This assignment looks longer than it is, simply because the setup for some of the problems is quite wordy. Do not be intimidated by this; the actual math to be done is not long.

1. Suppose that there are 1000 switches in a row (numbered 1 to 1000), each of which is initially in the "off" position. One thousand gnomes (numbered 1 through 1000) enter the room one by one, and each gnome flips some of the switches before leaving. Gnome 1 flips every switch, gnome 2 flips every even-numbered switch, gnome 3 flips every switch numbered with a multiple of 3, and so on. How many switches will be in the "on" position when all 1000 gnomes have passed through?
   Hint. First try working out what happens to the first 30 or so switches, and look for a pattern.

2. Recall that in RSA, the "public key" is a pair (m, e) (the modulus and the encrypting exponent), and the "private key" is a number f (the deciphering exponent) that is chosen so that
$$ef \equiv 1 \pmod{\phi(m)}.$$
   Euler's theorem guarantees that for any s *relatively prime to* m, $s^{ef} \equiv s \pmod{m}$, which is the key fact that allows the recipient to decrypt messages.

   (a) Show that if the modulus m is equal to a product pq of two different primes (as it is in RSA), then in fact $s^{ef} \equiv s \pmod{m}$) for *all* integers s, not just those relatively prime to m.
   Hint. Use the Chinese Remainder Theorem, and consider the two primes separately.

   (b) Suppose that $m = 45, e = 5$, and $f = 5$. Show that $ef \equiv 1 \pmod{\phi(m)}$, but that there is an integer s such that $s^{ef} \not\equiv s \pmod{m}$.

   Note. In fact, the congruence $s^{ef} \equiv s \pmod{m}$ (needed for RSA to work) is valid (for *all* s, not just those coprime to m) if and only if m is "square free," meaning that it is not divisible by any squares besides 1. I encourage you to try to prove this.

3. *Digital Signatures.*
   The main application of RSA is *encryption*, where one agent wishes to send a message to another agent (across a public channel) without eavesdroppers being able to tell what the message says. This problem discusses a second application, where RSA is used for *authentication*. Now the goal is not to keep a message secret, but instead to allow the recipient to verify that the message has not been forged by a third party. This problem describes a simplified version of RSA signatures. Once we've covered chapters 28 and 29, we'll discuss a signature algorithm that is more common in practice.

   Suppose that Alice has a public key (m, e), and that only she knows the private key f. Alice wishes to send a non-secret message s to Bob (where s is an integer between 0 and $m - 1$ inclusive). Meanwhile, a third agent, Mallory, has forged her own version of the message s (which might, for example, contain a virus).

   Bob receives both versions of s, but he cannot tell which sender is Alice and which is Mallory. In order to resolve this conundrum, he announces the following: each sender must send him a second number, t (called the "signature"). Bob will compute $t^e \mod m$ (where (m, e) is

Alice's public key). If he discovers that $t^e \equiv s \pmod{m}$, he will conclude that that message s was legitimate. Otherwise he will regard it as a forgery.

(a) What should Alice do to compute the number t? (This computation is called "signing" the message.)

(b) Why isn't Mallory able to forge a signature for her fake message s?

Note 1. This method of authenticating messages does have a security flaw: Mallory could choose t first, and then simply compute s from it. The pair (s, t) would then appear to Bob as a legitimate signed message from Alice. The downside for Mallory is that she doesn't get to decide what s says, so it will almost certainly be gibberish (which will clue Bob in that it's not from Alice after all). However, this flaw can be eliminated using something called a "hash function."

Note 2. The security of this system depends on the fact that Bob can have faith that (m, e) really is *Alice's* public key. So the integrity of the public key must be verified in advance. This is sometimes achieved by having a trusted third party, named Trent, meeting Alice in person (sometimes at an event called a "key signing party") and then signing her public key. This ensures that anyone with faith in Trent's public key can also have faith in Alice's public key.

4. (a) For each prime number p less than 20, make a list of the quadratic residues and quadratic non-residues of p.

(b) Let $A(p)$ denote the sum of the quadratic resides modulo p, and let $B(p)$ denote the sum of the non-resides. Compute $A(p)$ and $B(p)$ for all primes less than 20. For which of these primes does $A(p)$ equal $B(p)$?

(c) Make a conjecture about which primes p have $A(p) = B(p)$.