

# Introduction à la programmation

## Premiers exercices sur les listes

### Détecter si une liste a un seul élément

Écrire une fonction *EstSingleton* qui, étant donnée une liste  $l$  renvoie *true* si la  $l$  contient un seul élément et *false* sinon.

### Le deuxième élément d'une liste

Écrire une fonction *DeuxiemeElement* qui, à partir d'une liste ayant au moins deux éléments renvoie le deuxième élément de la liste.

### Le nombre d'éléments d'une liste

Écrire une fonction *longueur* qui, étant donnée une liste  $l$ , renvoie le nombre d'éléments de cette liste. Quel devrait être le type de cette fonction, autrement dit sur quelles listes pourra-t-on utiliser cette fonction ?

### La somme des éléments d'une liste

Écrire une fonction *somme* qui, étant donnée une liste  $l$ , renvoie la somme de ses éléments. Même question que précédemment : sur quels types de listes peut-on utiliser cette fonction ?

### La concaténation de deux listes

Écrire une fonction *ConcatListe* qui, à partir de deux listes renvoie une troisième liste constituée de la première liste et de la deuxième liste mises bout à bout. Par exemple *ConcatListe [1;2;3] [4;5;6]* est la liste  $[1;2;3;4;5;6]$ . Utiliser le filtrage explicite.

### L'appartenance à une liste

Écrire une fonction *Membre* qui, à partir d'un élément  $x$  et d'une liste  $l$  renvoie *true* si  $x$  appartient à  $l$  et *false* sinon.

### Tester l'égalité entre deux listes

Écrire une fonction *MemeListe* qui, à partir de deux listes  $l_1$  et  $l_2$  renvoie *true* si les deux listes sont égales et *false* sinon. Utiliser le filtrage explicite pour cette fonction

### Comparer la longueur de deux listes

Écrire une fonction *MemeLongueur* qui, à partir de deux listes  $l_1$  et  $l_2$  renvoie *true* si les deux listes ont le même nombre d'éléments et *false* sinon. On pourrait réaliser cela en appelant deux fois la fonction *longueur* (cf. la troisième question) mais alors la durée de calcul sera la durée de parcours des deux listes. Est-ce qu'il ne serait pas possible de proposer un algorithme plus efficace ?

**Renverser une liste**

Écrire une fonction *Renverser* qui, à partir d'une liste renvoie une autre liste contenant les mêmes éléments mais dans l'ordre inverse.