Written problems:

1. Textbook exercise 6.1 (Elliptic curve arithmetic over $\mathbb{R}$)
2. Textbook exercise 6.5, parts (a) and (b) (Listing the points of an EC over $\mathbb{Z}/p\mathbb{Z}$)

   Hint. You can save some time by making two lists in advance: values of y^2 for various y and values of $x^3 + Ax + B$ for various values of x, then checking for numbers occurring in both lists)
3. Textbook exercise 6.6(a) (addition table for an elliptic curve over $\mathbb{Z}/5\mathbb{Z}$)
4. Textbook exercise 6.9 (listing all solutions n to an equation $Q = n \cdot P$ on an elliptic curve).
5. Textbook exercise 6.16. (A more concise way to send EC points; you should read Proposition 2.26 to do part (b))

Programming problems:

1. 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.
2. Devise a method to create "blind forgeries" for a given DSA public key. That is, write a function `dsaBlind(p,q,g,A)` given p, g and A as in DSA, generate integers (d, s_1, s_2) such that (s_1, s_2) is a valid signature for d for the verification key A. You will likely want to adapt the strategy from one of last weeek's problems from Elgamal to DSA. Your method should be non-deterministic; the grading script will give the same test case multiple times to check that the same answer is not returned each time.
3. Write a function `ecAdd(P,Q,A,B,p)` to compute the sum $P \oplus Q$ of two points on the Elliptic Curve over $\mathbb{Z}/p\mathbb{Z}$ defined by $Y^2 \equiv X^3 + AX + B \pmod{p}$. You may assume that P and Q are both valid points on the curve[1]. The points P and Q will be either pairs (x, y) of elements of $\mathbb{Z}/p\mathbb{Z}$, or the integer `0` (as a stand-in for the point $\mathcal{O}$ at infinity), and the function should return the result in the same format.
4. Write a function `ecMult(n,P,A,B,p)` that computes an integer multiple $n \cdot P$ of a point P on an elliptic curve $Y^2 \equiv X^3 + AX + B \pmod{p}$. Points will be formatted (x, y), with $0 \leq x, y < p$, while the point at infinity should be denoted simply as 0. Your code will need to be able to scale to very large values of n; I suggest adapting the fast-powering algorithm from modular arithmetic to elliptic curves.

[1] Though of course if you were using this code in real life, you should add some error handling that checks this.