

Studente: Lorenzo Calisti

Matricola: 271250

Corso di Programmazione Procedurale e Logica
Progetto per la sessione Invernale 2015/2016

Docente: Prof. Marco Bernando

Specifica del problema

SPECIFICA DEL PROBLEMA

Write an ANSI C library that handles complex numbers by exporting the following functions. The first C function gets from the keyboard a complex number in algebraic (respectively, polar) form and prints to the screen the equivalent complex number in polar (respectively, algebraic) form. The second C function gets from the keyboard two complex numbers in algebraic (respectively, polar) form and prints to the screen their sum in algebraic (respectively, polar) form. The third C function gets from the keyboard two complex numbers in algebraic (respectively, polar) form and prints to the screen their difference in algebraic (respectively, polar) form. The fourth C function gets from the keyboard two complex numbers in algebraic (respectively, polar) form and prints to the screen their product in algebraic (respectively, polar) form. The fifth C function gets from the keyboard two complex numbers in algebraic (respectively, polar) form and prints to the screen their quotient in algebraic (respectively, polar) form.

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

ANALISI DEL PROBLEMA

Definizione di un numero complesso in forma algebrica: $z = a + ib$

Definizione di un numero complesso in forma trigonometrica: $z = \varphi[\cos(\theta) + i \sin(\theta)]$

1. Cambio di forma di un numero complesso

Input: Un numero complesso in forma algebrica (rispettivamente, trigonometrica).

Output: Il numero complesso corrispondente in forma trigonometrica (rispettivamente, algebrica).

Relazioni Input/Output:

Da forma algebrica a forma trigonometrica

$$\text{Modulo: } \varphi = \sqrt{a^2 + b^2}$$

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

Da forma trigonometrica a forma algebrica

$$a = \varphi \cos(\theta)$$

$$b = \varphi \sin(\theta)$$

2. Somma di due numeri complessi

Input: Due numeri complessi in forma algebrica (rispettivamente, trigonometrica)

Output: Il numero complesso risultante dalla somma dei due input

Relazioni Input/Output:

Somma di due numeri in forma algebrica

$$z = a + ib \qquad w = c + id$$

$$z + w = (a + c) + i(b + d)$$

Somma di due numeri in forma trigonometrica

Occorre riscrivere i due numeri in forma estesa moltiplicando il modulo per i due addendi tra parentesi

$$z = \varphi_1 \cos(\theta_1) + i\varphi_1 \sin(\theta_1)$$

$$w = \varphi_2 \cos(\theta_2) + i\varphi_2 \sin(\theta_2)$$

e poi sommare le parti reali e le parti immaginarie in modo del tutto analogo alla forma algebrica.

3. Differenza di due numeri complessi

Input: Due numeri complessi in forma algebrica (rispettivamente, trigonometrica)

Output: Il numero complesso risultante dalla differenza

Relazioni Input/Output:

Differenza di due numeri in forma algebrica

$$z = a + ib \qquad w = c + id$$

$$z - w = (a - c) + i(b - d)$$

Differenza di due numeri in forma trigonometrica

Occorre riscrivere i due numeri in forma estesa moltiplicando il modulo per i due addendi tra parentesi

$$z = \varphi_1 \cos(\theta_1) + i\varphi_1 \sin(\theta_1)$$

$$w = \varphi_2 \cos(\theta_2) + i\varphi_2 \sin(\theta_2)$$

e poi sottrarre le parti reali e le parti immaginarie in modo del tutto analogo alla forma algebrica.

4. Prodotto di due numeri complessi

Input: Due numeri complessi in forma algebrica (rispettivamente, trigonometrica)

Output: Il numero complesso risultante dal prodotto

Relazioni Input/Output:*Prodotto di due numeri in forma algebrica*

$$z = a + ib \qquad w = c + id$$

$$z \cdot w = (ac - bd) + i(ad + bc)$$

Prodotto di due numeri in forma trigonometrica

$$z = \varphi_1[\cos(\theta_1) + i \sin(\theta_1)] \qquad w = \varphi_2[\cos(\theta_2) + i \sin(\theta_2)]$$

$$z \cdot w = \varphi_1 \varphi_2 [\cos(\theta_1 + \theta_2) + i \sin(\theta_1 + \theta_2)]$$

5. Rapporto di due numeri complessi

Input: Due numeri complessi in forma algebrica (rispettivamente, trigonometrica)**Output:** Il numero complesso risultante dal rapporto**Relazioni Input/Output:***Rapporto di due numeri in forma algebrica*

$$z = a + ib \qquad w = c + id$$

$$\frac{z}{w} = \frac{ac + bd}{c^2 + d^2} + i \frac{bc - ad}{c^2 + d^2}$$

Rapporto di due numeri in forma trigonometrica

$$z = \varphi_1[\cos(\theta_1) + i \sin(\theta_1)] \qquad w = \varphi_2[\cos(\theta_2) + i \sin(\theta_2)]$$

$$\frac{z}{w} = \frac{\varphi_1}{\varphi_2} [\cos(\theta_1 - \theta_2) + i \sin(\theta_1 - \theta_2)]$$

Progettazione dell'algoritmo

PROGETTAZIONE DELL'ALGORITMO

La libreria, denominata “libreria_complesi” esporta cinque funzioni e utilizza altre quindici funzioni non esportate che hanno il compito di acquisire gli input, restituire l'output e “alleggerire” il codice evitando ridondanze.

Gli input e gli output sono gestiti utilizzando un particolare tipo di dato formato da più strutture dati concatenate.

Il tipo di dato nominato “n_compl_t” è composto da una struttura contenente il valore della forma ed un'altra struttura la quale ha al suo interno i valori di parte reale e parte immaginaria per la forma algebrica e i valori di modulo e argomento per la forma trigonometrica.

La funzione per acquisire i valori non viene esportata, ma è presente in tutte quelle esportate poiché, come dice il nome, si occupa di acquisire il valore di un solo numero complesso da tastiera (nei casi in cui i numeri da ottenere sono due la funzione viene invocata due volte); per fare ciò chiede preventivamente all'utente la forma in cui si trova il numero e secondo la risposta domanda il valore della parte reale e immaginaria oppure del modulo e dell'argomento.

Importante: il valore della parte reale, immaginaria e del modulo sono dei numeri reali approssimati alla seconda cifra decimale, mentre il valore dell'argomento è espresso in gradi sessagesimali (che saranno poi convertiti automaticamente dal programma in radianti nell'intervallo $(-\pi/2, +\pi/2]$ per poter eseguire i calcoli necessari).

La funzione per stampare a video il numero risultante (anch'essa non esportata) richiede un numero complesso in input all'invocazione (che sarà poi il numero complesso stampato a video) dopo di che controlla la forma del numero e stampa sullo schermo il numero nelle seguenti forme:

$$\begin{array}{ll} a + ib & \text{Se il numero è in forma algebrica} \\ \varphi [\cos(\theta) + i\sin(\theta)] & \text{Se il numero è in forma trigonometrica} \end{array}$$

Dove a, b, φ sono numeri reali (\mathbb{R}), approssimati alla seconda cifra decimale, e θ è un angolo espresso in gradi sessagesimali.

La prima funzione esportata si occupa di cambiare la forma di un numero complesso preso in input. Prima di tutto acquisisce il numero e ne controlla la forma a seconda della quale esegue gli opportuni calcoli descritti nella sezione ‘analisi del problema’ alla voce ‘cambio di forma di un numero complesso’; una volta convertito il numero lo restituisce a schermo utilizzando la funzione di stampa sopra descritta.

La seconda funzione esportata calcola la somma di due numeri complessi prendendo in input i due numeri grazie alla medesima funzione di acquisizione per poi controllare inizialmente la forma del primo numero e in seguito quella del secondo. Si precisa che questa funzione restituisce il risultato nella stessa forma del primo numero acquisito, dunque il secondo numero sarà cambiato di forma nel caso essa sia discordante.

Una volta che entrambi i numeri sono nella stessa forma si calcola la loro somma grazie alle formule di somma tra numeri complessi precedentemente descritte.

Analogamente alla seconda funzione la terza esportata esegue la differenza fra due numeri complessi. I meccanismi di calcolo sono identici anche se le formule utilizzate sono quelle per calcolare la differenza fra due numeri complessi. Si precisa nuovamente che questa funzione restituisce il risultato nella stessa forma del primo numero acquisito.

La quarta funzione esportata calcola il prodotto tra due numeri complessi presi in input dalla funzione

apposita e utilizzando gli stessi meccanismi di controllo della forma calcola il numero risultante applicando le proprietà del prodotto tra numeri complessi; anche in questo caso il risultato sarà espresso nella forma del primo numero acquisito.

La quinta ed ultima funzione esportata calcola il prodotto fra due numeri complessi presi in input; anch'essa si basa sulle stesse metodologie di controllo della forma, applicando le regole di calcolo del prodotto tra complessi per ottenere il risultato e stamparlo a video utilizzando la funzione di stampa, dunque anche questa funzione restituisce il risultato nella stessa forma del primo numero acquisito.

Le restanti tredici funzioni non esportate sono utilizzate all'interno delle altre funzioni per "semplificare" e "alleggerire" le istruzioni e i calcoli evitando numerose ripetizioni di codice.

Implementazione dell'algoritmo

IMPLEMENTAZIONE DELL'ALGORITMO

Contenuto del file **libreria_complessi.h**

```
/* **** */
/* Dichiarazione delle funzioni da esportare */
/* **** */

extern void cambia_forma(void); /* Cambia la forma in cui è visualizzato un numero complesso */

extern void somma_n_compl(void); /* Calcola la somma di due numeri complessi */

extern void differenza_n_compl(void); /* Calcola la differenza di due numeri complessi */

extern void prodotto_n_compl(void); /* Calcola il prodotto di due numeri complessi */

extern void rapporto_n_compl(void); /* Calcola rapporto di due numeri complessi */
```

Contenuto del file **makefile**

```
numeri_complessi: programma.o libreria_complessi.o makefile
    gcc -ansi -Wall -O programma.o libreria_complessi.o -o numeri_complessi -lm
programma.o: programma.c makefile
    gcc -ansi -Wall -O -c programma.c
libreria_complessi.o: libreria_complessi.c makefile
    gcc -ansi -Wall -O -c libreria_complessi.c
clear: programma.o libreria_complessi.o -lm
    rm -f programma.o libreria_complessi.o
clear_all: programma.o libreria_complessi.o numeri_complessi
    rm -f programma.o libreria_complessi.o numeri_complessi
```

Contenuto del file **libreria_complessi.c**

```
/* **** */
/* Libreria creata da Lorenzo Calisti */
/* **** */
/* **** */
/* Inclusione delle librerie standard "stdio.h" e "math.h" */
/* **** */
#include <stdio.h>
#include <math.h>

/* **** */
/* Definizione delle costanti simboliche */
/* **** */
#define PI_GRECO 3.141592 /* pigreco */
```

```

#define RAD_GRAD (180 / PI_GRECO) /*fatt. conversione rad/gradi*/
#define GRAD_RAD (PI_GRECO / 180) /*fatt. conversione gradi/rad*/

/*****/
/* Definizione dei tipi */
/*****/

/* Forma del numero complesso*/
typedef enum {forma_algebrica,
              forma_trigonometrica} forma_t;

/* Tipo forma algebrica */
typedef struct
{
    double parte_reale,
           parte_immag;
} f_algebrica_t;

/* Tipo forma trigonometrica */
typedef struct
{
    double modulo,
           argomento;
} f_trigonometrica_t;

/* Tipo numero complesso */
typedef struct
{
    forma_t forma;
    struct
    {
        f_algebrica_t f_algebrica;
        f_trigonometrica_t f_trigonometrica;
    } dati_forma;
} n_compl_t;
/*****/
/* Dichiarazione delle funzioni */
/*****/
/*-Funzioni da esportare-----*/

void cambia_forma(void);    /* Cambia la forma in cui è visualizzato un numero complesso*/

void somma_n_compl(void);   /* Calcola la somma di due numeri complessi */

void differenza_n_compl(void); /* Calcola la differenza di due numeri complessi */

void prodotto_n_compl(void); /* Calcola il prodotto di due numeri complessi */

void rapporto_n_compl(void); /* Calcola rapporto di due numeri complessi */

/*-Funzioni interne-----*/

```

```

void leggi_complesso(n_compl_t *); /* Acquisisce il numero complesso*/

int prendi_forma(void);           /*-----*/

n_compl_t prendi_valore_alg(void); /*-----*/

n_compl_t prendi_valore_trig(void); /*-----*/

void stampa_complesso(n_compl_t); /* Stampa il risultato delle operazioni*/

void stampa_alg(n_compl_t);       /*-----*/

void stampa_trig(n_compl_t);      /*-----*/

void converti_alg_trig(n_compl_t *); /* Converte da f.alg a f.trig*/

void converti_trig_alg(n_compl_t *); /* Converte da f.trig a f.alg*/

n_compl_t somma_f_alg(n_compl_t n1, /* Somma in f.alg*/
                      n_compl_t n2);

n_compl_t differenza_f_alg(n_compl_t n1, /* Differenza in f.alg*/
                           n_compl_t n2);

n_compl_t prodotto_f_alg(n_compl_t n1, /* Prodotto in f.alg*/
                         n_compl_t n2);

n_compl_t prodotto_f_trig(n_compl_t n1, /* Prodotto in f.trig*/
                          n_compl_t n2);

n_compl_t rapporto_f_alg(n_compl_t n1, /* Rapporto in f.alg*/
                        n_compl_t n2);

n_compl_t rapporto_f_trig(n_compl_t n1, /* Prodotto in f.trig*/
                         n_compl_t n2);

/*****/
/* Definizione delle funzioni */
/*****/
/*-Funzioni da esportare-----*/

/*****/
/* Funzione per cambiare la forma di un numero complesso */
/*****/
void cambia_forma(void)
{
    n_compl_t numero; /*var. di lavoro: numero complesso da convertire*/

    /* acquisizione del numero */
    leggi_complesso(&numero);

    /*controllo forma del numero*/

```

```

switch (numero.forma)
{
    case 0: /*se f.algebrica*/

        converti_alg_trig(&numero);
        break;

    case 1: /*se f.trigonometrica*/

        converti_trig_alg(&numero);
        break;
}

/*stampa il numero cambiato di forma*/
stampa_complesso(numero);
}
/*****
/* Funzione per sommare due numeri complessi */
*****/
void somma_n_compl(void)
{
    n_compl_t numero1, /*input: primo numero*/
                numero2; /*input: secondo numero*/
    n_compl_t risultato; /*output: il risultato della somma*/

    leggi_complesso(&numero1);      /*acquisizione del primo numero*/

    leggi_complesso(&numero2);      /*acquisizione del secondo numero*/

    /*controllo forma primo numero*/
    switch (numero1.forma)
    {
        case 0: /*se il num.1 è in f.algebrica*/

            /*controllo forma secondo numero*/
            switch (numero2.forma)
            {
                case 0: /*se il num.2 è in f.algebrica*/

                    /*invoca la funzione per sommare*/
                    risultato = somma_f_alg(numero1,
                                            numero2);

                    break;

                case 1: /*se il num.2 è in f.trigonometrica*/

                    /*cambio di forma trig-alg*/
                    converti_trig_alg(&numero2);

                    /*invoca la funzione per sommare*/
                    risultato = somma_f_alg(numero1,
                                            numero2);

                    break;
            }
        }
    }
}

```

```

    }

    /*assegna al ris. la forma alg.*/
    risultato.forma = forma_algebrica;

    break;

case 1:      /*se il num. 1 è in f.trigonometrica*/

    /*assegna momentaneamente al ris. la forma alg.*/
    risultato.forma = forma_algebrica;

    /*convertito in forma algebrica il primo numero*/
    converti_trig_alg(&numero1);

    /*controllo forma secondo numero*/
    switch (numero2.forma)
    {
        case 0: /*se il num.2 è in f.algebrica*/

            /*invoca la funzione per sommare*/
            risultato = somma_f_alg(numero1,
                                    numero2);

            break;

        case 1: /*se il num.2 è in f.trigonometrica*/

            /*cambio di forma trig-alg*/
            converti_trig_alg(&numero2);

            /*invoca la funzione per sommare*/
            risultato = somma_f_alg(numero1,
                                    numero2);

            break;

    }

    /*convertito il risultato in f.trigonometrica*/
    converti_alg_trig(&risultato);
    break;
}

/*stampa il risultato della somma*/
stampa_complesso(risultato);
}

/*****/
/* Funzione per sottrarre due numeri complessi */
/*****/
void differenza_n_compl(void)
{
    n_compl_t numero1, /*input: primo numero*/
               numero2; /*input: secondo numero*/

```



```

n_compl_t risultato; /*output: il risultato della differenza*/

leggi_complesso(&numero1); /*acquisizione del primo numero*/

leggi_complesso(&numero2); /*acquisizione del secondo numero*/

/*controllo forma primo numero*/
switch (numero1.forma)
{
    case 0: /*se il num.1 è in f.algebrica*/

        /*controllo forma secondo numero*/
        switch (numero2.forma)
        {
            case 0: /*se il num.2 è in f.algebrica*/
                /*invoca funz. per sottrarre*/
                risultato = differenza_f_alg(numero1,
                                            numero2);

                break;

            case 1: /*se il num.2 è in f.trigonometrica*/

                /*cambio di forma trig-alg*/
                converti_trig_alg(&numero2);

                /*invoco funz. per sottrarre*/
                risultato = differenza_f_alg(numero1,
                                            numero2);

                break;

        }

        /*assegna al ris. la forma alg.*/
        risultato.forma = forma_algebrica;

        break;

    case 1: /*se il num. 1 è in f.trigonometrica*/

        /*assegna momentaneamente al ris. la forma alg.*/
        risultato.forma = forma_algebrica;

        /*convertito in forma algebrica il primo numero*/
        converti_trig_alg(&numero1);

        /*controllo forma secondo numero*/
        switch (numero2.forma)
        {
            case 0: /*se il num.2 è in f.algebrica*/

                /*invoca la funz. per sottrarre*/
                risultato = differenza_f_alg(numero1,
                                            numero2);

                break;

```

```

        case 1: /*se il num.2 è in f.trigonometrica*/

            converti_trig_alg(&numero2);/*cambio di forma trig-alg*/

            /*invoco la funz. per sottrarre*/
            risultato = differenza_f_alg(numero1,
                                         numero2);

            break;

        }

        /*convertito il risultato in f.trigonometrica*/
        converti_alg_trig(&risultato);
        break;
    }

    /*stampa il risultato della differenza*/
    stampa_complesso(risultato);
}

/*****
/* Funzione per moltiplicare due numeri complessi */
*****/
void prodotto_n_compl(void)
{
    n_compl_t numero1, /*input: primo numero*/
               numero2; /*input: secondo numero*/
    n_compl_t risultato; /*output: il risultato del prodotto*/

    leggi_complesso(&numero1);    /*acquisizione del primo numero*/

    leggi_complesso(&numero2);    /*acquisizione del secondo numero*/

    /*controllo forma primo numero*/
    switch (numero1.forma)
    {
        case 0: /*se il num.1 è in f.algebrica*/

            /*controllo forma secondo numero*/
            switch (numero2.forma)
            {
                case 0: /*se il num.2 è in f.algebrica*/

                    /*invoco la funzione per moltiplicare*/
                    risultato = prodotto_f_alg(numero1,
                                                numero2);

                    break;

                case 1: /*se il num.2 è in f.trigonometrica*/

                    /*cambio di forma trig-alg*/
                    converti_trig_alg(&numero2);

```

```

        /*invoco la funzione per moltiplicare*/
        risultato = prodotto_f_alg(numero1,
                                   numero2);
        break;
    }

    /*assegna al ris. la forma alg.*/
    risultato.forma = forma_algebrica;

    break;

case 1:      /*se il num. 1 è in f.trigonometrica*/

    /*contollo forma secondo numero*/
    switch (numero2.forma)
    {
        case 0: /*se il num.2 è in f.algebrica*/

            /*cambio di forma alg-trig*/
            converti_alg_trig(&numero2);

            /*invoco la funz. per moltiplicare*/
            risultato = prodotto_f_trig(numero1,
                                       numero2);
            break;

            case 1: /*se il num.2 è in f.trigonometrica*/

                /*invoco la funz. per moltiplicare*/
                risultato = prodotto_f_trig(numero1,
                                           numero2);
                break;
        }

        /*assegna al ris. la forma trig.*/
        risultato.forma = forma_trigonometrica;

        break;
    }

    /*stampa il risultato del prodotto*/
    stampa_complesso(risultato);
}

/*****
/* Funzione per dividere due numeri complessi */
*****/

void rapporto_n_compl(void)
{
    n_compl_t numero1, /*input: primo numero*/
               numero2; /*input: secondo numero*/
    n_compl_t risultato; /*output: il risultato del rapporto*/

```

```

leggi_complesso(&numero1);      /*acquisizione del primo numero*/

leggi_complesso(&numero2);      /*acquisizione del secondo numero*/

/*controllo forma primo numero*/
switch (numero1.forma)
{
    case 0:      /*se il num.1 è in f.algebrica*/

        /*controllo forma secondo numero*/
        switch (numero2.forma)
        {
            case 0: /*se il num.2 è in f.algebrica*/

                /*invoco la funz. per dividere*/
                risultato = rapporto_f_alg(numero1,
                                           numero2);

                break;

            case 1: /*se il num.2 è in f.trigonometrica*/

                /*cambio di forma trig-alg*/
                converti_trig_alg(&numero2);

                /*invoco la funz. per dividere*/
                risultato = rapporto_f_alg(numero1,
                                           numero2);

                break;

        }

        /*assegna al ris. la forma alg.*/
        risultato.forma = forma_algebrica;

        break;

    case 1:      /*se il num. 1 è in f.trigonometrica*/

        /*controllo forma secondo numero*/
        switch (numero2.forma)
        {
            case 0: /*se il num.2 è in f.algebrica*/

                /*cambio di forma alg-trig*/
                converti_alg_trig(&numero2);

                /*invoco la funz. per dividere*/
                risultato = rapporto_f_trig(numero1,
                                           numero2);

                break;

            case 1: /*se il num.2 è in f.trigonometrica*/

                /*invoco la funz. per dividere*/

```

```

        risultato = rapporto_f_trig(numero1,
                                   numero2);
        break;
    }

    /*assegna al ris. la forma trig.*/
    risultato.forma = forma_trigonometrica;

    break;
}

/*stampa il risultato della somma*/
stampa_complesso(risultato);
}

/*-Funzioni interne-----*/

/*-----*/
/* Funzione per acquisire un numero complesso */
/*-----*/
void leggi_complesso(n_compl_t *num_comp)
{
    int scelta_forma; /*var. di lavoro: scelta della forma*/

    /*acquisizione della forma del numero */
    scelta_forma = prendi_forma();

    /*acquisizione dei valori del numero*/
    switch (scelta_forma)
    {
        case 1: /*se f.algebrica*/

            *num_comp = prendi_valore_alg();
            break;

        case 2: /*se f.trigonometrica*/

            *num_comp = prendi_valore_trig();
            break;
    }
}

/*-----*/
/* Funzione per scegliere la forma del numero compl. */
/*-----*/
int prendi_forma(void)
{
    int risposta, /*var. di lavoro: seleziona la forma del numero*/
        esito_lettura; /*var. di lavoro: esito della scanf*/
    char carattere; /*var. di lavoro: carattere nel buffer della tastiera*/

    /*Validazione stretta della scelta della forma*/

```

```

do
{
    printf("Selezionare la forma del ");
    printf("numero complesso (1 o 2):\n\n");
    printf("\t1. Forma Algebrica\n\n");
    printf("\t2. Forma Trigonometrica\n");

    esito_lettura = scanf("%d",
                        &risposta);

    if ((esito_lettura != 1) ||
        ((risposta != 1) &&
         (risposta != 2)))

    {
        printf("\nErrore! Input non accettabile! ");
        printf("[assicurarsi di avere inserito ");
        printf("1 oppure 2]\n");

        do
            carattere = getchar();

        while (carattere != '\n');

    }
}
while ((esito_lettura != 1) ||
        ((risposta != 1) &&
         (risposta != 2)));

return(risposta);
}

/*-----*/
/* Funzione per acquisire il valore in f.alg. del numero */
/*-----*/
n_compl_t prendi_valore_alg(void)
{
    n_compl_t num;    /*output: numero acquisito*/
    int esito_lettura; /*var. di lavoro: esito delle scanf*/
    char carattere;    /*var. di lavoro: carattere nel buffer*/

    /*assegnazione della forma alg. al tipo n_compl_t*/
    num.forma = forma_algebrica;

    /*Validazione stretta dell'input*/

    /*parte reale*/
    do
    {
        printf("Inserire il valore della parte reale: ");
        esito_lettura = scanf("%lf",
                            &num.dati_forma.f_algebrica.parte_reale);
    }
}

```

```

        if (esito_lettura != 1)
        {
            printf("\nAttenzione! Assicurarsi di aver inserito ");
            printf("un numero reale e non un carattere\n");
            do
                carattere = getchar();
            while (carattere != '\n');
        }
    }
while (esito_lettura != 1);

/*parte immaginaria*/
do
{
    printf("Inserire il valore della parte immaginaria: ");
    esito_lettura = scanf("%lf",
                        &num.dati_forma.f_algebrica.parte_imag);
    if (esito_lettura != 1)
    {
        printf("\nAttenzione! Assicurarsi di aver inserito ");
        printf("un numero reale e non un carattere\n");
        do
            carattere = getchar();
        while (carattere != '\n');
    }
}
while (esito_lettura != 1);

return(num);
}

/*-----*/
/* Funzione per acquisire il valore in f.trig. del numero */
/*-----*/
n_compl_t prendi_valore_trig(void)
{
    n_compl_t num;    /*output: numero acquisito*/
    int esito_lettura; /*var. di lavoro: esito delle scanf*/
    char carattere;    /*var. di lavoro: carattere nel buffer*/

    /*Assegnazione della forma trig. al tipo n_compl_t*/
    num.forma = forma_trigonometrica;

    /*Validazione stretta dell'input*/

    /*modulo*/
    do
    {
        printf("Inserire il valore del modulo: ");
        esito_lettura = scanf("%lf",
                            &num.dati_forma.f_trigonometrica.modulo);
        if (esito_lettura != 1)

```

```

        {
            printf("\nAttenzione! Assicurarsi di aver inserito ");
            printf("un numero reale e non un carattere\n");
            do
                carattere = getchar();
            while (carattere != '\n');
        }
    }
while (esito_lettura != 1);

/*argomento*/
do
{
    printf("Inserire il valore dell'argomento (in gradi): ");
    esito_lettura = scanf("%lf",
                        &num.dati_forma.f_trigonometrica.argumento);
    if (esito_lettura != 1)
    {
        printf("\nAttenzione! Assicurarsi di aver inserito ");
        printf("un numero reale e non un carattere\n");
        do
            carattere = getchar();
        while (carattere != '\n');
    }
}
while (esito_lettura != 1);

/*conversione dell'argomento da gradi a radianti*/
num.dati_forma.f_trigonometrica.argumento =
    (num.dati_forma.f_trigonometrica.argumento *
     GRAD_RAD);

return(num);
}

/*-----*/
/* Funzione per stampare il numero complesso risultante */
/*-----*/
void stampa_complesso(n_compl_t n)
{
    /*Controlla la forma del numero*/
    switch (n.forma)
    {
        case 0: /*se n è in f.algrbrica*/

            stampa_alg(n);
            break;

        case 1: /*se n è in f.trigonometrica*/

            stampa_trig(n);
            break;
    }
}

```



```

    }
}

/*-----*/
/* Stampa il numero in f.algebrica */
/*-----*/
void stampa_alg(n_compl_t num)
{
    char sgn; /*var. di lavoro: segno della p.immag*/

    /*se solo la parte reale è = 0*/
    if (num.dati_forma.f_algebrica.parte_reale == 0)
    {

        /*stampa il num in forma ib*/
        printf("Il numero complesso risultante ");
        printf("in forma algebrica è: ");
        printf("%.2fi\n",
            num.dati_forma.f_algebrica.parte_immag);
    }

    /*se solo la parte immaginaria è = 0*/
    else if (num.dati_forma.f_algebrica.parte_immag == 0)
    {

        /*stampa il num in forma a*/
        printf("Il numero complesso risultante ");
        printf("in forma algebrica è: ");
        printf("%.2f\n",
            num.dati_forma.f_algebrica.parte_reale);
    }

    /*se sia la p.reale che la p.immag sono != 0*/
    else
    {

        /*mette un segno a seconda del segno della p.immag*/
        if (num.dati_forma.f_algebrica.parte_immag >= 0)
            sgn = '+';
        else
            sgn = '-';

        /*stampa il num in forma a+ib*/
        printf("Il numero complesso risultante ");
        printf("in forma algebrica è: ");
        printf("%.2f %c %.2fi\n",
            num.dati_forma.f_algebrica.parte_reale,
            sgn,
            fabs(num.dati_forma.f_algebrica.parte_immag));
    }
}
}

```

```

/*-----*/
/* Stampa il numero in f.trigonometrica */
/*-----*/
void stampa_trig(n_compl_t num)
{
    printf("Il numero risultante ");
    printf("in forma trigonometrica è: ");
    printf("%.2f(cos%.2f° + i * sin%.2f°)\n",
        num.dati_forma.f_trigonometrica.modulo,
        (num.dati_forma.f_trigonometrica.argomento *
         RAD_GRAD),
        (num.dati_forma.f_trigonometrica.argomento *
         RAD_GRAD));
}

/*-----*/
/* Funzione per convertire da f. algebrica a trigonometrica */
/*-----*/
void converti_alg_trig(n_compl_t *num)
{
    /* assegnazione della forma trigonometrica */

    num->forma = forma_trigonometrica;

    /* calcolo del modulo */

    num->dati_forma.f_trigonometrica.modulo =
        sqrt (pow (num->dati_forma.f_algebrica.parte_reale, 2) +
              pow (num->dati_forma.f_algebrica.parte_immag, 2));

    /* calcolo dell'argomento */

    if (num->dati_forma.f_algebrica.parte_reale == 0) /*se a = 0 */
    {
        if (num->dati_forma.f_algebrica.parte_immag == 0) /* e b = 0 */

            num->dati_forma.f_trigonometrica.argomento = 0;

        else if (num->dati_forma.f_algebrica.parte_immag > 0) /* e b < 0 */

            num->dati_forma.f_trigonometrica.argomento = 1.5708;

        else /* e b > 0 */

            num->dati_forma.f_trigonometrica.argomento = -1.5708;

    }

    else if (num->dati_forma.f_algebrica.parte_reale > 0) /* se a > 0 e b qualsiasi */
    {
        num->dati_forma.f_trigonometrica.argomento =
            atan2 (num->dati_forma.f_algebrica.parte_immag,

```

```

num->dati_forma.f_algebrica.parte_reale);
}

else /* se a < 0 */
{
    if (num->dati_forma.f_algebrica.parte_imag >= 0) /* e b >= 0 */

        num->dati_forma.f_trigonometrica.argomento =
            atan2 (num->dati_forma.f_algebrica.parte_imag,
                num->dati_forma.f_algebrica.parte_reale) +
PI_GRECO;

    else /* e b < 0 */
        num->dati_forma.f_trigonometrica.argomento =
            atan2 (num->dati_forma.f_algebrica.parte_imag,
                num->dati_forma.f_algebrica.parte_reale) -
PI_GRECO;
}
}

/*-----*/
/* Funzione per convertire da f.trigonometrica a algebrica */
/*-----*/
void converti_trig_alg(n_compl_t *num)
{
    /* assegnazione della forma algebrica */

    num->forma = forma_algebrica;

    /*calcolo della parte reale*/

    num->dati_forma.f_algebrica.parte_reale =
        (cos(num->dati_forma.f_trigonometrica.argomento) *
            (num->dati_forma.f_trigonometrica.modulo));

    /*calcolo della parte immaginaria*/

    num->dati_forma.f_algebrica.parte_imag =
        (sin(num->dati_forma.f_trigonometrica.argomento) *
            (num->dati_forma.f_trigonometrica.modulo));
}

/*-----*/
/* Funzione per fare la somma in f.algebrica */
/*-----*/
n_compl_t somma_f_alg(n_compl_t n1,
                    n_compl_t n2 )
{
    n_compl_t somma;

    /*calcolo p.reale*/
    somma.dati_forma.f_algebrica.parte_reale =

```

```

        n1.dati_forma.f_algebrica.parte_reale +
        n2.dati_forma.f_algebrica.parte_reale;

    /*calcolo p.immaginaria*/
    somma.dati_forma.f_algebrica.parte_immag =
        n1.dati_forma.f_algebrica.parte_immag +
        n2.dati_forma.f_algebrica.parte_immag;

    return(somma);
}

/*-----*/
/* Funzione per fare la differenza in f.algebrica */
/*-----*/
n_compl_t differenza_f_alg(n_compl_t n1,
                           n_compl_t n2 )
{
    n_compl_t differenza; /*output: sottrazione */

    /*calcolo p.reale*/
    differenza.dati_forma.f_algebrica.parte_reale =
        n1.dati_forma.f_algebrica.parte_reale -
        n2.dati_forma.f_algebrica.parte_reale;

    /*calcolo p.immaginaria*/
    differenza.dati_forma.f_algebrica.parte_immag =
        n1.dati_forma.f_algebrica.parte_immag -
        n2.dati_forma.f_algebrica.parte_immag;

    return(differenza);
}

/*-----*/
/* Funzione per fare il prodotto in f.algebrica */
/*-----*/
n_compl_t prodotto_f_alg(n_compl_t n1,
                          n_compl_t n2 )
{
    n_compl_t prodotto;

    /*calcolo p.reale*/
    prodotto.dati_forma.f_algebrica.parte_reale =
        ((n1.dati_forma.f_algebrica.parte_reale *
          n2.dati_forma.f_algebrica.parte_reale) -
         (n1.dati_forma.f_algebrica.parte_immag *
          n2.dati_forma.f_algebrica.parte_immag));

    /*calcolo p.immaginaria*/
    prodotto.dati_forma.f_algebrica.parte_immag =
        ((n1.dati_forma.f_algebrica.parte_reale *
          n2.dati_forma.f_algebrica.parte_immag) +
         (n2.dati_forma.f_algebrica.parte_reale *

```

```

        n1.dati_forma.f_algebrica.parte_immag));

    return(prodotto);
}

/*-----*/
/* Funzione per fare il prodotto in f.trigonometrica */
/*-----*/
n_compl_t prodotto_f_trig(n_compl_t n1,
                          n_compl_t n2)
{
    n_compl_t prodotto_trig;

    /*calcolo modulo*/
    prodotto_trig.dati_forma.f_trigonometrica.modulo =
        n1.dati_forma.f_trigonometrica.modulo *
        n2.dati_forma.f_trigonometrica.modulo;

    /*calcolo argomeno*/
    prodotto_trig.dati_forma.f_trigonometrica.argomento =

        n1.dati_forma.f_trigonometrica.argomento +
        n2.dati_forma.f_trigonometrica.argomento;

    return(prodotto_trig);
}

/*-----*/
/* Funzione per fare il rapporto in f.algebrica */
/*-----*/
n_compl_t rapporto_f_alg(n_compl_t n1,
                        n_compl_t n2)
{
    n_compl_t rapporto;

    /*calcolo p.reale*/
    rapporto.dati_forma.f_algebrica.parte_reale =
        (((n1.dati_forma.f_algebrica.parte_reale *
            n2.dati_forma.f_algebrica.parte_reale) -
            (n1.dati_forma.f_algebrica.parte_immag *
            n2.dati_forma.f_algebrica.parte_immag)) /
            (pow (n2.dati_forma.f_algebrica.parte_reale, 2))+
            (pow (n2.dati_forma.f_algebrica.parte_immag,2))));

    /*calcolo p.immaginaria*/
    rapporto.dati_forma.f_algebrica.parte_immag =
        (((n1.dati_forma.f_algebrica.parte_reale *
            n2.dati_forma.f_algebrica.parte_immag) +
            (n2.dati_forma.f_algebrica.parte_reale *
            n1.dati_forma.f_algebrica.parte_immag)) /
            (pow (n2.dati_forma.f_algebrica.parte_reale, 2))+
            (pow (n2.dati_forma.f_algebrica.parte_immag,2))));

```

```

        return(rapporto);
    }

/*-----*/
/* Funzione per fare il rapporto in f.trigonometrica */
/*-----*/
n_compl_t rapporto_f_trig(n_compl_t n1,
                          n_compl_t n2)
{
    n_compl_t rapporto_trig;

    /*calcolo modulo*/
    rapporto_trig.dati_forma.f_trigonometrica.modulo =
        n1.dati_forma.f_trigonometrica.modulo /
        n2.dati_forma.f_trigonometrica.modulo;

    /*calcolo argomeno*/
    rapporto_trig.dati_forma.f_trigonometrica.argomento =
        n1.dati_forma.f_trigonometrica.argomento -
        n2.dati_forma.f_trigonometrica.argomento;

    return(rapporto_trig);
}

```

Contenuto del file **programma.c**

```

/*****/
/* Inclusione delle librerie */
/*****/

/*inclusione della libreria standard "stdio.h"*/
#include <stdio.h>

/*inclusione della libreria non standard "libreria_complessi.h"*/
#include "libreria_complessi.h"

/*****/
/* Definizione delle funzioni */
/*****/
void riesegui_operazione(void);

void scelta_operazione(void);

/*****/
/* Funzione main*/
/*****/

```

```

int main(void)
{
    riesegui_operazione();
    return(0);
}

/*****
/* Funzione principale */
*****/
void riesegui_operazione(void)
{
    char altre_op; /*input: eseguire altre op.*/
    int esito;      /*var.di lavoro: esito delle scanf*/
    char carattere; /*var. di lavoro: caratteri nel buffer*/

    scelta_operazione();

    do
    {
        printf("Eseguire altre operazioni? [y/n]\n");
        esito = scanf(" %c",
                      &altre_op);
        if ((esito != 1) ||
            (altre_op != 'y' &&
             altre_op != 'Y' &&
             altre_op != 'n' &&
             altre_op != 'N' ))
        {
            printf("Digitare 'y' oppure 'n'\n");
            do
                carattere = getchar();
            while (carattere != '\n');
        }
    }
    while ((esito != 1) ||
           (altre_op != 'y' &&
            altre_op != 'Y' &&
            altre_op != 'n' &&
            altre_op != 'N' ));

    do
        carattere = getchar();
    while (carattere != '\n');

    if (altre_op == 'y' ||
        altre_op == 'Y')
        riesegui_operazione();
}

/*****
/* Funzione per scegliere quale operazione eseguire */
*****/
void scelta_operazione(void)

```

```

{
    int risposta;    /*input: quale operazione eseguire*/
    int esito_lettura; /*var. di lavoro: esito delle scanf*/
    char carattere;  /*var. di lavoro: caratteri nel buffer della tastiera*/
                    /*var. di lavoro: array di stringhe*/
    char scelta[5][36] = {"1)Cambio di forma",
                          "2)Somma tra numeri complessi",
                          "3)Differenza tra numeri complessi",
                          "4)Prodotto tra numeri complessi",
                          "5)Rapporto tra numeri complessi"};

    /*scelta di operazione da effettuare*/
    do
    {
        /*scelta di una operazione*/
        printf("Quale operazione eseguire? [1-2-3-4-5]\n\n");
        printf("\t%s\n",
               scelta[0]);
        printf("\t%s\n",
               scelta[1]);
        printf("\t%s\n",
               scelta[2]);
        printf("\t%s\n",
               scelta[3]);
        printf("\t%s\n",
               scelta[4]);

        esito_lettura = scanf("%d",
                              &risposta);

        /*validazione stretta dell'input*/
        if ((esito_lettura != 1) ||
            ((risposta != 1) &&
             (risposta != 2) &&
             (risposta != 3) &&
             (risposta != 4) &&
             (risposta != 5)))
        {
            printf("Assicurarsi di avere inserito ");
            printf("un valore da 1 a 5 e riprovare\n");

            do
                carattere = getchar();
            while (carattere != '\n');
        }
    }
    while ((esito_lettura != 1) ||
           ((risposta != 1) &&
            (risposta != 2) &&
            (risposta != 3) &&
            (risposta != 4) &&
            (risposta != 5)));
}

```



```

/*si esegue una specifica funzione della libreria*/
/*"libreria_complessi" a seconda della risposta*/
switch (risposta)
{
    case 1: /*invoca la funz. cambio di forma*/

        printf("Esecuzione operazione %s\n",
               scelta[risposta-1]);
        cambia_forma();
        break;

    case 2: /*invoca la funz. per sommare*/

        printf("Esecuzione operazione %s\n",
               scelta[risposta-1]);
        somma_n_compl();
        break;

    case 3: /*invoca la funz. per sottrarre*/

        printf("Esecuzione operazione %s\n",
               scelta[risposta-1]);
        differenza_n_compl();
        break;

    case 4: /*invoca la funz. per moltiplicare*/

        printf("Esecuzione operazione %s\n",
               scelta[risposta-1]);
        prodotto_n_compl();
        break;

    case 5: /*invoca la funz. per dividere*/

        printf("Esecuzione operazione %s\n",
               scelta[risposta-1]);
        rapporto_n_compl();
        break;
}
}

```

*Nota: per eventuali verifiche sul codice si consiglia di leggere il contenuto dei file .c e .h allegati, poiché potrebbero esserci delle differenze di impaginazione tra quella versione e quella riportata nella relazione.

Testing del programma

TESTING DEL PROGRAMMA

Esito della compilazione tramite makefile secondo lo standard ANSI e con relativa segnalazione di warning dell'obiettivo "numeri_complessi".

```
File  Modifica  Visualizza  Terminale  Schede  Aiuto
lorenzo@lorenzo-VirtualBox:~/Scrivania/Progetti/Progetto_PPL$ make numeri_complessi
gcc -ansi -Wall -O -c programma.c
gcc -ansi -Wall -O -c libreria_complessi.c
gcc -ansi -Wall -O programma.o libreria_complessi.o -o numeri_complessi -lm
lorenzo@lorenzo-VirtualBox:~/Scrivania/Progetti/Progetto_PPL$
```

Come si può ben vedere il file eseguibile 'numeri_complessi' viene compilato correttamente senza nessun errore o warning.

Lancio dell'eseguibile e schermata iniziale.

```
lorenzo@lorenzo-VirtualBox:~/Scrivania/Progetti/Progetto_PPL$ ./numeri_complessi
Quale operazione eseguire? [1-2-3-4-5]

1)Cambio di forma
2)Somma tra numeri complessi
3)Differenza tra numeri complessi
4)Prodotto tra numeri complessi
5)Rapporto tra numeri complessi
.
```

Di seguito vengono riportati alcuni test, con relativi screenshot effettuati sul programma di testing definito "numeri_complessi" che utilizza le funzioni esportate dalla libreria "numeri_complessi.h"

Test n° 1

Alla scelta dell'operazione digitiamo 1, ovvero scegliamo di eseguire un cambio di forma, successivamente ci verrà chiesta la forma del numero da inserire, forma algebrica; diamo alla parte reale valore nullo e alla parte immaginaria un valore casuale.

Numero inserito: $+5i$

Risultato sperato: $5[\cos(90^\circ) + i\sin(90^\circ)]$

```
Esecuzione operazione 1)Cambio di forma
Selezionare la forma del numero complesso (1 o 2):

1. Forma Algebrica
2. Forma Trigonometrica
1
Inserire il valore della parte reale: 0
Inserire il valore della parte immaginaria: 5
Il numero risultante in forma trigonometrica è: 5.00(cos90.00° + i * sin90.00°)
Eeguire altre operazioni? [y/n]
```

Il risultato che otteniamo è proprio quello sperato.

Test n°2

Questa volta effettuiamo sempre il cambio di forma, ma inserendo un numero in forma trigonometrica con valori casuali per modulo e argomento.

Numero inserito: $5[\cos(45^\circ) + i\sin(45^\circ)]$

Risultato sperato: $\frac{5}{2}\sqrt{2} + \frac{5}{2}\sqrt{2}i$

```
Esecuzione operazione 1)Cambio di forma
Selezionare la forma del numero complesso (1 o 2):

    1. Forma Algebrica
    2. Forma Trigonometrica
2
Inserire il valore del modulo: 5
Inserire il valore dell'argomento (in gradi): 45
Il numero complesso risultante in forma algebrica è: 3.54 + 3.54i
Eeguire altre operazioni? [y/n]
```

Il risultato ottenuto è proprio quello sperato poiché $\frac{5}{2}\sqrt{2} \cong 3.54$.

Test n°3

Proviamo ora ad effettuare una somma; alla richiesta dell'operazione inseriamo 2. Prendiamo come caso particolare quello della somma tra un numero e il suo coniugato.

Somma inserita: $(4 + 3i) + (4 - 3i)$

Risultato sperato: 8

```
Esecuzione operazione 2)Somma tra numeri complessi
Selezionare la forma del numero complesso (1 o 2):

    1. Forma Algebrica
    2. Forma Trigonometrica
1
Inserire il valore della parte reale: 4
Inserire il valore della parte immaginaria: 3
Selezionare la forma del numero complesso (1 o 2):

    1. Forma Algebrica
    2. Forma Trigonometrica
1
Inserire il valore della parte reale: 4
Inserire il valore della parte immaginaria: -3
Il numero complesso risultante in forma algebrica è: 8.00
Eeguire altre operazioni? [y/n]
```

Test n°4

Eseguiamo la somma tra numeri in forma trigonometrica dove il primo numero ha modulo nullo

Somma inserita: $0[\cos(90^\circ) + i\sin(90^\circ)]$

Risultato sperato: $9[\cos(0^\circ) + i\sin(0^\circ)]$

```
Esecuzione operazione 2)Somma tra numeri complessi
Selezionare la forma del numero complesso (1 o 2):

    1. Forma Algebrica
    2. Forma Trigonometrica
2
Inserire il valore del modulo: 0
Inserire il valore dell'argomento (in gradi): 90
Selezionare la forma del numero complesso (1 o 2):

    1. Forma Algebrica
    2. Forma Trigonometrica
2
Inserire il valore del modulo: -9
Inserire il valore dell'argomento (in gradi): 0
Il numero risultante in forma trigonometrica è: 9.00(cos360.00° + i * sin360.00°)
Eseguire altre operazioni? [y/n]
```

Test n°5

Svolgiamo la differenza tra due numeri complessi in forme diverse (il primo numero in forma trigonometrica e il secondo in forma algebrica)

Differenza inserita: $\{3[\cos(210^\circ) + i\sin(210^\circ)]\} - (-2 - i)$

Risultato sperato: $0.78[\cos(-320^\circ) + i\sin(-320^\circ)]$

```
Esecuzione operazione 3)Differenza tra numeri complessi
Selezionare la forma del numero complesso (1 o 2):

    1. Forma Algebrica
    2. Forma Trigonometrica
2
Inserire il valore del modulo: 3
Inserire il valore dell'argomento (in gradi): 210
Selezionare la forma del numero complesso (1 o 2):

    1. Forma Algebrica
    2. Forma Trigonometrica
1
Inserire il valore della parte reale: -2
Inserire il valore della parte immaginaria: -1
Il numero risultante in forma trigonometrica è: 0.78(cos-320.10° + i * sin-320.10°)
Eseguire altre operazioni? [y/n]
```

Come possiamo ben vedere il risultato è della stessa forma del primo termine della sottrazione.

Test n°6

Differenza fra due numeri complessi in forma algebrica, con la parte reale del secondo numero nulla.

Differenza inserita: $(6 - 5i) - (-2i)$

Risultato sperato: $6 - 3i$

```
Esecuzione operazione 3) Differenza tra numeri complessi
Selezionare la forma del numero complesso (1 o 2):

    1. Forma Algebrica
    2. Forma Trigonometrica
1
Inserire il valore della parte reale: 6
Inserire il valore della parte immaginaria: -5
Selezionare la forma del numero complesso (1 o 2):

    1. Forma Algebrica
    2. Forma Trigonometrica
1
Inserire il valore della parte reale: 0
Inserire il valore della parte immaginaria: -2
Il numero complesso risultante in forma algebrica è: 6.00 - 3.00i
Eseguire altre operazioni? [y/n]
```

Test n°7

Prodotto tra due numeri qualunque in forma algebrica.

Prodotto inserito: $(5 - 2i) \cdot (3 + i)$

Risultato sperato: $17 - i$

```
Esecuzione operazione 4) Prodotto tra numeri complessi
Selezionare la forma del numero complesso (1 o 2):

    1. Forma Algebrica
    2. Forma Trigonometrica
1
Inserire il valore della parte reale: 5
Inserire il valore della parte immaginaria: -2
Selezionare la forma del numero complesso (1 o 2):

    1. Forma Algebrica
    2. Forma Trigonometrica
1
Inserire il valore della parte reale: 3
Inserire il valore della parte immaginaria: 1
Il numero complesso risultante in forma algebrica è: 17.00 - 1.00i
Eseguire altre operazioni? [y/n]
```

Test n°8

Prodotto fra due numeri in forma trigonometrica

Prodotto inserito: $\{-1[\cos(180^\circ) + i\sin(180^\circ)]\} \cdot \{2[\cos(45^\circ) + i\sin(45^\circ)]\}$

Risultato sperato: $-2[\cos(225^\circ) + i\sin(225^\circ)]$

```
Esecuzione operazione 5)Rapporto tra numeri complessi
Selezionare la forma del numero complesso (1 o 2):

    1. Forma Algebrica
    2. Forma Trigonometrica
2
Inserire il valore del modulo: -1
Inserire il valore dell'argomento (in gradi): 180
Selezionare la forma del numero complesso (1 o 2):

    1. Forma Algebrica
    2. Forma Trigonometrica
2
Inserire il valore del modulo: 2
Inserire il valore dell'argomento (in gradi): 45
Il numero risultante in forma trigonometrica è: -0.50(cos135.00° + i * sin135.00°)
Eseguire altre operazioni? [y/n]
```

Test n°9

Svolgiamo la divisione fra numeri complessi i cui termini sono gli stessi del prodotto effettuato nel test n°8

Prodotto inserito: $\{-1[\cos(180^\circ) + i\sin(180^\circ)]\} / \{2[\cos(45^\circ) + i\sin(45^\circ)]\}$

Risultato sperato: $-\frac{1}{2}[\cos(135^\circ) + i\sin(135^\circ)]$

```
Esecuzione operazione 4)Prodotto tra numeri complessi
Selezionare la forma del numero complesso (1 o 2):

    1. Forma Algebrica
    2. Forma Trigonometrica
2
Inserire il valore del modulo: -1
Inserire il valore dell'argomento (in gradi): 180
Selezionare la forma del numero complesso (1 o 2):

    1. Forma Algebrica
    2. Forma Trigonometrica
2
Inserire il valore del modulo: 2
Inserire il valore dell'argomento (in gradi): 45
Il numero risultante in forma trigonometrica è: -2.00(cos225.00° + i * sin225.00°)
Eseguire altre operazioni? [y/n]
```

Test n°10

Divisione tra due numeri complessi entrambi in forma algebrica, con parte reale del primo numero nulla.

Divisione inserita: $(+5i)/(3 + i)$

Risultato sperato: $0.44 + 2.67i$

```
Esecuzione operazione 5)Rapporto tra numeri complessi
Selezionare la forma del numero complesso (1 o 2):

    1. Forma Algebrica
    2. Forma Trigonometrica
1
Inserire il valore della parte reale: 0
Inserire il valore della parte immaginaria: 5
Selezionare la forma del numero complesso (1 o 2):

    1. Forma Algebrica
    2. Forma Trigonometrica
1
Inserire il valore della parte reale: 3
Inserire il valore della parte immaginaria: 1
Il numero complesso risultante in forma algebrica è: 0.44 + 2.67i
Esequire altre operazioni? [y/n]
```

Anche in questo caso possiamo notare che il risultato è quello sperato.

Dopo aver compiuto dieci test siamo abbastanza sicuri della correttezza del programma, anche perché i calcoli che effettua non sono molto complicati.

Per essere assolutamente certi che il programma restituisca dei dati corretti occorre verificare la sua correttezza mediante le triple di Hoare, questo è effettuato nella sezione “verifica del programma”.

Verifica del programma

VERIFICA DEL PROGRAMMA

Per verificare la correttezza del programma prendiamo in considerazione un insieme di tre istruzioni comparse nel programma, all'interno della funzione `converti_alg_trig`.

```
num.forma = forma_trigonometrica;

num.dati_forma.f_trigonometrica.modulo =
    sqrt (pow (num.dati_forma.f_algebrica.parte_reale, 2) +
          pow (num.dati_forma.f_algebrica.parte_imag, 2));

if (num.dati_forma.f_algebrica.parte_imag >= 0)
    num.dati_forma.f_trigonometrica.argomento =
        atan2 (num.dati_forma.f_algebrica.parte_imag,
               num.dati_forma.f_algebrica.parte_reale) + PI_GRECO;
else
    num.dati_forma.f_trigonometrica.argomento =
        atan2 (num.dati_forma.f_algebrica.parte_imag,
               num.dati_forma.f_algebrica.parte_reale) + PI_GRECO;
```

Queste istruzioni (S) non sono di input/output e le loro postcondizioni {Q} non sono delle tautologie, dunque è opportuno calcolare le loro precondizioni {P}.

Le prime due istruzioni sono dei semplici assegnamenti quindi calcolare le wp (S, R) consiste nel sostituire in R tutte le occorrenze di x con e R [e/x].

La terza istruzione è di selezione e si presenta nella forma “if (β) S₁ else S₂”; per calcolare la sua precondizione debole occorre lavorare sulle due parti come segue:

$$wp(S, R) = ((\beta) \rightarrow wp(S_1, R)) \wedge ((\neg \beta) \rightarrow wp(S_2, R))$$

Per poter lavorare meglio occorre rimuovere dal codice tutte le parti legate strettamente al linguaggio riportandole a stretto contatto con la matematica, quindi le tre istruzioni si presenteranno in questa forma:

forma = 1

$$modulo = \sqrt{parte\ reale^2 + parte\ immaginaria^2}$$

if (parte immaginaria \geq 0)

$$argomento = \arctan\left(\frac{parte\ immaginaria}{parte\ reale}\right) + \pi$$

else

$$argomento = \arctan\left(\frac{parte\ immaginaria}{parte\ reale}\right) - \pi$$

Accorciando i nomi delle variabili e sostituendo il numero di identificazione dell'elemento della struttura al posto del suo nome (nella prima formula al posto di “forma_trigonometrica” vi è 1 poiché è il secondo elemento del tipo di dato forma_t) è più semplice eseguire i calcoli.

Prima istruzione

S: $forma = 1$

R: $forma = 0 \vee forma = 1$

$wp(S, R) = (forma = 0 \vee forma = 1)_{forma,1} = (1 = 0 \vee 1 = 1) \equiv (falso \vee vero)$

Seconda istruzione

S: $modulo = \sqrt{parte\ reale^2 + parte\ immaginaria^2}$

R: $modulo \in \mathbb{R}$

$wp(S, R) = \left(\sqrt{parte\ reale^2 + parte\ immaginaria^2} \in \mathbb{R} \right)$
 $= vero, se ((parte\ reale)^2 \wedge (parte\ immaginaria)^2 \in \mathbb{R}) \wedge (parte\ reale^2 + parte\ immaginaria^2 \neq -1)$

Quindi l'ottenimento del risultato dipende dal valore di $(parte\ reale)^2$ e $(parte\ immag.^2)$ che devono appartenere ad \mathbb{R} e la loro somma deve essere diversa da (-1).

Terza istruzione

S: if $(parte\ immaginaria \geq 0)$

$argomento = \arctan\left(\frac{parte\ immaginaria}{parte\ reale}\right) + \pi$
else

$argomento = \arctan\left(\frac{parte\ immaginaria}{parte\ reale}\right) - \pi$

R: $argomento \in [\mathbb{R} \setminus (\frac{\pi}{2} + k\pi)]$

$wp(S, R) = ((parte\ immaginaria \geq 0) \rightarrow wp(S_1, R))$
 $\wedge ((parte\ immaginaria < 0) \rightarrow wp(S_2, R)) \equiv (vero \wedge vero) = vero$

$wp(S_1, R) = \left(\arctan\left(\frac{parte\ immaginaria}{parte\ reale}\right) + \pi \right) \in \mathbb{R}$
 $= vero\ se\ e\ solo\ se\ \arctan\left(\frac{parte\ immaginaria}{parte\ reale}\right) \neq \left(\frac{\pi}{2} + k\pi\right)$

$wp(S_2, R) = \left(\arctan\left(\frac{parte\ immaginaria}{parte\ reale}\right) - \pi \right) \in \mathbb{R}$
 $= vero\ se\ e\ solo\ se\ \arctan\left(\frac{parte\ immaginaria}{parte\ reale}\right) \neq \left(\frac{\pi}{2} + k\pi\right)$

Vero è la preconditione più debole poiché $(vero \wedge vero) \equiv vero$, quindi l'istruzione è sempre corretta rispetto alla postcondizione.