

2005年度 プログラミング演習 II 演習 8

2005年12月7日(水)

演習 1 以下は「配列の各要素の和を計算するプログラム (変数名を使用)」である。このソースプログラムを変更し、「配列の各要素の和を計算するプログラム (ポインタを使用)」を作成しなさい (ファイル名「p2exercise0802.c」)。ヒント：

- 第7行を「int *p;」と書き換える (ポインタの宣言)。
- 第8行と第9行の間に「p = a;」を挿入する (アドレスの代入)。* 配列なので「&」は不要。

● 配列の各要素の和を計算するプログラム (変数名を使用)

p2exercise0801.c

```
1: #include <stdio.h>
2:
3: int main(void)
4: {
5:     int a[] = {90, 76, 80, 65, 100, -1};           ← 「-1」は終端を示す
6:     int sum = 0;
7:     int i = 0;
8:
9:     while (a[i] != -1) {
10:         sum = sum + a[i];
11:         i++;
12:     }
13:     printf("sum = %d\n", sum);
14:
15:     return 0;
16: }
```

	a	← 配列の先頭アドレス				
変数名 (アドレス)	&a[0]	&a[1]	&a[2]	&a[3]	&a[4]	&a[5]
ポインタ (アドレス)	p	p+1	p+2	p+3	p+4	p+5
a[] (= a[6])	90	76	80	65	100	-1
変数名 (値)	a[0]	a[1]	a[2]	a[3]	a[4]	a[5]
ポインタ (値)	*p	*(p+1)	*(p+2)	*(p+3)	*(p+4)	*(p+5)

演習 2 以下の「ユークリッドの互除法を用いて最大公約数を求めるプログラム (変数名を使用)」を変更し、「ユークリッドの互除法を用いて最大公約数を求めるプログラム (ポインタを使用)」を作成しなさい (ファイル名「p2exercise0803.c」)。ヒント：

- 関数「gcd()」のプロトタイプを「void gcd(int x, int y, int *pans);」と宣言する。
- 関数「main()」の中で最大公約数を記憶する変数「ans」を宣言する。
- 演習 7 の **演習 2** の解答例を参考にすること (レポート 6 の **問 2**)。

● ユークリッドの互除法を用いて最大公約数を求めるプログラム (変数名を使用) p2exercise0501.c

```
1: #include <stdio.h>
2:
3: int gcd(int x, int y);
4:
5: int main(void)
6: {
7:     int x = 672, y = 204;
8:
9:     printf("gcd(%d,%d)=%d\n", x, y, gcd(x, y));
10:
11:     return 0;
12: }
13:
14: int gcd(int x, int y)
15: {
16:     int tmp;
17:
18:     while (y != 0) {
19:         tmp = x % y;
20:         x = y;
21:         y = tmp;
22:     }
23:
24:     return x;
25: }
```

演習 3 演習 6 の **演習 1** の「有理数 \mathbb{Q} 上で四則演算を行なうプログラム」を変更し、「有理数 \mathbb{Q} 上で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「p2exercise0804.c」)。
ヒント :

- 書き出しは以下のようなになる。
- 関数「gcd2()」はそのまま使用すればよい。
- 演習 7 の **演習 3** の解答例を参考にすること (レポート 6 の **問 3**)。

● 有理数 \mathbb{Q} 上で四則演算を行なうプログラム (構造体を使用)

p2exercise0804.c

```
1: #include <stdio.h>
2:
3: struct rational {                                ← rational (有理数)
4:     int nu;                                       ← numerator (分子) の略
5:     int de;                                       ← denominator (分母) の略
6: };
7:
8: int gcd2(int a, int b);
9: struct rational q_add(struct rational a, struct rational b);
10: struct rational q_sub(struct rational a, struct rational b);
11: struct rational q_mul(struct rational a, struct rational b);
12: struct rational q_div(struct rational a, struct rational b);
13:
14: int main(void)
15: {
16:     struct rational a, b, x;
17:     char op;
18:
19:     printf("?");
20:     scanf("(%d/%d)%c(%d/%d)", &a.nu, &a.de, &op, &b.nu, &b.de);
21:
22:     ⋮
23:     ⋮
24:     ⋮
25: }
```