

OCaml (あるいは F#) のプログラミングに関する課題 の略解
2015.4.23

- 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 n 。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。

答えの一例

```
let rec f n m =
  if m = 0 then 0
  else if (n mod m = 0) then
    (f n (m-1)) + 1
  else
    f n (m-1)
let num_of_div n =
  f n n
```

- 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 n 。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。

答えの一例

```
let rec factor n =
  if (n mod 2 > 0) then 0
  else
    (factor (n / 2)) + 1
```

- 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。

答えの一例

```
let rec nth vec i =
  match vec with
  | [ ] -> failwith "length of vector is too short"
```

```

    | v :: vec2 -> if i = 0 then v else nth vec2 (i - 1)
let array_ref ary i j =
    nth (nth a i) j
let rec vlength vec =
    match vec with
    | [ ] -> 0
    | v :: vec2 -> 1 + (vlength vec2)
let rec inner_product vec1 vec2 =
    match (vec1,vec2) with
    | ([],_) -> 0
    | (v1::vec1tail, v2::vec2tail)
        -> v1 * v2 + (visit vec1tail vec2tail)
    | _ -> failwith "length mismatch error "
let get_row ary i = nth ary i
let rec get_column ary j = ...
let matmul a b =
    let len = vlength a in
    for i = 0 to len -1 do
    for j = 0 to len -1 do
        inner_product
            (get_row ary i)
            (get_column ary j)
    ...

```

- 課題 4 (上のものでは物足りない人だけに向けた 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。

答えは省略