Devoir surveillé N°1 (1h)

La calculatrice ainsi que le mémo ne sont pas autorisés. Le barème est indicatif.

La présentation, la rédaction, la lisibilité des scripts entrent pour une part importante dans l'appréciation des copies.

Exercice I (8 points)

Faire les questions 1 à 8 du QCM joint en annexe.
Attention de ne rien écrire dans les cases des questions 9 à 12 !

Exercice II (12 points)

On rappelle si besoin quelques notions élémentaires sur les listes en python :

- Une liste se définit avec des crochets. Exemples : `L=[ 5 , 2 , 7 ]` ou `L=[]` pour une liste vide.
- On peut accéder à un élément d'une liste en utilisant son indice (sans oublier que la numérotation commence à 0). Exemple : `L[1]` donne 2.
- On peut obtenir la longueur (taille) de la liste avec la fonction « len ».
  Par exemple : `len(L)` donne ici 3.
- On peut rajouter un élément en fin de liste avec la méthode « .append ».
  Par exemple : `L.append(1)` modifie la liste L pour donner : `L=[ 5 , 2 , 7 , 1 ]`
- On peut déterminer si un élément appartient à une liste avec l'opérateur « in ». Exemple : `2 in L` renvoie `True` alors que `9 in L` renvoie `False`.

À noter : pour chaque fonction demandée, on n'oubliera pas de préciser les *docstrings* correspondantes... Pour la première on sera précis et complet, pour les suivantes on pourra se contenter d'un mini résumé.

Q 9 Écrire une fonction `nb_occurrences(L : list, x)->int` qui prend en argument une liste L et un objet x et qui renvoie le nombre d'occurrences de x dans L.
Par exemple : `nb_occurrences([2,5,8,5],5)` doit renvoyer 2 car il y a 2 fois l'objet 5 dans la liste.

Réponse Q9

```
def nb_occurences(L : list, x) -> int:
    """ Cette fonction reçoit une liste L et un élément x et elle
    renvoie le nombre de fois où x apparait dans la liste.
    """
    n = 0
    for elt in L:
        if elt == x :
            n += 1
    return n
```

Q 10 Écrire une fonction `liste_place(L : list,x)->list` qui prend en argument une liste L et un objet x et qui retourne la liste (éventuellement vide) des indices des positions de x dans L.
Par exemple : `liste_place([2,5,8,5],5)` doit renvoyer la liste `[1,3]` puisque l'objet 5 apparait deux fois dans la liste, en positions d'indice 1 et 3.

Réponse Q10

```
def liste_place(L : list, x) -> list:
    """ renvoie la liste des places de x..."""
    places = []
    for i in range(len(L)):
        if L[i] == x :
            places.append(i)
    return places
```

Q 11 Écrire une fonction `supprimer(L : list, indices : list)-> list` qui prend en arguments deux listes L et indices et retourne une **nouvelle** liste L′ obtenue à partir de L en enlevant les termes dont les indices figurent dans indices.
Par exemple : `supprimer([2,5,8,5],[1,3])` doit renvoyer la liste `[2,8]`.

Réponse Q11

```
def supprimer(L : list, indices : list)-> list :
    """ supprime les termes dont les indices sont dans 'indices'...
    """
    L_prime = []
    for i in range(len(L)):
        if i not in indices :
            L_prime.append(L[i])
    return L_prime
```

Q 12 On suppose dans cette question que les variables `L=[1,4,3,5,6,3,4]` et `x=3` sont définies.
En utilisant les fonctions précédentes (même si vous n'avez pas réussi à les coder), écrire une instruction **en une seule ligne**, qui permet d'afficher la liste obtenue à partir de L dont toutes les occurrences de x ont été supprimées.

Réponse Q12

```
print(supprimer(L,liste_place(L,x)))
```