

## OCaml (あるいは F#) のプログラミングに関する課題

2015.4.23

この授業は、OCaml/F# などの関数型プログラミング言語を習う授業ではなく、これらを道具として使って、一般のプログラミング言語の様々な機能を記述します。

したがって、OCaml/F# の全ての機能 (素晴らしい機能) を網羅するつもりはまったくなく、ここで使うのは、「OCaml/F# のなるべく小さなサブセット (他のプログラム言語の処理を記述するのに必要なだけの最小限のサブセット)」です。

この授業で、これらの言語に興味を持った人は、インターネットあるいは市販参考書を見て自分でプログラムを書いてみるとよいでしょう。

- 課題 1. ウェブページのリンク先の OCaml プログラムをすべて実行し、コメントや計算結果を参考に意味を理解せよ。必要ならば、プログラムの一部を改変して実行してみなさい。(この部分は、レポートとしての提出は不要です。ただし、必ず自分の手でやっておくことが必要です。)

- 課題 2 (OCaml を初めてさわったという人向け)

以下の関数を書きなさい。入力は一正の整数 1 つ。出力はその (正の) 約数の個数。

たとえば、28 を入力すると (約数は 1,2,4,7,14,28 なので) 6 が返り、13 を入力すると (約数は 1,13 なので) 2 が返る。なお、厳密に言えば -2 や -7 も 28 の約数であるが、ここでは、正の約数の個数のみを数えることにする。

なお、このような関数は、`int -> int` という型をもつはずである。

- 課題 2' (OCaml を初めてさわったという人向け; 課題 2 より少し簡単)

以下の関数を書きなさい。入力は一正の整数 1 つ。出力は、それが、2 で何回割り切れるか、という回数。

たとえば、56 を入力すると ( $56 = 2^3 * 7$  なので) 3 が返り、13 を入力すると (13 は 2 で割り切れないので) 0 が返る。

- 課題 3 (OCaml 経験者向け)

2次元の正方行列で要素がすべて整数であるものを「整数のリストのリスト」として表現する。たとえば、`[[1;2;3];[4;5;6];[7;8;9]]` は 3\*3 行列である。このような行列を 2つ入力としてもらい、それら 2つの行列の積を返す関数を定義せよ。(返すものも行列である。

なお、このような関数は、`(int list list) * (int list list) -> (int list list)` という型をもつはずである。

- 課題 4 (上のものでは物足りない人だけにに向けた optional 課題)

行列に対して、逆行列を返す関数を書いてください。

提出先: Manaba システム, 提出締切: 来週月曜深夜

上記の課題 2 か 課題 3 のどちらか一方を書いて提出してください。課題 4 もできた人は、それに追加して (1つのレポートとして) 出してください。