

第07回の演習課題

ex07-3.c

2つの整数 n , $skip$ の値を入力すると1から n まで、 $skip$ ごとの値に対する掛け算九九表を出力するプログラム **ex07-3.c** を作成し、`kiso2` コマンドを用いて提出しなさい。ただし、表は乗数 被乗数と積との対応関係が区別できるように、それぞれの値を適当な固定文字幅で出力する。また、入力された n , $skip$ のいずれかが1未満の値の場合は、何も出力せずに終了するものとする。

実行例：

```
t190900@s01cd0542-160:~/kiso2-2019/ex07$ ./ex07-3<code>
n skip? 0 1
t190900@s01cd0542-160:~/kiso2-2019/ex07$ ./ex07-3<code>
n skip? 1 0
t190900@s01cd0542-160:~/kiso2-2019/ex07$ ./ex07-3<code>
n skip? 1 1
      1
    1 1
t190900@s01cd0542-160:~/kiso2-2019/ex07$ ./ex07-3<code>
n skip? 9 1
      1  2  3  4  5  6  7  8  9
    1  1  2  3  4  5  6  7  8  9
    2  2  4  6  8 10 12 14 16 18
    3  3  6  9 12 15 18 21 24 27
    4  4  8 12 16 20 24 28 32 36
    5  5 10 15 20 25 30 35 40 45
    6  6 12 18 24 30 36 42 48 54
    7  7 14 21 28 35 42 49 56 63
    8  8 16 24 32 40 48 56 64 72
    9  9 18 27 36 45 54 63 72 81
t190900@s01cd0542-160:~/kiso2-2019/ex07$ ./ex07-3<code>
n skip? 10 2
      1  3  5  7  9
    1  1  3  5  7  9
    3  3  9 15 21 27
    5  5 15 25 35 45
    7  7 21 35 49 63
    9  9 27 45 63 81
t190900@s01cd0542-160:~/kiso2-2019/ex07$ ./ex07-3<code>
n skip? 20 3
      1  4  7 10 13 16 19
    1  1  4  7 10 13 16 19
    4  4 16 28 40 52 64 76
    7  7 28 49 70 91 112 133
   10 10 40 70 100 130 160 190
```

13	13	52	91	130	169	208	247
16	16	64	112	160	208	256	304
19	19	76	133	190	247	304	361

演習課題 ex07-4.c

2つの整数 *from*, *to* の値を入力すると、角度 *x* に対して *from* 度から *to* 度の範囲で10度刻みに $y = \sin(x)$ のグラフをテキスト表示するプログラム **ex07-4.c** を作成し、[kiso2コマンド](#)を用いて提出しなさい。ただし、グラフの *x* 軸を端末の縦（列）方向、*y* 軸を横（行）方向に表示し、 $y = \sin(x)$ の値を示すグラフを “*” で、 $y = -1, 0, 1$ 位置に軸線を “|” で表示し、“*” と “|” が同じ位置に表示すべき場合には “*” のみが出力されるものとする。また、このプログラムは、*from*, *to* の入力とグラフの出力を、*from* < *to* である限り何度も繰り返すものとする。

実行例：

```
t190900@s01cd0542-160:~/kiso2-2019/ex07$ ./ex07-4
from? 0
to? 100
  -1          0          1
  +-----+-----+
  0 |          *          | 0.000000
  10 |          | *        | 0.173648
  20 |          |  *      | 0.342020
  30 |          |   *    | 0.500000
  40 |          |    *   | 0.642788
  50 |          |     *  | 0.766044
  60 |          |      * | 0.866025
  70 |          |       *| 0.939693
  80 |          |        *| 0.984808
  90 |          |         *| 1.000000
 100 |          |          *| 0.984808
from? -180
to? 180
  -1          0          1
  +-----+-----+
-180 |          *          | -0.000000
-170 |          | *        | -0.173648
-160 |          |  *      | -0.342020
-150 |          |   *    | -0.500000
-140 |          |    *   | -0.642788
-130 |          |     *  | -0.766044
-120 |          |      * | -0.866025
-110 |          |       *| -0.939693
-100 |          |        *| -0.984808
 -90 |          |         *| -1.000000
 -80 |          |          *| -0.984808
 -70 |          |           *| -0.939693
 -60 |          |            *| -0.866025
 -50 |          |             *| -0.766044
 -40 |          |              *| -0.642788
```

```

-30 |          *          |          | -0.500000
-20 |           *        |          | -0.342020
-10 |            *       |          | -0.173648
  0 |             *      |          |  0.000000
 10 |              *     |          |  0.173648
 20 |               *    |          |  0.342020
 30 |                *   |          |  0.500000
 40 |                 *  |          |  0.642788
 50 |                  * |          |  0.766044
 60 |                   *|          |  0.866025
 70 |                    *|          |  0.939693
 80 |                     *|          |  0.984808
 90 |                      *|          |  1.000000
100 |                       *|          |  0.984808
110 |                        *|          |  0.939693
120 |                         *|          |  0.866025
130 |                          *|          |  0.766044
140 |                           *|          |  0.642788
150 |                            *|          |  0.500000
160 |                             *|          |  0.342020
170 |                              *|          |  0.173648
180 |                               *|          |  0.000000
from? 90
to? 90

```

`printf()` 関数の変換指定子は、その変換指定子で表示する数値などの文字幅を % 直後の数字で指定することができます。たとえば、`%10d` は整数値を右寄せ10文字幅で表示し、文字幅を超えた部分は空白文字が表示されます。より具体的には、`a=10` を `%10d` で表示すると “10” (空白8文字 + ”10” で合計10文字)が表示されます。ただし、もし表示する数値などが文字幅を超えた場合は、`%d` が指定されたときと同じように(左寄せに必要な文字幅ぴったり)になります。

実数値を表示するための `%f` についても、`%10f` (10文字幅)や、`%10.2f` (10文字幅で小数点以下2桁) のように文字幅を指定することができます。ただし、桁数ではなく、あくまで文字幅を指定するものなので、小数点も1文字と数えられますので注意が必要です。

From:
<https://133.83.80.10/> - **www-slab.math**

Permanent link:
<https://133.83.80.10/lecture/kiso2/2018/ex07>

Last update: **2019/09/23 13:24**

