

## 第04回の演習課題

### 演習課題 ex04-3.c

*double* 型変数  $x$  と *float* 型変数  $y$  を両方共 1.0 で初期化し、この2つの変数  $x$  と  $y$  の値が等しいあいだ  $x$  と  $y$  の値を共に2倍にしながら双方の値を出力することを繰り返すプログラム

**ex04-3.c** を作成し、kiso2コマンドを用いて提出しなさい。

実行例：

```
t180900@s01cd0542-160:~/kiso2-2018/ex04$ ./ex04-3
x=2.000000
y=2.000000
x=4.000000
y=4.000000
x=8.000000
y=8.000000
x=16.000000
y=16.000000
x=32.000000
y=32.000000
x=64.000000
y=64.000000
x=128.000000
y=128.000000
:
```

### 演習課題 ex04-4.c

ASCII文字コード表をすべて出力するため、次のようなプログラムを考える。 *char* 型変数  $c$  を 0 で初期化し、 $c$  が 127 未満である間、 $c$  の値を1ずつ増やしながら (加算しながら  $c$  の値を *int* 型の整数値と *char* 型の文字として `printf()` 関数で出力する。このようなプログラム **ex04-4.c** を作成し、kiso2コマンドを用いて提出しなさい。

実行例：

```
t180900@s01cd0542-160:~/kiso2-2018/ex04$ ./ex04-4
0 =>
1 =>
2 =>
3 =>
4 =>
5 =>
6 =>
7 =>
8 =>
9 =>
10 =>
```

```
11 =>
:
119 => w
120 => x
121 => y
122 => z
123 => {
124 => |
125 => }
126 => ~
```

ASCIIコードには、`printf()` 関数で出力したときに端末に表示できる文字（印字可能文字と呼ばれます）以外の文字コードが含まれています。そのような文字コードを `printf()` 関数で出力しても、端末には何も表示されません。また、文字コードには、改行 '\n' やタブ '\t' などの制御文字コードも含まれていません。

## 演習課題 ex04-5.c

演習 [ex03-6.c](#) では、`int` 型の変数と `double` 型の変数が保持できる値の範囲を `limits.h` や `float.h` をインクルードすることで参照できることを確認した。同様に、整数型変数 `char`, `short`, `int`, `long`, `long long` が表現できる最大、最小の値と、実数型変数 `float`, `double`, `long double` が表現できる最大、最小(もっとも0に近い)の値を `printf()` 関数で出力するプログラム [ex04-5.c](#) を作成し、[kiso2コマンドを用いて提出しなさい。](#)

実行例：

```
t180900@s01cd0542-160:~/kiso2-2018/ex04$ ./ex04-5
char型の最大値は 127
char型の最小値は -128
short型の最大値は 32767
short型の最小値は -32768
int型の最大値は 2147483647
int型の最小値は -2147483648
long型の最小値は -9223372036854775808
long型の最大値は 9223372036854775807
:
```

## 演習課題 ex04-6.c

`int` 型の変数 `sec` に秒数を0位上の値として入力すると、秒、分、時間の単位で値を出力するプログラム [ex04-6.c](#) を作成し、[kiso2コマンドを用いて提出しなさい。](#) ただし、出力する値は秒、分、時間の値をそれぞれ `$s`, `m`, `h$` とすると `0 \le s \le 59, 0 \le m \le 59, 0 \le h` の範囲とし、最も大きな時間単位の値が0である場合は値を出力しなくても（あるいは出力しても）良い。

実行例：

```
t180900@s01cd0542-160:~/kiso2-2018/ex04$ ./ex04-6
sec? 0
t180900@s01cd0542-160:~/kiso2-2018/ex04$ ./ex04-6
```

```
sec? 1
1
t180900@s01cd0542-160:~/kiso2-2018/ex04$ ./ex04-6
sec? 59
59
t180900@s01cd0542-160:~/kiso2-2018/ex04$ ./ex04-6
sec? 60
0
1
t180900@s01cd0542-160:~/kiso2-2018/ex04$ ./ex04-6
sec? 119
59
1
t180900@s01cd0542-160:~/kiso2-2018/ex04$ ./ex04-6
sec? 3599
59
59
t180900@s01cd0542-160:~/kiso2-2018/ex04$ ./ex04-6
sec? 3600
0
0
1
t180900@s01cd0542-160:~/kiso2-2018/ex04$ ./ex04-6
sec? 123456
36
17
34
```

ex04-6.c のプログラムソースは、第4回までの講義内容（while 文での繰り返し）のみで書くことが可能ですが、必要に応じて他のC言語の制御構造（if 文など）を使っても構いません。余裕のある人は、この講義ですでに触れている内容だけでプログラムを作成してみてください。

From:

<https://www-slab.math.ryukoku.ac.jp/> - **www-slab.math**

Permanent link:

<https://www-slab.math.ryukoku.ac.jp/lecture/kiso2/ex04>



Last update: **2018/10/11 17:06**