

TD n° 1 : Types, fonctions et récursivité

EXERCICE 1 *Portée statique en CAML*

Prévoir la réponse de CAML pour chacune des phrases dans la séquence suivante :

```
let a = 2;;
let f x = a * x;;
let a = 3 in f 1;;
let a = 3;;
f 1;;
```

EXERCICE 2 *Typage*

Quel est le type des fonctions suivantes ?

```
let f1 n m = if n = m then n else m;;
let f2 x y = let z = x + 1 in y || z > 10;;
let f3 x y = x y;;
let f4 x y = x (x y);;
let f5 x y = x (y x);;
let f6 x y z = x y z;;
let f7 (x, y, z) = function x -> (x, y, z);;
let f8 f g x = f x + g x;;
let f9 f g x = f (g x);;
```

EXERCICE 3

Que pensez-vous de `let max3 x y z = max max x y z;;` ?

EXERCICE 4 *Curryfication*

Quel est le type des deux fonctions suivantes ?

```
let add1 (x, y) = x + y;;
let add2 x y = x + y;;
```

Écrire une fonction `uncurry` qui prend en argument une fonction curryfiée à deux arguments (par exemple `add2`) et retourne la même fonction non curryfiée (par exemple `add1`), ainsi que sa réciproque `curry`. Quels sont leurs types ?

EXERCICE 5

Déterminer le type et expliquer ce que calcule la fonction suivante :

```
let rec composition f = function
| 0 -> (function x -> x)
| n ->
    (function x -> (f ((composition f (n - 1)) x)))
;;
```

EXERCICE 6 *Nombre de chiffres d'un entier*

Écrire une fonction `nombre_chiffres : int -> int` donnant le nombre de chiffres d'un entier naturel écrit en base 10.

EXERCICE 7 *Multiplication du paysan russe*

On multiplie deux entiers a et b de la manière suivante : on divise a par 2 tant que c'est possible, en doublant b , sinon on décrémente a et on ajoute b au résultat. Implémenter cet algorithme en CAML, montrer qu'il termine et qu'il renvoie le résultat escompté.