

## Les FLOTS [streams]

Nous ne travaillerons que sur des **flots infinis** ! Il sera donc en principe inutile de tester si le flot est vide.

⌘ **Exercice 12.1** Définissez la fonction (`sfilter pred s`) du cours page 14. Testez-la en faisant afficher le flot des nombres de Fibonacci impairs.

⌘ **Exercice 12.2** a) Sur le modèle du flot des entiers INT [cours p. 15], définir le flot IMPAIR de tous les nombres impairs  $\geq 1$  :

> (`sprint IMPAIR`)  
[1,3,5,7,9,11,13,15,17,19,...]

b) Sur le modèle du flot de Fibonacci [cours p. 16], définir le flot FACT des factorielles :

> (`sprint FACT`)  
[1,1,2,6,24,120,720,5040,40320,362880,...]

c) Toujours sur le même modèle, définir un flot (`rand n`) d'entiers aléatoires dans  $[0, n-1]$  :

> (`sprint (rand 100)`)  
[95,12,51,23,5,75,43,1,94,37,...]

⌘ **Exercice 12.3** a) Ecrire une version (`smap2 f S1 S2`) acceptant une fonction binaire `f`, donc généralisant la version usuelle vue en cours à la page 14. Les flots `S1` et `S2` sont supposés infinis.

b) En déduire les définitions de (`s+ S1 S2`), de (`s- S1 S2`) et de (`s* S1 S2`) qui opèrent terme à terme sur deux flots.

c) Définir le flot constant DEUX = [2,2,2,2,2,...] à partir du flot constant ONE. En déduire une autre définition implicite du flot IMPAIR en pensant : « IMPAIR + DEUX  $\approx$  IMPAIR »...

d) Définir le flot INT en s'inspirant de la méthode utilisée en c).

e) Définir le flot infini CARRES = [(0 0),(1 1),(2 4),(3 9),(4 16),(5 25),...]

⌘ **Exercice 12.4** Définir de manière implicite [cf cours page 17] :

a) le flot infini des factorielles [1, 1, 2, 6, 24, 120, ...]

b) le flot infini des nombres de Fibonacci [0, 1, 1, 2, 3, 5, 8, 13, ...]. Le 10000<sup>ème</sup> nombre de Fibonacci est-il long à calculer ?

⌘ **Exercice 12.5** a) Par calculs sur papier uniquement, répondez à la question suivante : décrire les éléments du flot `S` défini en Scheme par : (`define S (scons 1 (s+ S S))`)

b) Vérifiez ensuite votre réponse au toplevel en faisant afficher cette série. Indication : écrivez le flot sous la forme  $[a_0, a_1, a_2, \dots]$  puis calculez un à un les premiers éléments  $a_i$  de ce flot.

**Compléments : le livre de cours PCPS § 17.4.**