Programmation 3 : feuille 6

Variables et programmation itérative

Exercice 6 .1 *Environnement lexical permanent*

1. Définir une fonction `ticket` qui retourne `1` la première fois qu'on l'appelle, `2` la deuxième *etc* ...

```
CL-USER> (ticket)
1
CL-USER> (ticket)
2
CL-USER> (ticket)
3
```

2. Compléter l'implémentation de manière à pouvoir remettre le compteur à zéro.

```
CL-USER> (ticket)
1
CL-USER> (ticket)
2
CL-USER> (raz)
0
CL-USER> (ticket)
1
```

Exercice 6 .2 *Définition de variables spéciales*

1. Implémenter la fonction associée à la formule de la période du pendule $T = 2 * \pi * \sqrt{L/g}$ (avec g variable globale valant 9.81).
2. On se transporte sur la Lune, ou l'accélération de la pesanteur est de 1.62 Faut-il modifier l'implémentation de la fonction précédente ?

Exercice 6 .3

Le langage APL est le précurseur du logiciel Matlab largement utilisé par les ingénieurs et les scientifiques aujourd'hui. Le but de l'exercice est d'implémenter cettela fonction `iota` d'APL de manière itérative.

1. Écrire une version de base `iota (n)` qui retourne la liste des entiers de `0` à `n-1`.
2. Rajouter un paramètre optionel indiquant la valeur du premier élément (`0` par défaut).
3. Rajouter un deuxième paramètre optionel correspondant au pas (`1` par défaut).
4. Faire une nouvelle version en utilisant des paramètres mot-clé au lieu des paramètres optionels.

Exercice 6 .4

Écrire une version itérative de la fonction `triangle` de la Feuille 5.

Exercice 6 .5

Réécrire les fonctions récursives de la Feuille 2 de manière itérative.

- `derivee-n-approx`
- `op-prod`
- `serie`

Exercice 6 .6

Écrire une fonction `test-randomize-list (n)` où $n \in \mathbb{N}$ qui teste le fonctionnement de la fonction `random-list` (définie dans la Feuille 3) en appelant `n` fois la fonction `test-randomize-list-once`.

Exercice 6 .7

Écrire une fonction `test-mapcar-fill(n)` où $n \in \mathbb{N}$, dont le but est de tester `mapcar-fill` (définie dans la Feuille 5) en comparant (dans n cas choisis aléatoirement) son résultat sur deux listes de même taille avec celui retourné par `mapcar` sur les mêmes listes.