

RELAZIONE PROGETTO PPL

Davide Cortellucci

Matricola 260321

Specifica del problema

Scrivere una libreria ANSI C che gestisce i numeri complessi esportando le seguenti funzioni. La prima funzione C acquisisce da tastiera un numero complesso in forma algebrica (rispettivamente, trigonometrica) e stampa a video il numero complesso equivalente in forma trigonometrica (rispettivamente, algebrica). La seconda funzione C acquisisce da tastiera due numeri complessi in forma algebrica (rispettivamente, trigonometrica) e stampa a video la loro somma in forma algebrica (rispettivamente, trigonometrica). La terza funzione C acquisisce da tastiera due numeri complessi in forma algebrica (rispettivamente, trigonometrica) e stampa a video la loro differenza in forma algebrica (rispettivamente, trigonometrica). La quarta funzione C acquisisce da tastiera due numeri complessi in forma algebrica (rispettivamente, trigonometrica) e stampa a video il loro prodotto in forma algebrica (rispettivamente, trigonometrica). La quinta funzione C acquisisce da tastiera due numeri complessi in forma algebrica (rispettivamente, trigonometrica) e stampa a video il loro quoziente in forma algebrica (rispettivamente, trigonometrica).

Analisi del problema

1. Prima funzione:

- input: un numero complesso in forma algebrica (rispettivamente, trigonometrica);
- output: tale numero convertito in forma trigonometrica (rispettivamente, algebrica);
- relazioni: dato un numero $z = a + ib$ in forma algebrica, con a parte reale e b parte immaginaria, il corrispettivo in forma trigonometrica sarà un numero della forma $z = r[\cos(\theta) + i \sin(\theta)]$, con r modulo e θ argomento, questi ultimi ricavabili dalle seguenti formule:

$$r = \sqrt{a^2 + b^2}$$

Per $\theta \in (0, 2\pi]$:

$$\theta = \begin{cases} \frac{\pi}{2} & \text{se } a = 0, b > 0 \\ \frac{3\pi}{2} & \text{se } a = 0, b < 0 \\ \text{non definito} & \text{se } a = 0, b = 0 \\ \arctan\left(\frac{b}{a}\right) & \text{se } a > 0, b \geq 0 \\ \arctan\left(\frac{b}{a}\right) + 2\pi & \text{se } a > 0, b < 0 \\ \arctan\left(\frac{b}{a}\right) + \pi & \text{se } a < 0, b \text{ qualsiasi} \end{cases}$$

2. Seconda funzione:

- input: due numeri complessi in forma algebrica (rispettivamente, trigonometrica);
- output: somma di tali numeri in forma algebrica (rispettivamente, trigonometrica);
- relazioni: dati $z_1 = a + ib$ e $z_2 = c + id$, $z_1 + z_2 = (a + c) + i(b + d)$ per la forma algebrica, mentre per quella trigonometrica, dati $z_1 = r_1[\cos(\theta_1) + i \sin(\theta_1)]$ e $z_2 = r_2[\cos(\theta_2) + i \sin(\theta_2)]$, $z_1 + z_2 = [r_1 \cos(\theta_1) + r_2 \cos(\theta_2)] + i[r_1 \sin(\theta_1) + r_2 \sin(\theta_2)]$.

3. Terza funzione:

- input: due numeri complessi in forma algebrica (rispettivamente, trigonometrica);
- output: differenza di tali numeri in forma algebrica (rispettivamente, trigonometrica);
- relazioni: dati $z_1 = a + ib$ e $z_2 = c + id$, $z_1 - z_2 = (a - c) + i(b - d)$ per la forma algebrica, mentre per quella trigonometrica, dati $z_1 = r_1[\cos(\theta_1) + i \sin(\theta_1)]$ e $z_2 = r_2[\cos(\theta_2) + i \sin(\theta_2)]$, $z_1 - z_2 = [r_1 \cos(\theta_1) - r_2 \cos(\theta_2)] + i[r_1 \sin(\theta_1) - r_2 \sin(\theta_2)]$.

4. Quarta funzione:

- input: due numeri complessi in forma algebrica (rispettivamente, trigonometrica);
- output: prodotto di tali numeri in forma algebrica (rispettivamente, trigonometrica);

- relazioni: dati $z_1 = a + ib$ e $z_2 = c + id$, $z_1 z_2 = (ac - bd) + i(ad + bc)$ per la forma algebrica, mentre per quella trigonometrica, dati $z_1 = r_1[\cos(\theta_1) + i \sin(\theta_1)]$ e $z_2 = r_2[\cos(\theta_2) + i \sin(\theta_2)]$, $z_1 z_2 = r_1 r_2 [\cos(\theta_1 + \theta_2) + i \sin(\theta_1 + \theta_2)]$.

5. Quinta funzione:

- input: due numeri complessi in forma algebrica (rispettivamente, trigonometrica);
- output: quoziente di tali numeri in forma algebrica (rispettivamente, trigonometrica);
- relazioni: dati $z_1 = a + ib$ e $z_2 = c + id$, $\frac{z_1}{z_2} = (ac - bd) + i(ad + bc)$ per la forma algebrica, mentre per quella trigonometrica, dati $z_1 = r_1[\cos(\theta_1) + i \sin(\theta_1)]$ e $z_2 = r_2[\cos(\theta_2) + i \sin(\theta_2)]$, $\frac{z_1}{z_2} = \frac{r_1}{r_2} [\cos(\theta_1 - \theta_2) + i \sin(\theta_1 - \theta_2)]$.

Progettazione dell'algoritmo

Scelte di progetto

- Utilizzo due strutture, una per i numeri complessi in forma algebrica ed una per quelli in forma trigonometrica. La prima ha un campo per la parte reale ed uno per la parte immaginaria, mentre la seconda ha un campo per il modulo ed uno per l'argomento.
- I numeri in forma algebrica sono rappresentati nella forma $a + ib$, dove a è la parte reale, i è l'unità immaginaria e b la parte immaginaria. I numeri in forma trigonometrica sono rappresentati nella forma $r[\cos(\theta) + i \sin(\theta)]$, dove r è il modulo e θ è l'argomento.
- I numeri vengono stampati con 2 cifre decimali.
- Si richiede all'utente di inserire l'argomento dei numeri complessi in forma trigonometrica in radianti; tale argomento verrà stampato a video sempre in radianti.
- Nella prima funzione, durante la conversione da forma algebrica a trigonometrica, i calcoli per la conversione tengono conto del fatto che $\theta \in (0, 2\pi]$.
- Nelle funzioni che calcolano la somma e la differenza di due numeri complessi, nel caso in cui essi siano espressi in forma trigonometrica, tali numeri vengono convertiti in forma algebrica e ne viene computata la somma.
- Nella funzione che calcola il quoziente di due numeri complessi si controlla che il divisore sia diverso da zero. Se non lo è, la funzione comunica all'utente che la divisione non è effettuabile.

Passi dell'algoritmo

1. Definizione delle strutture
2. Dichiarazione delle funzioni
3. Prima funzione:
 - Inizializzazione strutture
 - Richiesta forma (algebrica o trigonometrica) dall'utente
 - Acquisizione numero e stampa a video
 - Conversione e stampa a video
4. Seconda funzione
 - Inizializzazione strutture
 - Richiesta forma (algebrica o trigonometrica) dall'utente
 - Acquisizione numeri e stampa a video
 - Calcolo somma e stampa a video
5. Terza funzione

- Inizializzazione strutture
- Richiesta forma (algebrica o trigonometrica) dall'utente
- Acquisizione numeri e stampa a video
- Calcolo differenza e stampa a video

6. Quarta funzione

- Inizializzazione strutture
- Richiesta forma (algebrica o trigonometrica) dall'utente
- Acquisizione numeri e stampa a video
- Calcolo prodotto e stampa a video

7. Quinta funzione

- Inizializzazione strutture
- Richiesta forma (algebrica o trigonometrica) dall'utente
- Acquisizione numeri e stampa a video
- Controllo divisore ed eventuale calcolo del quoziente e successiva stampa a video

Implementazione dell'algoritmo

File lib_num_complessi.c

```
/* Direttive al preprocessore */

#include <stdio.h>
#include <math.h>

#define PI_GRECO 3.14

/* Definizione dei tipi */

typedef struct num_compl_alg
{
    double parte_reale,
           parte_imm;
}num_compl_alg_t;

typedef struct num_compl_tri
{
    double modulo,
           argomento;
}num_compl_tri_t;

/* Dichiarazione delle funzioni */

void acquisisci_e_stampa(void);
void inizializza_strutt(num_compl_alg_t,
                       num_compl_tri_t);
void somma_num_compl(void);
void inizializza_compl_alg(num_compl_alg_t,
                           num_compl_alg_t);
void stampa_alg(num_compl_alg_t);
void stampa_tri(num_compl_tri_t);
num_compl_alg_t acquisisci_compl_alg(void);
num_compl_tri_t acquisisci_compl_tri(void);
void differenza_num_compl(void);
void prodotto_num_compl(void);
int convalida_scelta(void);
void rapporto_num_compl(void);

/* Definizione delle funzioni */

void inizializza_strutt(num_compl_alg_t compl_1,
                       num_compl_tri_t compl_2)
{
    compl_1.parte_reale = 0;
    compl_1.parte_imm = 0;
```

```

    compl_2.modulo = 0;
    compl_2.argomento = 0;
}

/*****/

void inizializza_compl_alg(num_compl_alg_t num_1,
                           num_compl_alg_t num_2)
{
    num_1.parte_reale = 0;
    num_1.parte_imm = 0;

    num_2.parte_reale = 0;
    num_2.parte_imm = 0;
}

/*****/

int convalida_scelta(void)
{
    int esito_lettura;      /* lavoro: validazione stretta */
    char caratt_rimosso;    /* lavoro: validazione stretta */
    int scelta = 0;         /* output: scelta dell'utente */

    do                      /** VALIDAZIONE STRETTA **/
    {
        printf("Digitare 1 per la forma algebrica o 2 per\nla forma trigonometrica: ");
        esito_lettura = scanf("%d",
                               &scelta);
        if (esito_lettura != 1 || scelta < 1 || scelta > 2)
        {
            printf("Input non valido!\n");
            do
            {
                caratt_rimosso = getchar();
                while (caratt_rimosso != '\n');
            }
        }
        while (esito_lettura != 1 || scelta < 1 || scelta > 2);    /** FINE VALIDAZIONE STRETTA **/

    }

    return(scelta);
}

/*****/

void stampa_alg(num_compl_alg_t numero)
{
    char parentesi_1 = '(',      /* lavoro: stampa a video */
        parentesi_2 = ')';      /* lavoro: stampa a video */

```



```

/* stampo nella forma a + ib = numero complesso */
if (numero.parte_imm >= 0)
    printf("%.2f + %.2f i.\n\n",
           numero.parte_reale,
           numero.parte_imm);
else
    printf("%.2f + %c%.2f%c i.\n\n",
           numero.parte_reale,
           parentesi_1,
           numero.parte_imm,
           parentesi_2);
}

/*****/

void stampa_tri(num_compl_tri_t numero)
{
    /* stampo nella forma r[cos(teta) + i sin(teta)] */
    if (numero.argomento == 10) /* argomento indefinito
                                   (a = 0, b = 0) */
        printf("indefinito (parte reale ed immaginaria
               uguali a zero!)\n\n");
    else
    {
        printf("%.2f[cos(%.2f) + i sin(%.2f)].\n",
               numero.modulo,
               numero.argomento,
               numero.argomento);
        printf("***   Notare che l'argomento e' espresso
               in radianti!   ***\n");
    }
}

/*****/

num_compl_alg_t acquisisci_compl_alg(void)
{
    int esito_lettura;          /* lavoro: validazione stretta */
    char caratt_rimosso;        /* lavoro: validazione stretta */
    num_compl_alg_t numero;     /* numero in forma algebrica */

    /* acquisisco la parte reale */
    do                          /** VALIDAZIONE STRETTA **/
    {
        printf("Inserire la parte reale: ");
        esito_lettura = scanf("%lf",
                               &numero.parte_reale);
        if (esito_lettura != 1)
        {

```

```

        printf("Input non valido!\n");
        do
            caratt_rimosso = getchar();
            while (caratt_rimosso != '\n');
        }
    }
    while (esito_lettura != 1); /** FINE VALIDAZIONE STRETTA **/

    /* acquisisco la parte immaginaria */
    do
        /** VALIDAZIONE STRETTA **/
    {
        printf("Inserire la parte immaginaria (escludendo i,
            che verra' aggiunta automaticamente): ");
        esito_lettura = scanf("%lf",
            &numero.parte_imm);
        if (esito_lettura != 1)
        {
            printf("Input non valido!\n");
            do
                caratt_rimosso = getchar();
                while (caratt_rimosso != '\n');
            }
        }
    }
    while (esito_lettura != 1); /** FINE VALIDAZIONE STRETTA **/

    return(numero);
}

/*****

num_compl_tri_t acquisisci_compl_tri(void)
{
    int esito_lettura;          /* lavoro: validazione stretta */
    char caratt_rimosso;        /* lavoro: validazione stretta */
    num_compl_tri_t numero;     /* memorizza il numero in forma
                                trigonometrica */

    do
        /** VALIDAZIONE STRETTA **/
    {
        printf("Inserire il modulo (si ricorda che deve essere
            reale positivo): ");
        esito_lettura = scanf("%lf",
            &numero.modulo);
        if (esito_lettura != 1 || numero.modulo < 0)
        {
            printf("Input non valido!\n");
            do
                caratt_rimosso = getchar();
                while (caratt_rimosso != '\n');
            }
        }
    }
}

```

```

}
while (esito_lettura != 1 ||
        numero.modulo < 0);    /** FINE VALIDAZIONE STRETTA **/

do                                /** VALIDAZIONE STRETTA **/
{
    printf("Inserire l'argomento (in radianti): ");
    esito_lettura = scanf("%lf",
                        &numero.argomento);
    if (esito_lettura != 1 || numero.argomento < 0 ||
        numero.argomento > 6.28)
    {
        printf("Input non valido!\n");
        do
        {
            caratt_rimosso = getchar();
            while (caratt_rimosso != '\n');
        }
    }
}
while (esito_lettura != 1 || numero.argomento < 0 ||
        numero.argomento > 6.28);
    /** FINE VALIDAZIONE STRETTA **/

return(numero);
}

/*****

void acquisisci_e_stampa(void)
{
    num_compl_alg_t forma_alg;    /* forma algebrica */
    num_compl_tri_t forma_tri;    /* forma trigonometrica */
    int scelta = 0;                /* scelta tra forma algebrica(1)
                                    e trigonometrica(2) */

    /** inizializzo le strutture **/
    inizializza_strutt(forma_alg, forma_tri);

    /** scelta della forma **/
    printf("Scegliere se si vuole esprimere il numero
           complesso in forma algebrica o trigonometrica.\n\n");
    scelta = convalida_scelta();

    /** acquisizione della forma e relativa conversione **/
    if (scelta == 1)                /* forma algebrica */
    {
        /* acquisisco il numero dall'utente */
        forma_alg = acquisisci_compl_alg();

        /* stampo nella forma a + ib = numero complesso */
        printf("\nIl numero in forma algebrica inserito e' ");
    }
}

```

```

        stampa_alg(forma_alg);

    /** conversione **/
    /* ricavo il modulo del numero */
    forma_tri.modulo = sqrt( pow(forma_alg.parte_reale, 2)
                            + pow(forma_alg.parte_imm, 2) );

    /* ricavo l'argomento del numero (distinguo 6 casi) */
    if (forma_alg.parte_reale == 0
        && forma_alg.parte_imm > 0) /* a = 0, b > 0 */
        forma_tri.argomento = PI_GRECO / 2;
    else if (forma_alg.parte_reale == 0
        && forma_alg.parte_imm < 0) /* a = 0, b < 0 */
        forma_tri.argomento = 3 * (PI_GRECO / 2);
    else if (forma_alg.parte_reale == 0
        && forma_alg.parte_imm == 0) /* a = 0, b = 0 */
        forma_tri.argomento = 10; /* 10 sta per "non
                                   definito" */
    else if (forma_alg.parte_reale > 0
        && forma_alg.parte_imm >= 0) /* a > 0, b >= 0 */
        forma_tri.argomento = atan(forma_alg.parte_imm
                                   / forma_alg.parte_reale);
    else if (forma_alg.parte_reale > 0
        && forma_alg.parte_imm < 0) /* a > 0, b < 0 */
        forma_tri.argomento = atan(forma_alg.parte_imm
                                   / forma_alg.parte_reale)
                               + (2 * PI_GRECO);
    else if (forma_alg.parte_reale < 0) /* a < 0, b assume
                                         qualsiasi valore */
        forma_tri.argomento = atan(forma_alg.parte_imm
                                   / forma_alg.parte_reale)
                               + PI_GRECO;

    /* stampo nella forma r[cos(teta) + i sin(teta)] */
    printf("L'equivalente in forma trigonometrica e' ");
    stampa_tri(forma_tri);
}
else /* forma trigonometrica */
{
    /* acquisisco il numero dall'utente */
    forma_tri = acquisisci_compl_tri();

    /* stampo a video il numero complesso inserito
       in forma trigonometrica */
    printf("\nIl numero in forma trigonometrica inserito e' ");
    stampa_tri(forma_tri);

    /** conversione **/
    forma_alg.parte_reale = forma_tri.modulo
                          * cos(forma_tri.argomento);

```

```

        forma_alg.parte_imm = forma_tri.modulo
                        * sin(forma_tri.argomento);

        /* stampo nella forma a + ib = numero complesso */
        printf("L'equivalente in forma algebrica e' ");
        stampa_alg(forma_alg);
    }
}/** fine funzione acquisisci_e_stampa **/

/*****

void somma_num_compl(void)
{
    num_compl_alg_t alg_1,      /* forma algebrica */
                        alg_2,  /* forma algebrica */
    somma_alg;                /* somma di due numeri
                                in forma algebrica */
    num_compl_tri_t tri_1,     /* forma trigonometrica */
                        tri_2; /* forma trigonometrica */
    num_compl_alg_t addendo_1, /* forma trigonometrica
                                convertito in forma
                                algebrica */
                        addendo_2, /* forma trigonometrica
                                convertito in forma
                                algebrica */
                        somma_tri; /* somma di due numeri compl.
                                in forma trigonometrica */
    int scelta = 0;            /* scelta tra forma algebrica
                                (1) e trigonometrica(2) */

    /** inizializzo le strutture **/
    inizializza_strutt(alg_1, tri_1);
    inizializza_strutt(alg_2, tri_2);
    inizializza_compl_alg(addendo_1, addendo_2);
    inizializza_compl_alg(somma_alg, somma_tri);

    /** scelta della forma **/
    printf("Scegliere se si vuole effettuare la somma tra numeri
            complessi in forma algebrica o trigonometrica.\n");
    scelta = convalida_scelta();

    if (scelta == 1)          /* forma algebrica */
    {
        /** acquisizione della forma algebrica e relativa somma **/
        printf("Primo addendo...\n");
        alg_1 = acquisisci_compl_alg();
        printf("Secondo addendo...\n");
        alg_2 = acquisisci_compl_alg();

        /* stampo a video il primo addendo */

```

```

printf("\nIl primo addendo inserito e' ");
stampa_alg(alg_1);

/* stampo a video il secondo addendo */
printf("Il secondo addendo inserito e' ");
stampa_alg(alg_2);

/* calcolo la somma in forma algebrica e la stampo */
somma_alg.parte_reale = alg_1.parte_reale
                      + alg_2.parte_reale;
somma_alg.parte_imm = alg_1.parte_imm
                     + alg_2.parte_imm;

printf("La somma dei numeri complessi inseriti e' ");
stampa_alg(somma_alg);
}
else /* forma trigonometrica */
{
    /** acquisizione della forma trigonometrica **/
    printf("Primo addendo...\n");
    tri_1 = acquisisci_compl_tri();
    printf("Secondo addendo...\n");
    tri_2 = acquisisci_compl_tri();

    /* stampo a video il primo addendo */
    printf("\nIl primo addendo inserito e' ");
    stampa_tri(tri_1);

    /* stampo a video il secondo addendo */
    printf("Il secondo addendo inserito e' ");
    stampa_tri(tri_2);

    /* calcolo somma in forma trigonometrica e stampo */
    /** conversione addendi in forma algebrica **/
    addendo_1.parte_reale = tri_1.modulo
                          * cos(tri_1.argomento);
    addendo_1.parte_imm = tri_1.modulo
                        * sin(tri_1.argomento);

    addendo_2.parte_reale = tri_2.modulo
                          * cos(tri_2.argomento);
    addendo_2.parte_imm = tri_2.modulo
                        * sin(tri_2.argomento);

    /** calcolo della somma in forma algebrica **/
    somma_tri.parte_reale = addendo_1.parte_reale
                          + addendo_2.parte_reale;
    somma_tri.parte_imm = addendo_1.parte_imm
                        + addendo_2.parte_imm;
}

```

```

        printf("La somma dei numeri complessi inseriti e' ");
        stampa_alg(somma_tri);
    }
}/** fine funzione somma_num_compl **/

/*****/

void differenza_num_compl(void)
{
    num_compl_alg_t minuendo_alg, /* forma algebrica */
        sottraendo_alg, /* forma algebrica */
        differenza_alg; /* differenza di due numeri
                        in forma algebrica */
    num_compl_tri_t minuendo_tri, /* forma trigonometrica */
        sottraendo_tri; /* forma trigonometrica */
    num_compl_alg_t operando_1, /* forma trigonometrica
                        convertito in forma
                        algebrica */
        operando_2, /* forma trigonometrica
                        convertito in forma
                        algebrica */
        differenza_tri; /* somma di due numeri compl.
                        in forma trigonometrica */
    int scelta = 0; /* scelta tra forma algebrica
                    (1) e trigonometrica(2) */

    /** inizializzo le strutture **/
    inizializza_strutt(minuendo_alg, minuendo_tri);
    inizializza_strutt(sottraendo_alg, sottraendo_tri);
    inizializza_compl_alg(operando_1, operando_2);
    inizializza_compl_alg(differenza_alg, differenza_tri);

    /** scelta della forma **/
    printf("Scegliere se si vuole effettuare la differenza tra
           numeri complessi in forma algebrica
           o trigonometrica.\n");
    scelta = convalida_scelta();

    if (scelta == 1) /* forma algebrica */
    {
        /** acquisizione della forma algebrica e relativa diff. **/
        printf("Minuendo...\n");
        minuendo_alg = acquisisci_compl_alg();
        printf("Sottraendo...\n");
        sottraendo_alg = acquisisci_compl_alg();

        /* stampo a video il minuendo */
        printf("\nIl minuendo inserito e' ");
        stampa_alg(minuendo_alg);
    }
}

```

```

/* stampo a video il sottraendo */
printf("Il sottraendo inserito e' ");
stampa_alg(sottraendo_alg);

/* calcolo la differenza in forma algebrica e la stampo */
differenza_alg.parte_reale = minuendo_alg.parte_reale
                          - sottraendo_alg.parte_reale;
differenza_alg.parte_imm = minuendo_alg.parte_imm
                          - sottraendo_alg.parte_imm;

printf("La differenza dei numeri complessi inseriti e' ");
stampa_alg(differenza_alg);
}
else /* forma trigonometrica */
{
    /** acquisizione della forma trigonometrica **/
    printf("Minuendo...\n");
    minuendo_tri = acquisisci_compl_tri();
    printf("Sottraendo...\n");
    sottraendo_tri = acquisisci_compl_tri();

    /* stampo a video il minuendo */
    printf("\nIl minuendo inserito e' ");
    stampa_tri(minuendo_tri);

    /* stampo a video il sottraendo */
    printf("Il sottraendo inserito e' ");
    stampa_tri(sottraendo_tri);

    /* calcolo la diff. in forma trigonometrica e stampo */
    /** conversione operandi in forma algebrica **/
    operando_1.parte_reale = minuendo_tri.modulo
                          * cos(minuendo_tri.argomento);
    operando_1.parte_imm = minuendo_tri.modulo
                          * sin(minuendo_tri.argomento);

    operando_2.parte_reale = sottraendo_tri.modulo
                          * cos(sottraendo_tri.argomento);
    operando_2.parte_imm = sottraendo_tri.modulo
                          * sin(sottraendo_tri.argomento);

    /** calcolo diff. in forma algebrica e stampa **/
    differenza_tri.parte_reale = operando_1.parte_reale
                              - operando_2.parte_reale;
    differenza_tri.parte_imm = operando_1.parte_imm
                              - operando_2.parte_imm;

    printf("La differenza dei numeri complessi inseriti e' ");
    stampa_alg(differenza_tri);
}

```



```

}
/*****/

void prodotto_num_compl(void)
{
    num_compl_alg_t moltiplicando_alg, /* forma algebrica */
                    moltiplicatore_alg, /* forma algebrica */
                    prodotto_alg;      /* forma algebrica */
    num_compl_tri_t moltiplicando_tri, /* forma trig. */
                    moltiplicatore_tri, /* forma trig. */
                    prodotto_tri;      /* forma trig. */
    int scelta = 0;                    /* scelta tra forma
                                        algebrica(1) e
                                        trigonometrica(2) */

    /** inizializzo le strutture **/
    inizializza_strutt(moltiplicando_alg, moltiplicando_tri);
    inizializza_strutt(moltiplicatore_alg, moltiplicatore_tri);
    inizializza_strutt(prodotto_alg, prodotto_tri);

    /** scelta della forma **/
    printf("Scegliere se si vuole effettuare il prodotto tra
           numeri complessi in forma algebrica o
           trigonometrica.\n");
    scelta = convalida_scelta();

    if (scelta == 1) /* forma algebrica */
    {
        /** acquisizione della forma algebrica e prodotto **/
        printf("Moltiplicando...\n");
        moltiplicando_alg = acquisisci_compl_alg();
        printf("Moltiplicatore...\n");
        moltiplicatore_alg = acquisisci_compl_alg();

        /* stampo a video il moltiplicando */
        printf("\nIl moltiplicando inserito e' ");
        stampa_alg(moltiplicando_alg);

        /* stampo a video il moltiplicatore */
        printf("Il moltiplicatore inserito e' ");
        stampa_alg(moltiplicatore_alg);

        /* calcolo prodotto in forma algebrica e stampo */
        prodotto_alg.parte_reale = (moltiplicando_alg.parte_reale
                                    * moltiplicatore_alg.parte_reale)
                                   - (moltiplicando_alg.parte_imm
                                    * moltiplicatore_alg.parte_imm);
        prodotto_alg.parte_imm = (moltiplicando_alg.parte_reale
                                   * moltiplicatore_alg.parte_imm)
                                  + (moltiplicando_alg.parte_imm

```

```

        * moltiplicatore_alg.parte_reale);

        printf("Il prodotto dei numeri complessi inseriti e' ");
        stampa_alg(prodotto_alg);
    }
    else /* forma trigonometrica */
    {
        /** acquisizione della forma trigonometrica **/
        printf("Moltiplicando...\n");
        moltiplicando_tri = acquisisci_compl_tri();
        printf("Moltiplicatore...\n");
        moltiplicatore_tri = acquisisci_compl_tri();

        /* stampo a video il moltiplicando */
        printf("\nIl moltiplicando inserito e' ");
        stampa_tri(moltiplicando_tri);

        /* stampo a video il moltiplicatore */
        printf("Il moltiplicatore inserito e' ");
        stampa_tri(moltiplicatore_tri);

        /* calcolo prodotto in forma trig. e stampo */
        prodotto_tri.modulo = moltiplicando_tri.modulo
                             * moltiplicatore_tri.modulo;
        prodotto_tri.argomento = moltiplicando_tri.argomento
                                + moltiplicatore_tri.argomento;

        printf("Il prodotto dei numeri complessi inseriti e' ");
        stampa_tri(prodotto_tri);
    }
}/** fine funzione prodotto_num_compl ***/

/*****

void rapporto_num_compl(void)
{
    num_compl_alg_t dividendo_alg,    /* forma algebrica */
                    divisore_alg,     /* forma algebrica */
                    rapporto_alg;     /* forma algebrica */
    num_compl_tri_t dividendo_tri,    /* forma trigonometrica */
                    divisore_tri,     /* forma trigonometrica */
                    rapporto_tri;     /* forma trigonometrica */
    int scelta = 0;                   /* scelta tra forma
                                     algebrica(1) e
                                     trigonometrica(2) */

    /** inizializzo le strutture **/
    inizializza_strutt(dividendo_alg, dividendo_tri);
    inizializza_strutt(divisore_alg, divisore_tri);
    inizializza_strutt(rapporto_alg, rapporto_tri);

```

```

/** scelta della forma */
printf("Scegliere se si vuole effettuare il rapporto tra
      numeri complessi in forma algebrica
      o trigonometrica.\n");
scelta = convalida_scelta();

if (scelta == 1)          /* forma algebrica */
{
    /** acquisizione della forma algebrica e rapporto */
    printf("Dividendo...\n");
    dividendo_alg = acquisisci_compl_alg();
    printf("Divisore...\n");
    divisore_alg = acquisisci_compl_alg();

    /* controllo che il denominatore sia diverso da zero */
    if (divisore_alg.parte_reale == 0 &&
        divisore_alg.parte_imm == 0) /* se denominatore = 0 */
        printf("Il divisore inserito e' uguale a zero,
        pertanto la divisione non e' possibile.\n");
    else /* altrimenti proseguo e calcolo il rapporto */
    {
        /* stampo a video il dividendo */
        printf("\nIl dividendo inserito e' ");
        stampa_alg(dividendo_alg);

        /* stampo a video il divisore */
        printf("Il divisore inserito e' ");
        stampa_alg(divisore_alg);

        /* calcolo rapporto in forma algebrica e stampo */
        rapporto_alg.parte_reale = ( (dividendo_alg.parte_reale
            * divisore_alg.parte_reale)
            + (dividendo_alg.parte_imm
            * divisore_alg.parte_imm) )
            / (pow(divisore_alg.parte_reale
                , 2)
            + pow(divisore_alg.parte_imm
                , 2));
        rapporto_alg.parte_imm = ( (dividendo_alg.parte_imm
            * divisore_alg.parte_reale)
            - (dividendo_alg.parte_reale
            * divisore_alg.parte_imm) )
            / ( pow(divisore_alg.parte_reale,
                2)
            + pow(divisore_alg.parte_imm,
                2) );

        printf("Il rapporto dei numeri complessi inseriti e' ");
        stampa_alg(rapporto_alg);
    }
}

```

```

    }
}
else /* forma trigonometrica */
{
    /** acquisizione della forma trigonometrica **/
    printf("Dividendo...\n");
    dividendo_tri = acquisisci_compl_tri();
    printf("Divisore...\n");
    divisore_tri = acquisisci_compl_tri();

    /* controllo che il denominatore sia diverso da zero */
    if (divisore_tri.modulo == 0) /* se divisore = 0 */
        printf("Il divisore inserito e' uguale a zero,
            pertanto la divisione non e' possibile.\n");
    else
    {
        /* stampo a video il dividendo */
        printf("\nIl dividendo inserito e' ");
        stampa_tri(dividendo_tri);

        /* stampo a video il divisore */
        printf("Il divisore inserito e' ");
        stampa_tri(divisore_tri);

        /* calcolo rapporto in forma trigonometrica e stampo */
        rapporto_tri.modulo = dividendo_tri.modulo
            / divisore_tri.modulo;
        rapporto_tri.argomento = dividendo_tri.argomento
            - divisore_tri.argomento;

        printf("Il rapporto dei numeri complessi inseriti e' ");
        stampa_tri(rapporto_tri);
    }
}
}/** fine funzione rapporto_num_compl ***/

```

```

/*****/

```

File lib_num_complessi.h

```

/* ridefinizione delle costanti simboliche */

#define PI_GRECO 3.14

/* ridefinizione dei tipi */

typedef struct num_compl_alg
{
    double parte_reale,
        parte_imm;
}

```

```

}num_compl_alg_t;

typedef struct num_compl_tri
{
    double  modulo,
           argomento;
}num_compl_tri_t;

/* ridichiarazione delle funzioni da esportare */

extern void acquisisci_e_stampa(void); /* acquisizione */
extern void somma_num_compl(void);    /* somma */
extern void differenza_num_compl(void); /* differenza */
extern void prodotto_num_compl(void);  /* prodotto */
extern void rapporto_num_compl(void);  /* divisione */

```

Makefile

```

progetto_2016: lib_num_complessi.o progetto_2016.o Makefile
    gcc -ansi -Wall -O lib_num_complessi.o progetto_2016.o
    -o progetto_2016 -lm

lib_num_complessi.o: lib_num_complessi.c
                        lib_num_complessi.h Makefile
    gcc -ansi -Wall -O -c lib_num_complessi.c

progetto_2016.o: progetto_2016.c lib_num_complessi.h Makefile
    gcc -ansi -Wall -O -c progetto_2016.c

pulisci:
    rm -f lib_num_complessi.o progetto_2016.o

pulisci_tutto:
    rm -f progetto_2016 lib_num_complessi.o progetto_2016.o

```

File progetto_2016.c

```

/* direttive al preprocessore */

#include <stdio.h>
#include "lib_num_complessi.h"

/* dichiarazione delle funzioni */
int acquisisci_scelta(void); /* acquisisce il valore scelta */

/* definizione delle funzioni */
int acquisisci_scelta(void)
{
    int esito_lettura;          /* lavoro: validazione stretta */
    char caratt_rimosso;        /* lavoro: validazione stretta */

```

```

int scelta = 0;                /* output: scelta dell'utente */

do                            /** VALIDAZIONE STRETTA **/
{
    esito_lettura = scanf("%d",
                          &scelta);
    if (esito_lettura != 1 || scelta < 1 || scelta > 5)
    {
        printf("Input non valido!\n");
        do
        {
            caratt_rimosso = getchar();
            while (caratt_rimosso != '\n');
        }
    }
    while (esito_lettura != 1 || scelta < 1
          || scelta > 5);      /** FINE VALIDAZIONE STRETTA **/

    return(scelta);
}

/** MAIN **/

int main(void)
{
    int scelta = 0;            /* scelta dell'utente */

    printf("\t*****\n");
    printf("\t** Programma-test per \"lib_num_complessi.h\" **\n");
    printf("\t*****\n");
    printf("\nDigitare uno dei seguenti numeri, ciascuno
           corrispondente alla funzione indicata:\n");
    printf("1 - per convertire un numero complesso dalla forma
           algebrica alla trigonometrica e viceversa;\n");
    printf("2 - per calcolare la somma tra due numeri
           complessi;\n");
    printf("3 - per calcolare la differenza tra due numeri
           complessi;\n");
    printf("4 - per calcolare il rapporto tra due numeri
           complessi;\n");
    printf("5 - per calcolare il quoziente tra due numeri
           complessi.\n...");

    /* acquisizione e convalida della scelta */
    scelta = acquisisci_scelta();

    switch (scelta)
    {
        case 1:
            acquisisci_e_stampa();
            break;
    }
}

```

```

        case 2:
            somma_num_compl();
            break;
        case 3:
            differenza_num_compl();
            break;
        case 4:
            prodotto_num_compl();
            break;
        case 5:
            rapporto_num_compl();
            break;
        default:
            printf("Si e' verificato un errore.\n");
            break;
    }

    printf("\tEsecuzione terminata!\n");

    return(0);
}

```

Testing del programma

- Conversione da forma algebrica a trigonometrica (prima funzione)

	Forma algebrica	Forma trigonometrica
1	$-8.00 + 0.00i$	$8.00[\cos(3.14) + i \sin(3.14)]$
2	$0.00 + 0.00i$	indefinito

- Conversione da forma trigonometrica ad algebrica (prima funzione)

	Forma trigonometrica	Forma algebrica
3	$2.00[\cos(1.57) + i \sin(1.57)]$	$0.00 + 2.00i$
4	$2.00[\cos(0) + i \sin(0)]$	$2.00 + 0.00i$

- Somma algebrica (seconda funzione)

Primo addendo	Secondo addendo	Somma
$0.00 + 5.00i$	$1.00 + 2.00i$	$1.00 + 7.00i$
$2.00 + 11.00i$	$-4.00 + 0.00i$	$-2.00 + 11.00i$

- Somma trigonometrica (seconda funzione)

Primo addendo	Secondo addendo
$4.00[\cos(3.14) + i \sin(3.14)]$	$3.00[\cos(4.71) + i \sin(4.71)]$
Somma	
$-4.01 + (-2.99)i$	

- Differenza algebrica (terza funzione)

Minuendo	Sottraendo	Differenza
$0.00 + 0.00i$	$-1.00 + 5.00i$	$1.00 + (-5.00)i$

- Differenza trigonometrica (terza funzione)

Minuendo	Sottraendo
$3.16[\cos(1.25) + i \sin(1.25)]$	$4.47[\cos(5.82) + i \sin(5.82)]$
Differenza	
$-3.00 + 5.00i$	

- Prodotto algebrico (quarta funzione)

Moltiplicando	Moltiplicatore	Prodotto
$1.00 + 1.00i$	$1.00 + 3.00i$	$-2.00 + 4.00i$

- Prodotto trigonometrico (quarta funzione)

Moltiplicando	Moltiplicatore
$8.00[\cos(3.14) + i \sin(3.14)]$	$2.00[\cos(1.57) + i \sin(1.57)]$
Prodotto	
$16.00[\cos(4.71) + i \sin(4.71)]$	

- Quoziente algebrico (quinta funzione)

Dividendo	Divisore	Quoziente
$1.00 + 1.00i$	$1.00 - 1.00i$	$0.00 + 1.00i$

- Quoziente trigonometrico (quinta funzione)

Dividendo	Divisore
$2.00[\cos(3.14) + i \sin(3.14)]$	0
Quoziente	
Impossibile calcolare (divisore = 0)	

Verifica del programma