

BACCALAURÉAT

SESSION 2023

Épreuve de l'enseignement de spécialité

NUMÉRIQUE et SCIENCES INFORMATIQUES

Partie pratique

Classe Terminale de la voie générale

Sujet n°27

DURÉE DE L'ÉPREUVE : 1 heure

**Le sujet comporte 3 pages numérotées de 1 / 3 à 3 / 3
Dès que le sujet vous est remis, assurez-vous qu'il est complet.**

Le candidat doit traiter les 2 exercices.

EXERCICE 1 (4 points)

Écrire une fonction `recherche_min` qui prend en paramètre un tableau, non vide, de nombres non trié `tab`, et qui renvoie l'indice de la première occurrence du minimum de ce tableau. Les tableaux seront représentés sous forme de listes Python.

Exemples :

```
>>> recherche_min([5])
0
>>> recherche_min([2, 4, 1])
2
>>> recherche_min([5, 3, 2, 2, 4])
2
```

EXERCICE 2 (4 points)

On considère la fonction `separe` qui prend en argument un tableau `tab` dont les éléments sont des 0 et des 1 et qui sépare les 0 des 1 en plaçant les 0 en début de tableau et les 1 à la suite.

```
def separe(tab):
    gauche = 0
    droite = ...
    while gauche < droite:
        if tab[gauche] == 0:
            gauche = ...
        else:
            tab[gauche], tab[droite] = ...
            droite = ...
    return tab
```

Exemples :

```
>>> separe([1, 0, 1, 0, 1, 0, 1, 0])
[0, 0, 0, 0, 1, 1, 1, 1]

>>> separe([1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0])
[0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1]
```

Description d'étapes effectuées par la fonction `separe` sur le tableau ci-dessous :

```
tab = [1, 0, 1, 0, 1, 0, 1, 0]
```

Etape 1 : on regarde la première case, qui contient un 1 : ce 1 va aller dans la seconde partie du tableau final et on l'échange avec la dernière case.

Il est à présent bien positionné : on ne prend plus la dernière case en compte.

```
tab = [0, 0, 1, 0, 1, 0, 1, 1]
```

Etape 2 : on regarde à nouveau la première case, qui contient maintenant un 0 : ce 0 va aller dans la première partie du tableau final et est bien positionné : on ne prend plus la première case en compte.

```
tab = [0, 0, 1, 0, 1, 0, 1, 1]
```

Etape 3 : on regarde la seconde case, qui contient un 0 : ce 0 va aller dans la première partie du tableau final et est bien positionné : on ne prend plus la seconde case en compte.

```
tab = [0, 0, 1, 0, 1, 0, 1, 1]
```

Etape 4 : on regarde la troisième case, qui contient un 1 : ce 1 va aller dans la seconde partie du tableau final et on l'échange avec l'avant-dernière case.

Il est à présent bien positionné : on ne prend plus l'avant-dernière case en compte.

```
tab = [0, 0, 1, 0, 1, 0, 1, 1]
```

Et ainsi de suite...

```
tab = [0, 0, 0, 0, 1, 1, 1, 1]
```

Compléter la fonction `separe` présentée à la page précédente.