

Regroupement juin 2022

Programmation fonctionnelle

Epreuve 1

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

1. Soient deux listes plates, la première étant une liste de noms et la seconde une liste de cours, construire récursivement une liste comme suit :

liste_noms : (nom1 nom2... nomN)

liste_cours : (cours1 cours2... coursN)

=> ((nom1 (cours1) (cours2) ... (coursN)) ... (nom2 (cours1) (cours2)... (coursN)))

2. Etant donné une troisième liste :

((note1 note2... noteN) (note1 note2... noteN)...)

contenant les notes obtenues par les étudiants, chaque sous-liste étant associée à un étudiant par sa position (la première sous-liste pour le premier étudiant, la deuxième pour le deuxième, etc.), et dans chaque sous-liste, chaque note est associée à un cours par sa position (la première note pour le premier cours, la deuxième pour le deuxième, etc.).

Construire à partir de la liste produite en (1) et de cette dernière liste :

.
=> ((nom1 (cours1 . note1) (cours2 . note2) (cours3 . note3))
 (nom1 (cours1 . note1) (cours2 . note2) (cours3 . Note3))
 ...)

Epreuve 2

A partir d'un arbre (liste quelconque), remplacez par zéro tous les atomes qui ne sont pas des nombres, à l'aide d'une fonction chirurgicale récursive.

Astuce 1 : Pour savoir si un élément est un nombre, utilisez la fonction `numberp` (qui ramène nil si son argument n'est pas un nombre).

Astuce 2 : Commencez par écrire une fonction non chirurgicale (qui construit sans modifier).