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 'pow.3p' command

$ man pow.3p

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

pow, powf, powl ? power function

SYNOPSIS

```
#include <math.h>
double pow(double x, double y);
float powf(float x, float y);
long double powl(long double x, long double y);
```

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 the value of x raised to the power y, xy. If x is negative, the application shall ensure that y is an integer value.

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 value of x raised to the power y.

For finite values of x < 0, and finite non-integer values of y, a do? main error shall occur and either a NaN (if representable), or an im? plementation-defined value shall be returned.

If the correct value would cause overflow, a range error shall occur and pow(), powf(), and powl() shall return ?HUGE_VAL, ?HUGE_VALF, and ?HUGE_VALL, respectively, with the same sign as the correct value of the function.

If the correct value would cause underflow, and is not representable, a range error may occur, and pow(), powf(), and powl() shall return 0.0, or (if IEC 60559 Floating-Point is not supported) an implementation-de? fined value no greater in magnitude than DBL_MIN, FLT_MIN, and LDBL_MIN, respectively.

For y < 0, if x is zero, a pole error may occur and pow(), powf(), and powl() shall return ?HUGE_VAL, ?HUGE_VALF, and ?HUGE_VALL, respec? tively. On systems that support the IEC 60559 Floating-Point option, if x is ?0, a pole error shall occur and pow(), powf(), and powl() shall return ?HUGE_VAL, ?HUGE_VALF, and ?HUGE_VALL, respectively if y is an odd integer, or HUGE_VAL, HUGE_VALF, and HUGE_VALL, respectively if y is not an odd integer.

If x or y is a NaN, a NaN shall be returned (unless specified elsewhere in this description).

For any value of y (including NaN), if x is +1, 1.0 shall be returned.

For any value of x (including NaN), if y is ?0, 1.0 shall be returned.

For any odd integer value of y > 0, if x is ?0, ?0 shall be returned.

For y > 0 and not an odd integer, if x is ?0, +0 shall be returned.

If x is -1, and y is ?Inf, 1.0 shall be returned.

For |x| < 1, if y is -Inf, +Inf shall be returned.

For |x| > 1, if y is -Inf, +0 shall be returned.

For |x| < 1, if y is +Inf, +0 shall be returned.

For |x| > 1, if y is +Inf, +Inf shall be returned.

For y an odd integer < 0, if x is -Inf, -0 shall be returned.

For y < 0 and not an odd integer, if x is -Inf, +0 shall be returned.

For y an odd integer > 0, if x is -Inf, -Inf shall be returned.

For y > 0 and not an odd integer, if x is -Inf, +Inf shall be returned.

For y < 0, if x is +Inf, +0 shall be returned.

For y > 0, if x is +Inf, +Inf shall be returned.

If the correct value would cause underflow, and is representable, a range error may occur and the correct value shall be returned.

ERRORS

These functions shall fail if:

Domain Error

The value of x is negative and y is a finite non-integer. If the integer expression (math_errhandling & MATH_ERRNO) is non-zero, then errno shall be set to [EDOM]. If the integer expression (math_errhandling & MATH_ERREXCEPT) is non-zero, then the invalid floating-point exception shall be raised.

Pole Error The value of x is zero and y is negative.

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.

Range Error The result overflows.

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 overflow floating-point exception shall be raised.

These functions may fail if:

Pole Error The value of x is zero and y is negative.

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.

Range Error The result underflows.

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

exp(), feclearexcept(), fetestexcept(), isnan()

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

COPYRIGHT

event of any discrepancy between this version and the original IEEE and The Open Group Standard, the original IEEE and The Open Group Standard is the referee document. The original Standard can be obtained online at http://www.opengroup.org/unix/online.html .

Any typographical or formatting errors that appear in this page are most likely to have been introduced during the conversion of the source files to man page format. To report such errors, see https://www.ker?nel.org/doc/man-pages/reporting_bugs.html .