BACCALAUREAT

SESSION 2022

Épreuve de l'enseignement de spécialité

NUMERIQUE et SCIENCES INFORMATIQUES

Partie pratique

Classe Terminale de la voie générale

Sujet n°26

DUREE DE L'EPREUVE : 1 heure

Le sujet comporte 2 pages numérotées de 1 / 2 à 2 / 2
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 `RechercheMin` qui prend en paramètre un tableau 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 liste Python.

Exemples :

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

EXERCICE 2 (4 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):
    i = 0
    j = ...
    while i < j :
        if tab[i] == 0 :
            i = ...
        else :
            tab[i], tab[j] = ...
            j = ...
    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]
```

```python
def indice_du_min(tab):
    n = len(tab)
    if n == 0:
        return None

    i_min = 0
    v_min = tab[0]

    for i in range(1, n):
        if tab[i] < v_min:
            i_min = i
            v_min = tab[i]

    return i_min


assert indice_du_min([5]) == 0
assert indice_du_min([2, 4, 1]) == 2
assert indice_du_min([5, 3, 2, 2, 4]) == 2
```

```python
def separe(tab):
    i = 0
    j = len(tab) - 1
    while i < j :
        if tab[i] == 0 :
            i = i + 1
        else :
            tab[i], tab[j] = tab[j], tab[i]
            j = j - 1
    return tab


assert separe([1, 0, 1, 0, 1, 0, 1, 0]) == [0, 0, 0, 0, 1, 1, 1, 1]
assert 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]
```