Name: ________________________________

- The exam is *double-sided*. Make sure to read both sides of each page.
- The time limit is 50 minutes.
- No calculators are permitted.
- You are permitted one page of notes, front and back.
- The textbook’s summary tables for the systems we have studied are provided on the last sheet. You may detach this sheet for easier reference.
- For any problem asking you to write a program, you may write in a language of your choice or in pseudocode, as long as your answer is sufficiently specific to tell the runtime of the program.
This page intentionally left blank.
(1) Alice and Bob are using Elgamal encryption, with public parameters $p = 29, g = 19$. The summary table for Elgamal is on the back page of this packet. There is also a multiplication table for $\mathbb{Z}/29\mathbb{Z}$, so that you do not need to do the arithmetic by hand.

(a) Alice chooses the private key $a = 9$. What is her public key? Express your answer as an integer in \{0, 1, 2, \ldots, 28\}.

Part (b) on reverse side. (3 points)
(b) Bob sends ciphertext (8, 25) to Alice. What is the plaintext? Express your answer as an integer in \{0, 1, 2, \cdots, 28\}. 

(3 points)
(2) (a) Suppose that $a \mid m$. Prove that the congruence $ax \equiv ab \pmod{m}$ holds if and only if the congruence $x \equiv b \pmod{m/a}$ holds (all variables are integers).

Part (b) on reverse side. (3 points)
(b) Suppose that $\gcd(a, m) = 1$. Prove that the congruence $ax \equiv ab \pmod{m}$ holds if and only if the congruence $x \equiv b \pmod{m}$ holds.

(3 points)
(3) Write a program that reduces breaking Diffie-Hellman key exchange to breaking Elgamal encryption.

More precisely: suppose that Eve has written a function break_elg\( (p, g, A, c1, c2) \) with the following behavior: if the arguments are as in Table 2.3 (back of the packet), then this function will return \( m \). Make use of this function to write a function break_dh\( (p, g, A, B) \), which accepts arguments as labeled in Table 2.2 and returns the corresponding shared secret.

You may use any functions that are built into Python (or any language you have written your homework in), plus the hypothetical function break_elg. You may also assume that you have already written a function ext_euclid\( (a, b) \), with the following behavior: given two positive integers \( a, b \), this function returns a list of three integers \( [u, v, d] \), where \( d = \gcd(a, b) \) and \( au + bv = d \).

For full points, your program must require at most \( O(\log p) \) arithmetic operations (not counting any operations needed to compute break_elg).
Additional space for problem 3.
Let $p$ be a prime number, and $g \in (\mathbb{Z}/p\mathbb{Z})^*$. 

(a) Prove that if the order of $g \pmod{p}$ is 17, then $p \equiv 1 \pmod{17}$. 

Part (b) on reverse side. (3 points)
(b) Prove conversely that if $p \equiv 1 \pmod{17}$, then there exists some element $g \in (\mathbb{Z}/p\mathbb{Z})^*$ of order 17.
(5) Suppose that Alice and Bob use the following variant of Elgamal. The parameters
$p, g$ are as in table 2.3, and Alice chooses a secret key $a$ and public key $A$ in the same
manner as in table 2.3. However, instead of following table 2.3, Bob computes his
ciphertext as followings: he chooses a random element $k$, and computes
\[ c_1 \equiv A^k \pmod{p}, \]
\[ c_2 \equiv m \cdot g^k \pmod{p}, \]
then sends $(c_1, c_2)$ to Alice.

Explain how Alice can efficiently decipher messages, i.e. determine $m$ from $(c_1, c_2)$. You will need to place a restriction on Alice’s original choice of private key $a$ in order
for decryption to be possible; clearly state this restriction.
Additional space for problem 5.
Additional space for work.
Additional space for work.
### Diffie–Hellman key exchange

<table>
<thead>
<tr>
<th>Alice</th>
<th>Bob</th>
</tr>
</thead>
<tbody>
<tr>
<td>Choose a secret integer (a).</td>
<td>Choose a secret integer (b).</td>
</tr>
<tr>
<td>Compute (A = g^a \pmod{p}).</td>
<td>Compute (B = g^b \pmod{p}).</td>
</tr>
</tbody>
</table>

### Encryption

**Choose plaintext \(m\).**
- Choose random element \(k\).
- Use Alice’s public key \(A\) to compute \(c_1 = g^k \pmod{p}\) and \(c_2 = m g^k \pmod{p}\).
- Send ciphertext \((c_1, c_2)\) to Alice.

### Decryption

**Compute \((c_1)^{-1} \cdot c_2 \pmod{p}\).**
- This quantity is equal to \(m\).
This page intentionally left blank.