Written problems:

1. Textbook exercise 4.6 (ElGamal signature examples)
2. Textbook exercise 4.7 (ElGamal "blind signatures")

Programming problems:

1. When using ElGamal digital signatures, it is essential that Samantha always generates her ephemeral key at random (much like in ElGamal encryption). In this problem, you will study why it is particularly dangerous to use the same ephemeral key twice. You will be given the public ElGamal parameters `p,g`, Alice's public key `A`, two documents `d1,d2`, and valid signatures `(s11,s12),(s21,s22)` for the two documents (respectively). The two signatures were generated using the same ephemeral key. Write a function `extractKey(p,g,A,d1,s11,s12,d2,s21,s22)` that extracts and returns Alice's private key `a` from this information.

   Hint: if you carefully manipulate the two congruences Samantha used to sign the documents, you can derive a congruence of the form $ma \equiv b \pmod{p-1}$, where m and b are values you can compute and a is the private key that you are trying to find. Unfortunately, it is possible that m is not invertible modulo $p-1$. You can use the solution to an earlier problem to "solve" this congruence to obtain a congruence that may not determine a uniquely; you'll need to figure out how to get from here to the specific value of a.

2. Suppose that Samantha and Victor are using a variant of Elgamal signatures, in which the verification congruence that Victor will use is $s_1^{s_1} \cdot g^{s_2} \equiv A^d \pmod{p}$. Write a function `signElGamalVariation(p,g,a,d)`, which produces a valid signature in this system, given the public parameters p, g, Samantha's secret signing key a, and a document d.