

TD n° 2 : Fonctions et récursivité

EXERCICE 1 *Définition de fonctions*

Déterminer le type et le comportement des fonctions suivantes :

OCAML

```
let fonction1 = fun f -> f 0;;
let fonction2 f = f 0;;
let fonction3 f g = fun x -> (f x) + (g x);;
let fonction4 f g x = (f x) + (g x);;
let fonction5 = fonction3 (fun x -> x);;
let fonction6 = fun f -> fun x -> f x;;
let fonction7 f = fun x -> f x;;
let fonction8 f x = f x;;
```

EXERCICE 2 *Composition*

Quel est le type de la fonction suivante ?

OCAML

```
let compose f g = fun x -> f (g x)
```

Remarque 1

PYTHON

```
def compose(f, g):
    def f_o_g(x):
        return f(g(x))
    return f_o_g
```

OCAML

```
let compose f g =
  let f_o_g x =
    f (g x)
  in
  f_o_g
```

EXERCICE 3

Proposer une fonction CAML pour chacun des types suivants :

- `int -> (int -> int)`
- `(int -> int) -> int`
- `int -> int -> int`

EXERCICE 4 *Typage*

Quel est le type des fonctions suivantes ?

OCAML

```
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 z = (x y) z;;
let f5 x y z = x (y z);;
let f6 x y z = x y z;;
let f7 (x, y, z) = fun x -> (x, y, z);;
```

EXERCICE 5

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

EXERCICE 6

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

OCAML

```
let rec composition f n =
  if n = 0 then
    fun x -> x
  else
    fun x -> (f ((composition f (n - 1)) x))
```

EXERCICE 7 *Curryfication*

Quel est le type des deux fonctions suivantes ?

OCAML

```
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`). Écrire sa réciproque `curry`. Quels sont leurs types ?

EXERCICE 8 *Nombre de chiffres d'un entier*

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

EXERCICE 9 *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.

EXERCICE 10 *Moyenne arithmético-géométrique (à faire avec CAML)*

Soit $a, b \in \mathbb{R}_+$, on définit deux suites positives $(u_n)_{n \in \mathbb{N}}$ et $(v_n)_{n \in \mathbb{N}}$, de premiers termes $u_0 = a$ et $v_0 = b$ et satisfaisant les relations de récurrence :

$$u_{n+1} = \sqrt{u_n v_n} \quad v_{n+1} = \frac{u_n + v_n}{2}$$

1. Montrer que les suites $(u_n)_{n \in \mathbb{N}}$ et $(v_n)_{n \in \mathbb{N}}$ sont adjacentes.
2. Pour $a = 1$ et $b = 2$, calculer u_{10} et v_{10} . A-t-on $(u_{10}) = (v_{10})$? Expliquer.

3. Écrire une fonction `suites` telle que `suites a b` renvoie le couple des suites $((u_n)_{n \in \mathbb{N}}, (v_n)_{n \in \mathbb{N}})$. Quel est son type ?
4. En utilisant la fonction précédente, écrire une fonction `termes` telle que `termes a b n` renvoie le couple des n -ième termes (u_n, v_n) .
5. Calculer u_{10} et v_{10} pour $(a, b) \in \{(1, 1000), (2, 487), (500, 501)\}$.

EXERCICE 11

Que font les fonctions f et g ?

1.

OCAML

```
let rec f n = (n = 0) || g (n - 1)
and g n = (n <> 0) && f (n - 1);;
```

2.

OCAML

```
let rec f n = if n = 0 then 1 else 3 + g (n - 1)
and g n = if n = 0 then 0 else 1 + f (n - 1);;
```

3. (à implémenter en CAML)

$$f : (m, n) \rightarrow \begin{cases} n & \text{si } m = 0 \\ g(m - 1, n + 1) & \text{sinon} \end{cases}$$

$$g : (m, n) \rightarrow \begin{cases} n & \text{si } m = 0 \\ f(m - 1, n) + 1 & \text{sinon} \end{cases}$$