Integers Modulo m

The Euclidean algorithm for integers leads to the notion of congruence of two integers modulo a given integer.

Congruence Modulo m

Two integers a and b are **congruent modulo** m if and only if $a - b$ is a multiple of m, in which case we write $a \equiv b\,(\mathrm{mod}\,m)$. Thus, $15 \equiv 33\,(\mathrm{mod}\,9)$, because $15 - 33 = -18$ is a multiple of 9. Given integers a and m, the **mod function** is given by $a \bmod m = b$ if and only if $a \equiv b\,(\mathrm{mod}\,m)$ and $0 \leq b \leq m - 1$; hence, $a \bmod m$ is the smallest **nonnegative residue** of a modulo m.

The underlying computer algebra system does not understand the congruence notation $a \equiv b(\mathrm{mod}\,m)$, but it does understand the function notation $a \bmod m$. This section shows how to translate problems in algebra and number theory into language that will be handled correctly by the computational engine.

Note that mod is a function of two variables, with the function written between the two variables. This usage is similar to the common usage of $+$, which is also a function of two variables with the function values expressed as $a + b$, rather than the usual functional notation $+(a, b)$.

Traditionally the congruence notation $a \equiv b\,(\mathrm{mod}\,m)$ is written with the $\mathrm{mod}\,m$ enclosed inside parentheses since the $\mathrm{mod}\,m$ clarifies the expression $a \equiv b$. In this context, the expression $b\,(\mathrm{mod}\,m)$ never appears without the preceding $a \equiv$. On the other hand, the mod function is usually written in the form $a \bmod m$ without parentheses.

> **To evaluate the mod function**

1. Leave the insertion point in the expression $a \bmod b$.
2. Choose **Evaluate.**

▸ **Evaluate**

$$23 \bmod 14 = 9$$

If a is positive, you can also find the smallest nonnegative residue of a modulo m by applying **Expand** to the quotient $\frac{a}{m}$.

▸ **Expand**

$$\frac{23}{14} = 1\frac{9}{14}$$

Since $1\frac{9}{14} = 1 + \frac{9}{14}$, multiplication of $\frac{23}{14} = 1 + \frac{9}{14}$ by 14 shows that $23 \bmod 14 = 9$.

In terms of the floor function $\lfloor x \rfloor$, the mod function is given by $a \bmod m = a - \lfloor \frac{a}{m} \rfloor m$.

▸ **Evaluate**

$$23 - \left\lfloor \frac{23}{14} \right\rfloor 14 = 9$$

Multiplication Tables Modulo m

You can make tables that display the products modulo m of pairs of integers from the set $\{0, 1, 2, \ldots, m - 1\}$.

> **To get a multiplication table modulo m with $m = 6$**

1. Define the function $g(i,j) = (i - 1)(j - 1)$.
2. From the **Matrices** submenu, choose **Fill Matrix**.
3. Select **Defined by Function.**
4. Enter g in the **Enter Function Name** box.
5. Select 6 rows and 6 columns.
6. Choose **OK**.
7. Type $\mathrm{mod}\,6$ at the right of the matrix. (Because the insertion point is in mathematics mode; *mod* automatically turns gray.)
8. Choose **Evaluate.**

▸ **Evaluate**

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 2 & 4 & 6 & 8 & 10 \\ 0 & 3 & 6 & 9 & 12 & 15 \\ 0 & 4 & 8 & 12 & 16 & 20 \\ 0 & 5 & 10 & 15 & 20 & 25 \end{bmatrix} \bmod 6 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 2 & 4 & 0 & 2 & 4 \\ 0 & 3 & 0 & 3 & 0 & 3 \\ 0 & 4 & 2 & 0 & 4 & 2 \\ 0 & 5 & 4 & 3 & 2 & 1 \end{bmatrix}$$

A more efficient way to generate the same multiplication table is to define $g(i,j) = (i-1)(j-1) \bmod 6$ and follow steps 2-6 above.

You can also find this matrix as the product of a column matrix with a row matrix.

▸ **Evaluate**

$$\begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{bmatrix} \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 2 & 4 & 6 & 8 & 10 \\ 0 & 3 & 6 & 9 & 12 & 15 \\ 0 & 4 & 8 & 12 & 16 & 20 \\ 0 & 5 & 10 & 15 & 20 & 25 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 2 & 4 & 6 & 8 & 10 \\ 0 & 3 & 6 & 9 & 12 & 15 \\ 0 & 4 & 8 & 12 & 16 & 20 \\ 0 & 5 & 10 & 15 & 20 & 25 \end{bmatrix} \bmod 6 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 2 & 4 & 0 & 2 & 4 \\ 0 & 3 & 0 & 3 & 0 & 3 \\ 0 & 4 & 2 & 0 & 4 & 2 \\ 0 & 5 & 4 & 3 & 2 & 1 \end{bmatrix}$$

Make a copy of this last matrix. From the **Edit** menu, choose **Insert Row(s)** and add a new row at the top (position 1); choose **Insert Column(s)**.and add a new column at the left (position 1); fill in the blanks and change the new row and column to **Bold** font, to get the following multiplication table modulo 6:

×	**0**	**1**	**2**	**3**	**4**	**5**
0	0	0	0	0	0	0
1	0	1	2	3	4	5
2	0	2	4	0	2	4
3	0	3	0	3	0	3
4	0	4	2	0	4	2
5	0	5	4	3	2	1

From the table, we see that $2 \cdot 4 \bmod 6 = 2$ and $3 \cdot 3 \bmod 6 = 3$.

A clever approach, that creates this table in essentially one step, is to define

$$g(i,j) = |i-2||j-2| \bmod 6$$

Choose **Fill Matrix** from the **Matrices** submenu, choose **Defined by Function** from the dialog box, specify g for the function, and set the matrix size to 7 rows and 7 columns. Then replace the digit 1 in the upper left corner by $\times$ and change the first row and column to **Bold** font, as before.

You can generate an addition table by defining $g(i,j) = i + j - 2 \bmod 6$.

Example *If p is a prime, then the integers modulo p form a field, called a* **Galois field** *and denoted GF_p. For the prime $p = 7$, you can generate the multiplication table by defining $g(i,j) = (i-1)(j-1) \bmod 7$ and choosing* **Fill Matrix** *from the* **Matrix** *submenu, then* *selecting* **Defined by function** *from the dialog box. You can generate the addition table in a similar manner using the function* $f(i,j) = i + j - 2 \bmod 7$.

×	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6
2	0	2	4	6	1	3	5
3	0	3	6	2	5	1	4
4	0	4	1	5	2	6	3
5	0	5	3	1	6	4	2
6	0	6	5	4	3	2	1

+	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	2	3	4	5	6	0
2	2	3	4	5	6	0	1
3	3	4	5	6	0	1	2
4	4	5	6	0	1	2	3
5	5	6	0	1	2	3	4
6	6	0	1	2	3	4	5

Inverses Modulo m

If $ab \bmod m = 1$, then b is called an **inverse of** a **modulo** m, and we write $a^{-1} \bmod m$ for the least positive residue of b. The computation engine also recognizes both of the forms $1/a \bmod m$ and $\frac{1}{a} \bmod m$ for the inverse modulo m.

Note that $a^{-1} \bmod m$ exists if and only if a is **relatively prime** to m; that is, it exists if and only if $\gcd(a,m) = 1$. Thus, modulo 6, only 1 and 5 have inverses. Modulo any prime, every nonzero residue has an inverse. In terms of the multiplication table modulo m, the integer a has an inverse modulo m if and only if 1 appears in row $a \bmod m$ (and 1 appears in column $a \bmod m$).

> **To compute the inverse of** $a \bmod m$ **if** $\gcd(a,m) = 1$

1. Enter the inverse in one of the forms $a^{-1} \bmod m$, $1/a \bmod m$, or $\frac{1}{a} \bmod m$.
2. Place the insertion point in the expression and choose **Evaluate**.

▸ **Evaluate**

$5^{-1} \bmod 7 = 3$ $\qquad \frac{1}{5} \bmod 7 = 3$ $\qquad 1/5 \bmod 7 = 3$

This calculation satisfies the definition of inverse, because $5 \cdot 3 \bmod 7 = 1$.

▸ **Evaluate**

$23^{-1} \bmod 257 = 190$

$\frac{1}{5} \bmod 6 = 5$

The notations $ab^{-1} \bmod m$, $a/b \bmod m$, and $\frac{a}{b} \bmod m$ are all interpreted as $a(b^{-1} \bmod m) \bmod m$; that is, find the inverse of b modulo m, multiply the result by a, and then reduce the product modulo m.

▸ **Evaluate**

$3/23 \bmod 257 = 56$

$\frac{2}{5} \bmod 6 = 4$

Solving Congruences Modulo m

> **To solve a congruence of the form** $ax \equiv b \pmod{m}$

- Multiply both sides by $a^{-1} \bmod m$ to get $x = b/a \bmod m$.

The congruence $17x \equiv 23 \pmod{127}$ has a solution $x = 91$, as the following two evaluations illustrate.

▸ **Evaluate**

$23/17 \bmod 127 = 91$

Check this result by substitution back into the original congruence.

▸ **Evaluate**

$$17 \cdot 91 \bmod 127 = 23$$

Note that, since 91 is a solution to the congruence $17x \equiv 23 \pmod{127}$, additional solutions are given by $91 + 127n$, where n is any integer. In fact, $x \equiv 91 \pmod{127}$ is just another way of writing $x = 91 + 127n$ for some integer n.

Pairs of Linear Congruences

Since linear congruences of the form $ax \equiv b \pmod{m}$ can be reduced to simple congruences of the form $x \equiv c \pmod{m}$, we consider systems of congruences in this latter form.

> **To solve a pair of linear congruences** $x \equiv c \pmod{m}$ **and** $x \equiv d \pmod{n}$

1. Check that $\gcd(m, n) = 1$ so that a solution exists.
2. Rewrite the congruences as equations $x = km + c$, $x = rn + d$, whence $km + c = rn + d$.
3. Rewrite this equation as the congruence $km \equiv (d - c) \bmod n$ and divide both sides by m to solve for k.
4. Place the insertion point in the congruence $k \equiv (d - c)/m \bmod n$ and choose **Evaluate**.
5. Using the computed value for k, place the insertion point in the equation $x = km + c$ and choose **Evaluate**.
6. The complete set of solutions are the solutions of $x \equiv (km + c) \bmod mn$, with $k \equiv (d - c)/m \bmod n$.

Example *Consider the system of two congruences*

$$x \equiv 45 \pmod{237}$$

$$x \equiv 19 \pmod{419}$$

Checking, $\gcd(237, 419) = 1$*, so* 237 *and* 419 *are relatively prime. The first congruence can be rewritten in the form* $x = 45 + 237k$ *for some integer* k*. Substituting this value into the second congruence, we see that*

$$45 + 237k = 19 + 419r$$

for some integer r*. This last equation can be rewritten in the form* $237k = 19 - 45 \bmod 419$*, which has the solution*

$$k = (19 - 45)/237 \bmod 419 = 60$$

Hence,

$$x = 45 + 237 \cdot 60 = 14265$$

Checking, $14265 \bmod 237 = 45$ *and* $14265 \bmod 419 = 19$*.*

Example *The complete set of solutions is given by*

$$x = 14265 + 237 \cdot 419s \equiv 14265 \pmod{99303}$$

Thus, the original pair of congruences has been reduced to a single congruence,

$$x \equiv 14265 \pmod{99303}$$

In general, if m and n are relatively prime, then one solution to the pair

$$x \equiv a \pmod{m}$$

$$x \equiv b \pmod{n}$$

is given by

$$x = a + m[(b - a)/m \bmod n]$$

A complete set of solutions is given by

$$x = a + m[(b - a)/m \bmod n] + rmn$$

where r is an arbitrary integer.

Systems of Linear Congruences

You can reduce systems of any number of congruences to a single congruence by solving systems of congruences two at a time. The **Chinese remainder theorem** states that, if the moduli are relatively prime in pairs, then there is a unique solution modulo the product of all the moduli.

> **To solve a system of linear congruences** $x \equiv c_i \pmod{m_i}$, $i = 1, 2, \ldots, t$

1. Check that $\gcd(m_i, m_j) = 1$ for every $i \neq j$ so that a solution exists.
2. Solve the congruences one pair at a time to obtain a complete solution.

Example *Consider the system of three linear congruences*

$$x \equiv 45 \pmod{237}$$

$$x \equiv 19 \pmod{419}$$

$$x \equiv 57 \pmod{523}$$

Checking, $\gcd(237 \cdot 419, 523) = 1$ *and* $\gcd(237, 419) = 1$*; hence this system has a solution. The first two congruences can be replaced by the single congruence* $x \equiv 14265 \pmod{99303}$*; hence the three congruences can be replaced by the pair*

$$x \equiv 14265 \pmod{99303}$$

$$x \equiv 57 \pmod{523}$$

As before, $14265 + 99303k = 57 + 523r$ *for some integers* k *and* r*. Thus,* $k = (57 - 14265)/99303 \bmod 523 = 134$*; hence* $x = 14265 + 99303 \cdot 134 = 13320867$*. This system of three congruences can thus be reduced to the single congruence*

$$x \equiv 13320867 \pmod{51935469}$$

Extended Precision Arithmetic

Computer algebra systems support exact sums and products of integers that are hundreds of digits long.

> **To do extended precision arithmetic**

1. Generate a set of mutually relatively prime bases, and do modular arithmetic modulo all of these bases.
2. Solve the resulting system of linear congruences.

For example, consider the vector

$$(997, 999, 1000, 1001, 1003, 1007, 1009)$$

of bases. Factorization shows that the entries are pairwise relatively prime.

▸ **Factor**

$$\begin{bmatrix} 997 \\ 999 \\ 1000 \\ 1001 \\ 1003 \\ 1007 \\ 1009 \end{bmatrix} = \begin{bmatrix} 997 \\ 3^3 37 \\ 2^3 5^3 \\ 7 \times 11 \times 13 \\ 17 \times 59 \\ 19 \times 53 \\ 1009 \end{bmatrix}$$

Consider the two numbers 23890864094 and 1883289456. You can represent these numbers by reducing the numbers modulo each of the bases. Thus,

$$23890864094 \leftrightarrow \begin{bmatrix} 23890864094 \bmod 997 & = 350 \\ 23890864094 \bmod 999 & = 872 \\ 23890864094 \bmod 1000 & = 94 \\ 23890864094 \bmod 1001 & = 97 \\ 23890864094 \bmod 1003 & = 879 \\ 23890864094 \bmod 1007 & = 564 \\ 23890864094 \bmod 1009 & = 218 \end{bmatrix}$$

$$1883289456 \leftrightarrow \begin{bmatrix} 1883289456 \bmod 997 & = 324 \\ 1883289456 \bmod 999 & = 630 \\ 1883289456 \bmod 1000 & = 456 \\ 1883289456 \bmod 1001 & = 48 \\ 1883289456 \bmod 1003 & = 488 \\ 1883289456 \bmod 1007 & = 70 \\ 1883289456 \bmod 1009 & = 37 \end{bmatrix}$$

Thus, the product $23890864094 \cdot 1883289456$ is represented by the vector

$$\begin{bmatrix} 350 \cdot 324 \bmod 997 & = & 739 \\ 872 \cdot 630 \bmod 999 & = & 909 \\ 94 \cdot 456 \bmod 1000 & = & 864 \\ 97 \cdot 48 \bmod 1001 & = & 652 \\ 879 \cdot 488 \bmod 1003 & = & 671 \\ 564 \cdot 70 \bmod 1007 & = & 207 \\ 218 \cdot 37 \bmod 1009 & = & 1003 \end{bmatrix}$$

The product $23890864094 \cdot 1883289456$ is now a solution to the system

$$\begin{aligned} x &\equiv 739 \pmod{997} \\ x &\equiv 909 \pmod{999} \\ x &\equiv 864 \pmod{1000} \\ x &\equiv 652 \pmod{1001} \\ x &\equiv 671 \pmod{1003} \\ x &\equiv 207 \pmod{1007} \\ x &\equiv 1003 \pmod{1009} \end{aligned}$$

Powers Modulo m

> **To calculate large powers modulo m**

- Place the insertion point in an expression of the form $a^n \bmod m$ and choose **Evaluate**.

Example *Define* $a = 2789596378267275$*,* $n = 3848590389047349$*, and* $m = 2838490563537459$*.*
Applying the command **Evaluate** *to* $a^n \bmod m$ *yields the following:*

$$a^n \bmod m = 26220\,18141\,09828$$

Fermat's Little Theorem states that, if p is prime and $0 < a < p$, then

$$a^{p-1} \bmod p = 1$$

The integer 1009 is prime, and the following is no surprise.

▸ **Evaluate**

$2^{1008} \bmod 1009 = 1$

Generating Large Primes

There is not a built-in function to generate large primes, but the underlying computational system does have such a function. The following is an example of how to define functions that correspond to existing functions in the underlying computational system. (Click here for a general discussion of how to access such functions.) In this example, $p(x)$ is defined as the **Scientific WorkPlace (Notebook) Name** for the MuPAD function, **nextprime(x)**, which generates the first prime greater than or equal to x.

> **To define $p(x)$ as the next-prime function**

1. From the **Definitions** submenu, choose **Define MuPAD Name**.
2. Enter `nextprime(x)` as the **MuPAD Name**.
3. Enter $p(x)$ as the **Scientific Notebook (WorkPlace) Name**.
4. Under **The MuPAD Name is a Procedure**, check **That is Built In to MuPAD or is Automatically Loaded**.
5. Choose **OK**.

Test the function using **Evaluate**.

▸ **Evaluate**

$p(5) = 5$

$p(500) = 503$

$p(8298) = 8311$

$p(273849728952758923) = 273\,849\,728\,952\,758\,923$

Example *The Rivest-Shamir-Adleman (RSA) cipher system is based directly on Euler's theorem and requires a pair of large primes. First, generate a pair of large primes—say,*

$$q = p(20934834573) = 20934834647$$

and

$$r = p(2593843747347) = 2593843747457$$

(In practice, larger primes are used; such as, $q \approx 10^{100}$ and $r \approx 10^{100}$.) Then

$$\begin{aligned} n &= qr \\ &= 20934834647 \cdot 2593843747457 \\ &= 543\,01689\,95316\,71217\,42679 \end{aligned}$$

and the number of positive integers $\leq n$ and relatively prime to n is given by

$$\begin{aligned} \varphi(n) &= (q-1)(r-1) \\ &= 20934834646 \cdot 2593843747456 \\ &= 543\,01689\,95055\,23431\,60576 \end{aligned}$$

Let

$$x = 29384737849576728375$$

be plaintext (suitably generated by a short section of English text). Long messages must be broken up into small enough chunks that each plaintext integer x is smaller than the modulus n. Choose E to be a moderately large positive integer that is relatively prime to $\varphi(n)$, for example, $E = 1009$. The ciphertext is given by

$$y = x^E \bmod n = 20636340188476258131729$$

Let

$$D = 1009^{-1} \bmod \varphi(n) = 42515693816587067 48945$$

Then friendly colleagues can recover the plaintext by calculating

$$z = y^D \bmod n = 29384737849576728375$$

Related topics

Index