

BACCALAUREAT

SESSION 2022

Épreuve de l'enseignement de spécialité

NUMERIQUE et SCIENCES INFORMATIQUES

Partie pratique

Classe Terminale de la voie générale

Sujet n°22

DUREE DE L'ÉPREUVE : 1 heure

**Le sujet comporte 2 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)

Programmer une fonction `renverse`, prenant en paramètre une chaîne de caractères non vide `mot` et renvoie une chaîne de caractères en inversant ceux de la chaîne `mot`.

Exemple :

```
>>> renverse("informatique")
"euqitamrofni"
```

EXERCICE 2 (4 points)

Un nombre premier est un nombre entier naturel qui admet exactement deux diviseurs distincts entiers et positifs : 1 et lui-même.

Le crible d'Ératosthène permet de déterminer les nombres premiers plus petit qu'un certain nombre `N` fixé.

On considère pour cela un tableau `tab` de `N` booléens, initialement tous égaux à `True`, sauf `tab[0]` et `tab[1]` qui valent `False`, 0 et 1 n'étant pas des nombres premiers.

On parcourt alors ce tableau de gauche à droite.

Pour chaque indice `i` :

- si `tab[i]` vaut `True` : le nombre `i` est premier et on donne la valeur `False` à toutes les cases du tableau dont l'indice est un multiple de `i`, à partir de `2*i` (c'est-à-dire `2*i`, `3*i` ...).

- si `tab[i]` vaut `False` : le nombre `i` n'est pas premier et on n'effectue aucun changement sur le tableau.

On dispose de la fonction `crible`, incomplète et donnée ci-dessous, prenant en paramètre un entier `N` strictement positif et renvoyant un tableau contenant tous les nombres premiers plus petits que `N`.

```
def crible(N):
    """
    Renvoie un tableau contenant tous les nombres premiers plus petits que N
    """
    premiers = []
    tab = [True] * N
    tab[0], tab[1] = False, False
    for i in range(..., N):
        if tab[i] == ...:
            premiers.append(...)
            for multiple in range(2*i, N, ...):
                tab[multiple] = ...
```

```
    return premiers
```

```
assert crible(40) == [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37]
```

Compléter le code de cette fonction.

```
"""
Author: Pascal Padilla
Source: correction de l'exercice 1 du sujet 22 des épreuves pratiques NSI 2022
Remarque : je propose 2 versions supplémentaires
"""
```

```
from doctest import testmod
```

```
def renverse(chaine: str) -> str:
    """
    renvoie une chaîne de caractères en inversant `chaine`

    Args:
        chaine (str): chaîne de caractères non vide

    Returns:
        str: chaîne écrite inversée

    Tests et exemple:
    >>> renverse("informatique")
    'euqitamrofni'
    """
    # initialisation de la sortie avec une chaîne vide
    chaine_inverse = ""

    # parcours de chaque lettre de chaîne du début vers la fin
    for i in range(len(chaine)):
        lettre = chaine[i]

        # ajout de la lettre courante au début de la chaîne inversée
        chaine_inverse = lettre + chaine_inverse

    return chaine_inverse

def renverse_pythonnesque_1(chaine):
    """
    renvoie une chaîne de caractères en inversant `chaine`

    Args:
        chaine (str): chaîne de caractères non vide

    Returns:
        str: chaîne écrite inversée

    Tests et exemple:
    >>> renverse_pythonnesque_1("informatique")
    'euqitamrofni'
    """
    return chaine[::-1] # hors programme

def renverse_pythonnesque_2(chaine):
    """
    renvoie une chaîne de caractères en inversant `chaine`

    Args:
        chaine (str): chaîne de caractères non vide

    Returns:
        str: chaîne écrite inversée

    Tests et exemple:
    >>> renverse_pythonnesque_2("informatique")
    'euqitamrofni'
    """
    chaine_inverse = ""
    for lettre in chaine:
        chaine_inverse = lettre + chaine_inverse
```

```
    return chaine_inverse

# tests avec des affichages
print(renverse("informatique"))

# tests avec des assertions
assert renverse("informatique") == 'euqitamrofni'

# tests avec doctest
testmod()
```

"""

Author: Pascal Padilla

Source: correction de l'exercice 2 du sujet 22 des épreuves pratiques NSI 2022

"""

```
def crible(N):
    """renvoie un tableau contenant tous les nombres premiers plus petit que N"""
    # tableau de nombres premiers initialement vide
    premiers = []

    # initialisation du tableau de booléens
    tab = [True] * N
    tab[0], tab[1] = False, False

    # parcours du tableau à partir de la 3ème case
    # (car 0 et 1 sont traités)
    for i in range(2, N):
        # le nombre courant est premier
        if tab[i] == True:
            # ajoute le nombre courant dans le tableau à renvoyer
            # (attention: c'est le nombre i qu'il faut ajouter
            # et pas tab[i] qui vaut 'True'...)
            premiers.append(i)

            # met à False tous les multiples à venir du nombre courant
            # partie délicate:
            # -> pour accéder aux multiples, on parcourt tous les
            # nombres de 2i à la fin (N) MAIS pas de 1 en 1 !
            # Il faut les parcourir de i en i.
            # Ainsi, en commençant à 2i, on arrive à 2i + i => 3i
            # puis 3i + i => 4i
            # etc.
            for multiple in range(2*i, N, i):
                # on met chaque multiple à faux car il n'est pas premier
                # (ben ouaip, c'est un multiple !)
                tab[multiple] = False

    # renvoyer le tableau des nombres premiers complété au fur et à mesure
    return premiers

assert crible(40) == [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37]

# tests avec un affichage
print(crible(40))
```