or rejects each transaction, and determines the current list of owners after all transactions are completed. The signature system will be ECDSA with Bitcoin's curve (Secp256k1), and the hash function will be SHA-256 (details and some sample code in the online statement).