

2009 年度 情報科学演習 演習 5

2009 年 6 月 10 日 (水)

演習 1 以下の (1)～(5) のプログラムを入力しなさい。さらに、コンパイルし、オブジェクトモジュールの実行しなさい。なお、「新訂 新 C 言語入門 ビギナー編」に実行結果が載っている場合は、オブジェクトモジュールの実行結果と同じになることを確認しなさい。

- (1) ■ p.205 の「変数のアドレスを見る」(ファイル名「`exercise0501.c`」)を入力しなさい。
- (2) ■ p.212 の「配列のアドレス表現を確認する」(ファイル名「`exercise0502.c`」)を入力しなさい。
- (3) ■ p.214 の「ポインタ演算をする」(ファイル名「`exercise0503.c`」)を入力しなさい。
- (4) ■ p.231 の「構造体を使ったプログラム」(ファイル名「`exercise0504.c`」)を入力しなさい。
- (5) ■ p.233 の「ふたつのメンバ指定方法を見る」(ファイル名「`exercise0505.c`」)を入力しなさい。

演習 2 演習 4 の **演習 4** の「複素数の四則演算を行なうプログラム 1」を変更し、「複素数の四則演算を行なうプログラム 3 (ポインタを使用)」(ファイル名「`exercise0506.c`」)を作成しなさい。ヒント：

- ■ p.220 のリスト「ポインタ引数で値を戻す関数」を参考にする。
- グローバル変数 `x, y` の代わりに関数「`main()`」の中でローカル変数 `x, y` を宣言する。
- 各関数の引数に `double` 型のポインタ `*px, *py` を追加する。

【参考】以下のプログラムは、関数から関数への渡し値と戻り値を様々な方法で行ったものである。各プログラムを比較し、その違いをよく理解しておくこと。

- 「複素数の四則演算を行なうプログラム 1 (グローバル変数を使用)」
- 「複素数の四則演算を行なうプログラム 2 (配列を使用)」 * 配列の先頭アドレスを渡す。
- 「複素数の四則演算を行なうプログラム 3 (ポインタを使用)」
- 「複素数の四則演算を行なうプログラム 4 (構造体を使用)」

演習 3 演習 4 の **演習 4** の「複素数の四則演算を行なうプログラム 1」を変更し、「複素数の四則演算を行なうプログラム 4 (構造体を使用)」(ファイル名「**exercise0507.c**」)を作成しなさい。ヒント：書き出しは以下のようになる (■ p.235～p.236 のリスト「構造体を引数と戻り値にする」とレポート 6 の **問 5** を参考にする)。

● 複素数の四則演算を行なうプログラム 4 (構造体を使用)

exercise0507.c

```

1: #include <stdio.h>
2:
3: struct complex { /* (re) + (im)i */
4:     double re;           ← real part (実部) の略
5:     double im;          ← imaginary part (虚部) の略
6: };
7:
8: struct complex add(struct complex a, struct complex b);
9: struct complex sub(struct complex a, struct complex b);
10: struct complex mul(struct complex a, struct complex b);
11: struct complex div(struct complex a, struct complex b);
12:
13: int main(void)
14: {
15:     struct complex a = {1.2, -0.5}; /* a = (1.2) + (-0.5)i */
16:     struct complex b = {-2.2, 6.8}; /* b = (-2.2) + (6.8)i */
17:     struct complex c;
18:
19:     c = add(a, b);
20:     printf("((%f)+(%f)i)+((%f)+(%f)i)=((%f)+(%f)i)%n",
21:           a.re, a.im, b.re, b.im, c.re, c.im);
22:     c = sub(a, b);
23:     printf("((%f)+(%f)i)-((%f)+(%f)i)=((%f)+(%f)i)%n",
24:           a.re, a.im, b.re, b.im, c.re, c.im);
25:     c = mul(a, b);
26:     printf("((%f)+(%f)i)*((%f)+(%f)i)=((%f)+(%f)i)%n",
27:           a.re, a.im, b.re, b.im, c.re, c.im);
28:     c = div(a, b);
29:     printf("((%f)+(%f)i)/((%f)+(%f)i)=((%f)+(%f)i)%n",
30:           a.re, a.im, b.re, b.im, c.re, c.im);
31:
32:     return 0;
33: }
34:
35: struct complex add(struct complex a, struct complex b)
36: {
    :
    :

```