

DEVOIR SURVEILLÉ N°3 (2h)

La calculatrice ainsi que le mémo **ne sont pas** autorisés.

Les notions évaluées dans ce devoir sont les suivantes :

- Les listes python et les tableaux numpy.
- La programmation par les tests.
- Utilisation des dictionnaires python.
- La présentation, la rédaction, la lisibilité des scripts

Exercice I (4 points)

Les questions sont indépendantes.

- Q 1** On propose la fonction `f` ci-dessous. Le programmeur n'a pas travaillé proprement et a omis la **signature** (c'est à dire les types des arguments reçus et renvoyés) et la **docstring** de cette fonction. Reconnaître le rôle de cette fonction et proposer sur votre copie une version avec un code plus propre, c'est à dire, **renommer** cette fonction avec un nom bien choisi, préciser sa **signature**, ajouter une **docstring** et proposer des noms de variables plus explicites :

```
def f(a):  
    b = a[0]  
    for i in a :  
        if i > b :  
            b = i  
    return b
```

- Q 2** On a reçu de la part d'une équipe de programmeurs une fonction `mean`. On n'a pas accès à son implémentation, mais l'équipe a bien fait son travail puisque lorsque l'on tape dans la console `help(mean)` on obtient l'affichage ci-dessous :

```
>>> help(mean)  
Help on function mean in module __main__:  
mean(l: list) -> float  
    reçoit une liste d entiers et renvoie  
    la moyenne arithmétique de cette liste
```

Écrire une fonction `test_mean()` qui permet de réaliser (avec le mot clef `assert`) un jeu de tests *raisonnable* (disons 3 tests) pour s'assurer que la fonction fait bien ce qu'elle est sensée faire. On n'oubliera pas d'ajouter un affichage dans le cas où tous les tests sont bien passés.

Exercice II (11 points)

Les questions sont indépendantes.

- Q 3** Donner une instruction python qui permet de créer par **compréhension** la liste `L3` des carrés des entiers naturels, pour les entiers entre 0 et 10 inclus.
- Q 4** On considère une fonction dont la signature est `f(x : int) -> int`. Donner une instruction (avec une boucle ou directement par compréhension) qui permet d'obtenir la liste `L4` des images par la fonction `f` des entiers impairs compris entre 2 et 10 inclus.
- Q 5** Donner le contenu de la variable `L5` après avoir exécuté l'instruction :
- ```
L5 = [i**2 for i in range(10) if i%3==0]
```
- Q 6** On considère la liste `L6 = [ 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h' ]`.  
À l'aide de **slices** donner les instructions python pour obtenir :
- a. `L61=['a', 'b', 'c']`   b. `L62=['g', 'h']`   c. `L63=['b', 'd', 'f', 'h']`   d. `L64=['c', 'd']`
- Q 7** On considère le programme python suivant :

```
import numpy as np
tab0 = np.array([1,2,3,4,5])
tab1 = np.arange(1,5,0.5)
tab2 = tab0*2-1
tab3 = tab0[:2]
tab4 = np.arange(1,9).reshape(2,4)
```

Donner les valeurs des variables `tab1`, `tab2`, `tab3` et `tab4`.

**Exercice III (7 points)**

On considère dans cet exercice l'élection d'un vainqueur à l'issue d'un vote. Les résultats du vote sont stockés dans une **liste python** : chaque vote exprimé est le nom d'un ou d'une candidate.

Par exemple, les résultats pourraient correspondre à la liste :

```
urne = ['A', 'A', 'A', 'B', 'C', 'B', 'C', 'B', 'C', 'B']
```

indiquant que 3 candidats ont obtenus au moins un vote chacun : A, B et C.

On cherche à déterminer le ou les candidats ayant obtenu le plus de suffrages. Pour cela, on propose d'écrire deux fonctions :

- La fonction `depouille(urne : list) -> dict` doit permettre de compter le nombre de votes exprimés pour chaque artiste. Elle prend en paramètre un tableau et renvoie le résultat dans un dictionnaire dont les clés sont les noms des candidats et les valeurs le nombre de votes en leur faveur. Avec la variable `urne` ci-dessus, le dictionnaire renvoyé devrait donc être :

```
{'A': 3, 'B': 4, 'C': 3}
```

- La fonction `vainqueurs(election)` doit désigner le nom du ou des gagnants. Elle prend en paramètre un dictionnaire non vide dont la structure est celle du dictionnaire renvoyé par la fonction `depouille` et renvoie une liste python. Cette liste peut donc contenir plusieurs éléments s'il y a des artistes ex-aequo. Avec le dictionnaire ci-dessus, la liste renvoyée devrait être : `['B']`

On rappelle que l'on peut initialiser un dictionnaire en utilisant une compréhension de dictionnaire. Par exemple les instructions :

```
lst = ['A', 'B', 'B', 'C']
dic = {c:0 for c in lst}
```

permettent de créer le dictionnaire `dic` contenant :

```
{'A': 0, 'B': 0, 'C': 0}
```

**Q8** Compléter la fonction `depouille` en précisant sur votre copie les instructions à écrire dans les zones 1 et 2 ci-dessous :

```
def depouille(urne):
 '''prend en paramètre une liste de suffrages et renvoie un
 dictionnaire avec le nombre de voix pour chaque candidat'''
 resultat = [Zone 1]
 for bulletin in urne:
 [Zone 2]
 return resultat
```

**Q9** Compléter la fonction `vainqueurs` en précisant sur votre copie les instructions à écrire dans les zones 3, 4 et 5 ci-dessous :

```
def vainqueurs(election):
 '''prend en paramètre un dictionnaire non vide avec le nombre de voix
 pour chaque candidat et renvoie la liste des vainqueurs'''
 nmax = 0
 for nvoix in election.values():
 if [Zone 3]
 nmax = [Zone 4]
 liste_finale = [nom for nom in election if [Zone 5]]
 return liste_finale
```

**Q10** Compléter le code ci-dessous qui doit faire afficher la liste du ou des vainqueurs du vote représenté par la liste `urne` ci-dessous :

```
urne = ['A', 'A', 'A', 'B', 'C', 'B', 'C', 'B', 'C', 'B']
[Zone 6]
```