

---

**UE ELFE - Programmation Fonctionnelle**


---

durée 1h – documents autorisés

**Contrôle de TD****Exercice 1 :** *Typage d'expressions***Q 1 .** Calculez le type des expressions suivantes :

- a. `fun x -> x +. 1. ;;`
- b. `fun f x y -> (f x y) + 2 ;;`
- c. `fun f g x -> f (g (f x)) ;;`

**Q 2 .** Donnez une expression pour chacun des types suivants :

- a. `int * int -> int`
- b. `int -> (int -> int) -> int`
- c. `( 'a -> 'b -> 'c ) -> ( 'a -> 'b ) -> 'a -> 'c`
- d. `( 'a -> 'a -> int ) -> ( 'a -> 'a ) -> 'a -> int`

**Exercice 2 :** *Listes*

Proposez une écriture récursive des fonctions suivantes :

**Q 1 .** `count l x` : la fonction *count* est de type `'a list -> 'a -> int`. Cette fonction compte, dans une liste *l*, le nombre d'éléments qui sont égaux à *x*.**Q 2 .** `flatten l` : la fonction *flatten* est de type `'a list list -> 'a list`. Cette fonction construit la liste des éléments des listes contenues dans *l*.

Par exemple :

$$\text{flatten} \left[ \left[ [1; 2]; []; [3]; []; [4]; [5] \right] \right] = \left[ [1; 2]; []; [3]; [4]; [5] \right]$$

**Exercice 3 :** *Palindromes sur l'ADN*Une chaîne d'ADN peut être représentée par une chaîne de caractères sur un alphabet  $\{A, T, G, C\}$ . Cette chaîne *s* de longueur *l* est un palindrome si elle vérifie le prédicat suivant :

$$\forall i \in [0..l-1] \quad (s[i], s[l-1-i]) \in \text{Complementaires}$$

L'ensemble des couples de caractères *Complementaires* est  $\{ ('A', 'T'), ('T', 'A'), ('C', 'G'), ('G', 'C') \}$ 

Voici des exemples de palindromes : "CG", "TCGA", "GTCGAC", "GGTCGACC", "AGGTCGACCT". En revanche, les chaînes "CC", "CT", "CA", "TCGT" n'en sont pas.

**Q 1 .** Réalisez la fonction `Coupe.Extremities s`. Cette fonction est de type `string -> string` et retourne la chaîne *s* tronquée de son premier et dernier caractère. Vous utiliserez les fonctions prédéfinies *String.sub s p l* (retourne la sous-chaîne de *s* de longueur *l* débutant à la position *p*) et *String.length s*.**Q 2 .** Réalisez la fonction `Extremities s`. Cette fonction est de type `string -> (char*char)` et retourne la paire de caractères situés aux extrémités gauche et droite de la chaîne *s*. Vous utiliserez les fonctions prédéfinies *String.get s i* (retourne le caractère à la position *i* sur la chaîne *s*) et *String.length s*.**Q 3 .** Réalisez la fonction `Complementaire c`. Cette fonction est de type `(char*char) -> bool` et retourne *true* si la paire de caractères *c* est complémentaire, *false* autrement. Vous utiliserez un filtrage de motif.**Q 4 .** En remarquant que les palindromes ainsi définis sont de longueur paire, proposez une fonction `Palindrome s` qui vérifie si une chaîne *s* est un palindrome.

# Solutions

## Exercise 1

### Q 1.

#### Solution

- a. `float -> float`
- b. `('a -> 'b -> int) -> 'a -> 'b -> int`
- c. `('a -> 'b) -> ('b -> 'a) -> 'a -> 'b`

### Q 2.

#### Solution

- a. `fun (x,y) -> x + y ;;` ou autre
- b. `fun x f -> f x + x ;;` ou autre
- c. `fun f g x -> f x (g x) ;;` ou autre
- d. `fun f g x -> f x (g x) + f (g x) x;;` ou autre

## Exercise 2

### Q 1.

#### Solution

```
let rec count1 l x = if l = [] then 0
                      else if List.hd l = x
                           then 1 + count1 (List.tl l) x
                           else count1 (List.tl l) x;;

let rec count2 l x = match l with
| [] -> 0
| h::t -> if h = x
          then 1 + count2 t x
          else count2 t x;;
```

### Q 2.

#### Solution

```
let rec flatten1 l = if l = []
                     then []
                     else (List.hd l) @ flatten1 (List.tl l);;

let rec flatten2 l = match l with
| [] -> []
| h::t -> h @ flatten2 t;;
```

## Exercise 3

#### Solution

```
let Coupe_Extremites s = String.sub s 1 (String.length s - 2);;
```

#### Solution

```
let Extremites s = String.get s 0 , String.get s (String.length s - 1);;
```

#### Solution

```
let Complementaire c = match c with
| ('A', 'T') | ('T', 'A') | ('G', 'C') | ('C', 'G') -> true
| _ -> false ;;
```

#### Solution

```
let rec Palindrome s = (String.length s = 0) ||
  (String.length s >= 2) &&
  Complementaire (Extremites s) &&
  Palindrome (Coupe_Extremites s);;
```