

Fiche de TD no. 1

Exercice 1 : *Expressions de liste.* Une liste s'écrit entre crochets, avec les éléments de la liste séparés par des virgules.

1. Rappelez ce que font les opérateurs `[]`, `:`, `++`, `..` et les fonctions `head`, `tail`, `reverse`.
2. Évaluez les expressions de liste suivantes :
 - (a) `1:[2]`
 - (b) `[3,4]++[1,2]`
 - (c) `[3..10]`
 - (d) `tail [1..4] ++ 5:[]`
 - (e) `head [1..4] : [5]`
 - (f) `reverse [1..4] ++ [5]`

Exercice 2 : *Définition de fonctions sur les listes.* Rappelez ce que font les fonctions `head`, `tail`, `reverse`, `length`, `drop`, `take`, `!!`, définie dans `prelude.hs` et vue en cours.

1. La fonction `last`, définie dans `prelude.hs`, sélectionne le dernier élément d'une liste. Montrez comment cette fonction peut se définir en utilisant les fonctions sur les listes présentées dans le cours.
2. Pouvez vous penser à d'autres définitions de `last` ?
3. De façon semblable, montrez comment la fonction `init`—du `prelude.hs`, qui enlève le dernier élément d'une liste—peut se définir de plusieurs façons.

Exercice 3 : *Fonctions sur les listes.* Une chaîne de caractères est une liste de caractères. On souhaite définir une fonction appelée `palindrome` qui répond à la question si une chaîne de caractères est un palindrome.

1. Quel est le type de cette fonction ? (Après avoir explicité le domaine et le codomaine de la fonction, écrivez l'expression de type appropriée).
2. Proposez une définition de cette fonction.

Exercice 4 : *Maths et Haskell.*

1. Réécrivez, en notation mathématique, les expressions suivantes de Haskell :
 - (a) `f 3 + g 9 *8`,
 - (b) `f 22 33 * (g 44 h 55)`,
 - (c) `f 22 33 * g 44 (h 55)`.

Étés vous capables de deviner le type de f , g , et h , dans les trois expressions Haskell ci-dessus ?

2. Réécrivez, dans le langage Haskell, les expressions mathématiques suivantes :
 - (a) $f(g(y, z), 33)$,
 - (b) $8g(44, 23)$,
 - (c) $f(8\pi(33 + 21))$.

Exercice 5. Quel est le type des valeurs suivantes ?

```
['a', 'b', 'c']
('a', 'b', 'c')
[(False, '0'), (True, '1')]
([False, True], ['0', '1'])
[tail, init, reverse]
```

Exercice 6. (Fonctions) Considérez les définitions suivantes :

```

second xs = head (tail xs)
appl (f,x) = f x
pair x y = (x,y)
mult x y = x * y
double = mult 2
palindrome xs = reverse xs == xs
twice f x = f (f x)
incrAll xs = map (+1) xs
norme xs = sqrt (sum (map f xs)) where f x = x^2

```

1. Calculez les types de ces fonctions, n'oubliant pas les *contraintes de classe*.
2. Quelles sont les fonctions de deux arguments ? Si une telle fonction n'est pas en *forme curriifiée*, donnez une définition équivalente qui soit en forme curriifiée.
3. Quelles sont les fonctions d'*ordre supérieur* ?
4. Quelles sont les fonctions *polymorphes* ?

Exercice 7 : Compréhension. On définit une image comme une suite de lignes, chaque ligne étant une suite de 0 et 1. Ainsi, nous allons représenter en Haskell une image comme un valeur de type `[[Int]]`, c'est-à-dire, comme une liste de listes d'entiers. Par exemple, le triangle sur la gauche sera représentée par l'expression Haskell sur la droite :

```

      * * *
        * *   [[1,1,1], [0,1,1], [0,0,1]]
          *

```

Lisez attentivement le script Haskell suivant :

```

1 inv :: Int -> Int
2 inv 0 = 1
3 inv x = 0

5 inverser :: [[Int]] -> [[Int]]
6 inverser image = map (map inv) image

8 refleter :: [[Int]] -> [[Int]]
9 refleter image = reverse image

11 eln :: Int -> [a] -> a
12 eln n xs = xs !! n

14 colonne :: [[Int]] -> Int -> [Int]
15 colonne image n = map (eln n) image

17 transposer :: [[Int]] -> [[Int]]
18 transposer image = map (colonne image) [0..(length (head image) -1)]

```

1. Soulignez toutes les constructions que n'ont pas encore été présentes en cours.
2. Quel est le type de l'expression `map inv` ? Et celui de l'expression `eln n` ?
3. Nous pouvons supprimer du script la définition de la fonction `eln`—qui est tout à fait semblable à l'opérateur binaire `!!`—si, à la ligne 15, on remplace l'expression `eln n` par une lambda-expression. Expliquez comment on peut faire exactement.
4. Expliquez ce qui est achevé par ces définitions, en terme de transformations d'images.
5. Complétez ce script avec laa définition d'une fonction qui fait une rotation à gauche de l'image. Comment définir une fonction qui fait la rotation à droite ?