

NAME

log1p, log1pf, log1pl - logarithm of 1 plus argument

SYNOPSIS

```
#include <math.h>
```

```
double log1p(double x);
float log1pf(float x);
long double log1pl(long double x);
```

Link with *-lm*.

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

```
log1p():
    _BSD_SOURCE || _SVID_SOURCE || _XOPEN_SOURCE >= 500 ||
    _XOPEN_SOURCE && _XOPEN_SOURCE_EXTENDED || _ISOC99_SOURCE ||
    _POSIX_C_SOURCE >= 200112L;
    or cc -std=c99
log1pf(), log1pl():
    _BSD_SOURCE || _SVID_SOURCE || _XOPEN_SOURCE >= 600 || _ISOC99_SOURCE ||
    _POSIX_C_SOURCE >= 200112L;
    or cc -std=c99
```

DESCRIPTION

$\log_1 p(x)$ returns a value equivalent to

$\log(1 + x)$

It is computed in a way that is accurate even if the value of x is near zero.

RETURN VALUE

On success, these functions return the natural logarithm of $(1 + x)$.

If x is a NaN, a NaN is returned.

If x is positive infinity, positive infinity is returned.

If x is -1, a pole error occurs, and the functions return **-HUGE_VAL**, **-HUGE_VALF**, or **-HUGE_VALL**, respectively.

If x is less than -1 (including negative infinity), a domain error occurs, and a NaN (not a number) 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 less than -1

An invalid floating-point exception (**FE_INVALID**) is raised.

Pole error: x is -1

A divide-by-zero floating-point exception (**FE_DIVBYZERO**) is raised.

These functions do not set *errno*.

ATTRIBUTES

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

The **log1p()**, **log1pf()**, and **log1pl()** functions are thread-safe.

CONFORMING TO

C99, POSIX.1-2001.

SEE ALSO

[exp\(3\)](#), [expm1\(3\)](#), [log\(3\)](#)

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/>.