

## Fichiers et dictionnaires (suite du précédent TD)

Le but de ce TD est d'écrire la procédure « inverse » de la procédure de lecture de profils d'expression faite au TD précédent (`lireProfils`). Il s'agit maintenant d'écrire dans un fichier à partir d'une liste de dictionnaires représentant des profils d'expression de gènes. Par exemple en partant de la liste de dictionnaires suivante :

```
[ { "nom": "Yea302", "00h00": 6.3, "01h21": 1.4, "02h32": 3.0, "02h40": 1.4, "03h35": 0.7,
"06h45": 5.6, "07h50": 0.7, "09h12": 9.0 } , { "nom": "TEL1", "00h00": 2.4, "05h28": 0.6,
"12h02": 3.9, "12h48": 5.5, "20h51": 10.2 } , { "nom": "SR077", "00h00": 36.0, "03h08": 55.2,
"10h00": 59.8, "15h33": 100.0, "18h25": 100.0 } ]
```

On voudra écrire dans un fichier les lignes suivantes :

```
nom = Yea302
00h00 : 6.3
01h21 : 1.4
02h32 : 3.0
02h40 : 1.4
03h35 : 0.7
06h45 : 5.6
07h50 : 0.7
09h12 : 9
nom = TEL1
00h00 : 2.4
05h28 : 0.6
12h02 : 3.9
12h48 : 5.5
20h51 : 10.2
nom = SR077
00h00 : 36
03h08 : 55.2
10h00 : 59.8
15h33 : 100
18h25 : 100
```

**Nota** : sauvegardez toutes vos fonctions pour pouvoir les réutiliser lors du prochain TD.

**Exercice 1** : Le premier problème rencontré est que les dictionnaires « mélangent » les champs et les énumèrent donc dans n'importe quel ordre. Commencez par le vérifier en examinant le résultat de la fonction `lireProfils` écrite au précédent TD. Profitez-en pour définir une variable globale `yea` contenant le premier dictionnaire de la liste (celui de nom `Yea302`); ce sera utile pour tester les exercices suivants.

**Exercice 2** : Pour résoudre ce problème d'ordre, écrivez une fonction `horloge` qui prend en entrée un dictionnaire `d` comme précédemment, et fournit en sortie la liste triée des temps du dictionnaire.

INDICATION : `sorted` est une fonction qui prend une liste en entrée et retourne cette liste triée par ordre croissant (dans la plupart des autres langages de programmation, cette fonction se nomme « `sort` » et « `sorted` » est plutôt la fonction booléenne qui dit si une liste est triée).

**Exercice 3** : Écrivez une procédure `ecrireProfils` qui prend en entrée un nom de fichier `f` et une liste de dictionnaires `ld` représentant des profils d'expression comme dans le précédent TD, et crée le fichier de nom `f` contenant successivement les profils d'expression, comme dans l'exemple précédent, avec les temps en ordre croissant pour chaque gène.

**Exercice 4** : Écrivez une fonction `horaire` qui prend en entrée deux nombres entiers `X` et `Y`, qui fournit un message d'erreur si `X` n'est pas une heure valide (c'est-à-dire si elle n'est pas comprise entre 0 et 23) ou si `Y` n'est pas un nombre de minutes valide (c'est-à-dire compris entre 0 et 59), et qui retourne dans tous les autres cas la chaîne de caractères normalisée correspondante. Par exemple `horaire(2,28)` retourne `"02h28"` et `horaire(19,7)` retourne `"19h07"`.