

## TD Structure de données : Les files

Pour chaque exercice, on utilisera un objet <type 'list'> pour représenter une file et toutes les méthodes disponibles sur ces objets.

### Exemple

```
>>> file = [5, 3, 7, 1]
>>> print("Le premier élément de la file :", file[0])
Le premier élément de la file : 5
>>> print("Le dernier élément :", file[-1])
Le dernier élément : 1
>>> file.append(8)
>>> print(file)
[5, 3, 7, 1, 8]
>>> sortie = file.pop(0)
>>> print(sortie)
5
```

### Exercice 1 :

1. En vous inspirant de ce que l'on a vu pour la classe Pile(), réaliser cette implémentation.

```
class File:
    """
    classe File
    création d'une instance File avec une liste
    """
    def __init__(self):
        """
        Initialisation d'une file vide
        """
        .....
    def vide(self):
        """
        teste si la file est vide
        """
        .....
    def enfiler(self, x):
        """
        enfile l'élément x
        """
        .....
    def defiler(self):
        """
        defile
        """
        .....
    def taille(self):
        """
        renvoie le nombre d'éléments de la file
        """
```

```

        """
        .....
def sommet(self):
    """
    sommet de la file sans defiler
    """
    .....

```

2. Écrire les instructions permettant de :

- créer une file
- la remplir avec les entiers 0,2,4,6,8
- la faire afficher
- "défiler" la file en faisant afficher l'élément récupéré
- afficher sa taille

### Exercice 2 :

Réaliser une file avec deux piles. Dans cette version, on a besoin d'une pile pour ajouter les nouveaux éléments et d'une seconde pour prendre les éléments à retirer.

```

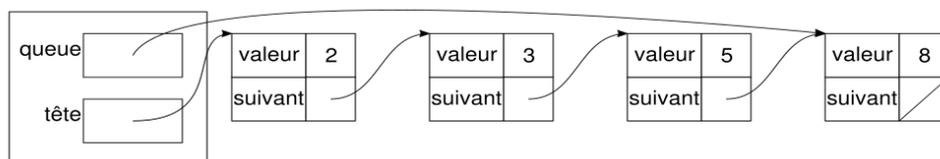
class File :
    def __init__(self):
        self.entree = Pile()
        self.sortie = Pile()
    def vide(self) :
        .....
    def ajouter(self, x):
        .....
    def retirer(self):
        .....

```

Tester votre implémentation avec les instructions précédentes.

### Exercice 3 :

Refaire le même exercice avec une liste chaînée.



### Exercice 4 :

On dispose d'une file, écrire une fonction qui renvoie la file inversée (l'élément de la tête sera situé à la queue et ainsi de suite). On utilisera une pile et seulement les méthodes associées aux piles et aux files.