

## TD révisions 2

- On cherche à construire la liste des nombres entiers qui sont égaux à la somme de leurs chiffres mis au cube (par exemple 153) inférieurs à un nombre donné. Définir les fonctions intermédiaires nécessaires à la réalisation de cette fonction et implémenter l'ensemble.

```
(armstrong 1000) → (0 1 153 370 371 407)
```

- Définir une fonction qui compte en profondeur le nombre d'entiers d'une liste quelconque divisibles par un entier donné.

```
(compter-divisibles-prof '(a 3 6 (7 j (t 9)) 4 12) 3) → 4
```

- On considère des arbres où les valeurs des nœuds sont des couples (listes de deux éléments). Le but de l'exercice est de décrypter un mot caché. Il faut parcourir l'arbre suivant un chemin donné et récupérer le 1er ou le 2ème élément de la liste suivant l'indication associée au chemin. Le chemin ainsi que l'élément à choisir sont donnés par une liste de couples. On suppose que le chemin reste dans l'arbre.

```
(motcache '((p i)((e u)((t z)())((s r)())((z d)())((u o)())()))
           '((1 g)(1 d)(2 d)(2 d)(1 d))) → (p e r d u)
```

- Soit une liste de listes. Définir en utilisant map une fonction qui ajoute au début de chaque sous-liste sa longueur.

```
(longueur_en_tete '((4 z)(3 a &)(1 2 3 4 5 6)))
→ ((2 4 z) (3 3 a &) (6 1 2 3 4 5 6))
```

- Définir une fonction qui calcule la moyenne des valeurs d'un arbre de nombres en faisant un seul parcours de l'arbre.

- En utilisant la fonction abstraite sur les arbres vue en cours, définir une fonction qui enlève 1 à toutes les valeurs impaires d'un arbre.