

A.D.T. Complex

Created by Gianluca DeLucia
(N86001826)
gian.delucia@studenti.unina.it

SCOPO:

Sviluppare ogni possibile operazione su due numeri complessi.

SPECIFICHE:

```
complex somma (complex x, complex y, complex z);  
complex sottrazione (complex x, complex y, complex z);  
complex moltiplicazione (complex x, complex y, complex z);  
complex divisione (complex x, complex y, complex z);  
double modulo (complex x, complex y, double &modulo1, double  
                                                         &modulo2);  
void coniugato (complex x, complex y, complex &z, complex &v);
```

DESCRIZIONE:

a) Background del problema:

Operare:

- 1) Somma tra due numeri complessi
- 2) Sottrazione tra due numeri complessi
- 3) Moltiplicazione tra due numeri complessi
- 4) Divisione tra due numeri complessi
- 5) Modulo di due numeri complessi
- 6) Coniugato di due numeri complessi

b) Algoritmo:

```
function Somma(in: X, Y, Z)  
  var X, Y, Z :Complex  
  begin  
    Z.Re := X.Re+Y.Re;  
    Z.Im := X.Im+Y.Im;  
  end  
end Somma  
  
function Sottrazione(in: X, Y, Z)  
  var X, Y, Z :Complex  
  begin  
    Z.Re := X.Re-Y.Re;  
    Z.Im := X.Im-Y.Im;  
  end  
end Sottrazione
```

```

function Moltiplicazione(in: X, Y, Z)
  var X, Y, Z :Complex
  begin
    Z.Re := X.Re*Y.Re;
    Z.Im := X.Im*Y.Im;
  end
end Moltiplicazione

```

```

function Divisione(in: X, Y, Z)
  var X, Y, Z :Complex
  begin
    Z.Re := X.Re/Y.Re;
    Z.Im := X.Im/Y.Im;
  end
end Divisione

```

```

function modulo (in: x, y; in/out: modulo1,modulo2)
  var x, y : Complex
  var modulo1,modulo2 : real
  begin
    modulo1 = sqrt[(x.re)^2 + (x.im)^2];
    modulo2 = sqrt[(y.re)^2 + (y.im)^2];
  end
end modulo

```

```

function coniugato(in: x, y; in/out: z,v)
  var x, y ,z, v : Complex
  begin
    z.re := (x.re);
    z.im := -(x.im);

    v.re := (y.re);
    v.im := -(y.im);
  end
end coniugato

```

RIFERIMENTI BIBLIOGRAFICI:

<http://www.federica.unina.it/corsi/programmazione-ii-laboratorio/>
 A. Murli, G. Laccetti, Laboratorio di Programmazione I, Ed. Liguori.
 Criscuolo, F.P. Tramontano, Introduzione alla Programmazione, Algoritmi Imperativi in C++, Ed. Manna.
 G. Dromey, Algoritmi fondamentali, Ed. Jackson.
 B. Kernighan, D. Ritchie, Linguaggio C, Ed. Jackson.
 H. M. Deitel, P. J. Deitel, Corso completo di programmazione, Ed. Apogeo.

LISTA DEI PARAMETRI:

```

type Complex : record
  Re : real
  Im : real
end

```

```

complex x,y,z,v;
char risp;
double modulo1, modulo2;

```

INDICATORI DI ERRORI:

Nessuno.

PROCEDURE AUSILIARIE:

Nessuna.

RACCOMANDAZIONI SULL'USO:

Non utilizzare numeri che abbiano più di otto cifre.

ESEMPIO D'USO:

a) Esempio di programma chiamante:

```

void Adt_complex ()
{
    int i;
    int scelta;
    complex x,y,z,v;
    char risp;
    double modulo1, modulo2;

    cout<<"\t\t\t\tA.D.T. Complex\n\n\n";

    do{
        cout<<"Immettere due numeri complessi su cui eseguire le operazioni.\n\n";
        cout<<"\tX ===> ";
        carica_complex (x);
        cout<<"\n\n\tY ===> ";
        carica_complex (y);

        cout<<"\n\nScegli quale operazione svolgere:\n\n";
        cout<<"\t1) Somma\n\t2) Sottrazione\n\t3) Moltiplicazione\n\t4) Divisione\n\t5) Modulo\n\t6) Coniugato\n\n ===> ";
        cin>>scelta;

        switch (scelta)
        {
            case (1):
            {
                cout<<"\n ===> SOMMA ";
                z = somma (x,y,z);
                cout<<" ===> Z = ("<<z.re<<" , "<<z.im<<"i)";
            }break;

            case (2):
            {
                cout<<"\n ===> SOTTRAZIONE ";
                z = sottrazione (x,y,z);
                cout<<" ===> Z = ("<<z.re<<" , "<<z.im<<"i)";
            }break;

```

```

        case (3):
        {
            cout<<"\n ===>>  MOLTIPLICAZIONE ";
            z = moltiplicazione(x,y,z);
            cout<<" ===>>  Z = ("<<z.re<<" , "<<z.im<<"i )";
        }break;

        case (4):
        {
            cout<<"\n ===>>  DIVISIONE ";
            z = divisione (x,y,z);
            cout<<" ===>>  Z = ("<<z.re<<" , "<<z.im<<"i )";
        }break;

        case (5):
        {
            cout<<"\n ===>>  MODULO ";
            modulo(x,y,modulo1,modulo2);
            cout<<" ===>>  Modulo X = "<<modulo1;
            cout<<"\n          ===>>  Modulo Y = "<<modulo2

        }break;

        case (6):
        {
            cout<<"\n ===>>  CONIUGATO ";
            coniugato(x,y,z,v);
            cout<<" ===>>  Coniugato di X = ("<<z.re<<" , "<<z.im<<"i )";
            cout<<"\n===>>  Coniugato di Y = ("<<v.re<<" , "<<v.im<<"i )";
        }

    }

    cout<<"\n\nVuoi fare un'altra operazione <s/n> ? ";
    cin>>risp;
    system ("cls");



}while(risp == 's');



```

b) Esempio di esecuzione:

A.D.T. Complex

Immettere due numeri complessi su cui eseguire le operazioni.

X ===>>  Parte Reale = 2
  Parte immaginaria = 3

Y ===>>  Parte Reale = 5
  Parte immaginaria = 4

Scegli quale operazione svolgere:

- 1) Somma
- 2) Sottrazione
- 3) Moltiplicazione
- 4) Divisione

- 5) Modulo
- 6) Coniugato

====> 3

MOLTIPLICAZIONE =====> $Z = (10, 12i)$

Vuoi fare un'altra operazione? <s/n>