

## Piles & Files : Exercices

### Exercice 1 :

On rappelle ci-contre le code de la classe `File` implémentée en TD/TP.

Décrire les modifications à apporter à la classe `File` pour qu'elle dispose d'un attribut privé `_longueur` indiquant le nombre d'éléments qu'elle contient. Cet attribut :

- sera modifié à chaque enfilage ou défilage d'un élément,

- sera renvoyé à l'utilisateur par la méthode `longueur()`.

```
1 from module_pile import Pile
2
3 class File:
4     '''Classe implémentant une structure de file grâce à deux piles'''
5
6     def __init__(self):
7         self._entree = Pile()
8         self._sortie = Pile()
9
10    def _transfert(self):
11        while not self._entree.est_vide():
12            self._sortie.empiler(self._entree.depiler())
13
14    def enfiler(self, element):
15        self._entree.empiler(element)
16
17    def defiler(self):
18        assert not self.est_vide()
19        if self._sortie.est_vide():
20            self._transfert()
21        return self._sortie.depiler()
22
23    def est_vide(self):
24        return (self._entree.est_vide() and self._sortie.est_vide())
```

### Exercice 2 :

On considère une chaîne de caractères comportant à la fois des parenthèses rondes ( et ) ainsi que des parenthèses carrées [ et ]. On dit que la chaîne est bien parenthésée si chaque ouvrante est associée à une unique fermante de même forme et réciproquement.

Ecrire une fonction prenant en paramètres une chaîne de caractères contenant entre autres les parenthèses décrites et qui renvoie `True` si la chaîne est bien parenthésée et `False` sinon.

### Exercice 2-bis :

On dit qu'une chaîne de caractères comprenant, entre autres choses, des parenthèses ( et ) est bien parenthésée lorsque chaque parenthèse ouvrante est associée à une unique fermante et réciproquement.

Ecrire une fonction prenant en paramètres une chaîne bien parenthésée `s` et l'indice `i` d'une parenthèse fermante et qui renvoie l'indice de la parenthèse ouvrante associée.