

BACCALAURÉAT

SESSION 2024

Épreuve de l'enseignement de spécialité

NUMÉRIQUE et SCIENCES INFORMATIQUES

Partie pratique

Classe Terminale de la voie générale

Sujet n°48

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.**

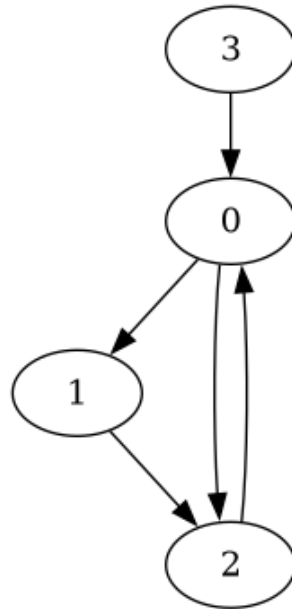
Le candidat doit traiter les 2 exercices.

EXERCICE 1 (10 points)

On considère dans cet exercice un graphe orienté représenté sous forme de listes d'adjacence.

On suppose que les sommets sont numérotés de 0 à $n - 1$.

Par exemple, le graphe suivant:



est représenté par la liste d'adjacence suivante:

```
adj = [[1, 2], [2], [0], [0]]
```

Rappel : cela signifie que les voisins sortants du sommet i sont les sommets de la liste $adj[i]$.

Écrire une fonction `voisins_entrants(adj, x)` qui prend en paramètre le graphe donné sous forme de liste d'adjacence et qui renvoie une liste contenant les voisins entrants du sommet x , c'est-à-dire les sommets y tels qu'il existe une arête de y vers x .

Exemples:

```
>>> voisins_entrants([[1, 2], [2], [0], [0]], 0)
[2, 3]
>>> voisins_entrants([[1, 2], [2], [0], [0]], 1)
[0]
```

EXERCICE 2 (10 points)

On considère dans cet exercice la suite de nombres suivante : 1, 11, 21, 1211, 111221, ...

Cette suite est construite ainsi : pour passer d'une valeur à la suivante, on la lit et on l'écrit sous la forme d'un nombre. Ainsi, pour 1211 :

- on lit *un 1, un 2, deux 1* ;
- on écrit donc en nombre *1 1, 1 2, 2 1* ;
- puis on concatène *111221*.

Compléter la fonction `nombre_suivant` qui prend en entrée un nombre sous forme de chaîne de caractères et qui renvoie le nombre suivant par ce procédé, encore sous forme de chaîne de caractères.

```
def nombre_suivant(s):
    '''Renvoie le nombre suivant de celui représenté par s
    en appliquant le procédé de lecture.'''
    resultat = ''
    chiffre = s[0]
    compte = 1
    for i in range(...):
        if s[i] == chiffre:
            compte = ...
        else:
            resultat += ... + ...
            chiffre = ...
        ...
    lecture_... = ... + ...
    resultat += lecture_chiffre
    return resultat
```

Exemples

```
>>> nombre_suivant('1211')
'111221'
>>> nombre_suivant('311')
'1321'
```