

BACCALAUREAT

SESSION 2022

Épreuve de l'enseignement de spécialité

NUMERIQUE et SCIENCES INFORMATIQUES

Partie pratique

Classe Terminale de la voie générale

Sujet n°6

DUREE DE L'ÉPREUVE : 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 `maxi` qui prend en paramètre une liste `tab` de nombres entiers et qui renvoie un couple donnant le plus grand élément de cette liste ainsi que l'indice de la première apparition de ce maximum dans la liste.

Exemple :

```
>>> maxi([1,5,6,9,1,2,3,7,9,8])
(9,3)
```

EXERCICE 2 (4 points)

La fonction `recherche` prend en paramètres deux chaînes de caractères `gene` et `seq_adn` et renvoie `True` si on retrouve `gene` dans `seq_adn` et `False` sinon.

Compléter le code Python ci-dessous pour qu'il implémente la fonction `recherche`.

```
def recherche(gene, seq_adn):
    n = len(seq_adn)
    g = len(gene)
    i = ...
    trouve = False
    while i < ... and trouve == ... :
        j = 0
        while j < g and gene[j] == seq_adn[i+j]:
            ...
        if j == g:
            trouve = True
        ...
    return trouve
```

Exemples :

```
>>> recherche("AATC", "GTACAAATCTTGCC")
True
>>> recherche("AGTC", "GTACAAATCTTGCC")
False
```

```

"""
Author: Pascal Padilla
Source: correction de l'exercice 1 du sujet 6 des épreuves pratiques NSI 2022
"""

from doctest import testmod

def maxi(tab: list) -> tuple[int, int]:
    """Renvoie le plus grand élément du tableau 'tab'
    et son indice.

    Args:
        tab (list): tableau à parcourir

    Returns:
        tuple[int, int] : couple de nombres entiers :
            * valeur maximale du tableau
            * indice de cette valeur

    Exemples et tests:
    >>> maxi([1,5,6,9,1,2,3,7,9,8])
    (9, 3)
    """

    # invariant de boucle
    # * v_max est la valeur maximale de la portion tab[0..i]
    # * i_max est l'indice de v_max

    # initialisation (avec la première valeur du tableau):
    # * i : 0
    # * v_max : tab[0]
    # * i_max : 0
    i_max = 0
    v_max = tab[0]

    # condition d'arrêt (tout le tableau est parcouru):
    # * i == len(tab)
    for i in range(1, len(tab)):

        # mise à jour de v_max et i_max si la
        # valeur courante du tableau est plus grande
        # que le maximum trouvé jusqu'à présent
        if tab[i] > v_max:
            i_max = i
            v_max = tab[i]

    # sortie de boucle : tout le tableau est parcouru
    # et grâce aux invariants, nous avons la valeur
    # maximale et son indice.
    return (v_max, i_max)

# Tests de l'énoncé: méthode classique
assert maxi([1,5,6,9,1,2,3,7,9,8]) == (9,3)

# Tests de l'énoncé: méthode avec doctest
testmod()

```

```

from doctest import testmod

def recherche(gene, seq_adn):
    """ Recherche la présence de la chaîne gene dans la chaîne seq_adn.

    Args:
        gene (str): chaîne de caractère à rechercher
        seq_adn (str): chaîne de caractère à explorer

    Returns:
        bool: True ssi gene est présent dans seq_adn

    Tests et Exemples:
    >>> recherche("AATC", "GTACAAATCTTGCC")
    True
    >>> recherche("AGTC", "GTACAAATCTTGCC")
    False
    """
    n = len(seq_adn)
    g = len(gene)
    # compteur de la lettre courante dans seq_adn
    i = 0
    trouve = False
    # la boucle se perpétue tant que :
    # * le compteur i ne dépasse pas la valeur limite
    # * ET que le gène na pas encore été trouvé
    while i < n - g + 1 and trouve == False :
        j = 0
        while j < g and gene[j] == seq_adn[i+j]:
            # pour les j premières recherche, le gene correspond
            # et donc on va explorer le caractère suivant
            j = j + 1
        if j == g:
            trouve = True
            # l'exploration du caractère i est terminée
            # que gene ai été trouvé ou pas, on va tenter
            # d'explorer le caractère suivant
            i = i + 1

    return trouve

assert recherche("AATC", "GTACAAATCTTGCC") == True
assert recherche("AGTC", "GTACAAATCTTGCC") == False

testmod()

```