BACCALAURÉAT

SESSION 2024

Épreuve de l'enseignement de spécialité

NUMÉRIQUE et SCIENCES INFORMATIQUES

Partie pratique

Classe Terminale de la voie générale

Sujet n°25

DURÉE DE L'ÉPREUVE : 1 heure

Le sujet comporte 4 pages numérotées de 1 / 4 à 4 / 4
Dès que le sujet vous est remis, assurez-vous qu'il est complet.

Le candidat doit traiter les 2 exercices.

EXERCICE 1 (10 points)

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

Exemples :

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

EXERCICE 2 (10 points)

On considère la fonction `separe` ci-dessous 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):
    '''Separe les 0 et les 1 dans le tableau tab'''
    gauche = 0
    droite = ...
    while gauche < droite:
        if tab[gauche] == 0 :
            gauche = ...
        else :
            tab[gauche] = ...
            tab[droite] = ...
            droite = ...
    return tab
```

Compléter la fonction `separe` ci-dessus.

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, 1]
```

Description d'étapes effectuées par la fonction `separe` sur le tableau ci-dessous, les caractères `^` indiquent les cases pointées par les indices `gauche` et `droite` :

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

- Étape 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]
       ^                 ^
```

- Étape 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]
          ^              ^
```

- Étape 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]
             ^           ^
```

- Étape 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]
```