

NAME

rint, rintf, rintl, llrint, llrintf, llrintl - round to nearest integer

SYNOPSIS

```
#include <math.h>

long int rint(double x);
long int rintf(float x);
long int rintl(long double x);

long long int llrint(double x);
long long int llrintf(float x);
long long int llrintl(long double x);
```

Link with *-lm*.

Feature Test Macro Requirements for glibc (see [feature_test_macros\(7\)](#)):

All functions shown above:

```
_XOPEN_SOURCE >= 600 || _ISOC99_SOURCE || _POSIX_C_SOURCE >= 200112L;
or cc -std=c99
```

DESCRIPTION

These functions round their argument to the nearest integer value, using the current rounding direction (see [fesetround\(3\)](#)).

Note that unlike the [rint\(3\)](#) family of functions, the return type of these functions differs from that of their arguments.

RETURN VALUE

These functions return the rounded integer value.

If *x* is a NaN or an infinity, or the rounded value is too large to be stored in a *long* (*long long* in the case of the **ll*** functions), then a domain error occurs, and the return value is unspecified.

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 a NaN or infinite, or the rounded value is too large
An invalid floating-point exception (**FE_INVALID**) is raised.

These functions do not set *errno*.

VERSIONS

These functions first appeared in glibc in version 2.1.

ATTRIBUTES

Multithreading (see [pthreads\(7\)](#))

The [rint\(\)](#), [rintf\(\)](#), [rintl\(\)](#), [llrint\(\)](#), [llrintf\(\)](#), and [llrintl\(\)](#) functions are thread-safe.

CONFORMING TO

C99, POSIX.1-2001.

SEE ALSO

[ceil\(3\)](#), [floor\(3\)](#), [lround\(3\)](#), [nearbyint\(3\)](#), [rint\(3\)](#), [round\(3\)](#)

COLOPHON

This page is part of release 3.74 of the Linux *man-pages* project. A description of the project, information about reporting bugs, and the latest version of this page, can be found at <http://www.kernel.org/doc/man-pages/>.