Lists

1 Liste Itérative → Python

Type abstrait : Liste itérative	**Python : type list**
λ : Liste	`type(L) = list`
$\lambda \leftarrow$ *liste-vide*	`L = []`
longueur(λ)	`len(L)`
ième(λ, k)	`L[k]`

Exercice 1.1 (ALGO ↦ Python)

Traduire les algorithmes suivants en Python.

1.
```
fonction compte(Element e, Liste λ) : entier
   variables
      entier    i, cpt
debut
   cpt ← 0
   pour i ← 1 jusqu'a longueur(λ) faire
      si e = ieme(λ, i)) alors
         cpt ← cpt + 1
   fin pour
   retourne cpt
fin
```

2.
```
fonction est-present(Element e, Liste λ) : booleen
   variables
      entier    i
debut
   i ← 1
   tant que (i <= longueur(λ)) et (e <> ieme(λ, i)) faire
      i ← i + 1
   fin tant que
   retourne (i <= longueur(λ))
fin
```

Exercice 1.2 (Construire une liste)

```
>>> help(list.append)
Help on method_descriptor:
append(...)
    L.append(object) -> None -- append object to end
>>> L = []
>>> L.append(1)
>>> L.append(2)
>>> L.append(3)
>>> L
[1, 2, 3]
```

Écrire une fonction qui retourne une nouvelle liste de n valeurs *val*.

Exercice 1.3 (Type abstrait $\mapsto$ Python)

Implémenter les opérations suivantes en Python.

1. *L'opération supprimer*

OPÉRATIONS
supprimer : Liste $\times$ Entier $\rightarrow$ Liste

PRÉCONDITIONS
supprimer(λ, k) **est-défini-ssi** $1 \leq k \leq longueur(\lambda)$

AXIOMES
$\lambda \neq$ *liste-vide* & $1 \leq k \leq longueur(\lambda) \Rightarrow longueur(supprimer(\lambda, k)) = longueur(\lambda) - 1$
$\lambda \neq$ *liste-vide* & $1 \leq k \leq longueur(\lambda)$ & $1 \leq i < k$
$\Rightarrow ième(supprimer(\lambda, k), i) = ième(\lambda, i)$
$\lambda \neq$ *liste-vide* & $1 \leq k \leq longueur(\lambda)$ & $k \leq i \leq longueur(\lambda) - 1$
$\Rightarrow ième(supprimer(\lambda, k), i) = ième(\lambda, i + 1)$

AVEC
λ : Liste
k, i : Entier

2. *L'opération insérer*

OPÉRATIONS
insérer : Liste $\times$ Entier $\times$ Élément $\rightarrow$ Liste

PRÉCONDITIONS
insérer(λ, k, e) **est-défini-ssi** $1 \leq k \leq longueur(\lambda) + 1$

AXIOMES
$1 \leq k \leq longueur(\lambda) + 1 \Rightarrow longueur(insérer(\lambda, k, e)) = longueur(\lambda) + 1$
$1 \leq k \leq longueur(\lambda) + 1$ & $1 \leq i < k \Rightarrow ième(insérer(\lambda, k, e), i) = ième(\lambda, i)$
$1 \leq k \leq longueur(\lambda) + 1$ & $k = i \Rightarrow ième(insérer(\lambda, k, e), i) = e$
$1 \leq k \leq longueur(\lambda) + 1$ & $k < i \leq longueur(\lambda) + 1$
$\Rightarrow ième(insérer(\lambda, k, e), i) = ième(\lambda, i - 1)$

AVEC
λ : Liste
k, i : Entier
e : Élément

Comment implémenter ces opérations "en place" ?

2 Problèmes

Exercice 2.1 (Histogramme)

1. Écrire une fonction qui donne un histogramme des caractères présents dans une chaîne de caractères : une liste de longueur 256 qui donne pour chaque caractère son nombre d'occurences dans la chaîne.
2. Écrire une fonction qui compte le nombre de caractères différents dans une chaînes de caractères.
3. Écrire une fonction qui retourne le caractère le plus fréquent d'une chaîne, ainsi que son nombre d'occurrences.

Exercice 2.2 (Kaprekar)

Procédé de Kaprekar :

Soit un nombre entier à p chiffres.

— Prendre les chiffres pour former deux nombres : le plus grand et le plus petit.

— Soustraire en complétant éventuellement par des 0 à gauche pour obtenir à nouveau un nombre de p chiffres.

— Recommencer le procédé avec le résultat.

Remarques :

— Nous travaillons ici toujours avec p chiffres. Cela signifie que si un résultat intermédiaire est inférieur à 10^{p-1} (par exemple 999 lorsque $p = 4$), les deux nombres seront construits à partir des chiffres du nombre complétés par des 0 (ici 999 et 9990).

— Les chiffres du nombres de départ ne doivent pas être tous égaux.

— Le procédé de Kaprekar peut être utilisé avec un nombre de chiffres p quelconque. Selon ce nombre p, on s'arrêtera avec une valeur fixe, ou on obtiendra un cycle.

Écrire un script Python qui applique le procédé de Kaprekar et affiche les différentes valeurs calculées.

Exercice 2.3 (Ératosthène)

Écrire une fonction qui donne la liste de tous les nombres premiers jusqu'à une valeur n donnée. Utiliser la méthode du "crible d'Eratosthène" (voir wikipedia).