

Regroupement juin 2021

Programmation fonctionnelle

Epreuve 1

Soient deux listes plates, la première étant une liste de produits et la seconde une liste de prix, ayant le même nombre d'éléments.

1. Construire récursivement une liste dont chaque élément est un doublet ; le premier élément de ce doublet est le produit et le deuxième le prix.

(produit produit ... produit)

(prix prix ... prix)

=> ((produit . prix) (produit . prix) ... (produit . prix))

2. Écrire une fonction qui ramène le prix d'un produit à partir de cette dernière liste :

(prix produit liste) => prix du produit

Epreuve 2

A partir d'un arbre (liste quelconque), supprimer les noeuds dont la valeur n'est pas un nombre, 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).