

Exercice 1

Écrire une fonction `RechercheMinMax` qui prend en paramètre un tableau de nombres non triés `tab`, et qui renvoie la plus petite et la plus grande valeur du tableau sous la forme d'un dictionnaire à deux clés 'min' et 'max'. Les tableaux seront représentés sous forme de liste Python.

Exemples :

```
1 >>> tableau = [0, 1, 4, 2, -2, 9, 3, 1, 7, 1]
2 >>> resultat = rechercheMinMax(tableau)
3 >>> resultat
4 {'min': -2, 'max': 9}
5 >>> tableau = []
6 >>> resultat = rechercheMinMax(tableau)
7 >>> resultat
8 {'min': None, 'max': None}
```

Exercice 2

La fonction `tri_insertion` suivante prend en argument une liste `L` et trie cette liste en utilisant la méthode du tri par insertion.

2.1 Compléter cette fonction pour qu'elle réponde à la spécification demandée.

```
1 def tri_insertion(L):
2     n = len(L)
3     # cas du tableau vide
4     if ...:
5         return L
6     for j in range(1,n):
7         e = L[j]
8         i=j
9         # A l'étape j, le sous-tableau L[0,j-1] est trié
10        # et on insère L[j] dans ce sous-tableau en déterminant
11        # le plus petit i tel que 0 <= i <= j et L[i-1] > L[j].
12        while i > 0 and L[i-1] > ...:
13            i = ...
14        # si i != j, on décale le sous tableau L[i,j-1] 'dun cran
15        # vers la droite et on place L[j], stocké dans e, en position i
16        # la boucle sur k descend de la valeur j à i+1
17        if i != j:
18            for k in range(j,i,...):
19                L[k] = L[...]
20            L[i] = ...
21
22    return L
```

Exemples :

```
1 >>> tri_insertion([2,5,-1,7,0,28])
2 [-1, 0, 2, 5, 7, 28]
3 >>> tri_insertion([10,9,8,7,6,5,4,3,2,1,0])
```

4 [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

- 2.2 Utiliser l'un des exemples ci-dessus pour créer un test d'assertion et tester votre fonction. Mettre ce test dans le corps du programme, en dehors de toute fonction.