

NAME

`fma`, `fmaf`, `fmal` – floating-point multiply and add

SYNOPSIS

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

```
double fma(double x, double y, double z);
```

```
float fmaf(float x, float y, float z);
```

```
long double fmal(long double x, long double y, long double z);
```

Link with `-lm`.

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

```
fma(), fmaf(), fmal():
```

```
_ISOC99_SOURCE || _POSIX_C_SOURCE >= 200112L
```

DESCRIPTION

These functions compute $x * y + z$. The result is rounded as one ternary operation according to the current rounding mode (see **fenv(3)**).

RETURN VALUE

These functions return the value of $x * y + z$, rounded as one ternary operation.

If x or y is a NaN, a NaN is returned.

If x times y is an exact infinity, and z is an infinity with the opposite sign, a domain error occurs, and a NaN is returned.

If one of x or y is an infinity, the other is 0, and z is not a NaN, a domain error occurs, and a NaN is returned.

If one of x or y is an infinity, and the other is 0, and z is a NaN, a domain error occurs, and a NaN is returned.

If x times y is not an infinity times zero (or vice versa), and z is a NaN, a NaN is returned.

If the result overflows, a range error occurs, and an infinity with the correct sign is returned.

If the result underflows, a range error occurs, and a signed 0 is returned.

ERRORS

See **math_error(7)** for information on how to determine whether an error has occurred when calling these functions.

The following errors can occur:

Domain error: $x * y + z$, or $x * y$ is invalid and z is not a NaN

An invalid floating-point exception (**FE_INVALID**) is raised.

Range error: result overflow

An overflow floating-point exception (**FE_OVERFLOW**) is raised.

Range error: result underflow

An underflow floating-point exception (**FE_UNDERFLOW**) is raised.

These functions do not set *errno*.

VERSIONS

These functions first appeared in glibc in version 2.1.

ATTRIBUTES

For an explanation of the terms used in this section, see **attributes(7)**.

Interface	Attribute	Value
fma() , fmaf() , fmal()	Thread safety	MT-Safe

CONFORMING TO

C99, POSIX.1-2001, POSIX.1-2008.

SEE ALSO

remainder(3), **remquo(3)**

COLOPHON

This page is part of release 5.02 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 <https://www.kernel.org/doc/man-pages/>.