

TD 2 - Boucles et listes

On justifiera la validité et la terminaison des fonctions.

Boucles

- 1 – Soit u la suite réelle définie par $u_0 = 1$ et pour tout $n \in \mathbb{N}$, $u_{n+1} = f(u_n)$ avec $f : x \mapsto \frac{1}{4}x^2$.
Ecrire une fonction `suiteu` telle que pour un `int n`, `suiteu(n)` renvoie le terme de rang `n` de la suite u .
- 2 – Ecrire une fonction `recur2` telle que pour un `int n`, `recur2(n)` renvoie le terme de rang `n` de la suite v définie par :
 $v_0 = 1, v_1 = 3$, et $\forall n \in \mathbb{N}, v_{n+2} = v_{n+1} - 2v_n$.
- 3 – Ecrire une fonction `recherche` telle que pour une liste `l` et un objet `a`, `recherche(l,a)` renvoie `True` si `a` est un élément de `l` et `False` sinon.
- 4 – Ecrire une fonction `nombre` telle que pour une liste `l` et un objet `a`, `nombre(l,a)` renvoie le nombre d'occurrences de `a` dans `l`.
- 5 – Ecrire une fonction `premier` telle que pour un `int n`, `premier(n)` renvoie `True` si `n` est premier ou `False` sinon.

Listes

On pourra utiliser les commandes présentées par des exemples ci-dessous :

```
>>> [1,2,3] + [4,5]
[1, 2, 3, 4, 5]
>>> [1,2,3] * 4
[1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3]
>>> len([5,2,-1])
3
>>> l = [4,2,-1]
>>> l.append(5)
>>> l
[4,2,-1,5]
```

- 5 – Ecrire une fonction qui à une liste renvoie la somme de ses termes.
- 6 – Ecrire une fonction qui à une liste renvoie le plus petit de ses éléments.
Ecrire ensuite une fonction, en adaptant la fonction précédente, qui à une liste renvoie l'indice du plus petit élément.
- 7 – Ecrire une fonction `nb_el` qui à une liste renvoie le nombre de ses éléments distincts.
Par exemple, `nb_el([5,2,-1,2,5,2])` renvoie 2.
On pourra créer une liste auxiliaire initialement vide (`[]`) qu'on remplira des éléments distincts de la liste argument et utiliser la fonction `recherche` ci-dessus.
- 8 – Ecrire une fonction `card_max` qui à une liste renvoie l'élément de la liste ayant le plus grand nombre d'occurrences et ce nombre.
On pourra dans un premier temps utiliser la fonction `nombre` et l'appliquer à chaque élément de la liste.
On s'inspirera ensuite de la fonction `nb_el` pour écrire une fonction qui nécessite moins d'opérations.