

BACCALAURÉAT

SESSION 2024

Épreuve de l'enseignement de spécialité

NUMÉRIQUE et SCIENCES INFORMATIQUES

Partie pratique

Classe Terminale de la voie générale

Sujet n°33

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 (10 points)

Programmer une fonction `renverse` qui prend en paramètre une chaîne de caractères `mot` et qui renvoie cette chaîne de caractères en ordre inverse.

Exemple :

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

EXERCICE 2 (10 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 petits qu'un certain nombre n fixé.

On considère pour cela un tableau `tab` de n booléens (type `list`), 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 et 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 $2*i, 3*i \dots$).
- 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`, donnée ci-dessous et à compléter, 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....  
            multiple = ...  
            while multiple < n:  
                tab[multiple] = ...  
                multiple = ...  
    return premiers
```

Exemples :

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