

TP 1 PARTIE B

Algorithmique et structures des données

L3I MIAE 2025-26

La Partie B du TP doit être rendu sur AMETICE en 2 fichiers :

- Le premier fichier nommé: VOTRENOM_TP1_B_QGEN.py
- Le deuxième fichier nommé : VOTRENOM_TP1_B_CLPERS.py

Exercice 5 (fichier VOTRENOM_TP1_B_QGEN)

Définir une version *générique* de tri rapide `trirapide_gen` qui peut trier **toute** liste des données.

L'algorithme prend comme paramètre additionnel une fonction `compare`.

On suppose que l'appel

`compare(a, b)` retourne :

`True` si **a** est *strictement plus grand* de **b** dans **l'ordre considéré** et `False` sinon.

Par exemple (pas significatif) ! pour trier d'une façon **croissante** une liste d'entiers on passera à l'algorithme la fonction :

```
def compare_int(a,b):  
    if a > b:  
        return True  
    else:  
        return False
```

et pour trier une liste d'entiers d'une façon **décroissante** on passera à l'algorithme la fonction

```
def comparinv_int(a,b):  
    if a < b:  
        return True  
    else:  
        return False
```

Les appels seront respectivement

```
trirapide(liste, p, r, compare)  
trirapide(liste, p, r, comparinv)
```

Vous pouvez tester l'algorithme avec les fonctions de comparaison ci-dessus, mais le fichier à rendre doit contenir **seulement** `trirapide_gen` et les fonctions qui sont appelées par celui-ci.

Exercice 6 (fichier VOTRENOM_TP1_B_CLPERS.py)

1) Définir la classe `Personne` avec attributs

- `nom`
- `age`
- `salaire`

indiquant le nom, age et salaire (mensuel) d'une personne. La classe contient seulement deux méthodes `init` et `affiche`:

```
__init__(self, nom, age, salaire)  
affiche(self)
```

2) générer une liste de personnes (au moins 8)

3) Utiliser l'algorithme `trirapide_gen` pour trier la liste :

- a) En ordre croissant d'age
- b) En ordre décroissant de salaire

Il faut importer l'algorithme par l'instruction *au début du fichier* :

```
from VOTRENOM_TP1_B_QGEN import myquicksortgen
```