Parcours de listes

Les deux types de parcours de liste ont une complexité en O(n).

Le **parcours par indice**, avec lequel la variable de boucle va prendre successivement tous les indices valides pour cette liste. On utilise la fonction `len` sur la liste dans le `range`.

```
ma_liste = ['NSI', 16, 12.4, (1, 4)]
for i in range( len(ma_liste) ) :
    print(ma_liste[i])
```

```
NSI
16
12.4
(1, 4)
```

Le **parcours par valeur**, avec lequel la variable de boucle va prendre successivement toutes les valeurs contenues dans la liste. On n'utilise cette fois pas de range, mais directement la liste.

```
ma_liste = ['NSI', 16, 12.4, (1, 4)]
for e in ma_liste :
    print(e)
```

```
NSI
16
12.4
(1, 4)
```

Exemple

```
def MoyennePonderee(lNote : list, lCoef : list) -> float :
    """
    Calcul d'une moyenne pondérée avec deux listes.
    >>> MoyennePonderee([5, 10, 15], [1, 2, 2])
    11.0
    """

    # Précondition
    assert type(lNote) == list and type(lCoef) == list
    assert len(lNote) == len(lCoef)
    for e in lNote: # Parcours par valeur
        assert type(e) == int or type(e) == float
    for e in lCoef: # Parcours par valeur
        assert type(e) == int or type(e) == float

    # Calculs
    lRes = [0, 0]
    for i in range(len(lNote)): # Parcours par indice de deux listes
        lRes[0] += lNote[i] * lCoef[i]
        lRes[1] += lCoef[i]
    return lRes[0] / lRes[1]
```