Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'log1p.3p' command

$ man log1p.3p

LOG1P(3P) POSIX Programmer's Manual LOG1P(3P)

PROLOG

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

NAME

log1p, log1pf, log1pl ? compute a natural logarithm

SYNOPSIS

```
#include <math.h>
double log1p(double x);
float log1pf(float x);
long double log1pl(long double x);
```

DESCRIPTION

The functionality described on this reference page is aligned with the ISO C standard. Any conflict between the requirements described here and the ISO C standard is unintentional. This volume of POSIX.1?2017 defers to the ISO C standard.

These functions shall compute loge(1.0 + x).

An application wishing to check for error situations should set errno to zero and call feclearexcept(FE_ALL_EXCEPT) before calling these functions. On return, if errno is non-zero or fetestexcept(FE_INVALID | FE_DIVBYZERO | FE_OVERFLOW | FE_UNDERFLOW) is non-zero, an error has

occurred.

RETURN VALUE

Upon successful completion, these functions shall return the natural logarithm of 1.0 + x.

If x is -1, a pole error shall occur and log1p(), log1pf(), and log1pl() shall return -HUGE_VAL, -HUGE_VALF, and -HUGE_VALL, respec? tively.

For finite values of x that are less than -1, or if x is -Inf, a domain error shall occur, and either a NaN (if supported), or an implementa? tion-defined value shall be returned.

If x is NaN, a NaN shall be returned.

If x is ?0, or +Inf, x shall be returned.

If x is subnormal, a range error may occur and x should be returned.

If x is not returned, log1p(), log1pf(), and log1pl() shall return an implementation-defined value no greater in magnitude than DBL_MIN, FLT_MIN, and LDBL_MIN, respectively.

ERRORS

These functions shall fail if:

Domain Error

The finite value of x is less than -1, or x is -Inf.

If the integer expression (math_errhandling & MATH_ERRNO) is non-zero, then errno shall be set to [EDOM]. If the in? teger expression (math_errhandling & MATH_ERREXCEPT) is non-zero, then the invalid floating-point exception shall be raised.

Pole Error The value of x is -1.

If the integer expression (math_errhandling & MATH_ERRNO) is non-zero, then errno shall be set to [ERANGE]. If the integer expression (math_errhandling & MATH_ERREXCEPT) is non-zero, then the divide-by-zero floating-point exception shall be raised.

These functions may fail if:

Range Error The value of x is subnormal.

If the integer expression (math_errhandling & MATH_ERRNO) is non-zero, then errno shall be set to [ERANGE]. If the integer expression (math_errhandling & MATH_ERREXCEPT) is non-zero, then the underflow floating-point exception shall be raised.

The following sections are informative.

EXAMPLES

None.

APPLICATION USAGE

On error, the expressions (math_errhandling & MATH_ERRNO) and (math_er?rhandling & MATH_ERREXCEPT) are independent of each other, but at least one of them must be non-zero.

RATIONALE

None.

FUTURE DIRECTIONS

None.

SEE ALSO

feclearexcept(), fetestexcept(), log()

The Base Definitions volume of POSIX.1?2017, Section 4.20, Treatment of Error Conditions for Mathematical Functions, <math.h>

COPYRIGHT

files to man page format. To report such errors, see https://www.ker?

nel.org/doc/man-pages/reporting_bugs.html .