Suppose I wish to divide an integer \( d \) (divisor) into an integer \( n \). The answer may not be an integer, but we can nevertheless describe the results precisely:

**Theorem (The Division Algorithm):** Suppose that \( d \) and \( n \) are positive integers. Then there exists a unique pair of numbers \( q \) (called the quotient) and \( r \) (called the remainder) such that \( n = qd + r \) and \( 0 \leq r < d \).

For example, if we wish to divide 17 into 50, we can satisfy the equation \( 50 = 17q + r \) with \( q = 1, r = 33 \) or with \( q = 3, r = -1 \). But only \( q = 2, r = 16 \) satisfies the requirements of the theorem (why?). So – if we divide 50 by 17, then the quotient is 2 and remainder 16. We say that the remainder on division of 50 by 17 is 16.

If we divide \( d \) into \( n \), and the remainder is \( r = 0 \), we say that \( d \) is a **divisor** of \( n \) and we say that \( n \) is a **multiple** of \( d \). We write \( d \equiv n \pmod{m} \).

**Modular arithmetic.**

Much of modern cryptography is based on modular arithmetic, which we now briefly review. We say that \( a \equiv b \pmod{m} \) if the integers \( a \) and \( b \) differ by a multiple of \( m \). (In other words \( m \mid (b - a) \)). The number \( m \) is called the **modulus**, and we say \( a \) and \( b \) are congruent modulo \( m \).

For example, \( 3 \equiv 17 \pmod{2} \) because \( 17 - 3 \) is divisible by 2. Thus two numbers \( a \) and \( b \) satisfy \( a \equiv b \pmod{2} \) if and only if \( a \) and \( b \) are both even or both odd (they have the same parity). Similarly \( 5 \equiv 21 \pmod{4} \) and \( 17 \equiv 11 \pmod{3} \) and \( 35 \equiv -5 \pmod{8} \), but \( 11 \not\equiv 4 \pmod{3} \) and \( -141 \not\equiv 19 \pmod{30} \).

**Proposition 1.** The following two statements are equivalent:

1. \( a \equiv b \pmod{m} \).
2. \( a \) and \( b \) leave the same remainder when divided by \( m \).

Proof. (1) \(\Rightarrow\) (2). If \( a \equiv b \pmod{m} \) then \( a - b = mn \). Let \( r_b \) be the remainder when \( b \) is divided by \( m \). Then \( b = qm + r_b \) where \( 0 \leq r_b < m \). But then \( a = b + mn = qm + r_b + mn = (q + n)m + r_b \), showing that \( r_b \) is also the remainder when \( a \) is divided by \( m \).

(2) \(\Rightarrow\) (1). If \( a \) and \( b \) leave the same remainder \( r \) when divided by \( m \), then \( a = mq_a + r \) and \( b = mq_b + r \) (where \( 0 \leq r < m \)) and so \( a - b = mq_a - mq_b = m(q_a - q_b) \). Thus \( a \equiv b \pmod{m} \). \(\square\)

Thus \( 36 \equiv 87 \pmod{17} \) because \( 87 - 36 = 51 = 3 \cdot 17 \). But we might also check the congruence by computing the remainders when each of those two numbers is divided by 17. Thus

\[
36 = 2 \cdot 17 + 2
\]

and

\[
87 = 5 \cdot 17 + 2.
\]

They both leave the same remainder \( 2 \) when divided by 17, so they are congruent modulo 17.

**Proposition 2.** Let \( a, b, c, m \) and \( n \) be integers. If \( a \equiv b \pmod{m} \) and \( c \equiv d \pmod{m} \) then \( a + c \equiv b + d \pmod{m} \) and \( ac \equiv bd \pmod{m} \) and \( a^n \equiv c^n \pmod{m} \).
Proof. Note that \((a + c) - (b + d) = (a - b) + (c - d)\) (and the sum of two multiples of \(m\) is a multiple of \(m\)). Also note that \(ac - bd = a(c - d) + (a - b)d\) (again a sum of two multiples of \(m\)). The statements about powers are proved by induction. \(\square\)

Because of these theorems, we regularly do modular arithmetic by remainders. If, for example, we know that \(x = 13\) and \(y = 23\), and we want to test the statement \((x + 1)^3 \equiv (y + 3)^2 \pmod{11}\), we might do this in two ways.

1. Compute \(x + 1 = 14\) so \((x + 1)^3 = 14^3 = 2744\). Also \(y + 3 = 26\) so \((y + 3)^2 = 26^2 = 676\). Now we have to test whether \(2744 \equiv 676 \pmod{11}\). We compute \(2744 - 676 = 2068\) and divide by 11. Since \(2068 = 11 \cdot 188\), we see that the congruence is true. (Phew!)

2. Alternatively, we might do the computations on both expressions separately, but every time a number gets bigger than 11, we reduce it \(\pmod{11}\). Thus we compute \(x + 1 = 14 \equiv 3 \pmod{11}\) by remainder. Similarly \(y + 3 = 26 \equiv 4 \pmod{11}\). Now \(3^3 \equiv 27 \equiv 5 \pmod{11}\). Also \((y + 3)^2 \equiv 16 \equiv 5 \pmod{11}\). Since \((x + 1)^3\) and \((y + 3)^2\) are both congruent to 5 \(\pmod{11}\), the congruence is verified.

The second approach seems easier, since we keep the numbers smaller by using remainders.

The effectiveness of this approach becomes even more apparent when we compute large powers of a number.

**An example.** Suppose we want to determine whether \(7^{22} \equiv 3 \pmod{23}\). Obviously we’d rather not compute 7 to the 22nd power. That’s a lot of work (and it’s a 19 digit number). But notice that \(7^2 = 49 \equiv 3 \pmod{23}\). Thus \(7^4 \equiv (7^2)^2 \equiv 3^2 = 9 \equiv 9 \pmod{23}\). Then \(7^8 \equiv (7^4)^2 \equiv 9^2 = 81 \equiv 12 \pmod{23}\).

By always replacing a number by its remainder on division by 23, we’re keeping the numbers small, and we’ve already gotten up to \(7^8\). Next we compute \(7^{16} = (7^8)^2 \equiv 12^2 = 144 \equiv 6 \pmod{23}\).

Now we notice that \(7^{22} = 7^{16} \cdot 7^4 \cdot 7^2 \equiv 6 \cdot 9 \cdot 3 = 6 \cdot 27 \equiv 6 \cdot 4 = 24 \equiv 1 \pmod{23}\). So without computing that 19 digit number, we already know that \(7^{22}\) is congruent to 1 \(\pmod{23}\) (and not to 3).

This was an example of a very fundamental algorithm in modern cryptography called **modular exponentiation.** We’ll detail this algorithm later. For the moment, suffice to say that modern computers can easily compute \(a^b \pmod{m}\) for very large positive integers \(a\), \(b\) and \(m\). Suppose, for example, that we need to find \(13^{53}\) modulo 89 (these numbers can be larger – hundreds of digits, if we want).

In Maple, we may type

\[
13\&^\wedge 53 \text{ mod } 89;
\]

In Mathematica we type

\[
\text{PowerMod}[13, 53, 89]
\]

and in Sage, we simply type

\[
\text{pow}(13, 53, 89)
\]

and in each language, we instantly get the correct answer 58.

The following theorem will be useful in this context.

**Fermat’s Little Theorem.** Now that we can reliably take powers of numbers in a modular system, we may state a very old theorem – which is very relevant to modern cryptography. We
previously showed that $7^{22} \equiv 1 \pmod{23}$. Using the modular exponention algorithm above (or a symbolic algebra package), we may similarly verify the following:

$$1^{22} \equiv 1 \pmod{23},$$
$$2^{22} \equiv 1 \pmod{23},$$
$$\vdots$$
$$6^{22} \equiv 1 \pmod{23},$$

and

$$8^{22} \equiv 1 \pmod{23},$$
$$9^{22} \equiv 1 \pmod{23},$$
$$\vdots$$
$$21^{22} \equiv 1 \pmod{23}.$$ 

Something seems to be going on here. Remarkably, all of these numbers, when raised to the 22nd power, are congruent to 1 modulo 23. It turns out to depend on the fact that 23 is a prime.

Also

$$7^{12} \equiv 1 \pmod{13}$$

and

$$8^{28} \equiv 1 \pmod{29}$$

and

$$5^{100} \equiv 1 \pmod{101}.$$ 

These are all examples of a very old theorem which was actually proved by Pierre de Fermat in his lifetime:

**Fermat’s Little Theorem:** If $p$ is a prime number, and $n$ is an integer not divisible by $p$, then $n^{p-1} \equiv 1 \pmod{p}$. 

Proof. We first prove that $n^p \equiv n \pmod{p}$ without any restrictions on $n$. Clearly this latter statement is true for $n = 0$, so we proceed by induction on $n$. If $n^p \equiv n \pmod{p}$ then

$$(n + 1)^p = n^p + {p \choose 1} n^{p-1} + {p \choose 2} n^{p-2} + \ldots + 1^p$$

(by the Binomial Theorem). The first and last terms of the right-hand sum are congruent to $n + 1$ by assumption, and all the remaining binomial coefficients have $p$ in the numerator, and not in the denominator – so they’re all $\equiv 0 \pmod{p}$.

Having proved (by induction) that $n^p \equiv n \pmod{p}$ for all $n$, (equivalently that $p$ divides $n^p - n = n(n^{p-1} - 1)$), then we note that when $p$ does not divide $n$ it must divide the other factor $(n^{p-1} - 1)$, which completes the proof of the theorem. \(\square\)

This theorem has become the basis for a lot of monkey business in cryptography, as we shall see later. In particular, it can be used to test very large numbers (hundreds of digits) to verify
that they aren't prime. If $p$ is a large number, and we compute $2^{p-1} \pmod{p}$ (which we know can be done very quickly), and we don’t get 1, then we know that $p$ is not a prime.

**The Fermat Test:** Given a large integer $n$, we can rapidly compute $k^{n-1} \pmod{n}$ for various small integers $k$ (such as $k = 2$). If any of these fails to give 1, then we know $n$ is not a prime. On the other hand, if they all give the answer 1, then the test does not tell us whether $n$ is prime or not.

For example, if $p = 341$ and we compute $2^{341-1} \pmod{341}$, we get 1, which doesn’t tell us anything. But when we try $3^{341-1} \pmod{341}$ we get 56, telling us that $341$ is definitely not a prime.

Similarly, if $m = 1393541763680708315055726675067654123$ and we calculate (on a computer) $2^{(m-1)} \pmod{m}$, we obtain the remainder $2913797628388130905174177872112930$, which is not 1. Thus we immediately know that $m$ is not a prime number (but we may have to do a bit of work to actually factor it). That’s useful information when dealing with very large numbers (hundreds of digits).

On the other hand, if we let $n = 1574601601$ we find that $2^{(n-1)} \equiv 1 \pmod{n}$. Likewise, we may try $3^{(n-1)}$, which also gives 1, and $5^{(n-1)} \equiv 1$. This becomes discouraging, and we eventually give up. In fact the Fermat test simply doesn’t give us an answer for this number. We’d need to try another approach to determine whether $n$ is prime or not (it isn’t).

**Algebra with modular arithmetic.**

Working with modular equations with a fixed modulus $m$ (particularly if $m$ is a prime number) is a lot like doing high-school algebra – but in a small, finite universe (instead of the usual infinite set of numbers). The only possible remainders are $0, 1, 2, \ldots, m - 2, m - 1$, so everything can be done in terms of those numbers. We can add, subtract, multiply and divide, solve equations, factor polynomials, even use familiar rules like the quadratic formula (provided the modulus is an odd prime).

For example, suppose our modulus is 7 (a nice odd prime), and suppose we want to solve the congruence $2x + 5 \equiv x + 2 \pmod{7}$. Since we’re working with a fixed modulus (7), we might as well simply write $2x + 5 = x + 2$ (while remembering that all operations take place within the set \{0, 1, 2, 3, 4, 5, 6\}). Then, subtracting $x$ from both sides gives $x + 5 = 2$ and subtracting 5 from both sides gives $x = -3$. But -3 is the same as $4 \pmod{7}$. So our answer is $x \equiv 4 \pmod{7}$.

In effect, we’re doing arithmetic (and algebra) within the set \{0, 1, 2, 3, 4, 5, 6\} with operations defined by the following addition and multiplication tables:

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>6</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>
Now suppose we want to solve the equation \(2x + 3 = 5(x + 1) \pmod{7}\). Doing algebra, we get \(2x + 3 = 5x + 5\) so \((-3)x = 2\). But \((-3)\) is the same as 4 in mod 7, so we have \(4x = 2\). Now how do we divide both sides by the coefficient 4? We could just look at the multiplication table above. What \(x\) must be multiplied by 4 to get 2? This would look like

\[
\begin{array}{c|c}
\cdot & x \\
- & - - - - - - \\
4 & 2 \\
\end{array}
\]

in the table. In other words, we look across the row with the leading 4 in the multiplication table until we find a 2, then we go to the top of that column to find \(x\), and we see from the table that \(x = 4\).

We should probably check that \(x = 4\) satisfies our original equation. Note that \(2x + 3 = 8 + 3 = 11 \equiv 4 \pmod{7}\) and \(5(x + 1) = 5 \cdot 5 = 25 \equiv 4 \pmod{7}\). So the equation is satisfied (since both sides are equal to 4 mod 7).

But suppose we wished to solve the congruence \(32x \equiv 49 \pmod{239}\). Unfortunately, this would be a lot of work if we had to construct a table (239 by 239) and look across the row with the leading 32 to see what \(x\) must be multiplied by 32 to give 47. A computer could do it, but not if the modulus were a couple of hundred digits long. No amount of computing power could solve such a “lookup” problem with numbers that large. We need an **algorithm** for solving such problems.

**Division modulo** \(m\).

Suppose we wish to solve the problem \(ax \equiv b \pmod{m}\). For example, the congruence \(3x \equiv 1 \pmod{5}\) has one solution modulo 5, namely \(x \equiv 2 \pmod{5}\). Of course there are infinitely many integers \(x\) with satisfy the congruence \(x = \ldots -8, -3, 2, 7, 12, \ldots\), but the answer \(x \equiv 2 \pmod{5}\) completely characterizes the solution set.

Similarly, the congruence \(3x \equiv 7 \pmod{8}\) has solution \(x \equiv 5 \pmod{8}\). The congruence \(4x \equiv 2 \pmod{6}\) has two solutions modulo 6, namely \(x \equiv 2 \pmod{6}\) and \(x \equiv 5 \pmod{6}\). But the congruence \(4x \equiv 7 \pmod{10}\) has no solutions.

Before doing the work of finding solutions (which we’ll discuss in a moment), we’d like to know if there are any to be found. There’s a simple test. Recall that the greatest common divisor of two numbers \(m\) and \(n\), denoted \(\gcd(m, n)\), is the largest positive integer which is a divisor of both \(m\) and \(n\). Thus \(\gcd(30, 48) = 6\). The following theorem may be helpful:

**Proposition 3.** The congruence \(ax \equiv b \pmod{m}\) has solutions if and only if the greatest common divisor \(\gcd(a, m)\) is a divisor of \(b\). In case that is true, then the number of solutions, modulo \(m\), is \(\gcd(a, m)\).

For example, the congruence \(30x \equiv 18 \pmod{48}\) has solutions because \(\gcd(30, 48) = 6\) which is a divisor of 18. Furthermore, the number of solutions modulo 48 is exactly 6. We don’t yet know how to find them, but there are six solutions. On the other hand, the congruence \(6x \equiv 9 \pmod{10}\) has no solutions at all, because \(\gcd(6, 10) = 2\) which is not a divisor of 9.

**Algorithm for finding solutions:** How do we solve \(ax \equiv b \pmod{m}\), assuming it has solutions? After computing \(d = \gcd(a, m)\) and verifying that it is a divisor of \(b\), we may first divide \(a\), \(b\) and \(m\) by that common divisor. That gives us a new congruence \(a_1x \equiv b_1 \pmod{m_1}\) where the \(\gcd(a_1, m_1) = 1\) (which is certainly a divisor of \(b_1\)).
In other words, we may as well assume we’ll always be solving congruences of the form \( ax \equiv b \pmod{m} \) where \( a \) and \( m \) have no common divisors except \( \pm 1 \). But how do we proceed in that case?

Assume that \( a \) and \( b \) are remainders mod \( m \). So we assume that \( 0 \leq a < m \) and that \( 0 \leq b < m \). Note that \( ax \equiv b \pmod{m} \) is equivalent to saying \( ax + my = b \) for some integer \( y \). Thus it is equivalent to the congruence \( my \equiv b \pmod{a} \). Now since \( m \) is greater than \( a \) we can replace it by its remainder mod \( a \), and similarly with \( b \). This reduces the problem to one involving smaller numbers.

We keep doing that reduction until the problem becomes small enough to handle easily. But that’s a little vague. Let’s see it in action.

**Example.** Begin with \( 64x \equiv 98 \pmod{478} \). Note that \( \gcd(64,478) = 2 \) which is a divisor of 98, so we’re good to go. By proposition 3, we expect two solutions to this congruence – modulo 478. Now divide everything by 2 to obtain the equivalent congruence \( 32x \equiv 49 \pmod{239} \). This is the problem we have to solve.

This means that \( 32x + 239y = 49 \) (note \( y \) may be negative – we don’t care). But that equation \( 32x + 239y = 49 \) can be interpreted as another congruence: \( 239y \equiv 49 \pmod{32} \).

But now we can reduce 239 and 49 each modulo 32, so our new problem is \( 15y \equiv 17 \pmod{32} \). But that problem is still too hard, so we do the trick again. Note that \( 15y \equiv 17 \pmod{32} \) means \( 15y + 32z = 17 \) so (equivalently) \( 32z \equiv 17 \pmod{15} \), which reduces (mod 15) to \( 2z \equiv 2 \), which is easy enough to solve by inspection. \( z \equiv 1 \pmod{15} \).

But how does that get us the solution to our original problem? We introduce a new variable \( t \) at this point and write \( z = 1 + 15t \), and then we work our way back up through the equations above, solving for each variable in terms of \( t \). Thus

\[
15y + 32z = 17 \implies 15y + 32(1 + 15t) = 17 \implies y = -1 - 32t
\]

\[
32x + 239y = 49 \implies 32x + 239(-1 - 32t) = 49 \implies x = 9 + 239t
\]

In other words (bearing in mind that \( t \) can be any positive or negative integer), we’ve shown that \( x \equiv 9 \pmod{239} \) is the solution to our original problem. (Check it out!) To put it another way, we’ve found that \( \frac{49}{32} = 9 \) in the arithmetic modulo 239.

Recall that our original problem was \( 64x \equiv 98 \pmod{478} \). Note that \( x \equiv 9 \pmod{239} \) is equivalent to \( x \equiv 9 \) and \( x \equiv 248 \) modulo 478. That gives us the two solutions (predicted by Proposition 3) to our original congruence.

**Quadratic equations.** Suppose we want to solve the equation \( x(x + 2) \equiv 1 \pmod{7} \). This yields a quadratic equation \( x^2 + 2x = 1 \) or \( x^2 + 2x + 6 = 0 \pmod{7} \). Let’s use the quadratic formula. Thus the solution should be

\[
x = \frac{-2 \pm \sqrt{2^2 - 4 \cdot 1 \cdot 6}}{2} = \frac{5 \pm \sqrt{4 - 24}}{2} = \frac{5 \pm \sqrt{-3}}{2} = \frac{5 \pm \sqrt{-1} \cdot 2}{2} = \frac{6 \cdot 4}{2} = 2, 2
\]

so our two answers are 3 and 2. One may check that *both* of these number work in our original equation: \( x(x + 2) \equiv 1 \pmod{7} \). No other answers in the set \( \{0, 1, 2, 3, 4, 5, 6\} \) will work.
Miscellaneous challenge problems.

(In each case, find solutions in the set \{0, 1, 2, \ldots, m - 1\} where \(m\) is the given modulus.)

1. \(2^6 \pmod{7}\).
2. \(2^{12} \pmod{7}\).
3. \(2^{25} \pmod{7}\).
4. \(3^{84} \pmod{29}\).
5. \(11^{129} \pmod{100}\).
6. Solve \(2x + 3 = 5 \pmod{7}\).
7. Solve \(4x + 4 = x + 10 \pmod{11}\).
8. Solve \(27x \equiv 129 \pmod{509}\).
9. Find \(\frac{1}{27}\) in the system of arithmetic modulo 509.
10. Solve \(x(x - 7) \equiv 231 \pmod{509}\).
11. Find all three solutions (roots) of the equation \(x^3 + x^2 + x + 1 \equiv 4 \pmod{11}\).
12. Find any solutions which exist to the equation \(x^3 + x^2 + x + 1 \equiv 4 \pmod{101}\).
13. Find \(2^{6870610} \pmod{298723}\).
14. Find the solutions to \(x^2 + 3 \equiv 0 \pmod{5400979}\).
15. Find all cube roots of 5400978 modulo 5400979.
16. Let \(e = 1,000,003\) (a million and three) and let \(m = 22,645,783,325,411\). Find all integers \(x\) such that \(x^e \equiv 2 \pmod{m}\).

Have fun! Solve as many as you can.