

**NAME**

`cacos`, `cacosf`, `cacosl` – complex arc cosine

**SYNOPSIS**

```
#include <complex.h>
```

```
double complex cacos(double complex z);
```

```
float complex cacosf(float complex z);
```

```
long double complex cacosl(long double complex z);
```

Link with `-lm`.

**DESCRIPTION**

These functions calculate the complex arc cosine of  $z$ . If  $y = \text{cacos}(z)$ , then  $z = \text{ccos}(y)$ . The real part of  $y$  is chosen in the interval  $[0, \pi]$ .

One has:

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

**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
<code>cacos()</code> , <code>cacosf()</code> , <code>cacosl()</code>	Thread safety	MT-Safe

**CONFORMING TO**

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

**EXAMPLE**

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

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

**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/>.