

Programmation Fonctionnelle : TD3

EFREI - Paris
Récurtivité et listes

*
 * *

Appropriation du cours

Reprendre le cours 3 et tester en OCaml les exemples, en particulier vous devez très bien comprendre :

- La fonction `nbOccurences`
- La fonction `retireOccurence1`
- La fonction `plusGrand` (dans les deux versions, vérifier avec la trace que la version 2 entraîne moins de calculs).

Problème 1

1. Écrire la spécification puis le code d'une fonction `sommeListe1 l` qui retourne la somme des éléments d'une liste l d'entiers donnée en entrée.
2. Ecrire la spécification et le code d'une fonction `mulScalaire l lambda` qui multiplie par $\lambda \in \mathbb{R}$ chaque élément d'une liste l de réels donnée en entrée.

Problème 2

1. Écrire la spécification puis le code d'une fonction `sommeListe2 l1 l2` qui prend en paramètres deux listes l'entiers l_1 et l_2 et retourne la liste des sommes des éléments de même rang sur la plus courte des deux longueurs de liste (càd : $\min(|l_1|, |l_2|)$).
2. Donner une version `sommeListe3` où le résultat contient autant d'éléments que la plus longue des deux listes. Les éléments sans correspondance dans l'autre liste sont simplement copiés à la fin.

Problème 3

1. Écrire la spécification et le code d'une fonction `equal l1 l2` qui prend en entrée deux listes l_1 et l_2 et qui retourne vrai si et seulement l_1 et l_2 contiennent le même nombre d'éléments et que pour chaque indice, l'élément i de l_1 est égal à l'élément i de l_2 .

2. On donne la spécification suivante `miroir` :
$$\begin{cases} \text{List} < T > & \rightarrow \text{List} < T > \\ l = [x_0, \dots, x_{n-2}, x_{n-1}] & \mapsto l' = [x_{n-1}, x_{n-2}, \dots, x_0] \end{cases}$$

Écrire le code correspondant à `miroir` 1. On pourra utiliser l'opérateur de concaténation `@` d'OCaml.

3. Un *palindrome* est un mot qui peut être lu dans les deux sens (de droit à gauche ou de gauche à droite). Par exemple : “rotor”, “ete”, “callac” ou bien “mon nom”.

Écrire la spécification et le code d’une fonction `palindrome` 1 qui prend en entrée une liste de caractères *l* et retourne vrai si le mot formé par la liste est un palindrome et faux sinon.