

## TDTP 5 : listes de listes

---

### À faire pendant la séance

---

TD

#### Listes de listes : la suite de Conway

Devinette : comment se construit la suite de Conway ?

0, 10, 1110, 3110, 132110, 1113122110, 311311222110, ...

- Définir une fonction qui calcule un terme de la suite de Conway à partir du précédent.

`(conway '(1 1 1 0)) -> (3 1 1 0)`

On peut écrire cette fonction en remontant (en utilisant le résultat de l'appel récursif sur le reste de la liste) ou en descendant (en utilisant un compteur du nombre de répétitions de l'élément courant).

- Définir une fonction qui calcule les n premiers termes de la suite de Conway en fonction du premier terme.

`(conwayn 4 '(0)) -> ((0) (1 0) (1 1 1 0) (3 1 1 0))`

TP

#### D'autres listes de listes

- Définir une fonction qui insère un élément en tête de la sous-liste dont l'indice est passé en paramètre.

`(ajoute 'a '((e r) (r y b) (t e)) 2) -> ((e r) (a r y b) (t e))`

`(ajoute 'a '((e r) (r y b) (t e)) 0) -> ((a) (e r) (r y b) (t e))`

- Définir une fonction `(sp n x y)` qui, étant donnés deux nombres x et y, calcule les n premiers termes de la suite  $x_n = x_{n-1} + y_{n-1}$  et  $y_n = x_{n-1} * y_{n-1}$ .

`(sp 4 5 2) -> ((5 2) (7 10) (17 70) (87 1190))`

TP

#### Listes ordonnées

- Définir une fonction qui, étant donné un nombre n et une liste L d'entiers triée par ordre croissant, retourne la liste des éléments de L diviseurs de n. *Attention à ne pas parcourir la liste en totalité quand ce n'est pas nécessaire.*

`(diviseurs '(1 2 3 4 5 8 12 13 15 17) 12) -> (1 2 3 4 12)`

---

### Pour s'entraîner (exercices supplémentaires facultatifs)

---

La fonction prédéfinie `random` permet d'engendrer des nombres entiers au hasard : `(random x)` retourne un entier dans l'intervalle  $[0, x[$ .

- Définir une fonction qui retourne une liste de nombres entiers positifs pris au hasard. Les deux paramètres de cette fonction sont la longueur de la liste à construire et la valeur maximale des nombres à générer.

`(liste_random 5 10) -> (7 9 6 7 4)`

- Définir une fonction qui retourne une liste composée d'un nombre entier pris au hasard entre 0 et N, et d'un booléen indiquant si ce nombre est un multiple de trois ou de sept.

`(nb_test 10) -> (5 #f)`

`(nb_test 10) -> (6 #t)`

- Définir une fonction identique à la fonction `liste_random`, mais qui ne retourne que des nombres pairs.

`(liste_random_pairs 5 10) -> (4 10 6 4 6)`