

Re: Sign conventions for remainder

Source: <http://sci.tech-archive.net/Archive/sci.math/2004-06/3855.html>

From: David Eppstein (eppstein_at_ics.uci.edu)

Date: 06/17/04

Date: Thu, 17 Jun 2004 13:30:29 -0700

In article <200406171732.i5HHWRG37280@mickey.empros.com>, mstemper@siemens-emis.com (Michael Stemper) wrote:

> *Just for fun, I've been implementing a package to perform arithmetic on
> integers of arbitrary size. Addition, subtraction, and multiplication
> were all pretty straight-forward. (Subtraction was the easiest!) But,
> when I prepared to implement division, I realized that I'm not aware of
> the conventions for remainders.*
>
> *If the divisor and the dividend are both positive, I know that my result
> should be $d = n*q + r$, with $0 \leq r < n$. What if one of them is negative? Should
> the remainder still be in that range? Should it be in the negative of that
> range? What about if both the divisor and the dividend are negative?*
>
> *Yes, I am aware that there is no "right" answer to this, there are only
> conventions. But, that's exactly what I'm looking for. Conventions are
> usually established because they're useful.*

If the divisor is positive, the remainder should still be in the range $0 \leq r < n$, and $d = nq + r$ should still be true. If the divisor is negative, it's less clear to me what the right convention is -- the choice taken by Python is to use the same sign as the divisor, which seems reasonable enough to me.

The C-C++-Java convention of remainder having the same sign as dividend is wrong and should be avoided.

--

David Eppstein <http://www.ics.uci.edu/~eppstein/>
Univ. of California, Irvine, School of Information & Computer Science