

2005 年度 プログラミング演習 II 演習 7

2005 年 11 月 16 日 (水) (同日修正)

演習 1 以下は「配列の各要素のアドレスを観察するプログラム (変数名を使用)」である。このソースプログラムを変更し、「配列の各要素のアドレスを観察するプログラム (ポインタを使用)」を作成しなさい (ファイル名「p2exercise0702.c」)。ヒント：

- 第 6 行を「`int *p;`」と書き換える (ポインタの宣言)。 ← 修正
- 第 7 行と第 8 行の間に「`p = a;`」を挿入する (アドレスの代入)。 ← 追加
* 配列なので「`&`」は不要
- ■ p.214 のリストを参考にする。

● 配列の各要素のアドレスを観察するプログラム (変数名を使用)

p2exercise0701.c

```
1: #include <stdio.h>
2:
3: int main(void)
4: {
5:     int a[5] = {1, 2, 3, 4, 5};
6:     int i = 0;
7:
8:     while (1) {
9:         printf("%p: %d\n", &a[i], a[i]);
10:        if (a[i] == 5) break;
11:        ++i;
12:    }
13:
14:    return 0;
15: }
```

← 修正

* 「■」は教科書「新訂 新 C 言語入門 ビギナー編」を表す。

演習 2 演習 5 の **演習 3** の「複素数の四則演算を行なうプログラム」を変更し、「複素数の四則演算を行なうプログラム (ポインタを使用)」を作成しなさい (ファイル名「p2exercise0703.c」)。ヒント：

- グローバル変数 `x, y` の代わりに `main()` 関数の中でローカル変数 `x, y` を宣言する。 ← 修正
- ■ p.220 のリストを参考にする。

演習 3 演習 5 の **演習 3** の「複素数の四則演算を行なうプログラム」を変更し、「複素数の四則演算を行なうプログラム (構造体を使用)」を作成しなさい (ファイル名「p2exercise0704.c」)。ヒント：書き出しは以下ようになる (■ p.235～p.236 のリストを参考にする)。

● 複素数の四則演算を行なうプログラム (構造体を使用)

p2exercise0704.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 x;
18:
19:     x = add(a, b);
20:     printf("((%f)+(%f)i)+((%f)+(%f)i)=((%f)+(%f)i)%n",
21:           a.re, a.im, b.re, b.im, x.re, x.im);
22:     x = sub(a, b);
23:     printf("((%f)+(%f)i)-((%f)+(%f)i)=((%f)+(%f)i)%n",
24:           a.re, a.im, b.re, b.im, x.re, x.im);
25:     x = mul(a, b);
26:     printf("((%f)+(%f)i)*((%f)+(%f)i)=((%f)+(%f)i)%n",
27:           a.re, a.im, b.re, b.im, x.re, x.im);
28:     x = div(a, b);
29:     printf("((%f)+(%f)i)/((%f)+(%f)i)=((%f)+(%f)i)%n",
30:           a.re, a.im, b.re, b.im, x.re, x.im);
31:
32:     return 0;
33: }
34:
35: struct complex add(struct complex a, struct complex b)
36: {
37:     :
```