

# ALG. Piles Files. Exercices

Pour les exercices suivants, on utilisera le fichier `ALG.Piles_Files_Exercices` à monter sur Jupyter (cela évitera de tout recopier 😊). Il contient les classes `Stack` et `File`.

## Exercice 1 : de nouvelles méthodes pour la classe `Stack`

- 1/ **Ajouter** la méthode `clear(self)` qui permet de supprimer (proprement) tous les éléments d'une pile.
- 2/ **Ajouter** la méthode `size(self)` qui permet de donner le nombre d'éléments de la pile.
- 3/ **Proposer** un jeu de tests.
- 4/ Quelle est la **complexité** des deux méthodes précédentes ?

Aide : choisir la définition adaptée de la classe `Stack`.

## Exercice 2 : séparation d'une pile en deux

**Ecrire** une fonction qui prend en paramètre une pile contenant des nombres pairs et impairs et qui renvoie deux nouvelles piles : une contenant des nombres pairs et l'autre les nombres impairs.

**On veillera à ne pas modifier l'ordre initial des nombres.**

## Exercice 3 : vérification du bon parenthésage

On dispose d'une chaîne de caractères contenant uniquement des parenthèses ouvrantes et fermantes comme « `((()))((()))` ».

Un **parenthésage est correct** si, en parcourant la chaîne de parenthèses de gauche à droite, le nombre de parenthèses déjà ouvrantes est supérieur ou égal au nombre de parenthèses fermantes. De plus, dans la chaîne complète, il doit y avoir autant de parenthèses ouvrantes que de parenthèses fermantes.

- 1/ **Ecrire** un programme qui, en utilisant la classe `Stack`, parcourt la chaîne et empile une valeur (au choix) lorsque le caractère est « `(` » et dépile une valeur lorsque le caractère est « `)` ».
- 2/ **Tester** le code sur les chaînes « `(( ( ) ( ) ) )` », « `(( ) ) ( ) ( )` » et « `(( ( ) ) )` ». Que dire de ses trois cas ?
- 3/ Comment peut-on détecter le **bon parenthésage** ?
- 4/ (\*) **Modifier** le programme précédent pour qu'il prenne en compte également les crochets. Une expression comme « `( a ( [ b c ] d ) )` » est bien parenthésée mais pas « `( a [ ( b c ) ] d )` »

## Exercice 4 : affichage d'une file construite à partir de deux piles

- 1/ **Ajouter** la méthode `__str__(self)` qui permet d'afficher le contenu de la file comme une « queue ». Par exemple, si on y ajoute 1 puis « 3 » puis « kikoo », elle devra afficher 1 ; 3 ; kikoo.
- 2/ **Faire** le jeu de tests suivant :

```
# Jeu de tests
myFile = File()
myFile.add_back("45")
myFile.add_back("a")
myFile.add_back(-1)
print(myFile) # Attendu |-1|a|45|
myFile.pop_front()
print(myFile) # Attendu |-1|a|
myFile.add_back("yop")
myFile.add_back("0")
print(myFile) # Attendu [0|yop|-1|a|
```

Aide : penser à utiliser la fonction `deepcopy()` qui permet une copie en profondeur d'un objet, cela pourrait éviter des effets de bords.

Rappel : écrire l'instruction `from copy import deepcopy` pour pouvoir utiliser cette fonction.