Please write your hackerrank username somewhere on your problem set.
Written problems are due at the beginning of class on Friday, September 18. Programming problems must be electronically submitted by 10:50am according to the instructions on problem set 1. The submission page for the programming questions is as follows.

`https://www.hackerrank.com/math158-pset-2`

Written problems

1. Textbook exercise 1.15 (1.14 in first edition).
2. Textbook exercise 1.20 (1.19 in first edition).
3. Textbook exercise 1.22 (1.21 in first edition).
4. Consider the linear congruence $ax \equiv b \pmod{M}$. Prove that this congruence has a solution if and only if $\gcd(a, M)$ divides b.
5. Textbook exercise 1.34 (1.32 in first edition).
6. (a) Suppose that a and b are two integers such that $g^a \equiv 1 \pmod{m}$ and $g^b \equiv 1 \pmod{m}$. Prove that $g^{\gcd(a,b)} \equiv 1 \pmod{m}$.
   (b) Suppose $g \in (\mathbf{Z}/p\mathbf{Z})^\times$, where p is a prime. Let d be the smallest positive integer such that $g^d \equiv 1 \pmod{p}$ (called the *multiplicative order* of $g \pmod{p}$). Prove that d divides $p - 1$.
7. Textbook exercise 2.3 (same in first edition).
8. Textbook exercise 2.4 (same in first edition).

Programming problems

9. You play Bob in this problem. Write a program to perform Diffie-Hellman key exchange. Specifically, you will receive three integers from Alice: p, g, and A (as described in Table 2.2 of the textbook), where p is a 1024-bit prime. You must generate a number B to send to Alice, and also compute the shared secret S. You should look up how to generate random numbers. The autograder will check to make sure that your program is not deterministic.
10. Write a problem to solve linear congruences of the form $ax \equiv b \pmod{M}$. If the congruence has solutions, your program should give a single congruence $x \equiv c \pmod{N}$ that describes them all. If the congruence has no solutions, your program must detect this.
11. You play Eve in this problem. You have intercepted six encrypted messages sent from Alice to Bob. The cryptosystem they are using converts a string (the plaintext) into an integer (the ciphertext); so the data you have intercepted consists of six integers. You have also obtained the source code Alice and Bob are using for their encryption; it is reproduced below on the last page (you can also find it in the Python starter code on hackerrank). Alice and Bob have a secret key k, which is a 1024-bit integer.

    Write a program to break Alice and Bob's encryption, and print the original six plaintext messages.

Note. Because Eve accomplishes this attack using only a selection of a few enciphered messages, this is called a *ciphertext-only attack*.

Hint. You don't need to focus on what's going on the in `encode` and `decode` functions (they use some syntax that may not be familiar). All you need to know about them is that they convert strings into integers and back. Focus on what happens in the other two functions.

```
# Encodes a given string as an integer.
def encode(text):
        code = 0
        for (loc,ch) in enumerate(text):
                code += ord(ch)*(1 << loc*8)
        return code

# Decodes an integer to a string.
def decode(code):
        text = ''
        while code > 0:
                byte = code & 0xFF
                text += chr(byte)
                code >>= 8
        return text

# Enciphers a given string (text) using a secret key (a positive integer).
def encipher(text,key):
        return key*encode(text)

# Deciphers a given integer (cipher) to return the original string.
def decipher(cipher,key):
        return decode(cipher/key)
```