Modulo Operation (mod) Algorithm and Examples

In computing, the modulo (sometimes called modulus, or mod) operation finds the remainder of division of one number by another. Given two positive numbers, \(a\) (the dividend) and \(n\) (the divisor), a modulo \(n\) (abbreviated as \(a \mod n\)) is the remainder of the Euclidean division of \(a\) by \(n\). For instance, the expression "5 mod 2" would evaluate to 1 because 5 divided by 2 leaves a quotient of 2 and a remainder of 1, while "9 mod 3" would evaluate to 0 because the division of 9 by 3 has a quotient of 3 and leaves a remainder of 0; there is nothing to subtract from 9 after multiplying 3 times 3.

Note that doing the division with a calculator won't show the result referred to here by this operation; the quotient will be expressed as a decimal fraction. Although typically performed with \(a\) and \(n\) both being integers, many computing systems allow other types of numeric operands. The range of numbers for an integer modulo of \(n\) is 0 to \(n - 1\). \((n \mod 1)\) is always 0; \(n \mod 0\) is undefined, possibly resulting in a "Division by zero" error in computer programming languages. See modular arithmetic for an older and related convention applied in number theory. When either \(a\) or \(n\) is negative, the naive definition breaks down and programming languages differ in how these values are defined.

Example 1. \(a \mod n \equiv a - [n \times \text{int}(\frac{a}{n})]\)

\[
[269.86] - [n \times \text{int}(\frac{a}{n})] \\
= 269 - [100 \times \text{int}(\frac{269.86}{100})] \\
= 269 - [100 \times \text{int}(2.6986)] \\
= 269 - [100 \times 2] \\
= 69
\]

Example 2. \(M^b_i \equiv [100 \times 10^{(\log P_i) \mod 1}] \mod 100\)

\[
\text{Step 1.} \quad \log(269.86) \mod 1 \\
= 2.4311 \mod 1 \\
= 2.4311 - [1 \times \text{int}(2.4311)] \\
= 2.4311 - 2 \\
= 0.4311
\]

\[
\text{Step 2.} \quad M^b_i = [100 \times 10^{0.4311}] \mod 100 \\
= [100 \times 2.6983] \mod 100 \\
= 269.83 - [100 \times \text{int}(\frac{269.83}{100})] \\
= 269.83 - 200 \\
= 69
\]