NAME

isgreater, isgreaterequal, isless, islessequal, islessgreater, isunordered - floating-point relational tests without exception for NaN

SYNOPSIS

#include <math.h>

int isgreater(*x*, *y***);**

int isgreaterequal(*x*, *y***);**

int isless(*x*, *y***);**

int islessequal(*x*, *y***);**

int islessgreater(*x*, *y***);**

int isunordered(*x*, *y***);**

Link with *-lm*.

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

All functions described here:
_XOPEN_SOURCE >= 600 || _ISOC99_SOURCE || _POSIX_C_SOURCE >= 200112L;
or *cc -std=c99*

DESCRIPTION

The normal relation operations (like <, less than) will fail if one of the operands is NaN. This will cause an exception. To avoid this, C99 defines the macros listed below.

These macros are guaranteed to evaluate their arguments only once. The arguments must be of real floating-point type (note: do not pass integer values as arguments to these macros, since the arguments will *not* be promoted to real-floating types).

isgreater()
determines *(x)* > *(y)* without an exception if *x* or *y* is NaN.

isgreaterequal()
determines *(x)* >= *(y)* without an exception if *x* or *y* is NaN.

isless()
determines *(x)* < *(y)* without an exception if *x* or *y* is NaN.

islessequal()
determines *(x)* <= *(y)* without an exception if *x* or *y* is NaN.

islessgreater()
determines *(x)* < *(y)* || *(x)* > *(y)* without an exception if *x* or *y* is NaN. This macro is not equivalent to *x != y* because that expression is true if *x* or *y* is NaN.

isunordered()
determines whether its arguments are unordered, that is, whether at least one of the arguments is a NaN.

RETURN VALUE

The macros other than **isunordered**() return the result of the relational comparison; these macros return 0 if either argument is a NaN.

isunordered() returns 1 if *x* or *y* is NaN and 0 otherwise.

ERRORS

No errors occur.

ATTRIBUTES

Multithreading (see pthreads(7))

The **isgreater**(), **isgreaterequal**(), **isless**(), **islessequal**(), **islessgreater**(), and **isunordered**() macros are thread-safe.

CONFORMING TO

C99, POSIX.1-2001.

NOTES

Not all hardware supports these functions, and where hardware support isn't provided, they will be emulated by macros. This will result in a performance penalty. Don't use these functions if NaN is of no concern for you.

SEE ALSO

fpclassify(3), isnan(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/.