Please show your work and write only in **pen**. Notes are forbidden. Calculators, and all other electronic devices, are forbidden. Brains are encouraged, but at most one (your own!) may be used per exam.

1: Please compute the Jacobi symbol $\left( \frac{99}{1003} \right)$.

The key trick to remember with Jacobi symbols is to repeatedly factor out 2-parts and flip (via quadratic reciprocity) until the numbers get small enough to be trivial. Following the Theorem from Page 91 of our book (which will be reproduced for you on a cheat sheet), we proceed as follows:

$$\left( \frac{99}{1003} \right) = -\left( \frac{1003}{99} \right) \quad \text{(By Assertion (5), since } 99 = 1003 \equiv 3 \mod 4 \text{.)}$$

$$= -\left( \frac{13}{99} \right) \quad \text{(By Assertion (1), since } 1003 = 10 \cdot 99 + 13 \text{.)}$$

$$= -\left( \frac{99}{13} \right) \quad \text{(By Assertion (5), since } 13 = 1 \mod 4 \text{.)}$$

$$= -\left( \frac{8}{13} \right) \quad \text{(By Assertion (1), since } 99 = 7 \cdot 13 + 8 \mod 13 \text{.)}$$

$$= -\left( \frac{2}{13} \right) \quad \text{(Since } 8 = 2^2 \cdot 2 \text{ has a square root mod } 13 \text{ if and only if } 2 \text{ has a square root mod } 13 \text{.)}$$

$$= -(−1) \quad \text{(By Assertion (4), since } 13 = 5 \mod 8 \text{.)}^2$$

So the answer is $[1]$.

---

1 My edition of our book appears to have a typo in Assertion (4): the congruence should be mod 8, not mod $n$. 

---

Instructor: J. Maurice Rojas
Suppose \((p, \alpha, \beta) = (43933, 2, 3)\) is the public part of an instance of the El Gamal digital signature scheme, and you would like to break this instance, i.e., you would like to find the unique \(a \in \{2, \ldots, 43931\}\) such that \(\alpha^a = \beta\), so you can start forging signatures on messages.

(a): Please find \(a \mod 3\).
(b): Please find \(a \mod 7\).
(c): Suppose a friend tells you that \(a \mod 2092 = 186\). Find \(a\).

The Pohlig-Hellman Method (covered in class) tells us that, for small \(q\), we can efficiently recover the last base-\(q\) digit, \(a_0\), of \(a\). This boils down to computing 
\[\alpha_0 := \alpha^{q-1} \quad \text{and} \quad \beta_0 := \beta^{q-1},\]
and then a (small) brute-force search find which \(a_0 \in \{0, \ldots, q-1\}\) satisfies \(\alpha_0^{a_0} \equiv \beta_0 \mod p\). It will thus behoove us to find the binary expansions of \(\frac{43933-1}{3}\) and \(\frac{43933-1}{2}\): they are respectively  
\[14644 = 2^{13} + 2^{12} + 2^{11} + 2^8 + 2^5 + 2^4 + 2^2\]
and  
\[6276 = 2^{12} + 2^{11} + 2^7 + 2^2.\]

The squares mod 43933 needed to get us started are then the following:  
\[2^2 = 4, \quad 2^4 = 16, \quad 2^8 = 256, \quad 2^{16} = 21603, \quad 2^{25} = 33283, \quad 2^{32} = 31427, \quad 2^{64} = 42489, \quad 2^{128} = 20285, \quad 2^{256} = 4747, \quad 2^{512} = 40313, \quad 2^{1024} = 12366, \quad 2^{2048} = 31116, \quad 2^{4096} = 10002\]
and  
\[3^2 = 9, \quad 3^4 = 81, \quad 3^8 = 6561, \quad 3^{16} = 36314, \quad 3^{32} = 13668, \quad 3^{64} = 11108, \quad 3^{128} = 23800, \quad 3^{256} = 11831, \quad 3^{512} = 2023, \quad 3^{1024} = 6760, \quad 3^{2048} = 7280, \quad 3^{4096} = 15202, \quad 3^{8192} = 13224.\]

**Part (a):** Using our preliminary computations, we obtain  
\[\alpha_0 = 2^{2^{12}+2^{11}+3^2+2^1+3^2} = 10002 \cdot 31116 \cdot 12366 \cdot 20285 \cdot 33283 \cdot 21603 \cdot 16 = 34108 \mod 43933\]
and  
\[\beta_0 = 3^{2^{12}+2^{11}+3^2+2^1+3^2} = 13224 \cdot 15202 \cdot 7280 \cdot 11831 \cdot 13668 \cdot 36314 \cdot 81 = 34108 \mod 43933.\]

So then, it is clear straightaway that \(\alpha_0^1 = \beta_0\) and thus **\(a \mod 3\) is exactly 1.**

**Part (b):** Similar to Part (a), we obtain that  
\[\alpha_0 = 2^{6276} = 2^{12^1}+12^{11}+2^{12^1}+2^{11}+2^2 = 31116 \cdot 12366 \cdot 42489 \cdot 16 = 15760 \mod 43933\]
and  
\[\beta_0 = 3^{6276} = 3^{12^1}+2^{11}+3^2+2^1+3^2 = 15202 \cdot 7280 \cdot 23800 \cdot 81 = 24351 \mod 43933.\]

So now we start our brute-force search: \(\alpha_0^2 = 24351 \mod 43933\), so we need not go any further. **\(a \mod 7\) is thus clearly 2.**

**Part (c):** One could proceed directly from a 3 modulus version of the Chinese Remainder Theorem (CRT). However, it is easier here to simply find \(a \mod 21\) and then use the simpler version of the CRT we saw in class.

In particular, enumerating elements of the arithmetic progression 7\(j+2\) we obtain \(\{9, 16, 23, \ldots\}\) and it is then clear that 16 \(\equiv 1 \mod 3\). So \(a \equiv 16 \mod 21\). Since 21 \(\cdot 2092 = 43932\) and \(a \in \{0, \ldots, 43931\}\), it is then clear that the CRT applied to the moduli 21 and 2092 will give us \(a\).

A simple Extended Euclidean Algorithm calculation gives us  
\[797 \cdot 21 + (-8) \cdot 2092 = 1.\]

In other words, \(a \mod 43932\) should be 16 \(\cdot 2092 \cdot (-8) + 186 \cdot 21 \cdot 797 = 33658 \mod 43932\). So **\(a = 33658\).**

**Note:** The numbers you’d encounter in an actual final would be a lot smaller.
3: Please review your midterm (particularly correcting any errors you may have made) and your earlier homeworks and quizzes.
Consider $\mathbb{F}_{343}$ realized as $\mathbb{F}_7[t]/(t^3 + t + 1)$. Please express $\frac{1}{t}$, $\frac{1}{t+1}$, and $\frac{1}{t^2}$ as polynomials in $t$ of degree no more than 2.

This problem was done in class. However, for the sake of checking answers, here is what Maple says:

$$\frac{1}{t} = 6t^2 + 6 \text{ (or } -t^2 - 1)$$

$$\frac{1}{t+1} = t^2 + 6t + 2 \text{ (or } t^2 - t + 2)$$

$$\frac{1}{t^2} = t^2 + 6t + 1 \text{ (or } t^2 - t + 1)$$

Do please remember that while one can try to be clever with various algebraic identities, the safest fall-back to compute such inverses is the good old Extended Euclidean Algorithm.
Find the sum of the points \((1,5)\) and \((9,3)\) on the elliptic curve \(C\) defined by \(y^2 = x^3 + 2x + 3\) over \(\mathbb{F}_{19}\).

Applying the elliptic curve addition law specifically to our curve \(C\), we obtain that the group sum of \((x_1, y_1)\) and \((x_2, y_2)\) is \((x_3, y_3)\) where
\[
x_3 := m^2 - x_1 - x_2, \\
y_3 := m(x_1 - x_3) - y_1,
\]
and
\[
m = \frac{3x_1^2 + 2}{2y_1} \quad \text{or} \quad \frac{y_2 - y_1}{x_2 - x_1},
\]
according as \((x_1, y_1) = (x_2, y_2)\) or not.

(One should of course also remember that \((x_3, y_3)\) is the point at infinity, in the event of an infinite slope.)

So for the points at hand, we simply get
\[
m = \frac{3 \cdot 5}{9 - 1} = \frac{15}{8} = \frac{1}{4} = -5 = 14 \mod 19
\]
and thus
\[
x_3 = 14^2 - 1 - 9 = (-5)^2 - 1 - 9 = 25 - 1 - 9 = 15 \mod 19
\]
and
\[
y_3 = 14(1 - 15) - 5 = -5(-14) - 5 = -5 \cdot 5 - 5 = -30 = 8 \mod 19.
\]
So our answer is \([(15, 8)]\).
Suppose $A$ is a 3 digit integer of the form $21a$ where $a$ is unknown. Please find a value for $a$ such that there is a point on the elliptic curve defined by $y^2 = x^3 + 7x + 15$ (over $\mathbb{F}_{593}$) with $x$-coordinate $A$. Note: This kind of calculation, over much larger fields, is basic in converting plain-text to points on elliptic curves over finite fields.

This question is merely a Jacobi symbol question in disguise. In particular, once one tabulates the values of $x^3 + 7x + 15 \mod 593$ as $x$ ranges over 
\{210 + 0, 210 + 1, \ldots, 210 + 9\},

it is clear that there is a $y \in \mathbb{F}_{593}$ with $(x, y)$ lying on our curve if and only if $x^3 + 7x + 15$ has a square root mod 593.

So after some work, our first table is as follows:

<table>
<thead>
<tr>
<th>$x$</th>
<th>210</th>
<th>211</th>
<th>212</th>
<th>213</th>
<th>214</th>
<th>215</th>
<th>216</th>
<th>217</th>
<th>218</th>
<th>219</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x^3 + 7x + 15 \mod 593$</td>
<td>418</td>
<td>524</td>
<td>117</td>
<td>389</td>
<td>160</td>
<td>29</td>
<td>2</td>
<td>85</td>
<td>284</td>
<td>12</td>
</tr>
</tbody>
</table>

So after some work, our first table So now we merely compute Jacobi symbols. The resulting table is then the following:

<table>
<thead>
<tr>
<th>$x$</th>
<th>210</th>
<th>211</th>
<th>212</th>
<th>213</th>
<th>214</th>
<th>215</th>
<th>216</th>
<th>217</th>
<th>218</th>
<th>219</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(x^3 + 7x + 15)_{593}$</td>
<td>$-1$</td>
<td>$-1$</td>
<td>$-1$</td>
<td>$-1$</td>
<td>$1$</td>
<td>$1$</td>
<td>$-1$</td>
<td>$1$</td>
<td>$-1$</td>
<td>$-1$</td>
</tr>
</tbody>
</table>

So we see that any $a \in \{5, 6, 8\}$ would work.
The Kraitchik-Lehmer Theorem gives a quick way to certify that an integer \( N \) is prime: exhibit a \((k + 1)\)-tuple of integers \((a; p_1, \ldots, p_k)\) such that (1) \( a^{N-1} \equiv 1 \mod N \), (2) \( p_1, \ldots, p_k \) are all the prime divisors of \( N - 1 \), and (3) \( a^{(N-1)/p_i} \not\equiv 1 \mod N \) for all \( p_i \). Verifying properties (1)-(3) is then doable in time polynomial in \( \log N \). Note: The Kraitchik-Lehmer Theorem is recursive in the sense that the primality of each \( p_i \) (if not obvious) must be certified by its own \( k_i \)-tuple, involving smaller primes which may also have to be certified by their own certificates, and so on.

Verify that the primality of 5521 is certified by the 5-tuple \((11; 2, 3, 5, 23)\). In other words, do all the necessary computations implied by the Kraitchik-Lehmer Theorem.

Verifying Property (1) is simple: a direct computation, employing the binary expansion of 5520 as \( 2^4 + 2^7 + 2^8 + 2^{10} + 2^{12} \) and recursive squaring, gives us that \( 11^{5520} \equiv 1 \mod 5521 \). (Note in particular that one must compute \( 11^2, 11^2 \equiv (11^2)^2, 11^2 = (11^2)^2 \), \( \ldots \), \( 11^{2^{11}} = (11^{2^{11}})^2 \mod 5521 \) along the way.)

Verifying Property (2) is also easy, taking appropriate care with what is meant. In particular, we are merely given that 2, 3, 5, and 23 are the prime divisors of \( 5521 - 1 = 5520 \), but we don’t know their corresponding powers in the factorization of 5520. (Note that we should also check that 2, 3, 5, and 23 are prime, but these numbers are small enough to be known primes. In the case of a certificate involving larger primes, we would have needed certificates for the subsidiary \( p_i \).

Figuring out the powers of 2, 3, 5, and 23 dividing 5520 exactly is easy. First, 5520 is even, so \( 2 \mid 5520 \). Squaring to obtain higher powers of 2, we also see that \( 4 \mid 5520, 16 \mid 5520, \) but \( 256 \nmid 5520 \). So \( 2^4 \mid 5520 \) and we also easily see that \( 2^5 \nmid 5520 \). So the 2-part of 5520 is \( 2^4 \), and we can proceed with \( \frac{5520}{16} = 345 \).

The 3-part of 345 is easier to obtain: we immediately see that \( 3 \mid 345 \) but \( 9 \nmid 345 \). So the 3-part of 345 is just 3 and we can proceed with \( \frac{345}{3} = 115 \). The 5-part of 115 is just 5 and then we are left with \( \frac{115}{5} = 23 \). So \( 5520 = 2^4 \cdot 3 \cdot 5 \cdot 23 \), and we have thus verified Property (2).

To verify Property (3) we merely compute \( 11^{5520/2}, 11^{5520/3}, 11^{5520/5}, \) and \( 11^{5520/23} \mod 5521 \) via recursive squaring. In particular, the respective values are 5520, 5180, 1374, and 485, none of which is 1. So we are done.

Note: Clearly, this seems like an awful lot of effort to prove that 5521 is prime. However, for much larger numbers (e.g., with hundreds of digits and beyond), the certificates from the Kraitchik-Lehmer Theorem are much easier to verify than running any factorization algorithm. Also, the historical importance is that Pratt was able to prove around 1975 that the Kraitchik-Lehmer Theorem implies that primality detection lies in \( \text{NP} \).
In class, you’ve learned that computing square roots mod $N$ appears to be computationally difficult. In particular, one can factor large integers in probabilistic polynomial-time if computing square roots mod $N$ is doable in polynomial-time. Given that $N := 3972013$, and the following (shortened) list of square roots mod 3972013, show how to factor $N$.

<table>
<thead>
<tr>
<th>$x$</th>
<th>1</th>
<th>4</th>
<th>9</th>
<th>16</th>
<th>25</th>
<th>36</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sqrt{x}$</td>
<td>${\pm1, \pm661670}$</td>
<td>${\pm2, \pm1323340}$</td>
<td>${\pm3, \pm1985010}$</td>
<td>${\pm4, \pm1325333}$</td>
<td>${\pm5, \pm663663}$</td>
<td>${\pm6, \pm1993}$</td>
</tr>
</tbody>
</table>

You may recall from class that some classical factoring algorithms are based on computing $\gcd(N, x^2 - y^2)$ for random $x$ and $y$. The list above helps you avoid the work of making lots of random choices.

In particular, observing that $x^2 - y^2 = (x - y)(x + y)$, and observing that 36 has the smallest square roots in the list above, it appears that picking $x = 1993$ and $y = 6$ may be promising. Indeed, since $x^2 = y^2$ mod $N$, we must have that $N$ divides $(x - y)(x + y) = 1987 \cdot 1999$. Since $0 < 1987 \cdot 1999 < 2000 \cdot 2000 = 4,000,000$, we must then have that $N = 1987 \cdot 1999$. (One could also just multiply out 1987·1999 but it saves time to use the preceding inequalities.) Checking divisibility by the primes no greater than $\lfloor \sqrt{1999} \rfloor = 44$, we then see that 1987 and 1999 are prime and we have found the complete factorization of $N$. 
9: We’ve discussed $O$-bounds for the complexity of some basic arithmetic operations but one can be more precise: Via a 2008 paper of D. J. Bernstein (“Fast multiplication and its applications,” downloadable from http://cr.yp.to/arith.html), one can derive the following more explicit complexity upper bounds:

**Bound #1.** Multiplying two integers, each with at most $n$ digits, takes no more than $10n \log(n) \log(\log(n))$ bit operations.

**Bound #2.** Division with remainder, applied to two integers having no more than $n$ digits each, takes no more than $17n \log(n) \log(\log(n))$ bit operations.

**Bound #3.** The Jacobi symbol $\left( \frac{a}{N} \right)$, assuming $a$ and $N$ have no more than $n$ digits, can be computed within $20n^2$ bit operations.

When $N$ is prime and $a$ is an integer relatively prime to $N$, one can then easily prove that $a$ is a square if and only if $a^{(N-1)/2} \equiv 1 \pmod N$. So you can prove that a given $N$ is composite (without factoring) by exhibiting an $a$ that is a non-square mod $N$ with $a^{(N-1)/2} \equiv 1 \pmod N$. The **Solovay-Strassen Theorem**, combined with the Jacobi symbol $\left( \frac{a}{N} \right)$, gives a fast randomized compositeness test based on these ideas: when $N$ is odd and composite, at least half of the $a \in (\mathbb{Z}/n\mathbb{Z})^*$ satisfy $\left( \frac{a}{N} \right) \neq a^{(N-1)/2} \pmod N$.

We estimate the number of bit operations you need to prove that

975560017088786057282202253758453583177327752739921656394760090611523

is composite (if it really is composite) with probability at least $\frac{10^{23}}{1024}$.

Let $N$ denote the large number stated above. As discussed in class, the Solovay-Strassen Theorem implies that we can pick 10 random $a$ in $(\mathbb{Z}/N\mathbb{Z})^*$ and check the values of $\left( \frac{a}{N} \right)$ and $a^{(N-1)/2} \pmod N$ to verify compositeness. So then, if $N$ really is composite, the probability that we keep finding $\left( \frac{a}{N} \right) = a^{(N-1)/2}$ mod $N$ is at most $\frac{1}{1024}$. Put another way, with probability at least $\frac{10^{23}}{1024}$, we will actually find a certificate for the compositeness of $N$: an $a \in (\mathbb{Z}/N\mathbb{Z})^*$ with $\left( \frac{a}{N} \right) \neq a^{(N-1)/2} \pmod N$.

So now we merely need to estimate the number of bit operations to compute $\left( \frac{a}{N} \right)$ and $a^{(N-1)/2} \pmod N$, for $a$ potentially having as many digits as $N$, ten times. Let $\log_2(x)$ denote the base-2 log of $x$ and recall that $\log_2(10) < 3.322$. At this point, we need to make 3 hypotheses clear. **Note:** On an actual complexity question on the final, I would certainly make these hypotheses clearly known on a cheat sheet.

We now start counting bit operations. Counting digits, we see that our stated $N$ above (along with $(N-1)/2$) has exactly 69 digits.

So by Hypothesis 2, each computation of $\left( \frac{a}{N} \right)$ will take no more than $20 \cdot 69^2 = 95220$ bit operations.

Counting bit operations for computing $a^{(N-1)/2}$ is only slightly more involved. First, there is the computation of the binary expansion of $(N-1)/2$: this can be accomplished simply by converting digits, from low order to high order, successively adding bits in groups of 4. (This is because 0-9 involve at most 4 bits.) So converting $(N-1)/2$ takes no more than $(4+5) \cdot 69 = 621$ bit operations. (4 bit operations for the conversion of a digit to binary, +5 to add each new binary number to the current bit string.)

One then needs to compute $a^2$, $a^4 = (a^2)^2$, $a^8 = (a^4)^2$, $\ldots$, $a^{2^{229}} \pmod N$. This is because $N$ has no more than $\lceil 69 \cdot \log_2(10) \rceil = 230$ bits as a binary number. (The true number of bits is in fact 229 so we’re not too far off!) Thanks to Hypothesis 1, each squaring mod $N$ takes no more than $(10+17) \cdot 69 \log_2(69) \log_2(69)) < 27 \cdot 69 \cdot 6.11 \cdot 2.611 < 29721$ bit operations. So we need no more than $229 \cdot 29721 = 6806109$ bit operations to compute our dyadic powers of $a \pmod N$. To at last compute $a^{(N-1)/2}$
mod \( N \), we then observe that there are no more than 230 1s in the binary expansion of \((N-1)/2\). (There are actually 99.) So we need no more than 229 additional multiplications mod \( N \) to at last compute \( a^{(N-1)/2} \) mod \( N \). This means no worse than an additional \( 229 \cdot 27 \cdot 69 \lg(69) \lg(lg(69)) < 6806109 \) bit operations. In summary, computing \( a^{(N-1)/2} \) mod \( N \) takes no more than \( 621 + 6806109 + 6806109 = 13612763 \) bit operations.

To conclude, our total bit operation count is thus no worse than 
\[ 10(95220 + 13612763) = 137,079,830. \]

On most modern microprocessors, this means less than 0.1 seconds. However, this does not take into account cache usage, which makes the true time spent depend strongly on the underlying software implementation (and the skill and diligence of the programmer). Indeed, if one is truly serious about understanding the practical complexity of large integer calculations, one must also take into account the usage of memory.