Regroupement septembre 2016

Programmation fonctionnelle

Vous pouvez définir des fonctions auxiliaires si vous en avez besoin.

Epreuve 1

1. Soient deux listes plates, la première étant une liste de produits et la seconde une liste de prix.

Liste_produits : (produit produit ... produit)
Liste_prix : (prix prix ... prix)

Créer une fonction "associer" qui construit récursivement une liste dont chaque élément est un doublet ; le premier élément de ce doublet est le produit et le deuxième le prix. Il faut définir la fonction associer :

(associer Liste_produits Liste_prix)
=> ((produit . prix) (produit . prix) ... (produit . prix))

On laisse tomber les produits qui n'ont pas de prix et les prix qui n'ont pas de produits.

2. Créer une fonction chirurgicale récursive qui fait la même chose mais en modifiant directement Liste_produits :

(modifier Liste_produits Liste_prix)
=> ((produit . prix) (produit . prix) ... (produit . prix))

Liste_produits : ((produit . prix) (produit . prix) ... (produit . prix))

Astuce : utiliser rplaca.

Epreuve 2

On part d'un arbre contenant des produits groupés par catégories et dont les feuilles contiennent des prix ; par exemple :

Arbre_produits :
((fruit (pomme (golden 3.40) (boskop 2.5)) (raisin (blanc (muscat 4)...) (noir (chasselas 5.5))))
(légume (carotte 2)...))

Créer une liste des produits dont le prix est inférieur à une certaine valeur :

(inférieur 4 Arbre_produits) => ((golden 3.40) (boskop 2.5) (carotte 2) ...)

Durée de l'épreuve : 45mn