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 'j0.3p' command

$ man j0.3p

J0(3P) POSIX Programmer's Manual J0(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

j0, j1, jn ? Bessel functions of the first kind

SYNOPSIS

```
#include <math.h>
double j0(double x);
double j1(double x);
double jn(int n, double x);
```

DESCRIPTION

The j0(), j1(), and jn() functions shall compute Bessel functions of x of the first kind of orders 0, 1, and n, respectively.

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 relevant

Bessel value of x of the first kind.

If the x argument is too large in magnitude, or the correct result would cause underflow, 0 shall be returned and a range error may occur.

If x is NaN, a NaN shall be returned.

ERRORS

These functions may fail if:

Range Error The value of x was too large in magnitude, or an underflow occurred.

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.

No other errors shall occur.

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(), isnan(), y0()

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