

Programmation fonctionnelle TP 5

antoine.frenoy@inserm.fr
<http://perso.crans.org/frenoy/caml2013.html>

Jeudi 21 février 2013

Types et fonctions, encore une fois

Pour chacune des fonctions suivantes, donnez son type sans utiliser ocaml, puis vérifiez avec ocaml :

1. `let f1 (x,y) = (x (y+3))>1;;`
2. `let f2 l = 3::l;;`
3. `let f3 u = [u;5,2];;`
4. `let f4 l1 l2 = match l1,l2 with
|a::t1,b::t2->if (a<b) then List.append t1 t2 else List.append t2 t1
|_->[];;"`
5. `let f5 x = match x with |[]->0 |h::[] ->h 0 |h1::h2::t-> h1 (h2 0);;`

Pour chacun des types suivants, donnez une fonction de ce type puis vérifiez avec ocaml :

1. `'a -> 'a list`
2. `(int -> 'a) list -> 'a list`

Travailler sur les listes

1. Écrire une fonction prenant en argument une liste `l` de listes et indiquant si tous les éléments de `l` ont au moins 4 éléments.
2. Écrire une fonction prenant en argument une liste `l` de listes et triant les éléments de `l` selon leur nombre d'éléments.
3. Écrire une fonction prenant en argument une liste `l` de fonctions de types `int -> int` et renvoyant la même liste triée selon la valeur en 3 des fonctions de la liste.
4. Écrire une fonction prenant en argument une liste de listes et indiquant si elle contient la liste vide.

Utiliser la récursivité

1. Écrire une fonction prenant en argument un entier et indiquant si c'est un nombre premier.
2. Écrire une fonction prenant en arguments une liste et un entier `n` et effectuant une rotation de `n` éléments vers la gauche dans la liste.