

Compléments sur le TP3

Je comble ici quelques oublis du C.R. du TP3

7.7 Version maison de reverse

Je remets juste ce paragraphe en forme, le `sort` s'était mis au milieu de la version que je vous avais donné je crois. Je n'ai pas mis de corrigé pour le `sort` c'était volontaire.

Une solution est la suivante :

```
def mon_reverse(L):
    temp=L[:]# cree une shallow copy de L,
    for i in range(len(L)):
        L[i]=temp[len(L)-i-1]
```

Commentaires sur ce script :

- La variable `temp` est **temporaire** : cette copie de `L` va permettre de modifier `L` en gardant jusqu'au bout le souvenir de ce qu'il y a au départ de `L` grâce à `temp`.
- Dans la boucle `for`, le compteur `i` parcourt *tous* les indices de la liste : ces indices vont de 0 à `n-1` si on appelle `n=len(L)`.
- Dans cette boucle `for`, le `len(L)-i-1` parcourt les entiers de `n-1` à 0, donc `temp[len(L)-i-1]` parcourt les valeurs de la liste de l'entrée `n-1` à 0.
- Avez-vous compris pourquoi on a besoin de `temp` : que se passerait-il si on avait écrit en dernière ligne : `L[i]=L[len(L)-i-1]` ?

Remarque : on peut aussi faire un `reverse` qui n'utilise pas de liste auxiliaire en échangeant `L[i]` avec `L[n-1-i]` grâce à une troisième variable tampon... (moins cher en mémoire qu'une liste).

7.8 Version maison de insert

```
def moninsert(L,i,a):
    """insere objet a à l'indice i de L, en décalant les suivants, L est modifiée"""
    Queue=L[i:]
    L[i:]=[a]+Queue
```

8 Le module time

Attention : sur la dernière mise à jour de Python, le module `time` n'a plus de fonction `clock` mais une fonction `time` qui la remplace.