電子 1(木曜クラス)

第 12 回課題 12

(構造体、2010/7/8(木))

基本問題

12-1:複素数

本提出期限 2010/7/8(木) 22:00
再提出期限 2010/7/23(金)22:00

提出物 : Makefile、ソースファイル (complex.c)、入力ファイル (complex.in)、出力ファイル (complex.out)

複素数の四則演算 (加算、減算、乗算、除算) を行なうプログラムを作成せよ。ただし、要求仕様を全て満ように作成せよ．また、最後の実行例を参考にプログラムを作成せよ。

要求仕様 1(全体的な要求)

- プログラム全体の入力は，2 つの複素数を標準入力から行なう。各複素数は，実部 (double 型の値)，虚部 (double 型の値) の順に標準入力から入力する．
- プログラム全体の出力は全て標準出力に行なう。入力された 2 つの複素数に対して、4 つの演算を行なった結果すべて (それぞれは複素数) を出力する．各複素数は (実部，虚部) の形式で，実部，虚部共にも小数点以下 2 桁まで出力する．実行例参照.
- プログラムの内部仕様として、以下を満たすこと。
  - 複素数を表す構造体型として，struct complex 型を定義して利用せよ．struct complex 型は以下のように定義してあること．

```
/*構造体テンプレート定義*/
/*複素数を表す構造体*/
struct complex
{
    double real; /*実部*/
    double imag; /*虚部*/

};
```

  - 1 個の複素数を標準入力より入力する関数 scan_complex 作成せよ。scan_complex の仕様は後を参照.
  - 1 個複素数を標準出力へ出力する関数 print_complex を作れ．print_complex の仕様は後を参照.
  - 複素数の四則演算を行なう関数をそれぞれ作成せよ．加算を行なう関数 add_complex, 減算を行なう関数 sub_complex, 乗算を行なう関数 mul_complex, 除算を行なう関数 div_complex の各仕様は後の説明を参照.
  - グローバル変数を用いないこと。
  - main 関数中では，scanf 文を利用しないこと．
  - main 関数中では，pritnf 文を用いて複素数型の変数の値の表示を行なわない.
  - main 関数中では，複素数の四則演算を行なう関数を呼び出す前に，入力された複素数をチェックすること．入力値によっては，演算が不可能な場合がある．この場合には，可能な演算のみ行ない，不可能な演算については四則演算の関数を呼び出さずに適切なメッセージを標準出力に出力せよ．

要求仕様 **2**(関数 scan_complex への要求)

- 次のプロトタイプ宣言を持つ.

```
/*
機能説明：
複素数を標準入力から読み込む関数

仮引数：なし (void)

戻り値：標準入力から読み込んだ複素数

副作用:
標準入力から 2 つの double 値を読み込み,
1 つ目 double 値を戻り値の実部,
2 つ目 double 値を戻り値の虚部とする.
*/
struct complex scan_complex();
```

- 仮引数は無い.
- 戻り値は，標準入力より読み込まれた複素数 (struct complex) 型の値とする．標準入力より 2 つの実数 (double 値) を読みとり，1 番目の実数を実部，2 番目の実数を虚部として持つ複素数を返す.
- 関数 scan_complex 中では標準出力への出力は行なってはならない．例えば，printf 文による表示を行ってはならない.

要求仕様 3(関数 print_complex への要求)

- 次のプロトタイプ宣言を持つ.

```
/*
機能説明：
複素数を標準出力に出力する関数

仮引数：z:標準出力へ出力すべき複素数

戻り値：なし (void)

副作用:
仮引数で与えられた複素数 z を整形して標準出力へ出力する.
*/
void print_complex(struct complex z);
```

- 仮引数 z は，標準出力へ出力したい任意の複素数 (struct complex 型の値) とする.
- 戻り値は無い．すなわち，戻り値の型は，void 型とする.
- 仮引数 z で与えられた複素数の実部および虚部を，(実部，虚部) の形式で標準出力に出力する副作用を持つ．実部，虚部共にも小数点以下 2 桁まで出力する.
- 関数 print_complex 中では複素数の値以外の表示を行なってはならない．特に，“改行” を表示してはならない.
- 関数 print_complex 中では標準入力による値の読み込みは行なってはならない．例えば，scanf 文を用いてはならない.

要求仕様 4(関数 add_complex への要求)

- 次のプロトタイプ宣言を持つ.

```
/*
機能説明：
2 の複素数の和を求める関数

仮引数:
operand1：被演算項 1，和の左辺の複素数
operand2：被演算項 2，和の右辺の複素数

戻り値:
仮引数で与えられた 2 つの複素数の和 operand1+operand2
*/
struct complex add_complex(struct complex operand1,
                           stcuct complex prerand2);
```

- 仮引数は以下の意味を持つ.
  - 仮引数 operand1 は被演算項であり，演算の左辺の値 (複素数，struct complex 型の値) を表わす operand1 は，任意の複素数で良い.
  - 仮引数 operand2 は被演算項であり，演算の右辺の値 (複素数，struct complex 型の値) を表わす operand2 は，任意の複素数で良い.
- 戻り値として，加算結果 (operand1+operand2) の複素数 (struct complex 型の値) を返す.
- 関数 add_complex 中では標準入出力は行なってはならない．例えば，scanf 文や printf 文は用いてはならない.

要求仕様 5(関数 sub_complex への要求)

- 次のプロトタイプ宣言を持つ.

```
/*
機能説明：
2 の複素数の差を求める関数

仮引数:
operand1：被演算項 1，差の左辺の複素数
operand2：被演算項 2，差の右辺の複素数

戻り値:
仮引数で与えられた 2 つの複素数の差 operand1-operand2
*/
struct complex sub_complex(struct complex operand1,
                           struct complex prerand2);
```

- 仮引数は以下の意味を持つ.
  - 仮引数 operand1 は被演算項であり，演算の左辺の値 (複素数，struct complex 型の値) を表わす operand1 は，任意の複素数で良い.
  - 仮引数 operand2 は被演算項であり，演算の右辺の値 (複素数，strcut complex 型の値) を表わす operand2 は，任意の複素数で良い.
- 戻り値として，減算結果 (operand1-operand2) の複素数 (struct complex 型の値) を返す.
- 関数 sub_complex 中では標準入出力は行なってはならない．例えば，scanf 文や printf 文は用いてはならない.

要求仕様 6(関数 mul_complex への要求)

- 次のプロトタイプ宣言を持つ.

```
/*
機能説明：
2 の複素数の積を求める関数

仮引数:
operand1：被演算項 1，積の左辺の複素数
operand2：被演算項 2，積の右辺の複素数

戻り値:
仮引数で与えられた 2 つの複素数の積 operand1*operand2
*/
struct complex mul_complex(struct complex operand1,
                           struct complex prerand2);
```

- 仮引数は以下の意味を持つ.
  - 仮引数 operand1 は被演算項であり，演算の左辺の値 (複素数，strcut complex 型の値) を表わす operand1 は，任意の複素数で良い.
  - 仮引数 operand2 は被演算項であり，演算の右辺の値 (複素数，struct complex 型の値) を表わす operand2 は，任意の複素数で良い.
- 戻り値として，乗算結果 (operand1*operand2) の複素数 (struct complex 型の値) を返す.
- 関数 mul_complex 中では標準入出力は行なってはならない．例えば，scanf 文や printf 文は用いてはならない.

要求仕様 7(関数 div_complex への要求)

- 次のプロトタイプ宣言を持つ.

```
/*
機能説明：
2 の複素数の商を求める関数

仮引数:
operand1：被演算項 1，商の左辺の複素数
operand2：被演算項 2，商の右辺の複素数 ((0.0,0.0) 以外の値)

戻り値:
仮引数で与えられた 2 つの複素数の積 operand1*operand2
*/
struct complex div_complex(struct complex operand1,
                           struct complex prerand2);
```

- 仮引数は以下の意味を持つ.
  - 仮引数 operand1 は被演算項であり，演算の左辺の値 (複素数，struct complex 型の値) を表わす operand1 は，任意の複素数で良い.
  - 仮引数 operand2 は被演算項であり，演算の右辺の値 (複素数，Complex 型の値) を表わす operand2 に与えられる複素数は，operand2 は (0.0,0.0) であってはならない. operand2 が (0.0,0.0) であった場合は、戻り値はどのような値でも良い。この関数内でエラー処理する必要は無い.
- 戻り値として，除算結果 (operand1/operand2) の複素数 (struct complex 型の値) を返す.
- 関数 div_complex 中では標準入出力は行なってはならない．例えば，scanf 文や printf 文は用いてはならない.

complex.in 例

```
1.0 2.0
3.0 4.0
```

実行例

```
b11b0xx@tyy:~/prog/12$./complex < complex.in
複素数 a=? (空白区切り)
a=(  1.00,  2.00)
複素数 b=? (空白区切り)
b=(  3.00,  4.00)

a+b=(  4.00,  6.00)
a-b=( -2.00, -2.00)
a*b=( -5.00, 10.00)
a/b=(  0.44,  0.08)
b11b0xx@tyy:~/prog/12$
```