

BACCALAURÉAT

SESSION 2025

Épreuve de l'enseignement de spécialité

NUMÉRIQUE et SCIENCES INFORMATIQUES

Partie pratique

Classe Terminale de la voie générale

Sujet n°10

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)

Écrire une fonction recherche qui prend en paramètres un tableau tab de nombres entiers triés par ordre croissant et un nombre entier n, et qui effectue une recherche dichotomique du nombre entier n dans le tableau non vide tab.

Cette fonction doit renvoyer un indice correspondant au nombre cherché s'il est dans le tableau, None sinon.

Exemples :

```
>>> recherche([2, 3, 4, 5, 6], 5)
```

```
3
```

```
>>> recherche([2, 3, 4, 6, 7], 5) # renvoie None
```

EXERCICE 2 (10 points)

Le codage de César transforme un message en changeant chaque lettre en la décalant dans l'alphabet. Par exemple, avec un décalage de 3, le A se transforme en D, le B en E, ..., le X en A, le Y en B et le Z en C. Les autres caractères ('!', ' ?' ...) ne sont pas codés.

La fonction `position_alphabet` ci-dessous prend en paramètre un caractère `lettre` et renvoie la position de `lettre` dans la chaîne de caractères `alphabet` s'il s'y trouve.

La fonction `cesar` prend en paramètre une chaîne de caractères `message` et un nombre entier `decalage` et renvoie le nouveau message codé avec le codage de César utilisant le décalage `decalage`.

```
alphabet = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
```

```
def position_alphabet(lettre):  
    '''Renvoie la position de la lettre dans l'alphabet'''  
    return ord(lettre) - ord('A')  
  
def cesar(message, decalage):  
    '''Renvoie le message codé par la méthode de César  
    pour le decalage donné'''  
    resultat = ''  
    for ... in message:  
        if 'A' <= c and c <= 'Z':  
            indice = (...) % 26  
            resultat = resultat + alphabet[indice]  
        else:  
            resultat = ...  
    return resultat
```

Compléter la fonction `cesar`.

Exemples :

```
>>> cesar('BONJOUR A TOUS. VIVE LA MATIERE NSI !', 4)  
'FSRNSYV E XSYW. ZMZI PE QEXMIVI RWM !'  
>>> cesar('GTSOTZW F YTX. ANAJ QF RFYNJWJ SXN !', -5)  
'BONJOUR A TOUS. VIVE LA MATIERE NSI !'
```