

2011 年度 情報科学演習 演習 6

2011 年 6 月 15 日 (水)

演習 1 情報科学テキストの p.171 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 (ヘッダ)」(ファイル名「`myheader.h`」) と「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 (ソースプログラム)」(ファイル名「`gcd.c`」) をそれぞれ入力し、コマンド「`gcc -E gcd.c | less`」を実行してヘッダファイルがインクルードされていることを確認しなさい。もちろん、コンパイルして実行結果も確認すること。

演習 2 演習 5 の **演習 2** の「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`」) に含まれる複素数の四則演算の関数たちをヘッダファイル (ファイル名「`complex.h`」) にしなさい。なお、ヘッダファイルを利用した「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「`exercise0601.c`」) は以下になる。さらに、演習 1 と同様、コマンド「`gcc -E exercise0601.c | less`」を実行してヘッダファイルがインクルードされていることを確認しなさい。もちろん、コンパイルして実行結果も確認すること。

● 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)

`exercise0601.c`

```
1: #include <stdio.h>
2: #include "complex.h"
3:
4: int main(void)
5: {
6:     double a = 1.2, b = -0.5; /* a + bi */
7:     double c = -2.2, d = 6.8; /* c + di */
8:     double x, y;             /* x + yi */
9:
10:    add(a, b, c, d, &x, &y);
11:    printf("((%f)+(%f)i)+((%f)+(%f)i)=(%f)+(%f)i\n", a, b, c, d, x, y);
12:    sub(a, b, c, d, &x, &y);
13:    printf("((%f)+(%f)i)-((%f)+(%f)i)=(%f)+(%f)i\n", a, b, c, d, x, y);
14:    mul(a, b, c, d, &x, &y);
15:    printf("((%f)+(%f)i)*((%f)+(%f)i)=(%f)+(%f)i\n", a, b, c, d, x, y);
16:    div(a, b, c, d, &x, &y);
17:    printf("((%f)+(%f)i)/((%f)+(%f)i)=(%f)+(%f)i\n", a, b, c, d, x, y);
18:
19:    return 0;
20: }
```

演習 3 ■ p.256 の「マクロ機能を使う」(ファイル名「`exercise0602.c`」)を入力しなさい。さらに、コマンド「`gcc -E exercise0602.c | less`」を実行し、プログラムの先頭部分のマクロ定義「`#define`」によってプログラムが置換されていることを確認しなさい。もちろん、コンパイルして実行結果も確認すること。

演習 4 以下の (1)～(2) 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を実行しなさい。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.271 の「三角関数の使用例」(ファイル名「`exercise0603.c`」)を入力しなさい。

コンパイル方法：

```
$ gcc -o exercise0603.exe exercise0603.c -lm
                                     ↑ ライブラリ「libm.a」をリンクする
$
```

(2) ■ p.273 の「`rand` 関数の使用例」(ファイル名「`exercise0604.c`」)を入力しなさい。

【注意】Cygwin 環境では「`RAND_MAX=0x7FFFFFFF (2147483647)`」と定義されている。

演習 5 ■ p.271 のリスト「数学関数の使用例」を参考に、「平方根を求めるプログラム」(ファイル名「`exercise0605.c`」)を作成しなさい。ただし、出力結果が以下になること。ヒント：

- キャスト演算子 (■ p.84) を用いて整数型から倍精度浮動小数点型に変換する。
- 値を表示するための変換仕様には、それぞれ「`%2d`」, 「`%.15f`」を用いる。

出力結果：

```
$ ./exercise0605.exe
sqrt( 1)=1.0000000000000000
sqrt( 2)=1.414213562373095
sqrt( 3)=1.732050807568877
sqrt( 4)=2.0000000000000000
sqrt( 5)=2.236067977499790
sqrt( 6)=2.449489742783178
sqrt( 7)=2.645751311064591
sqrt( 8)=2.828427124746190
sqrt( 9)=3.0000000000000000
sqrt(10)=3.162277660168380
$
```