

第11回の演習課題

ex11-3.c

関数 `sqsum()` は、2つの `double` 型の値 `a`, `b` を引数として受け取ると $a^2 + b^2$ の値を `double` 型として返す関数である。関数 `sqsum()` を定義しなさい。また、2つの実数値 `x`, `y` を入力すると、この `sqsum()` を用いて $\sqrt{x^2+y^2}$ の値を計算し出力するプログラム **ex11-3.c** を作成し、`kiso2` コマンドを用いて提出しなさい。ただし、実数値 `x`, `y` の入力と計算結果の出力は繰り返し行えるものとし、`x`, `y` 両方に 0 が入力されるとプログラムは「終了します」と出力してから終了するものとする。

実行例：

```
t180900@s01cd0542-160:~/kiso2-2018/ex11$ ./ex11-3
x y? 1 1
1.414214
x y? -1 -1
1.414214
x y? 2 3
3.605551
x y? 0 9
9.000000
x y? 9 0
9.000000
x y? 12.7 3.1
13.072873
x y? 0 0
終了します。
```

ex10-8.c で定義した関数 `pexit()` が再利用できます。

ex11-4.c

関数 `absum()` が以下のように定義されている。関数 `absum()` を（変更せずそのまま）利用し、2つの整数 `a`, `b` の値を入力すると $\sum_{k=|a|+1}^{|b|} k$ の値を出力するプログラム **ex11-4.c** を作成し、`kiso2` コマンドを用いて提出しなさい。ただし $|a| < |b|$ とし、入力された値がこれを満たさない場合は、再度 `a`, `b` を入力するものとする。

[absum.c](#)

```
int absum(int n) {
    int i;
    int sum = 0;

    for (i=1; i<=abs(n); i++)
        sum += i;
    return sum;
}
```

実行例：

```
t180900@s01cd0542-160:~/kiso2-2018/ex11$ ./ex11-4
a b? 0 0
a b? -1 0
a b? -1 -1
a b? 0 -1
1
t180900@s01cd0542-160:~/kiso2-2018/ex11$ ./ex11-4
a b? 9 -10
10
t180900@s01cd0542-160:~/kiso2-2018/ex11$ ./ex11-4
a b? -9 10
10
t180900@s01cd0542-160:~/kiso2-2018/ex11$ ./ex11-4
a b? 123 -432
85902
```

関数 `abs()` の中で使われている (呼び出されている) 関数 `abs()` は、`stdlib.h` で宣言されている C 言語の標準関数です。

ex11-5.c

関数 `trcase()` は、1つの `char` 型の値を `c` を引数として受け取り、`c` が大文字アルファベット (A-Z) であれば小文字アルファベット (a-z) に変換し、小文字アルファベットであれば大文字アルファベットに変換して `char` 型の戻り値として返す関数である。ただし `trcase()` は、引数 `c` がアルファベットでなければ `c` をそのまま返す。関数 `trcase()` を定義しなさい。また、この関数 `trcase()` を用いて、キーボードから 1 文字を入力すると `trcase()` に従ってアルファベット文字の変換を行うプログラム **ex11-5.c** を作成し、[kiso2](#) コマンドを用いて提出しなさい。 ただし、アルファベットの変換はプログラムが強制終了されるまで繰り返し行うものとし、入力された文字がアルファベット以外である場合は何も出力しないものとする。

実行例 `^C` は Control+C で強制終了)

```
t180900@s01cd0542-160:~/kiso2-2018/ex11$ ./ex11-5
a
a => A
b
b => B
c
c => C
Z
Z => Z
A
A => a
B
B => b
Z
Z => z
[
]
1
2
```

```
3
^C
```

文字型の扱いは第4回の内容です。アルファベット文字とchar整数型との対応は、ASCII文字コードにしたがっています。

ex11-6.c

関数 `max()` は、2つの double型の値 `a`, `b` を引数として受け取り、そのうちの大きい方の値を返す double型の関数である。関数 `max4()` は、4つの double型の値 `a`, `b`, `c`, `d` を引数として受け取り、そのうちのもっとも大きい値を返す double型の関数である。関数 `max()`, `max4()` を定義しなさい。これらの関数を用いて、4つの実数値を入力するとその最大値を出力することを繰り返すプログラム **ex11-6.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4つの実数値がすべて等しい場合、プログラムは最大値を出力した後に終了するものとする。

実行例：

```
t180900@s01cd0542-160:~/kiso2-2018/ex11$ ./ex11-6
a b c d? 1 2 3 4
max: 4.000000
a b c d? 4.1 4 4 4
max: 4.100000
a b c d? -12.3 4.5 67.8 9.0
max: 67.800000
a b c d? 1 1 1 1.1
max: 1.100000
a b c d? 1 1 1 1
max: 1.000000
```

ex11-7.c

関数 `fact()` が以下のように定義されている。関数 `fact()` を（変更せずそのまま）利用し、キーボードから自然数 `n` の値を入力すると $n!$ (n の階乗) の値を出力するプログラム **ex11-7.c** を作成し、kiso2コマンドを用いて提出しなさい。ただし n に 0 未満の値が入力された場合は、プログラムは何も出力せずに終了するものとする。

fact.c

```
int fact(int n) {
    if (n == 0)
        return 1;           // 0! = 1
    else
        return n*fact(n-1); // n! = n * (n-1)!
}
```

実行例：

```
t180900@s01cd0542-160:~/kiso2-2018/ex11$ ./ex11-7
n? -1
```

```
t180900@s01cd0542-160:~/kiso2-2018/ex11$ ./ex11-7
n? 0
0! = 1
t180900@s01cd0542-160:~/kiso2-2018/ex11$ ./ex11-7
n? 1
1! = 1
t180900@s01cd0542-160:~/kiso2-2018/ex11$ ./ex11-7
n? 2
2! = 2
t180900@s01cd0542-160:~/kiso2-2018/ex11$ ./ex11-7
n? 3
3! = 6
t180900@s01cd0542-160:~/kiso2-2018/ex11$ ./ex11-7
n? 10
10! = 3628800
t180900@s01cd0542-160:~/kiso2-2018/ex11$ ./ex11-7
n? 19
19! = 109641728
```

ソースファイルで定義されたC言語の関数は、その関数が呼び出されると定義に基づいた関数の実体（コピー）がメモリ上に作成 処理され、またその関数呼び出しが終了するとその実体もメモリ上から開放（消去）されます。したがって、定義された関数は、プログラム上から（異なる実体として）複数同時に呼び出すことが可能です。そのため、この演習問題のようにその関数の中で自分自身を呼び出すこともできます。このような入れ子の関数呼び出しを「再帰呼び出し」といいます。

From:

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

Permanent link:

<https://www-slab.math.ryukoku.ac.jp/lecture/kiso2/ex11>



Last update: **2018/12/06 16:39**