

NAME

lgamma, **lgamma_r**, **lgammaf**, **lgammaf_r**, **lgammal**, **lgammal_r**, **gamma**, **gamma_r**, **gammaf**, **gammaf_r**, **tgamma**, **tgammaf**, **tgammal**, - log gamma functions, gamma function

LIBRARY

Math Library (libm, -lm)

SYNOPSIS

```
#include <math.h>
```

```
extern int signgam;
```

```
double
```

```
lgamma(double x);
```

```
double
```

```
lgamma_r(double x, int *signgamp);
```

```
float
```

```
lgammaf(float x);
```

```
float
```

```
lgammaf_r(float x, int *signgamp);
```

```
long double
```

```
lgammal(long double x);
```

```
long double
```

```
lgammal_r(long double x, int *signgamp);
```

```
double
```

```
gamma(double x);
```

```
double
```

```
gamma_r(double x, int *signgamp);
```

```
float
```

```
gammaf(float x);
```

```
float
```

gammalf_r(float *x*, int **signgamp*);

long double

tgamma(double *x*);

float

tgammalf(float *x*);

long double

tgammalf(long double *x*);

DESCRIPTION

lgamma(*x*), **lgammalf**(*x*), and **lgammalf**(*x*) return $\ln|\langle\text{Gamma}\rangle(x)|$. The external integer *signgam* returns the sign of $\langle\text{Gamma}\rangle(x)$.

lgamma_r(*x*, *signgamp*), **lgammalf_r**(*x*, *signgamp*), and **lgammalf_r**(*x*, *signgamp*) provide the same functionality as **lgamma**(*x*), **lgammalf**(*x*), and **lgammalf**(*x*), but the caller must provide an integer to store the sign of $\langle\text{Gamma}\rangle(x)$.

The **tgamma**(*x*), **tgammalf**(*x*), and **tgammalf**(*x*) functions return $\langle\text{Gamma}\rangle(x)$, with no effect on *signgam*.

gamma(), **gammalf**(), **gamma_r**(), and **gammalf_r**() are deprecated aliases for **lgamma**(), **lgammalf**(), **lgamma_r**(), and **lgammalf_r**(), respectively.

IDIOSYNCRASIES

Do not use the expression " $\text{signgam} \cdot \exp(\text{lgamma}(x))$ " to compute $g := \langle\text{Gamma}\rangle(x)$. Instead use a program like this (in C):

```
lg = lgamma(x); g = signgam*exp(lg);
```

Only after **lgamma**() or **lgammalf**() has returned can *signgam* be correct.

For arguments in its range, **tgamma**() is preferred, as for positive arguments it is accurate to within one unit in the last place. Exponentiation of **lgamma**() will lose up to 10 significant bits.

RETURN VALUES

gamma(), **gammalf**(), **gammalf**(), **gamma_r**(), **gammalf_r**(), **gammalf_r**(), **lgamma**(), **lgammalf**(), **lgammalf**(), **lgamma_r**(), **lgammalf_r**(), and **lgammalf_r**() return appropriate values unless an argument is out of range. Overflow will occur for sufficiently large positive values, and non-positive integers. For large non-integer negative values, **tgamma**() will underflow.

BUGS

To conform with newer C/C++ standards, a stub implementation for **tgammal** was committed to the math library, where **tgammal** is mapped to **tgamma**. Thus, the numerical accuracy is at most that of the 53-bit double precision implementation.

SEE ALSO

math(3)

STANDARDS

The **lgamma()**, **lgammaf()**, **lgammal()**, **tgamma()**, **tgammaf()**, and **tgammal()** functions are expected to conform to ISO/IEC 9899:1999 ("ISO C99").

HISTORY

The **lgamma()** function appeared in 4.3BSD. The **gamma()** function appeared in 4.4BSD as a function which computed $\langle \text{Gamma} \rangle(x)$. This version was used in FreeBSD 1.1. The name **gamma()** was originally dedicated to the **lgamma()** function, and that usage was restored by switching to Sun's `fdlibm` in FreeBSD 1.1.5. The **tgamma()** function appeared in FreeBSD 5.0.