



**Full credit is given to the above companies including the Operating System (OS) that this PDF file was generated!**

***Rocky Enterprise Linux 9.2 Manual Pages on command 'y1.3'***

**\$ man y1.3**

Y0(3)                      Linux Programmer's Manual                      Y0(3)

NAME

y0, y0f, y0l, y1, y1f, y1l, yn, ynf, ynl - Bessel functions of the second kind

SYNOPSIS

```
#include <math.h>

double y0(double x);
double y1(double x);
double yn(int n, double x);
float y0f(float x);
float y1f(float x);
float ynf(int n, float x);
long double y0l(long double x);
long double y1l(long double x);
long double ynl(int n, long double x);

Link with -lm.
```

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

y0(), y1(), yn():

```
_XOPEN_SOURCE
|| /* Since glibc 2.19: */ _DEFAULT_SOURCE
|| /* Glibc versions <= 2.19: */ _SVID_SOURCE || _BSD_SOURCE
```

y0f(), y0l(), y1f(), y1l(), ynf(), ynl():

```
_XOPEN_SOURCE >= 600
|| (_ISOC99_SOURCE && _XOPEN_SOURCE)
```

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

```
|| /* Glibc versions <= 2.19: */ _SVID_SOURCE || _BSD_SOURCE
```

## DESCRIPTION

The `y0()` and `y1()` functions return Bessel functions of  $x$  of the second kind of orders 0 and 1, respectively. The `yn()` function returns the Bessel function of  $x$  of the second kind of order  $n$ .

The value of  $x$  must be positive.

The `y0f()`, `y1f()`, and `ynf()` functions are versions that take and return float values. The `y0l()`, `y1l()`, and `ynl()` functions are versions that take and return long double values.

## RETURN VALUE

On success, these functions return the appropriate Bessel value of the second kind for  $x$ .

If  $x$  is a NaN, a NaN is returned.

If  $x$  is negative, a domain error occurs, and the functions return `-HUGE_VAL`, `-HUGE_VALF`, or `-HUGE_VALL`, respectively. (POSIX.1-2001 also allows a NaN return for this case.)

If  $x$  is 0.0, a pole error occurs, and the functions return `-HUGE_VAL`, `-HUGE_VALF`, or `-HUGE_VALL`, respectively.

If the result underflows, a range error occurs, and the functions return 0.0

If the result overflows, a range error occurs, and the functions return `-HUGE_VAL`, `-HUGE_VALF`, or `-HUGE_VALL`, respectively. (POSIX.1-2001 also allows a 0.0 return for this case.)

## 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:  $x$  is negative

`errno` is set to `EDOM`. An invalid floating-point exception (`FE_INVALID`) is raised.

Pole error:  $x$  is 0.0

`errno` is set to `ERANGE` and an `FE_DIVBYZERO` exception is raised (but see `BUGS`).

Range error: result underflow

`errno` is set to `ERANGE`. No `FE_UNDERFLOW` exception is returned by `fetestexcept(3)` for this case.

Range error: result overflow

`errno` is set to `ERANGE` (but see `BUGS`). An overflow floating-point exception

(FE\_OVERFLOW) is raised.

## ATTRIBUTES

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

??

?Interface ? Attribute ? Value ?

??

?y0(), y0f(), y0l() ? Thread safety ? MT-Safe ?

??

?y1(), y1f(), y1l() ? Thread safety ? MT-Safe ?

??

?yn(), ynf(), ynl() ? Thread safety ? MT-Safe ?

??

## CONFORMING TO

The functions returning double conform to SVr4, 4.3BSD, POSIX.1-2001, POSIX.1-2008. The others are nonstandard functions that also exist on the BSDs.

## BUGS

Before glibc 2.19, these functions misdiagnosed pole errors: errno was set to EDOM, in? stead of ERANGE and no FE\_DIVBYZERO exception was raised.

Before glibc 2.17, did not set errno for "range error: result underflow".

In glibc version 2.3.2 and earlier, these functions do not raise an invalid floating-point exception (FE\_INVALID) when a domain error occurs.

## SEE ALSO

j0(3)

## COLOPHON

This page is part of release 5.10 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/>.

2020-06-09

Y0(3)