

## 第12回の演習課題

### ex12-2.c

自然数  $n$  の値を入力し、つづけて  $n$  次元ベクトルの要素を実数値として繰り返し入力すると、それまでに入力された  $n$  次元ベクトルの和を出力するプログラム **ex12-2.c** を作成し、`kiso2` コマンドを用いて提出しなさい。ただし、入力された  $n$  の値が自然数でない場合は再び  $n$  の入力を行うものとし、また、入力された  $n$  次元ベクトルが0ベクトルであればプログラムを終了するものとする。

実行例：

```
t190900@s01cd0542-160:~/kiso2-2019/ex12$ ./ex12-2
n? 0
n? 1
elements? 1
(1.000000)
elements? 2.3
(3.300000)
elements? 3.4
(6.700000)
elements? 0
t190900@s01cd0542-160:~/kiso2-2019/ex12$ ./ex12-2
n? 3
elements? 1 2 3
(1.000000, 2.000000, 3.000000)
elements? 1.2 -2.3 -5.6
(2.200000, -0.300000, -2.600000)
elements? -93.1 23.6 54.12
(-90.900000, 23.300000, 51.520000)
elements? 0 0 1
(-90.900000, 23.300000, 52.520000)
elements? 0 1 0
(-90.900000, 24.300000, 52.520000)
elements? 1 0 0
(-89.900000, 24.300000, 52.520000)
elements? 0 0 0.0
```

実習室のC言語環境では、配列変数の宣言時のサイズ指定に変数を用いることが可能です。

### ex12-3.c

関数 `cnt()` は、`int` 型のグローバル変数 `count` を用いて次のように定義されている。また `<stdlib.h>` で宣言されるC言語の標準ライブラリの関数 `rand()` は、返り値として `0` から `RAND_MAX` の範囲の整数値を疑似乱数として返す関数である。

関数 `cnt()` を（変更せずそのまま）利用し、整数  $n$  の値を入力すると `cnt(n)` の値が（疑似乱数であ

`rand()` の戻り値以下の場合には `cnt()` を繰り返し呼び出すようなプログラム **ex12-3.c** を作成し、`kiso2` コマンドを用いて提出しなさい。ただし `n` が `rand()` の値を超えた場合は、グローバル変数 `count` の値を、`cnt()` 関数が呼び出された回数として出力してプログラムを終了するものとする。

`cnt.c`

```
void cnt(void) { count++; }
```

実行例：

```
t190900@s01cd0542-160:~/kiso2-2019/ex12$ ./ex12-3
n? 999999999
cnt() は 1 回呼び出されました。
t190900@s01cd0542-160:~/kiso2-2019/ex12$ ./ex12-3
n? 9999999
cnt() は 172 回呼び出されました。
t190900@s01cd0542-160:~/kiso2-2019/ex12$ ./ex12-3
n? 99999
cnt() は 19964 回呼び出されました。
t190900@s01cd0542-160:~/kiso2-2019/ex12$ ./ex12-3
n? 999
cnt() は 2088410 回呼び出されました。
t190900@s01cd0542-160:~/kiso2-2019/ex12$ ./ex12-3
n? 9
cnt() は 91538659 回呼び出されました。
```

`rand()` は擬似乱数を生成する関数であり、呼び出される度に異なる整数値を乱数として返します。したがって、上記の実行例と必ずしも同じ結果になるとは限りません。ただし、実習室環境ではおそらく同じ回数が表示されていると思います。異なる乱数列を生成したい場合は、`srand()` 関数を利用します。

## ex12-4.c

テント写像  $f(x)$  は、以下のように定義される区分線形写像である  $f(x) = \begin{cases} 2x, & x < \frac{1}{2}, \\ 2(1-x), & \frac{1}{2} \leq x, \end{cases}$  1つの実数を引数としてもち、テント写像  $f(x)$  にしたがって実数値を返す関数 `tent()` を定義しなさい。

定義された関数 `tent()` を用いて、初期値  $x_0$  の値を実数として  $0 \leq x \leq 1$  の範囲で与えると  $x_1=f(x_0)$ ,  $x_2=f(x_1)$ ,  $x_3=f(x_2)$ ,  $\dots$  によって決まる  $x_i$ , ( $i=0, 1, 2, \dots, 19$ ) の値を小数点以下20桁まで出力するプログラム **ex12-4.c** を作成し、`kiso2` コマンドを用いて提出しなさい。ただし、入力された  $x_0$  の値が  $0 \leq x \leq 1$  の範囲にない場合は再び  $x_0$  の入力を行うものとする。

実行例：

```
t190900@s01cd0542-160:~/kiso2-2019/ex12$ ./ex12-4
x0? -0.1
x0? 1.01
```

```
x0? 0.12
x0: 0.11999999999999999556
x1: 0.23999999999999999112
x2: 0.47999999999999998224
x3: 0.95999999999999996447
x4: 0.080000000000000007105
x5: 0.1600000000000000014211
x6: 0.3200000000000000028422
x7: 0.6400000000000000056843
x8: 0.719999999999999986313
x9: 0.5600000000000000227374
x10: 0.8799999999999999545253
x11: 0.24000000000000000909495
x12: 0.48000000000000001818989
x13: 0.96000000000000003637979
x14: 0.07999999999999992724042
x15: 0.15999999999999995448085
x16: 0.319999999999999970896170
x17: 0.639999999999999941792339
x18: 0.7200000000000000116415322
x19: 0.5599999999999999767169356
t190900@s01cd0542-160:~/kiso2-2019/ex12$ ./ex12-4
x0? 0.12000001
x0: 0.120000010000000000417
x1: 0.240000020000000000835
x2: 0.480000040000000001669
x3: 0.960000080000000003339
x4: 0.07999983999999993323
x5: 0.15999967999999986645
x6: 0.31999935999999973291
x7: 0.63999871999999946581
x8: 0.72000256000000106837
x9: 0.55999487999999786325
x10: 0.88001024000000427350
x11: 0.23997951999999145301
x12: 0.47995903999998290601
x13: 0.95991807999996581202
x14: 0.08016384000006837596
x15: 0.16032768000013675191
x16: 0.32065536000027350383
x17: 0.64131072000054700766
x18: 0.71737855999890598468
x19: 0.56524288000218803063
```

### ex12-5.c

グローバル変数を使わずに ex12-3.c と同じ動作を実現するため、関数 cnt() を次のようにプロトタイプ宣言した。関数 cnt() をプロトタイプ宣言にしたがって定義し、ex12-3.c と同じ入出力を行うプログラム **ex12-5.c** を作成して kiso2 コマンドを用いて提出しなさい。

cnt.c

```
int cnt(int);
```

実行例 :

```
t190900@s01cd0542-160:~/kiso2-2019/ex12$ ./ex12-5
n? 999999999
cnt() は 1 回呼び出されました。
t190900@s01cd0542-160:~/kiso2-2019/ex12$ ./ex12-5
n? 999999999
cnt() は 20 回呼び出されました。
t190900@s01cd0542-160:~/kiso2-2019/ex12$ ./ex12-5
n? 999999999
cnt() は 172 回呼び出されました。
t190900@s01cd0542-160:~/kiso2-2019/ex12$ ./ex12-5
n? 99
cnt() は 4880726 回呼び出されました。
t190900@s01cd0542-160:~/kiso2-2019/ex12$ ./ex12-5
n? 9
cnt() は 91538659 回呼び出されました。
```

## ex12-6.c

大きさ20の整数型の配列変数 `v[]` が以下のように初期化されている。キーボードから整数 `s` の値を入力すると、配列 `v[]` の要素に `s` が含まれていればその要素番号をすべて出力し、また `s` が含まれていなければ `s` がない旨を出力するプログラム **ex12-6.c** を作成して `kiso2` コマンドを用いて提出しなさい。

```
int v[] = {77, 37, 29, 54, 73, 64, 43, 53, 5, 52, 59, 49, 2, 93, 9, 20, 52, 75, 34, 99};
```

実行例 :

```
t190900@s01cd0542-160:~/kiso2-2019/ex12$ ./ex12-6
s? 0
0 は見つかりませんでした
t190900@s01cd0542-160:~/kiso2-2019/ex12$ ./ex12-6
s? 100
100 は見つかりませんでした
t190900@s01cd0542-160:~/kiso2-2019/ex12$ ./ex12-6
s? 2
2 は 12 番目に見つかりました
t190900@s01cd0542-160:~/kiso2-2019/ex12$ ./ex12-6
s? 52
52 は 9 番目に見つかりました
52 は 16 番目に見つかりました
t190900@s01cd0542-160:~/kiso2-2019/ex12$ ./ex12-6
s? 99
99 は 19 番目に見つかりました
```

### ex12-7.c

1以上の整数の値 n を入力すると n 列までのパスカルの三角形を表示するプログラム **ex12-7.c** を作成してkiso2コマンドを用いて提出しなさい。ただし n に1未満の値が入力された場合は、再度 n の入力を行うものとする。

実行例：

```

t190900@s01cd0542-160:~/kiso2-2019/ex12$ ./ex12-7
n? 0
n? 1
    1
t190900@s01cd0542-160:~/kiso2-2019/ex12$ ./ex12-7
n? 2
    1
   1 1
t190900@s01cd0542-160:~/kiso2-2019/ex12$ ./ex12-7
n? 10
           1
          1 1
         1 2 1
        1 3 3 1
       1 4 6 4 1
      1 5 10 10 5 1
     1 6 15 20 15 6 1
    1 7 21 35 35 21 7 1
   1 8 28 56 70 56 28 8 1
  1 9 36 84 126 126 84 36 9 1
t190900@s01cd0542-160:~/kiso2-2019/ex12$ ./ex12-7
n? 20
           1
          1 1
         1 2 1
        1 3 3 1
       1 4 6 4 1
      1 5 10 10 5 1
     1 6 15 20 15 6 1
    1 7 21 35 35 21 7 1
   1 8 28 56 70 56 28 8 1
  1 9 36 84 126 126 84 36 9 1
 10 45 10 1 1
 120 45 10 1
 165 55 11 1
           1 11 55 165 330 462 462 330
          1 12 66 220 495 792 924 792
         1 12 66 220 495 792 924 792
        1 12 66 220 495 792 924 792
       1 12 66 220 495 792 924 792
      1 12 66 220 495 792 924 792
     1 12 66 220 495 792 924 792
    1 12 66 220 495 792 924 792
   1 12 66 220 495 792 924 792
  1 12 66 220 495 792 924 792
 1 12 66 220 495 792 924 792

```

