

NAME

cacos, cacosf, cacosl – complex arc cosine

LIBRARY

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

SYNOPSIS

```
#include <complex.h>

double complex cacos(double complex z);
float complex cacosf(float complex z);
long double complex cacosl(long double complex z);
```

DESCRIPTION

These functions calculate the complex arc cosine of *z*. If *y* = *cacos*(*z*), then *z* = *ccos*(*y*). The real part of *y* is chosen in the interval [0,π].

One has:

$$cacos(z) = -i * \log(z + i * \sqrt{1 - z * z})$$

ATTRIBUTES

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

Interface	Attribute	Value
cacos() , cacosf() , cacosl()	Thread safety	MT-Safe

STANDARDS

C11, POSIX.1-2008.

HISTORY

glibc 2.1. C99, POSIX.1-2001.

EXAMPLES

```
/* Link with "-lm" */

#include <complex.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int
main(int argc, char *argv[])
{
    double complex z, c, f;
    double complex i = I;

    if (argc != 3) {
        fprintf(stderr, "Usage: %s <real> <imag>\n", argv[0]);
        exit(EXIT_FAILURE);
    }

    z = atof(argv[1]) + atof(argv[2]) * I;

    c = cacos(z);

    printf("cacos() = %6.3f %6.3f*i\n", creal(c), cimag(c));

    f = -i * clog(z + i * csqrt(1 - z * z));

    printf("formula = %6.3f %6.3f*i\n", creal(f), cimag(f));
```

```
        exit(EXIT_SUCCESS);  
    }
```

SEE ALSO**ccos(3), clog(3), complex(7)**