**Written problems:**

1. Samantha and Victor agree to the following digital signature scheme. The public parameters and key creation are identical to those of ECDSA. The verification procedure is different: to decide whether $(s_1, s_2)$ is a valid signature for a document $d$, Victor computes

   $$\begin{aligned} w_1 &\equiv s_1^{-1} d \pmod{q} \\ w_2 &\equiv s_1^{-1} s_2 \pmod{q}, \end{aligned}$$

   then he checks to see whether or not

   $$x(w_1 G \oplus w_2 V)\%q = s_1.$$

   If so, he regards $(s_1, s_2)$ as a valid signature for $d$.

   Determine a signing procedure for this signature scheme, i.e. a procedure that Samantha can follow to produce a valid signature on a specific document. Your procedure should be non-deterministic, like in ECDSA.

2. Consider the following variant of EC Diffie-Hellman key exhange, in which Alice and Bob only exchange individual numbers, rather than both coordinates of a point on an elliptic curve.

   - **Public parameter creation:** same as in table 6.5.
   - **Private computations:** same as in table 6.5.
   - **Public exchange of values:** Alice sends *the $x$-coordinate* of $Q_A$ to Bob; Bob sends *the $x$-coordinate* of $Q_B$ to Alice.
   - **Further private computations**: Both Alice and Bob determine the *$x$-coordinate* of $(n_A \cdot n_B)P$. This is their shared secret value.

   (a) Prove that if $Q, Q'$ are two points on an elliptic curve with the same $x$-coordinate, and $n$ is any integer, then $nQ$ and $nQ'$ also have the same $x$-coordinate.

   (b) Describe how Alice is able to (efficiently) determine the shared secret, using only the information that she knows. You may assume that Alice has an efficient algorithm to determine square roots modulo $p$.

   (c) What advantages, if any, does this system have over the usual ECDH system described in table 6.5?

**Programming problems:**

1. Write a function `ecDLP(P,Q,A,B,p,q)` to solve the elliptic curve discrete logarithm problem, in cases where the prime $p$ is up to 28 bits. Here, $P, Q$ are points on the curve $Y^2 \equiv X^3 + AX + B \pmod{p}$, and you are given, for convenience, the number $q$ of points on the curve (which you may assume to be prime). The function should return the minimum nonnegative $n$ such that $n \cdot P = Q$. Note that if you find any such solution $n'$, then you can find the minimum solution by computing $n' \pmod{q}$ (we will discuss why this is true in class soon).

   A naive trial-and-error approach will earn partial credit, but to solve all test cases I recommend adapting the BSGS algorithm to the setting of elliptic curves. Note: one issue you may encounter is that it is not possible to place a "list" (e.g. `[2,3]`) into a Python `dict`. To resolve this, make sure all of your elliptic curve points (besides $\mathcal{O}$) are represented as "tuples" instead (e.g. `(2,3)`, with parentheses instead of brackets).

2. Write a function `ecdsaVerify(V,d,s1,s2)` that determines whether $(s_1, s_2)$ is a valid ECDSA signature for the document $d$ and the public (verification) key $V$ (see p. 461 in the 1st edition or p. 322 in the 2nd edition for notation regarding ECDSA), using Elliptic curve "P-384." You should look up the specifics of curve P-384, e.g. by finding the relevant information in the standards document at the link below. You will also need to look up how to convert hexadecimal strings to integers in Python.

   `https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.186-4.pdf`

   You may enjoy looking through the standards document to see what other sorts of information it contains.