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

$$\begin{aligned} N &= 35 \\ e &= 5 \end{aligned}$$

   Bob sends the following ciphertext to Alice.

$$c = 17$$

   Use a brute-force approach to extract Alice's private key and determine Bob's plaintext m. Clearly show all steps. There is a multiplication table for $\mathbb{Z}/35\mathbb{Z}$ at the back of the exam packet, so you do not need to do those computations by hand (feel free to detach it for convenient reference).

2. [12 points] Solve the following system of three congruences. Show all steps, and answer in the form of a single congruence $x \equiv \cdots \pmod{\cdots}$ that is satisfied if and only if the original three congruences are satisfied.

$$\begin{aligned} x &\equiv 1 \pmod 2 \\ x &\equiv 1 \pmod 3 \\ x &\equiv 4 \pmod{13} \end{aligned}$$

3. [12 points] Samantha is using the Elgamal digital signature algorithm, with public parameters g, p, private key a, and public key A. Here we follow the notation of Table 4.2 (provided at the back of the exam packet). You may also assume that g is a primitive root modulo p (as in the summary table).

   Samantha signs a document D, publishing signature (S_1, S_2). Later, she signs a document D', publishing signature (S_1', S_2'). Unfortunately, Samantha has generated her ephemeral keys poorly! Eve notices this by observing that the following congruence holds.

$$S_1' \equiv S_1 \cdot g^2 \pmod{p}.$$

   For simplicity, you may also assume that $S_1 S_2' - S_1' S_2$ is a unit modulo $p - 1$.

   Help Eve steal Samantha's private key, by writing a formula for a that Eve could use to compute a using only published numbers and modular arithmetic.

4. This problem concerns some aspects of the Pohlig-Hellman algorithm. The purpose is to prove some basic facts discussed in describing that algorithm, using some specific numbers.

   Suppose that p is a prime number, and $g \in (\mathbb{Z}/p\mathbb{Z})^\times$ has order 143. Note that 143 factors as $11 \cdot 13$. Suppose that h is another element of $(\mathbb{Z}/p\mathbb{Z})^\times$, and that Eve wishes to find an integer n such that $g^n \equiv h \pmod{p}$.

Define four more elements of $(\mathbb{Z}/p\mathbb{Z})^\times$ as follows.

$$\begin{aligned} g_1 &\equiv g^{13} \pmod{p} \\ h_1 &\equiv h^{13} \pmod{p} \\ g_2 &\equiv g^{11} \pmod{p} \\ h_2 &\equiv h^{11} \pmod{p} \end{aligned}$$

(a) [4 points] Prove that $\mathrm{ord}_p(g_1) = 11$.
(Similarly, $\mathrm{ord}_p(g_2) = 13$; you do not need to prove this, but you may assume it in part b)

(b) [6 points] Suppose that $n_1, n_2 \in \mathbb{Z}$ satisfy $g_1^{n_1} \equiv h_1 \pmod{p}$ and $g_2^{n_2} \equiv h_2 \pmod{p}$. Prove that if an integer n satisfies $g^n \equiv h \pmod{p}$ 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 $g^n \equiv h \pmod{p}$.

Reference tables

Public parameter creation	
A trusted party chooses and publishes a (large) prime p and an integer g having large prime order in $\mathbb{F}_p^*$.	
Private computations	
Alice	**Bob**
Choose a secret integer a. Compute $A \equiv g^a \pmod p$.	Choose a secret integer b. Compute $B \equiv g^b \pmod p$.
Public exchange of values	
Alice sends A to Bob $\longrightarrow$ A	
B $\longleftarrow$ Bob sends B to Alice	
Further private computations	
Alice	**Bob**
Compute the number $B^a \pmod p$.	Compute the number $A^b \pmod p$.
The shared secret value is $B^a \equiv (g^b)^a \equiv g^{ab} \equiv (g^a)^b \equiv A^b \pmod p$.	

Table 2.2: Diffie–Hellman key exchange

Public parameter creation	
A trusted party chooses and publishes a large prime p and an element g modulo p of large (prime) order.	
Alice	**Bob**
Key creation	
Choose private key $1 \le a \le p-1$. Compute $A = g^a \pmod p$. Publish the public key A.	
Encryption	
	Choose plaintext m. Choose random element k. Use Alice's public key A to compute $c_1 = g^k \pmod p$ and $c_2 = mA^k \pmod p$. Send ciphertext (c_1, c_2) to Alice.
Decryption	
Compute $(c_1^a)^{-1} \cdot c_2 \pmod p$. This quantity is equal to m.	

Table 2.3: Elgamal key creation, encryption, and decryption

Bob	**Alice**
Key creation	
Choose secret primes p and q. Choose encryption exponent e with $\gcd(e, (p-1)(q-1)) = 1$. Publish $N = pq$ and e.	
Encryption	
	Choose plaintext m. Use Bob's public key (N, e) to compute $c \equiv m^e \pmod N$. Send ciphertext c to Bob.
Decryption	
Compute d satisfying $ed \equiv 1 \pmod{(p-1)(q-1)}$. Compute $m' \equiv c^d \pmod N$. Then m' equals the plaintext m.	

Table 3.1: RSA key creation, encryption, and decryption

Samantha	**Victor**
Key creation	
Choose secret primes p and q. Choose verification exponent e with $\gcd(e, (p-1)(q-1)) = 1$. Publish $N = pq$ and e.	
Signing	
Compute d satisfying $de \equiv 1 \pmod{(p-1)(q-1)}$. Sign document D by computing $S \equiv D^d \pmod N$.	
Verification	
	Compute $S^e \bmod N$ and verify that it is equal to D.

Table 4.1: RSA digital signatures

Public parameter creation	
A trusted party chooses and publishes a large prime p and primitive root g modulo p.	
Samantha	**Victor**
Key creation	
Choose secret signing key $1 \le a \le p-1$. Compute $A = g^a \pmod p$. Publish the verification key A.	
Signing	
Choose document D mod p. Choose random element $1 < k < p$ satisfying $\gcd(k, p-1) = 1$. Compute signature $S_1 \equiv g^k \pmod p$ and $S_2 \equiv (D - aS_1)k^{-1} \pmod{p-1}$.	
Verification	
	Compute $A^{S_1} S_1^{S_2} \bmod p$. Verify that it is equal to $g^D \bmod p$.

Table 4.2: The Elgamal digital signature algorithm

Public parameter creation	
A trusted party chooses and publishes large primes p and q satisfying $p \equiv 1 \pmod q$ and an element g of order q modulo p.	
Samantha	**Victor**
Key creation	
Choose secret signing key $1 \le a \le q-1$. Compute $A = g^a \pmod p$. Publish the verification key A.	
Signing	
Choose document D mod q. Choose random element $1 < k < q$. Compute signature $S_1 \equiv (g^k \bmod p) \bmod q$ and $S_2 \equiv (D + aS_1)k^{-1} \pmod q$.	
Verification	
	Compute $V_1 \equiv DS_2^{-1} \pmod q$ and $V_2 \equiv S_1 S_2^{-1} \pmod q$. Verify that $(g^{V_1} A^{V_2} \bmod p) \bmod q = S_1$.

Table 4.3: The digital signature algorithm (DSA)

Multiplication table modulo 35:

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
2	0	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34
3	0	3	6	9	12	15	18	21	24	27	30	33	1	4	7	10	13	16
4	0	4	8	12	16	20	24	28	32	1	5	9	13	17	21	25	29	33
5	0	5	10	15	20	25	30	0	5	10	15	20	25	30	0	5	10	15
6	0	6	12	18	24	30	1	7	13	19	25	31	2	8	14	20	26	32
7	0	7	14	21	28	0	7	14	21	28	0	7	14	21	28	0	7	14
8	0	8	16	24	32	5	13	21	29	2	10	18	26	34	7	15	23	31
9	0	9	18	27	1	10	19	28	2	11	20	29	3	12	21	30	4	13
10	0	10	20	30	5	15	25	0	10	20	30	5	15	25	0	10	20	30
11	0	11	22	33	9	20	31	7	18	29	5	16	27	3	14	25	1	12
12	0	12	24	1	13	25	2	14	26	3	15	27	4	16	28	5	17	29
13	0	13	26	4	17	30	8	21	34	12	25	3	16	29	7	20	33	11
14	0	14	28	7	21	0	14	28	7	21	0	14	28	7	21	0	14	28
15	0	15	30	10	25	5	20	0	15	30	10	25	5	20	0	15	30	10
16	0	16	32	13	29	10	26	7	23	4	20	1	17	33	14	30	11	27
17	0	17	34	16	33	15	32	14	31	13	30	12	29	11	28	10	27	9
18	0	18	1	19	2	20	3	21	4	22	5	23	6	24	7	25	8	26
19	0	19	3	22	6	25	9	28	12	31	15	34	18	2	21	5	24	8
20	0	20	5	25	10	30	15	0	20	5	25	10	30	15	0	20	5	25
21	0	21	7	28	14	0	21	7	28	14	0	21	7	28	14	0	21	7
22	0	22	9	31	18	5	27	14	1	23	10	32	19	6	28	15	2	24
23	0	23	11	34	22	10	33	21	9	32	20	8	31	19	7	30	18	6
24	0	24	13	2	26	15	4	28	17	6	30	19	8	32	21	10	34	23
25	0	25	15	5	30	20	10	0	25	15	5	30	20	10	0	25	15	5
26	0	26	17	8	34	25	16	7	33	24	15	6	32	23	14	5	31	22
27	0	27	19	11	3	30	22	14	6	33	25	17	9	1	28	20	12	4
28	0	28	21	14	7	0	28	21	14	7	0	28	21	14	7	0	28	21
29	0	29	23	17	11	5	34	28	22	16	10	4	33	27	21	15	9	3
30	0	30	25	20	15	10	5	0	30	25	20	15	10	5	0	30	25	20
31	0	31	27	23	19	15	11	7	3	34	30	26	22	18	14	10	6	2
32	0	32	29	26	23	20	17	14	11	8	5	2	34	31	28	25	22	19
33	0	33	31	29	27	25	23	21	19	17	15	13	11	9	7	5	3	1
34	0	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18

Multiplication table modulo 35, continued:

	18	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**	**33**	**34**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
2	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31	33
3	19	22	25	28	31	34	2	5	8	11	14	17	20	23	26	29	32
4	2	6	10	14	18	22	26	30	34	3	7	11	15	19	23	27	31
5	20	25	30	0	5	10	15	20	25	30	0	5	10	15	20	25	30
6	3	9	15	21	27	33	4	10	16	22	28	34	5	11	17	23	29
7	21	28	0	7	14	21	28	0	7	14	21	28	0	7	14	21	28
8	4	12	20	28	1	9	17	25	33	6	14	22	30	3	11	19	27
9	22	31	5	14	23	32	6	15	24	33	7	16	25	34	8	17	26
10	5	15	25	0	10	20	30	5	15	25	0	10	20	30	5	15	25
11	23	34	10	21	32	8	19	30	6	17	28	4	15	26	2	13	24
12	6	18	30	7	19	31	8	20	32	9	21	33	10	22	34	11	23
13	24	2	15	28	6	19	32	10	23	1	14	27	5	18	31	9	22
14	7	21	0	14	28	7	21	0	14	28	7	21	0	14	28	7	21
15	25	5	20	0	15	30	10	25	5	20	0	15	30	10	25	5	20
16	8	24	5	21	2	18	34	15	31	12	28	9	25	6	22	3	19
17	26	8	25	7	24	6	23	5	22	4	21	3	20	2	19	1	18
18	9	27	10	28	11	29	12	30	13	31	14	32	15	33	16	34	17
19	27	11	30	14	33	17	1	20	4	23	7	26	10	29	13	32	16
20	10	30	15	0	20	5	25	10	30	15	0	20	5	25	10	30	15
21	28	14	0	21	7	28	14	0	21	7	28	14	0	21	7	28	14
22	11	33	20	7	29	16	3	25	12	34	21	8	30	17	4	26	13
23	29	17	5	28	16	4	27	15	3	26	14	2	25	13	1	24	12
24	12	1	25	14	3	27	16	5	29	18	7	31	20	9	33	22	11
25	30	20	10	0	25	15	5	30	20	10	0	25	15	5	30	20	10
26	13	4	30	21	12	3	29	20	11	2	28	19	10	1	27	18	9
27	31	23	15	7	34	26	18	10	2	29	21	13	5	32	24	16	8
28	14	7	0	28	21	14	7	0	28	21	14	7	0	28	21	14	7
29	32	26	20	14	8	2	31	25	19	13	7	1	30	24	18	12	6
30	15	10	5	0	30	25	20	15	10	5	0	30	25	20	15	10	5
31	33	29	25	21	17	13	9	5	1	32	28	24	20	16	12	8	4
32	16	13	10	7	4	1	33	30	27	24	21	18	15	12	9	6	3
33	34	32	30	28	26	24	22	20	18	16	14	12	10	8	6	4	2
34	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1