

NAME

lgamma, lgammaf, lgammal, lgamma_r, lgammaf_r, lgammal_r, signgam – log gamma function

SYNOPSIS

```
#include <math.h>

double lgamma(double x);
float lgammaf(float x);
long double lgammal(long double x);

double lgamma_r(double x, int *signp);
float lgammaf_r(float x, int *signp);
long double lgammal_r(long double x, int *signp);

extern int signgam;
```

Link with `-lm`.

Feature Test Macro Requirements for glibc (see `feature_test_macros(7)`):

```
lgamma():
    _ISOC99_SOURCE || _POSIX_C_SOURCE >= 200112L || _XOPEN_SOURCE
    || /* Since glibc 2.19: */ _DEFAULT_SOURCE
    || /* Glibc versions <= 2.19: */ _BSD_SOURCE || _SVID_SOURCE
lgammaf(), lgammal():
    _ISOC99_SOURCE || _POSIX_C_SOURCE >= 200112L
    || /* Since glibc 2.19: */ _DEFAULT_SOURCE
    || /* Glibc versions <= 2.19: */ _BSD_SOURCE || _SVID_SOURCE
lgamma_r(), lgammaf_r(), lgammal_r():
    /* Since glibc 2.19: */ _DEFAULT_SOURCE
    || /* Glibc versions <= 2.19: */ _BSD_SOURCE || _SVID_SOURCE
signgam:
    _XOPEN_SOURCE
    || /* Since glibc 2.19: */ _DEFAULT_SOURCE
    || /* Glibc versions <= 2.19: */ _BSD_SOURCE || _SVID_SOURCE
```

DESCRIPTION

For the definition of the Gamma function, see `tgamma(3)`.

The `lgamma()`, `lgammaf()`, and `lgammal()` functions return the natural logarithm of the absolute value of the Gamma function. The sign of the Gamma function is returned in the external integer `signgam` declared in `<math.h>`. It is 1 when the Gamma function is positive or zero, `-1` when it is negative.

Since using a constant location `signgam` is not thread-safe, the functions `lgamma_r()`, `lgammaf_r()`, and `lgammal_r()` have been introduced; they return the sign via the argument `signp`.

RETURN VALUE

On success, these functions return the natural logarithm of $\Gamma(x)$.

If `x` is a NaN, a NaN is returned.

If `x` is 1 or 2, `+0` is returned.

If `x` is positive infinity or negative infinity, positive infinity is returned.

If `x` is a nonpositive integer, a pole error occurs, and the functions return `+HUGE_VAL`, `+HUGE_VALF`, or `+HUGE_VALL`, respectively.

If the result overflows, a range error occurs, and the functions return `HUGE_VAL`, `HUGE_VALF`, or `HUGE_VALL`, respectively, with the correct mathematical sign.

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:

Pole error: x is a nonpositive integer

errno is set to **ERANGE** (but see BUGS). A divide-by-zero floating-point exception (**FE_DIVBYZERO**) is raised.

Range error: result overflow

errno is set to **ERANGE**. An overflow floating-point exception (**FE_OVERFLOW**) is raised.

CONFORMING TO

The **lgamma()** functions are specified in C99, POSIX.1-2001, and POSIX.1-2008. *signgam* is specified in POSIX.1-2001 and POSIX.1-2008, but not in C99. The **lgamma_r()** functions are nonstandard, but present on several other systems.

BUGS

In glibc 2.9 and earlier, when a pole error occurs, *errno* is set to **EDOM**; instead of the POSIX-mandated **ERANGE**. Since version 2.10, glibc does the right thing.

SEE ALSO

tgamma(3)

COLOPHON

This page is part of release 5.05 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 <https://www.kernel.org/doc/man-pages/>.