ソフトウェア技法　第 2 回　201311367　佐藤良祐

課題 2.1

```
(* 実数のリストの要素の和を計算する関数 *)
let rec list_sums lst =
  match lst with
    [] -> 0.
  | x :: rest -> x +. list_sums rest;;

(* リストの長さを返す *)
let rec length lst =
  match lst with
    [] -> 0
  | _ :: rest -> 1 + length rest;;

(* 実数のリストの要素の平均値を返す *)
let rec average lst =
  let sum = list_sums lst in
  sum /. float_of_int (length lst);;
```

実行結果

```
# list_sums [1.; 2.; 3.; 4.; 5.];;
- : float = 15.
# average [1.; 2.; 3.; 4.; 5.];;
- : float = 3.
# let lst = [-1.1; -2.2; -3.; 5.; 10.];;
val lst : float list = [-1.1; -2.2; -3.; 5.; 10.]
# list_sums lst;;
- : float = 8.70000000000000107
# average lst;;
- : float = 1.74000000000000021
```

課題 2.2

```
(* リストの先頭n 個の要素を取り除いたリストを返す *)
let rec drop (lst, n) =
  match lst with
    [] -> []
  | x :: rest -> if n = 0 then lst else drop(rest, n-1);;
```

実行結果

```
# drop ([0; 1; 2; 3; 4;], 1);;
drop ([0; 1; 2; 3; 4;], 1);;
- : int list = [1; 2; 3; 4]
# drop ([0; 1; 2; 3; 4;], 3);;
drop ([0; 1; 2; 3; 4;], 3);;
- : int list = [3; 4]
# drop ([0; 1; 2; 3; 4;], 5);;
drop ([0; 1; 2; 3; 4;], 5);;
- : int list = []
# drop ([0; 1; 2; 3; 4;], 6);;
drop ([0; 1; 2; 3; 4;], 6);;
- : int list = []
```

課題 2.3

```
(* 整数のリストが与えられたとき,非負の要素と負の要素に分割する *)
let rec split_intlist lst =
  match lst with
    [] -> ([], [])
  | x :: rest ->
     let (gtez, ltz) = split_intlist rest in
     if x < 0 then (gtez, x :: ltz)
     else (x :: gtez, ltz);;
```

実行結果

```
# split_intlist [-1; 0; 10; 5; -3];;
- : int list * int list = ([0; 10; 5], [-1; -3])
```

課題 2.4

```
(* リストのリスト [[]]を連結する *)
let rec flatten lst =
  match lst with
    [] -> []
  | x :: rest -> x @ flatten rest;;
```

実行結果

```
flatten [[1; 2]; [3; 4]];;
- : int list = [1; 2; 3; 4]
# flatten [[1]; []; [0; 2]; [-1]];;
- : int list = [1; 0; 2; -1]
```