TP N°2 : MANIPULATIONS DE LISTE

I Création d'une liste

Une liste peut se créer à la main : `L = [1,2,3,4]`

OU

Une liste peut se créer **par compréhension** :

```
L = [i + 1 for i in range(4)].
```

Liste par compréhension

Supposons que T est une liste de nombres et f une fonction.
La liste Y des images est la liste :

$$\{f(t) \quad \text{pour} \quad t \quad \text{dans} \quad T\}.$$

On peut rajouter une condition :

```
[x for x in range(1,39) if x %3 == 0]
```

(reste de la division euclidienne de `x` par 3)

II Manipulation d'une liste

Une liste peut être vue comme un tiroir dans lequel chaque élément est contenu dans une boîte. Chaque boîte est numérotée :

- La première façon : la première porte le numéro 0, la deuxième 1, etc...
  Exemple 1 **:** `L=[1,2,3,4]` , L[0] vaut 1, `L[1]` vaut 2, `L[2]` vaut 3, `L[3]` vaut 4.

- La deuxième façon : la dernière porte le numéro -1, l'avant dernière le numéro -2 etc...
  Exemple 2 **:** `L=[1,2,3,4]` , L[-1] vaut 4, `L[-2]` vaut 3, `L[-3]` vaut 2, `L[-4]` vaut 1.

✓ **Les commandes possibles**

Les différentes méthodes sur les listes :

- `len(L)` renvoie la longueur de la chaîne soit le nombre d'éléments de L
- `L.remove(a)` supprime l'élément `a` qui apparait pour la première fois dans la liste `L`. Le contenu de `L` est modifié.
- `L.pop(i)` renvoie et supprime l'élément `L[i]` de `L`. Le contenu de `L` est modifié.
  `L.pop()` renvoie et supprime le dernier élément de la liste.
- `L.index(a)` renvoie le plus petit indice i tel que $L[i] = a$.
- `L*n` ou `n*L` répète n fois la liste `L`.
- `L.append(a)` ajoute à `L`, l'élément `a`. Le contenu de `L` est modifié.
- `L.extend(A)` ajoute à `L` les éléments de la liste `A`. Le contenu de `L` est modifié.
- `L + A` ajoute à `L` les éléments de la liste `A`. Cette instruction crée un nouvel objet.
- `L.sort()` trie la liste dans l'ordre croissant. Le contenu de `L` est modifié.
- `T = L.copy()` ou `T = L[:]` ou `T = L[0 : len(L)]` fait une copie de la liste `L` et crée le nouvel objet `T`. Une modification de L n'entraine pas de modification de `T` contrairement à l'instruction `T = L`, `T` est un alias de `L`, une modification de `L` entraine aussi une modification de `T`.
- `max(L)` , `min(L)` et `sum(L)` revoie le plus grand, le plus petit élément de la liste `L` et la somme de tous les éléments de `L`.

✓ **Extraire les éléments par « slicing »**

Considérons la liste `L=[4,5,6,7,8,9,10,11]`, à l'aide des instructions suivantes, on peut :

- Récupérer les éléments de `L` d'indice 0 jusqu'à 2 : `L[:3]`
- Récupérer les éléments de `L` à partir de l'indice 3 : `L[3:]`
- Récupérer les éléments de `L` d'indice 1 jusqu'à 3 : `L[1:4]`
- Récupérer les éléments de `L` d'indices pairs : `L[0::2]` ou `L[::2]`
  Récupérer les éléments de `L` d'indices pairs à partir de l'indice 4 : `L[4::2]`.
- Récupérer les éléments de `L` d'indices impairs : `L[1::2]`.
  Récupérer les éléments de `L` d'indices impairs à partir de l'indice 3 : `L[3::2]`.
- Récupérer les éléments d'indice i jusqu'à $j-1$ avec un pas p :`L[i:j:p]`.
- Renverser la liste `L[::-1]`.

III Exercices

✓ Création d'une liste

1

1) Créer une liste de n zéros à l'aide d'une fonction `nzeros(n)`
2) Ecrire une fonction `lstindice(L,x)` qui permet de renvoyer la liste des indices de L correspondant aux différentes positions de x.
   Par exemple si $L = [0, 1, 0, 2, 0, 3, 4]$ et $x = 0$, alors on veut obtenir $[0, 2, 4]$.
   On pourra construire cette liste soit par compréhension, soit autrement.
3) Ecrire une fonction `double(L)` qui renvoie une liste dont les éléments sont les éléments de L apparaissant deux fois d'affilée.
   Par exemple si $L = [0, 1, 1, 2, 0]$, `double(L)` doit renvoyer $[0, 0, 1, 1, 1, 1, 2, 2, 0, 0]$.

2 Ecrire une fonction `inverse(L)` qui permet de renvoyer la liste L en sens inverse.

3 Ecrire une fonction `pos(L)` qui renvoie la liste des éléments positifs de L.

4 Dans une urne composée de jetons numérotés de 1 à n. On effectue k tirages avec remise dans cette urne. Créer une fonction de paramètres n et k renvoyant la liste des k numéros tirés.
Puis créer une deuxième fonction de paramètre n retournant une façon de vider l'urne.
Indication : *On pourra utiliser l'instruction `randint(a,b)` du module random qui permet d'obtenir un nombre entier au hasard entre les entiers a et b.*

✓ Recherche dans une liste

5 Ecrire une fonction `appartient` de paramètre une liste `t` et un élément `elem` qui renvoie `True` si `elem` est dans la liste `t` et `False` sinon. (On pourra parcourir la liste.)

6 Ecrire une fonction `inclus(L1,L2)` qui vérifie si tous les éléments de la liste `L1` sont dans la liste `L2`

✓ Recherche du minimum dans une liste de nombres

7

1) Ecrire une fonction `minitab` de paramètre une liste `l` qui renvoie le minimum de cette liste. On ne cherche pas ici à connaître la position du minimum dans la liste, ni à utiliser la fonction min.
2) Ecrire une fonction `minitabindice` de paramètre une liste `l` qui renvoie le minimum de cette liste et le première indice de ce minimum.

8 Créer une fonction `comptage(l,elem)` qui renvoie le nombre de fois où elem `apparaît` dans la liste `l`. (le nombre d'occurrences de `elem` dans `l`.)

✓ Moyenne, produit d'une liste de nombres

9 Ecrire une fonction `moyenne` de paramètre une liste de nombres qui renvoie leur moyenne.

10 Ecrire une fonction `produit(l)` qui renvoie le produit de tous les termes de la liste l.

✓ Recherche du nombre maximal de zéros consécutifs dans une liste

11 Soient un entier naturel n et une liste `L` de longueur n dont les termes valent 0 ou 1. On cherche le nombre maximal de 0 consécutifs dans `L`.
Par exemple dans la liste L suivante :

`[0,1,1,1,0,0,0,1,1,0,0,0,0]`,

le nombre maximal de 0 consécutifs vaut 4.

1. Ecrire une fonction `nombreZeros(L,i)` qui admet pour argument une liste L de longueur n, un indice i compris entre 0 et $n-1$ qui retourne :
   - 0 si $L[i] = 1$,
   - les nombres de zéros consécutifs dans L à partir de `L[i]` inclus, si $L[i] = 0$.

   Par exemple, l'appel de `nombreZeros(L,4)` vaut 3.
2. Comment obtenir le nombre maximal de zéros consécutifs d'une liste connaissant la liste des `nombreZeros(L,i)` pour $0 \leqslant i \leqslant n-1$?
   En déduire une fonction `nombreZerosMax(L)`, de paramètre `L`, renvoyant le nombre maximal de zéros consécutifs dans la liste `L`. On utilisera la fonction `nombreZeros`.

✓ **Recherche d'un mot dans une chaine de caractère**

12 Ecrire une fonction `sousmot` de paramètres `mot` et `chaine` deux listes de chaines de caractères qui renvoie `True` si mot apparaît dans chaine et `False` sinon.