

[Help](#)

```
#ifndef __COMPLEX_H__
#define __COMPLEX_H__

#ifndef _FCOMPLEX_DECLARE_T_
typedef struct FCOMPLEX {double r,i;} fcomplex;
#define _FCOMPLEX_DECLARE_T_
#endif /* _FCOMPLEX_DECLARE_T_ */

#define CUNO (Complex(1,0))
#define CI (Complex(0,1))

#if defined(__STDC__) || defined(ANSI) || define
    d(NRANSI) /* ANSI */
double Real( fcomplex g );
double Imm( fcomplex g );
fcomplex Cadd(fcomplex a, fcomplex b);
fcomplex Csub(fcomplex a, fcomplex b);
fcomplex Cmul(fcomplex a, fcomplex b);
fcomplex Complex(double re, double im);
fcomplex Conjg(fcomplex z);
fcomplex Cdiv(fcomplex a, fcomplex b);
double Cabs(fcomplex z);
fcomplex Csqrt(fcomplex z);
fcomplex Csqrt();
fcomplex Clog(fcomplex z);
fcomplex Cexp(fcomplex z);
fcomplex RCmul(double a, fcomplex b);
double Carg(fcomplex a);
fcomplex Cgamma(fcomplex a);
double Cnp(int n, int p);
double fact(int n);
fcomplex cgammln(fcomplex xx);
#else /* ANSI */
/* traditional - K&R */
double Real( fcomplex g );
double Imm( fcomplex g );
fcomplex Cadd();
fcomplex Csub();
fcomplex Cmul();
```

```
fcomplex Complex();
fcomplex Conjg();
fcomplex Cdiv();
double Cabs();
fcomplex Csqrt();
fcomplex Clog(fcomplex z);
fcomplex Cexp(fcomplex z);
fcomplex Cpow(fcomplex z, fcomplex esp);
fcomplex RCmul();
double Carg();
fcomplex Cgamma();
double Cnp();
double fact();
fcomplex cgammln(fcomplex xx);
#endif /* ANSI */
#endif /* __COMPLEX_H_ */
```

References