

**NAME**

modf, modff, modfl – extract signed integral and fractional values from floating-point number

**LIBRARY**

Math library (*libm*, *-lm*)

**SYNOPSIS**

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

```
double modf(double x, double *iptr);
```

```
float modff(float x, float *iptr);
```

```
long double modfl(long double x, long double *iptr);
```

Feature Test Macro Requirements for glibc (see **feature\_test\_macros(7)**):

```
modff(), modfl():
```

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

```
|| /* Since glibc 2.19: */ _DEFAULT_SOURCE
```

```
|| /* glibc <= 2.19: */ _BSD_SOURCE || _SVID_SOURCE
```

**DESCRIPTION**

These functions break the argument *x* into an integral part and a fractional part, each of which has the same sign as *x*. The integral part is stored in the location pointed to by *iptr*.

**RETURN VALUE**

These functions return the fractional part of *x*.

If *x* is a NaN, a NaN is returned, and *\*iptr* is set to a NaN.

If *x* is positive infinity (negative infinity), +0 (−0) is returned, and *\*iptr* is set to positive infinity (negative infinity).

**ERRORS**

No errors occur.

**ATTRIBUTES**

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

Interface	Attribute	Value
<b>modf()</b> , <b>modff()</b> , <b>modfl()</b>	Thread safety	MT-Safe

**STANDARDS**

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

The variant returning *double* also conforms to SVr4, 4.3BSD.

**SEE ALSO**

**frexp(3)**, **ldexp(3)**