NAME

dinf − Double-precision +Infinity

SYNOPSIS

Fortran (77, 90, 95, HPF):

f77 *[flags] file(s) . . . -L/usr/local/lib -lgjl*

DOUBLE PRECISION FUNCTION dinf()

C (K&R, 89, 99), C++ (98):

cc *[flags] -I/usr/local/include file(s) . . . -L/usr/local/lib -lgjl*

Use

#include <gampsi.h>

to get this prototype:

fortran_double_precision dinf(void);

NB: The definition of C/C++ data types **fortran_** *xxx*, and the mapping of Fortran external names to C/C++ external names, is handled by the C/C++ header file. That way, the same function or subroutine name can be used in C, C++, and Fortran code, independent of compiler conventions for mangling of external names in these programming languages.

Last code modification: 12-Jun-2000

DESCRIPTION

Return double-precision +Infinity, or else on non-IEEE 754 systems, the largest representable floating-point number.

For IEEE 754 systems, each call to this function intentionally produces a trappable zero divide, rather than saving the computed value on the first call, and then just returning the saved value on subsequent calls.

Although every tested hardware implementation of double-precision IEEE 754 arithmetic correctly generates Infinity from zero divides, the quadruple-precision analog of this function exists because of at least one abberant software implementation of quadruple-precision arithmetic (on IBM RS/6000 AIX 4.x), which produces NaN, instead of Infinity, for the square of large numbers. Fortunately, it correctly produces Infinity for 1.0/0.0, so that is how we generate it here.

Relegating the computation of Infinity to a separate function also provides a convenient single debugger breakpoint location.

SEE ALSO

ainf(3), **qinf**(3), **isainf**(3), **isdinf**(3), **isqinf**(3).

AUTHORS

The algorithms and code are described in detail in the paper

Algorithm xxx: Quadruple-Precision Gamma(x) and psi(x) Functions for Real Arguments

in ACM Transactions on Mathematical Software, Volume ??, Number ??, Pages ????--???? and ????--????, 2001, by

Nelson H. F. Beebe
Center for Scientific Computing
University of Utah
Department of Mathematics, 110 LCB
155 S 1400 E RM 233
Salt Lake City, UT 84112-0090
Tel: +1 801 581 5254
FAX: +1 801 581 4148
Email: `beebe@math.utah.edu`, `beebe@acm.org`, `beebe@computer.org`
WWW URL: `http://www.math.utah.edu/~beebe`

and

James S. Ball
University of Utah
Department of Physics
Salt Lake City, UT 84112-0830

USA
Tel: +1 801 581 8397
FAX: +1 801 581 6256
Email: ball@physics.utah.edu
WWW URL: http://www.physics.utah.edu/people/faculty/ball.html