

BACCALAURÉAT

SESSION 2023

Épreuve de l'enseignement de spécialité

NUMÉRIQUE et SCIENCES INFORMATIQUES

Partie pratique

Classe Terminale de la voie générale

Sujet n°18

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 `max_et_indice` qui prend en paramètre une liste non vide `tab` de nombres entiers et qui renvoie la valeur du plus grand élément de cette liste ainsi que l'indice de sa première apparition dans cette liste.

L'utilisation de la fonction native `max` n'est pas autorisée.

Ne pas oublier d'ajouter au corps de la fonction une documentation et une ou plusieurs assertions pour vérifier les pré-conditions.

Exemples :

```
>>> max_et_indice([1, 5, 6, 9, 1, 2, 3, 7, 9, 8])
(9, 3)
>>> max_et_indice([-2])
(-2, 0)
>>> max_et_indice([-1, -1, 3, 3, 3])
(3, 2)
>>> max_et_indice([1, 1, 1, 1])
(1, 0)
```

EXERCICE 2 (4 points)

L'ordre des gènes sur un chromosome est représenté par un tableau `ordre` de n cases d'entiers distincts deux à deux et compris entre 1 et n .

Par exemple, `ordre = [5, 4, 3, 6, 7, 2, 1, 8, 9]` dans le cas $n=9$.

On dit qu'il y a un point de rupture dans `ordre` dans chacune des situations suivantes :

- la première valeur de `ordre` n'est pas 1 ;
- l'écart entre deux gènes consécutifs n'est pas égal à 1 ;
- la dernière valeur de `ordre` n'est pas n .

Par exemple, si `ordre = [5, 4, 3, 6, 7, 2, 1, 8, 9]` avec $n = 9$, on a

- un point de rupture au début car 5 est différent de 1
- un point de rupture entre 3 et 6 (l'écart est de 3)
- un point de rupture entre 7 et 2 (l'écart est de 5)
- un point de rupture entre 1 et 8 (l'écart est de 7)

Il y a donc 4 points de rupture.

Compléter les fonctions Python `est_un_ordre` et `nombre_points_rupture` proposées à la page suivante pour que :

- la fonction `est_un_ordre` renvoie `True` si le tableau passé en paramètre représente bien un ordre de gènes de chromosome et `False` sinon ;
- la fonction `nombre_points_rupture` renvoie le nombre de points de rupture d'un tableau passé en paramètre représentant l'ordre de gènes d'un chromosome.

```
def est_un_ordre(tab):
    '''
    Renvoie True si tab est de longueur n et contient tous les
    entiers de 1 à n, False sinon
    '''
    for i in range(1,...):
        if ...:
            return False
    return True

def nombre_points_rupture(ordre):
    '''
    Renvoie le nombre de point de rupture de ordre qui représente
    un ordre de gènes de chromosome
    '''
    assert ... # ordre n'est pas un ordre de gènes
    n = len(ordre)
    nb = 0
    if ordre[...] != 1: # le premier n'est pas 1
        nb = nb + 1
    i = 0
    while i < ...:
        if ... not in [-1, 1]: # l'écart n'est pas 1
            nb = nb + 1
        i = i + 1
    if ordre[...] != n: # le dernier n'est pas n
        nb = nb + 1
    return nb
```

Exemples :

```
>>> est_un_ordre([1, 6, 2, 8, 3, 7])
False
>>> est_un_ordre([5, 4, 3, 6, 7, 2, 1, 8, 9])
True
>>> nombre_points_rupture([5, 4, 3, 6, 7, 2, 1, 8, 9])
4
>>> nombre_points_rupture([1, 2, 3, 4, 5])
0
>>> nombre_points_rupture([1, 6, 2, 8, 3, 7, 4, 5])
7
>>> nombre_points_rupture([2, 1, 3, 4])
2
```