

## 第03回の基本課題

### 演習課題 ex03-1.c

1つの整数値をキーボードから入力し、その値を2倍した値を return 文で返すプログラム **ex03-1.c** を作成し、kiso2コマンドを用いて提出しなさい。

実行例：

```
t190900@s01cd0542-160:~/kiso2-2019/ex03$ ./ex03-1
1
t190900@s01cd0542-160:~/kiso2-2019/ex03$ echo $?
2
t190900@s01cd0542-160:~/kiso2-2019/ex03$ ./ex03-1
0
t190900@s01cd0542-160:~/kiso2-2019/ex03$ echo $?
0
t190900@s01cd0542-160:~/kiso2-2019/ex03$ ./ex03-1
9
t190900@s01cd0542-160:~/kiso2-2019/ex03$ echo $?
18
t190900@s01cd0542-160:~/kiso2-2019/ex03$ ./ex03-1
-1
t190900@s01cd0542-160:~/kiso2-2019/ex03$ echo $?
254
```

`main()` 関数の戻り値が負になった場合、シェル(端末)上では負の戻り値を正しく表示することができません。これはC言語が負の整数を2の補数という表現で取り扱っているためです。2の補数については、1年後期の「情報処理の基礎」で取り扱っています。実行例で 254 が表示される理由(正しくは -2)は、`main()` 関数が2の補数表現で -2 を返す一方で、シェルは \$? に保持された値を8ビットの正の整数(0~255)として表現するためです。

### 演習課題 ex03-2.c

次のような処理を行うプログラム **ex03-2.c** を作成し、kiso2コマンドを用いて提出しなさい。

1. 1つの int 型変数  $i$  を定義し、その値を1に初期化する。
2.  $i$  の値が 0 よりも大きい間、次の処理を繰り返す。
  1.  $i$  の値を2倍にする。
  2.  $i$  の値を `printf()` 関数で出力(表示)する。
3.  $i$  の値が 0 以下となって 2. の処理が終了したら `return` 文で 1 を返す。

実行例：

```
t190900@s01cd0542-160:~/kiso2-2019/ex03$ ./ex03-2
2
4
8
16
32
```

```
64
128
256
512
1024
2048
4096
8192
16384
32768
65536
131072
262144
524288
1048576
2097152
4194304
8388608
16777216
33554432
67108864
134217728
268435456
536870912
1073741824
-2147483648
```

幾つかの処理を「ある条件を満たすあいだ」繰り返したい場合、一般的には *while* 文を使います。前期の計算機基礎実習Iでは、[第7回](#) に学修した内容です。

[第03回演習課題へ](#)

From:

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

Permanent link:

<https://www-slab.math.ryukoku.ac.jp/lecture/kiso2/2018/base03>



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