

TD-TP n°2

Fonctions récursives, mémorisation.

À l'ouverture de l'éditeur *DrRacket*, assurez-vous de choisir le langage *Racket*. Sur Moodle, il y a une section *TP2* dans laquelle est contenu un fichier bleu nommé *TP2 - Rendu*, dans lequel vous pouvez déposer votre fichier.

Exercice 1. Que fait la fonction mystère ci-dessous ?

```
#lang racket
(define mystere
  (lambda (n)
    (if (= n 0)
        0
        (if (= (modulo n 2) 0)
            (+ n (mystere (- n 1)))
            (mystere (- n 1))))))
```

Écrivez une version améliorée de cette fonction en utilisant une mémorisation.

Exercice 2. Fonctions récursives.

1. Écrire une fonction récursive qui calcule la somme $\sum_{i=1}^n \frac{1}{i^2}$.
2. Écrire une fonction récursive (sans utiliser *expt*) qui permet de calculer une puissance x^y .
3. Écrire une fonction récursive qui teste si un nombre est une puissance de 5.
4. Écrire une fonction récursive (sans utiliser *quotient*) qui permet de renvoyer le quotient de la division euclidienne de a par b , où a et b sont deux entiers.
5. Écrire une fonction récursive (sans utiliser *gcd*) qui renvoie le PGCD de deux nombres en utilisant la méthode des différences. Principe : un nombre est un diviseur de 2 nombres a et b s'il est un diviseur de la différence entre a et b .

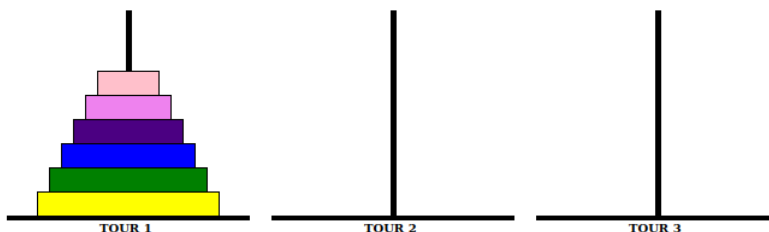
Exercice 3. Fonction de Tribonacci. On considère la suite de Tribonacci définie par

$$T(0) = 1, T(1) = 1, T(2) = 1 \text{ et pour } n \geq 3, T(n) = T(n-1) + T(n-2) + T(n-3).$$

1. Écrire une fonction récursive permettant de calculer $T(n)$.
2. Écrire une fonction permettant de calculer $T(n)$ en utilisant la récursivité terminale.
3. Comparer l'efficacité de ces deux fonctions pour différentes valeurs de n .

Exercice 4. Tours de Hanoï. Les tours de Hanoï sont un jeu de réflexion imaginé par le mathématicien français Édouard Lucas, consistant à déplacer des disques de diamètres différents d'une tour de « départ » à une tour d'« arrivée » en passant par une tour « intermédiaire », et ceci en un minimum de coups. Pour jouer, il faut respecter les règles suivantes:

- on ne peut déplacer plus d'un disque à la fois;
- on ne peut placer un disque que sur un autre disque plus grand que lui ou sur un emplacement vide.



Vous pouvez vous amuser à tester le jeu sur <http://championmath.free.fr/tourhanoi.htm> et changer le nombre n de disques. Écrire une fonction récursive qui renvoie le nombre minimum de coups nécessaires pour résoudre les tours de Hanoï avec n disques.

Exercice 5. Premières manipulations de listes. Écrivez et tester les instructions suivantes dans Racket:

- `(car '(1 2 3 4))`
- `(cdr '(1 2 3 4))`
- `(car (cdr '(1 2 3 4)))`
- `(cdr '(1 (2 3 4)))`
- `(cons 1 (2 3 4))`
- `(cons '(1 2) '(3 4))`
- `(null? '())`
- `(null? '(1 2))`

À votre avis, quel est le rôle des fonctions `car`, `cdr`, `cons` et `null?`. Que devraient renvoyer les instructions suivantes:

- `(cdr (car (cdr '(a (b c) (d e))))))`
- `(cons 2 (cdr (cdr '(0 0 1 0 0))))`
- `(cdr (car '(1 2 3 4)))`
- `(null? (cdr '(1 2 3)))`
- `(null? (car '(1 2 3)))`

Exercice 6. Fonctions de listes En vous inspirant de la fonction `minimum` vue en cours, essayez d'écrire une fonction récursive `appartient` prenant deux arguments: un nombre x et une liste L , et qui renvoie vrai si x appartient à la liste, faux sinon.