

DEUG 1^{ère} année
MIAS 1 et MASS 1 – Programmation fonctionnelle

TD n° 6

But : Récursivité profonde.

Exercice 1 : Donner la spécification et définir en scheme une fonction **Nbocc** qui renvoie le nombre d'occurrences d'un élément dans une liste en examinant tous les niveaux de la liste.

`(Nbocc 1 '(1 1 c 1 4 h)) → 3`

`(Nbocc 'a '(b a (2 d (g 4 a) (h a)) ((a)))) → 4`

Exercice 2 : Donner la spécification et définir en scheme une fonction qui supprime tous les nombres présents dans une liste quelconque.

`(supnombre '(1 1 c 1 4 h)) → (c h)`

`(supnombre '(b 9 (2 d (g 4 a) (h a)) (8))) → (b (d (g a) (h a))())`

Exercice 3 : Donner la spécification et définir en scheme un prédicat qui teste légalité de deux listes quelconques. On suppose bien entendu qu'on ne peut pas utiliser le prédicat prédéfini `equal?` pour tester l'égalité de deux listes.