

**NAME**

drem, dremf, drem1, remainder, remainderf, remainderl – floating-point remainder function

**LIBRARY**

Math library (*libm*, *-lm*)

**SYNOPSIS**

```
#include <math.h>

/* The C99 versions */
double remainder(double x, double y);
float remainderf(float x, float y);
long double remainderl(long double x, long double y);

/* Obsolete synonyms */
double drem(double x, double y);
float dremf(float x, float y);
long double drem1(long double x, long double y);
```

Feature Test Macro Requirements for glibc (see **feature\_test\_macros(7)**):

```
remainder():
    _ISOC99_SOURCE || _POSIX_C_SOURCE >= 200112L
    || _XOPEN_SOURCE >= 500
    || /* Since glibc 2.19: */ _DEFAULT_SOURCE
    || /* glibc <= 2.19: */ _BSD_SOURCE || _SVID_SOURCE

remainderf(), remainderl():
    _ISOC99_SOURCE || _POSIX_C_SOURCE >= 200112L
    || /* Since glibc 2.19: */ _DEFAULT_SOURCE
    || /* glibc <= 2.19: */ _BSD_SOURCE || _SVID_SOURCE

drem(), dremf(), drem1():
    /* Since glibc 2.19: */ _DEFAULT_SOURCE
    || /* glibc <= 2.19: */ _BSD_SOURCE || _SVID_SOURCE
```

**DESCRIPTION**

These functions compute the remainder of dividing  $x$  by  $y$ . The return value is  $x - n * y$ , where  $n$  is the value  $x / y$ , rounded to the nearest integer. If the absolute value of  $x - n * y$  is 0.5,  $n$  is chosen to be even.

These functions are unaffected by the current rounding mode (see **fenv(3)**).

The **drem()** function does precisely the same thing.

**RETURN VALUE**

On success, these functions return the floating-point remainder,  $x - n * y$ . If the return value is 0, it has the sign of  $x$ .

If  $x$  or  $y$  is a NaN, a NaN is returned.

If  $x$  is an infinity, and  $y$  is not a NaN, a domain error occurs, and a NaN is returned.

If  $y$  is zero, and  $x$  is not a NaN, a domain error occurs, and a NaN is returned.

**ERRORS**

See **math\_error(7)** for information on how to determine whether an error has occurred when calling these functions.

The following errors can occur:

Domain error:  $x$  is an infinity and  $y$  is not a NaN

*errno* is set to **EDOM** (but see **BUGS**). An invalid floating-point exception (**FE\_INVALID**) is raised.

These functions do not set *errno* for this case.

Domain error: *y* is zero

*errno* is set to **EDOM**. An invalid floating-point exception (**FE\_INVALID**) is raised.

## ATTRIBUTES

For an explanation of the terms used in this section, see **attributes(7)**.

Interface	Attribute	Value
<b>drem()</b> , <b>dremf()</b> , <b>dreml()</b> , <b>remainder()</b> , <b>remainderf()</b> , <b>remainderl()</b>	Thread safety	MT-Safe

## STANDARDS

The functions **remainder()**, **remainderf()**, and **remainderl()** are specified in C99, POSIX.1-2001, and POSIX.1-2008.

The function **drem()** is from 4.3BSD. The *float* and *long double* variants **dremf()** and **dreml()** exist on some systems, such as Tru64 and glibc2. Avoid the use of these functions in favor of **remainder()** etc.

## BUGS

Before glibc 2.15, the call

```
remainder(nan(" "), 0);
```

returned a NaN, as expected, but wrongly caused a domain error. Since glibc 2.15, a silent NaN (i.e., no domain error) is returned.

Before glibc 2.15, *errno* was not set to **EDOM** for the domain error that occurs when *x* is an infinity and *y* is not a NaN.

## EXAMPLES

The call "remainder(29.0, 3.0)" returns  $-1$ .

## SEE ALSO

**div(3)**, **fmod(3)**, **remquo(3)**