BACCALAURÉAT

SESSION 2023

Épreuve de l'enseignement de spécialité

NUMÉRIQUE et SCIENCES INFORMATIQUES

Partie pratique

Classe Terminale de la voie générale

Sujet n°11

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)

On modélise la représentation binaire d'un entier non signé par un tableau d'entiers dont les éléments sont 0 ou 1. Par exemple, le tableau `[1, 0, 1, 0, 0, 1, 1]` représente l'écriture binaire de l'entier dont l'écriture décimale est

`2**6 + 2**4 + 2**1 + 2**0 = 83`.

À l'aide d'un parcours séquentiel, écrire la fonction `convertir` répondant aux spécifications suivantes :

```
def convertir(tab):
    """
     tab est un tableau d'entiers, dont les éléments sont 0 ou 1,
     et représentant un entier écrit en binaire.
     Renvoie l'écriture décimale de l'entier positif dont la
     représentation binaire est donnée par le tableau tab
    """
```

Exemple :

```
>>> convertir([1, 0, 1, 0, 0, 1, 1])
83
>>> convertir([1, 0, 0, 0, 0, 0, 1, 0])
130
```

EXERCICE 2 (4 points)

La fonction `tri_insertion` suivante prend en argument une liste `tab` et trie cette liste en utilisant la méthode du tri par insertion. Compléter cette fonction pour qu'elle réponde à la spécification demandée.

On rappelle le principe du tri par insertion : on considère les éléments à trier un par un, le premier élément constituant, à lui tout seul, une liste triée de longueur 1. On range ensuite le second élément pour constituer une liste triée de longueur 2, puis on range le troisième élément pour avoir une liste triée de longueur 3 et ainsi de suite… A chaque étape, le premier élément de la sous-liste non triée est placé dans la sous-liste des éléments déjà triés de sorte que cette sous-liste demeure triée.
Le principe du tri par insertion est donc d'insérer à la n-ième itération, le n-ième élément à la bonne place.

```
def tri_insertion(tab):


n = len(tab)
for i in range(1, n):
    valeur_insertion = tab[...]
    # la variable j est utilisée pour déterminer où placer la
valeur à insérer
    j = ...
    # tant qu'on a pas trouvé la place de l'élément à insérer
    # on décale les valeurs du tableau vers la droite
    while j > ... and valeur_insertion < tab[...]:
        tab[j] = tab[j-1]
        j = ...
    tab[j] = ...
```

Exemple :

```
>>> liste = [9, 5, 8, 4, 0, 2, 7, 1, 10, 3, 6]

>>> tri_insertion(liste)

>>> liste
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```