Note: this exam was an open-book and open-notes take-home exam.

1. [12 points] Alice and Bob are using RSA encryption. Alice publishes the following public key.

$$\begin{aligned} N &= 64777 \\ e &= 11 \end{aligned}$$

Bob sends the following ciphertext to Alice.

$$c = 42675$$

Use a brute-force approach to extract Alice's private key and determine Bob's plaintext m. You should use a computer for the computations, but clearly explain what you have done and how you have used the computer to do it.

2. [12 points] Note (2024): our Midterm 2 will not cover Elliptic curves, so you can skip this question. You may want to return to it to review for the final exam. Let E be the elliptic curve over $\mathbb{R}$ defined by the equation

$$y^2 = x^3 - x + 1.$$

Let P be the point $(1, 1)$. Determine the point $(-3) \cdot P$. Do the arithmetic by hand, and show your computations (but you may use a computer to check your arithmetic).

3. [12 points] Suppose that Samantha and Victor use the following variation on DSA. The system uses the same public parameters p, q, g as DSA (notation as in Table 4.3 of the textbook). Rather that publishing a single verification key A, Samantha chooses *two* secret signing keys a_1, a_2, and publishes two verification keys A_1, A_2 such that

$$\begin{aligned} A_1 &\equiv g^{a_1} \pmod{p}, \text{ and} \\ A_2 &\equiv g^{a_2} \pmod{p}. \end{aligned}$$

A signature on a document D consists of a pair of integers (S_1, S_2). When he receives a document D with signature (S_1, S_2), Victor will use the following verification procedure.

- Compute $V_1 \equiv DS_2^{-1} \pmod{q}$ and $V_2 \equiv S_1 S_2^{-1} \pmod{q}$ (as in DSA).
- Verify that

$$\left((A_1^{V_1} A_2^{V_2})\%p\right)\%q = S_1.$$

  (If this equation is false, the signature is considered invalid.)

Devise an (efficient) *signing procedure* that Samantha could follow to produce valid signatures. Write out your procedure as a Python function (receiving a document D and the parameters and private keys as input), and prove that your program returns a valid signature according to Victor's procedure above.

I will not deduct points for syntax errors, as long as it is clear what you mean. You may use the built-in Python function `pow` for fast modular powers, and you may assume that you have implemented an efficient function `modinv` to compute modular inverses. *EDIT: you may also make use of any functions from the `random` library to generate random numbers.* Any other needed helper functions should be implemented in your written solution.

4. This problem concerns an adaptation of the Pohlig-Hellman algorithm to the Elliptic Curve Discrete Logarithm Problem (ECDLP).

   Suppose that E is an elliptic curve over $\mathbb{Z}/p\mathbb{Z}$, and $P \in E$ is a point of order 143. Note that 143 factors as $11 \cdot 13$. Suppose that Q is another point on the curve, and that Eve wishes to find an integer n such that $n \cdot P = Q$.

   Define four more points on E as follows.

$$\begin{aligned} P_1 &= 13 \cdot P \\ Q_1 &= 13 \cdot Q \\ P_2 &= 11 \cdot P \\ Q_2 &= 11 \cdot Q \end{aligned}$$

   (a) [4 points] Prove that $\mathrm{ord}_E(P_1) = 11$ and $\mathrm{ord}_E(P_2) = 13$.

   (b) [6 points] Suppose $Q \in E$ is another point on the curve, and that $n_1, n_2 \in \mathbb{Z}$ are integers such that $n_1 \cdot P_1 = Q_1$ and $n_2 \cdot P_2 = Q_2$. Prove that if an integer n satisfies $n \cdot P = Q$ then n must satisfy the following two congruences.

$$\begin{aligned} n &\equiv n_1 \pmod{11} \\ n &\equiv n_2 \pmod{13} \end{aligned}$$

   (The converse is also true, but you do not need to prove it).

   (c) [2 points] Briefly explain why part (b) may be useful to Eve in her attempt to solve $n \cdot P = Q$.