

2011 年度 情報科学演習 演習 2

2011 年 5 月 18 日 (水)

演習 1 以下のソースプログラムは「1 から 10 まで表示するプログラム」である。このソースプログラムを変更し、実行結果のように「アルファベットA からZ まで表示するプログラム 1」(ファイル名「`exercise0202.c`」)を作成しなさい。ヒント：

- A からZ の文字コードは「新訂 新 C 言語入門 ビギナー編」の見開きを見よ。
- `printf` で 1 文字を表示するには「`%c`」を使用する。
- キャスト演算子「`(char)`」を使用する (■ p.84 を参照)。

● 1 から 10 まで表示するプログラム

`exercise0201.c`

```
1: #include <stdio.h>
2:
3: int main(void)
4: {
5:     int i;
6:
7:     for (i = 1; i <= 10; i++) {
8:         printf("%d\n", i);
9:     }
10:
11:     return 0;
12: }
```

実行結果：

```
$ ./exercise0202.exe
A
B
... 省略 ...
Z

$
```

演習 2 演習 1 の `for` 文を使用した「アルファベットA からZ まで表示するプログラム 1」を参考に、`while` 文を使用した「アルファベットA からZ まで表示するプログラム 2」(ファイル名「`exercise0203.c`」) および `do-while` 文を使用した「アルファベットA からZ まで表示するプログラム 3」(ファイル名「`exercise0204.c`」)を作成しなさい。ヒント：■ p.129 の練習問題 1 を参考にせよ。

* 「■」は教科書「新訂 新 C 言語入門 ビギナー編」を表す。

演習 3 以下の「掛け算九九の表を表示するプログラム」を変更し、「16進数の掛け算表を16進数で表示するプログラム」(ファイル名「`exercise0206.c`」)を作成しなさい。なお、第10行の「`%3d`」の意味は、表示幅3桁・右づめで整数値を表示する。ヒント: `printf` で16進数を英大文字3桁で表示するには「`%3X`」を使用する。

● 掛け算九九の表を表示するプログラム

`exercise0205.c`

```
1: #include <stdio.h>
2:
3: int main(void)
4: {
5:     int i, j;
6:
7:     for (i = 1; i < 10; i++) {
8:         for (j = 1; j < 10; j++) {
9:             if (i <= j) {
10:                printf("%3d", i * j);
11:            }
12:            else {
13:                printf("  ");
14:            }
15:        }
16:        printf("\n");
17:    }
18:
19:    return 0;
20: }
```

実行結果 :

```
$ ./exercise0206.exe
1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
   4  6  8  A  C  E 10 12 14 16 18 1A 1C 1E
     9  C  F 12 15 18 1B 1E 21 24 27 2A 2D

... 省略 ...

          90 9C A8 B4
            A9 B6 C3
              C4 D2
                E1

$
```

演習 4 以下の「5 の平方根を小数点以下第 2 位まで求めるプログラム」は、レポート 2 の問 1 の (1) の「5 の平方根を小数点以下第 2 位まで求めるアルゴリズムの流れ図 (前判定)」をプログラムにしたものである。このプログラムを変更し、「1000 の平方根を少数第 2 位まで求めるプログラム」(ファイル名「`exercise0208.c`」)を作成しなさい。

● 5 の平方根を小数点以下第 2 位まで求めるプログラム

`exercise0207.c`

```
1: #include <stdio.h>
2:
3: int main(void)
4: {
5:     double m, n = 0, s = 0;
6:
7:     while (s <= 50000) {
8:         n = n + 1;
9:         m = 2 * n - 1;
10:        s = s + m;
11:    }
12:
13:    n = n - 1;
14:    n = n / 100;
15:
16:    printf("sqrt(5)=%f\n", n);
17:
18:    return 0;
19: }
```

実行結果 :

```
$ ./exercise0208.exe
sqrt(1000)=31.620000
$
```

考察 : $\sqrt{1000} = 10\sqrt{10}$ であることに着目し、 $\sqrt{10}$ の値を求めてから、その値を 10 倍するプログラムを作成せよ。

演習 5 以下は、演習 4 の「5 の平方根を小数点以下第 2 位まで求めるプログラム」を改良して、5 の平方根を小数点以下第 6 位まで求められるようにしたプログラムである。このプログラムをコンパイルすると以下の警告が発せられた (なお、実行結果は正しい結果が得られているようである)。今後、より精度の高い値を求めるためにはこのプログラムでは不十分である (計算量が膨大になる; 演習 4 の考察も考慮する必要がある)。そこで、テキスト (p.76) の後半に記載した、よ

り少ない計算量で精度の高い値を求めるアルゴリズムを採用したプログラム「5の平方根を小数点以下第14位まで求めるプログラム」(ファイル名「**exercise0210.c**」)を作成しなさい。

注意：データ型 **double** の仮数部の有効桁数は10進数で約15桁である。

● 5の平方根を小数点以下第6位まで求めるプログラム

exercise0209.c

```
1: #include <stdio.h>
2:
3: int main(void)
4: {
5:     double m, n = 0, s = 0;
6:
7:     while (s <= 50000000000000) {
8:         n = n + 1;
9:         m = 2 * n - 1;
10:        s = s + m;
11:    }
12:
13:    n = n - 1;
14:    n = n / 1000000;
15:
16:    printf("sqrt(5)=%16.14f\n", n);      ← ■ p.152 の精度指定子を参照
17:
18:    return 0;
19: }
```

実行結果：

```
$ gcc -o exercise0209.exe exercise0209.c
exercise0209.c: In function 'main':
exercise0209.c:7: warning: integer constant is too large for 'long' type

$ ./exercise0209.exe
sqrt(5)=2.236067000000000

$
```

実行結果：

```
$ ./exercise0210.exe
sqrt(5)=2.23606797749978

$
```