Correction du devoir n° 4 sur la stabilité des tris

Questions n°1 :

Oui le tri par insertion est bien un tri en place, car on trie la liste à l'intérieure d'elle-même, ce qui constitue une difficulté technique, mais permet de diminuer de moitié la mémoire nécessaire pour sctocker les données.

Ce tri est également stable.
L'ordre d'appartition de deux valeurs identiques est conservé après le tri.

En effet, l'algorithme parcourt les éléments dans l'ordre, et place la première des valeurs identiques rencontrées le plus à gauche possible dans la liste triée, comme le montre l'exemple suivant :

```
[7, 9, 10, 5, 12, 7, 2, 8]
[7, 9, 10, 5, 12, 7, 2, 8]
[7, 9, 10, 5, 12, 7, 2, 8]
[5, 7, 9, 10, 12, 7, 2, 8]
[5, 7, 9, 10, 12, 7, 2, 8]
[5, 7, 7, 9, 10, 12, 2, 8]
[2, 5, 7, 7, 9, 10, 12, 8]
[2, 5, 7, 7, 8, 9, 10, 12]
```

Questions n°2 :

En revanche, le tri par sélection n'est pas stable, comme le montre l'exemple simple suivant :

au départ la liste :	[2, 3, 3, 8, 2, 6, 1]
première étape :	[1, 3, 3, 8, 2, 6, 2]
deuxième étape :	[1, 2, 3, 8, 3, 6, 2]
troisième étape :	[1, 2, 2, 8, 3, 6, 3]
quatrième étape :	[1, 2, 2, 3, 8, 6, 3]
cinquième étape :	[1, 2, 2, 3, 3, 6, 8]
sixième étape :	[1, 2, 2, 3, 3, 6, 8]

Il est facile de le rendre stable : à chaque itération, on recherche le premier élément le plus petit de la partie non triée de la liste, et on le place juste avant le premier élément de la partie non triée de la liste.

En modifiant la fontion de tri par sélection ainsi, on rend le tri stable :

- On parcourt la liste du début à l'avant dernier terme, avec l'indice k allant de 0 à n - 2 :
- La plus petite valeur est celle d'indice k au début de chaque étape
- On recherche s'il existe une valeur plus petite que celle d'indice k dans le reste de la liste
- Si c'est le cas :
- On insère cette valeur à la position k dans la liste
- Puis on supprime cette valeur de la liste, qui est maintenant à la position k + 1

L'algorithme en pseudo code peut s'écrire à peu près ainsi :

```
liste[0, …, n - 1] est la liste à triée
n est la longueur de la liste

pour k allant de 0 à n - 2 faire :
        index_mini = k
        pour j allant de k + 1 à n - 1 faire :
                si  liste[ j ] < liste[ index_mini ] alors :
                        index_mini = j
        si index_mini  ≠  k alors :
                insérer l'élément liste[ index_mini ] à la position k dans la variable liste
                supprimer l'élément d'indice index_mini + 1 de la liste
```

En version Python, cela peut être écrit en utilisant les méthodes sur les listes.

```python
def tri_selection_stable(tableau):
    for k in range(len(tableau) - 1):
        print(tableau)
        index_mini = k
        for j in range(k + 1, len(tableau)):
            if tableau[j] < tableau[index_mini]:
                index_mini = j
        if index_mini != k:
            # on insère la plus petite valeur trouvée au début de la partie non triée de la liste
            tableau.insert(k, tableau[index_mini])
            # puis on supprime cet élément pour qu'il ne reste pas en double
            del tableau[index_mini + 1]
    print(tableau)
```