

NOM :

Prénom :

Classe :

Devoir Surveillé 1

Les réponses sont à écrire exclusivement sur ce document

Les différentes questions sont indépendantes les unes des autres.

Vous accorderez un soin particulier aux justifications que vous apporterez à vos réponses, qui auront une grande importance dans votre note.

1 Références partagées

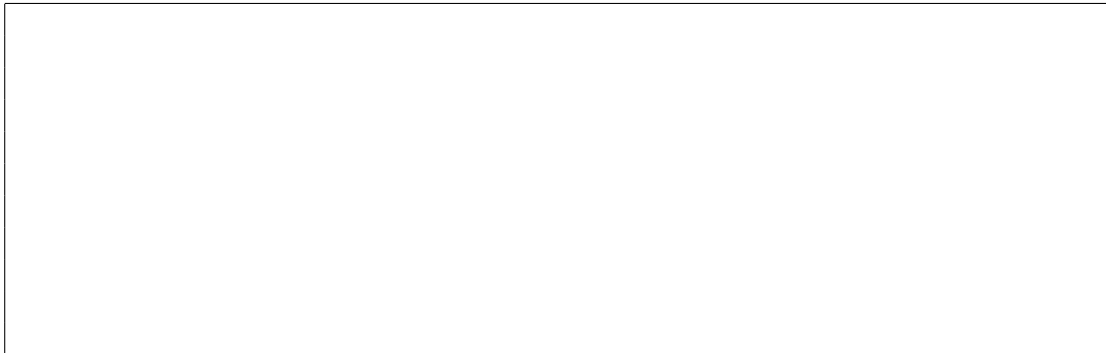
Dans chacun des trois cas suivants, qu'affiche `print(a[1])`. Justifier en décrivant (éventuellement à l'aide d'un dessin) les évolutions des états de la mémoire.

Question 1. Premier cas :

```
a=[1,3]
```

```
b=a
```

```
b[1]=6
```



Question 2. Deuxième cas :

```
a=["alpha","beta"]
```

```
b=a[:]
```

```
b[1]="gamma"
```

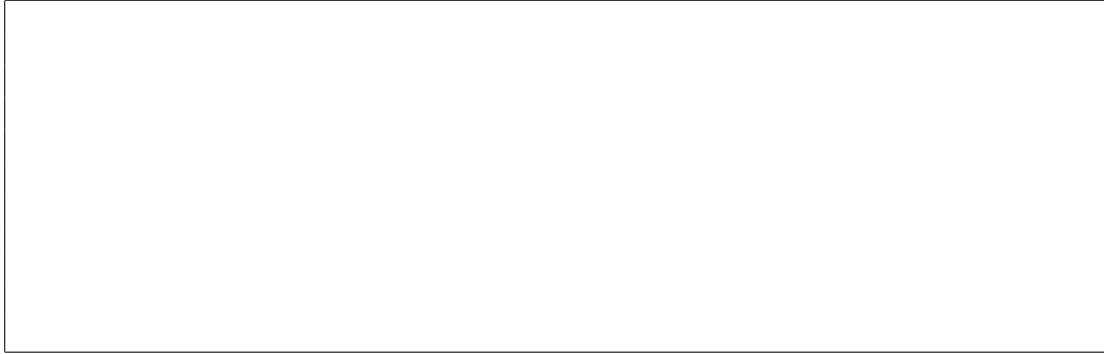


Question 3. Premier cas :

```
a=[True,[True]]
```

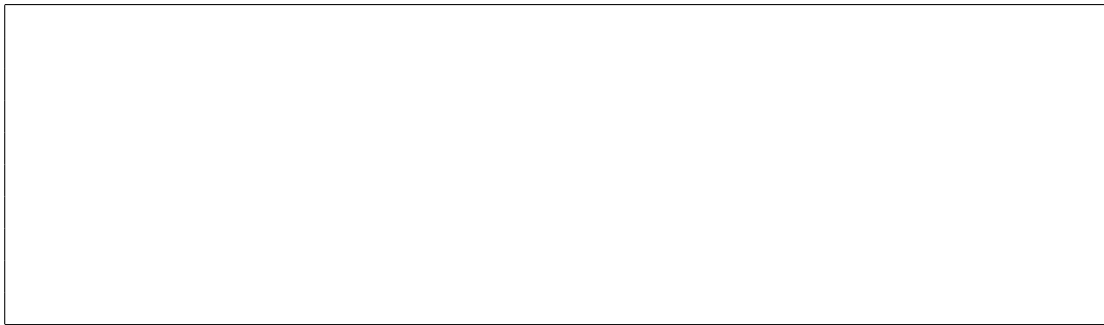
```
b=a[:]
```

```
b[1][0]=False
```



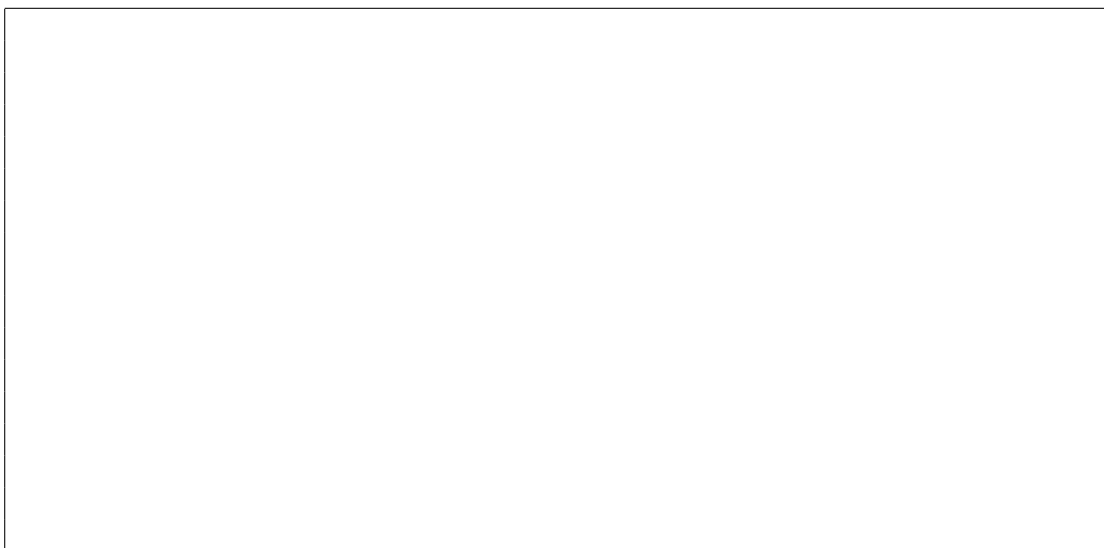
2 Parcours de listes

Question 4. Écrire une fonction `test01` qui prend en argument une liste `L` et renvoie le booléen `True` si cette liste ne contient que des 0 et des 1 et qui renvoie le booléen `False` sinon.



Question 5. Écrire une fonction `testlongueur` qui prend en argument une liste `L` et renvoie l'entier correspondant à la longueur de la plus grande sous-liste de `L` composée uniquement de 1.

Par exemple, `testlongueur([0,1,0,1,1,1,0,1,0,1,1,1,1])` renverra 4.



3 Chaîne de caractères

Question 6. Écrire une fonction `chaine`, `caractere` qui prend en argument une chaîne de caractères `chaine` et un caractère `caractere` et qui renvoie le booléen indiquant si `caractere` est présent dans `chaine`.

Question 7. Écrire une fonction `chaine`, `souschaine` qui prend en argument deux chaînes de caractères `chaine` et `souschaine` et qui renvoie le booléen indiquant si la sous-chaîne `souschaine` est présente dans `chaine`.

Question 8. Un fichier externe, nommé "smalltalk.txt" et stocké dans le même répertoire que les scripts Python contient un texte.

Écrire une suite d'instructions qui permet d'ouvrir ce fichier, de le lire et de tester si la chaîne de caractères "info" est présente dans le texte.

4 Listes de listes

Question 9. La fonction `zip` construit un itérateur associant en tuples des éléments de plusieurs listes en s'arrêtant quand la liste la plus courte est épuisée. Par exemple :

```
list(zip([1,3,5],[2,4,6,8])) = [(1,2), (3,4), (5,6)]
```

Écrire une fonction `monzip2` qui prend en arguments deux listes `L1` et `L2` et qui renvoie la liste de tuples de 2 éléments qui serait renvoyée par la commande `list(zip(L1,L2))`

Question 10. Écrire une fonction `monzip` qui prend en argument une liste de listes `LL` et qui renvoie la liste de tuples de $n=\text{len}(LL)$ éléments qui serait renvoyée par la commande `list(zip(LL))`