

## Regroupement septembre 2022

### Programmation fonctionnelle

#### Epreuve 1

Dans les différents exercices, vous pouvez, si besoin est, écrire des sous-fonctions pour vous simplifier le travail.

1. Soit une liste constituée de noms de personne (on suppose qu'il n'y a qu'un seul prénom et que le nom est représenté par un unique atome, même s'il a plusieurs parties), de la forme :

liste\_personnes : ((prénom . nom) (prénom . nom) ...)

Regroupez dans une liste tous les noms différents (si deux personnes ont le même nom, le nom doit n'apparaître qu'une fois) :

=> (nom nom ...)

Faites la même chose pour les prénoms.

2. Pour chaque nom, comptez le nombre de personnes ayant ce nom et générez la liste :

((nom . nombre) (nom . nombre)...) )

#### Epreuve 2

A partir d'un arbre (liste quelconque) ne contenant que des noms (pas de nombres), remplacez chaque sous-liste terminale (ne contenant pas de sous-liste) par le nombre d'éléments de cette sous-liste, à l'aide d'une fonction chirurgicale récursive. Nil sera considéré comme un atome, pas comme une liste.

Exemple : (a (b (c (d e f) r (e)) x (g h f)) i j (k l m n))

=> (a (b (c 3 r 1) x 3) i j 4)

Astuce 1 : Pour savoir la longueur d'une liste, utilisez la fonction length.

Astuce 2 : Commencez par écrire une fonction testant si une liste a des sous-listes.