## NAME

**hypot**, **hypotf**, **hypotl**, **cabs**, **cabsf**, **cabsl** **-** Euclidean distance and complex absolute value functions

## LIBRARY

Math Library (libm, -lm)

## SYNOPSIS

**#include <math.h>**

*double*
**hypot**(*double x*, *double y*);

*float*
**hypotf**(*float x*, *float y*);

*long double*
**hypotl**(*long double x*, *long double y*);

**#include <complex.h>**

*double*
**cabs**(*double complex z*);

*float*
**cabsf**(*float complex z*);

*long double*
**cabsl**(*long double complex z*);

## DESCRIPTION

The **hypot**(), **hypotf**() and **hypotl**() functions compute the sqrt(x*x+y*y) in such a way that underflow will not happen, and overflow occurs only if the final result deserves it. The **cabs**(), **cabsf**() and **cabsl**() functions compute the complex absolute value of *z*.

**hypot**(*infinity*, *v*) = **hypot**(*v*, *infinity*) = +infinity for all *v*, including NaN.

## ERROR (due to Roundoff, etc.)

Below 0.97 *ulps*. Consequently **hypot**(*5.0*, *12.0*) = 13.0 exactly; in general, hypot and cabs return an integer whenever an integer might be expected.

## NOTES

As might be expected, **hypot**(*v*, *NaN*) and **hypot**(*NaN*, *v*) are NaN for all *finite v*. But programmers might be surprised at first to discover that **hypot**(+-*infinity*, *NaN*) = +infinity. This is intentional; it happens because **hypot**(*infinity*, *v*) = +infinity for *all v*, finite or infinite. Hence **hypot**(*infinity*, *v*) is independent of *v*. Unlike the reserved operand fault on a VAX, the IEEE NaN is designed to disappear when it turns out to be irrelevant, as it does in **hypot**(*infinity*, *NaN*).

## SEE ALSO

carg(3), math(3), sqrt(3)

## STANDARDS

The **hypot**(), **hypotf**(), **hypotl**(), **cabs**(), **cabsf**(), and **cabsl**() functions conform to ISO/IEC 9899:1999 ("ISO C99").

## HISTORY

Both a **hypot**() function and a **cabs**() function appeared in Version 7 AT&T UNIX.