BACCALAUREAT

SESSION 2022

Épreuve de l'enseignement de spécialité

NUMERIQUE et SCIENCES INFORMATIQUES

Partie pratique

Classe Terminale de la voie générale

Sujet n°15

DUREE DE L'EPREUVE : 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)

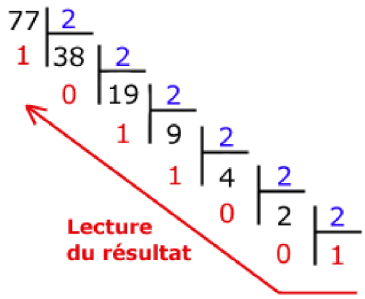
Écrire une fonction python appelée `nb_repetitions` qui prend en paramètres un élément `elt` et une liste `tab` et renvoie le nombre de fois où l'élément apparaît dans la liste.

Exemples :

```
>>> nb_repetitions(5,[2,5,3,5,6,9,5])
3
>>> nb_repetitions('A',['B','A','B','A','R'])
2
>>> nb_repetitions(12,[1,'!',7,21,36,44])
0
```

EXERCICE 2 (4 points)

Pour rappel, la conversion d'un nombre entier positif en binaire peut s'effectuer à l'aide des divisions successives comme illustré ici :



Voici une fonction python basée sur la méthode des divisions successives permettant de convertir un nombre entier positif en binaire :

```
def binaire(a):
    bin_a = str(...)
    a = a // 2
    while a ... :
        bin_a = ...(a%2) + ...
        a = ...
    return bin_a
```

Compléter la fonction `binaire`.

Exemples :

```
>>> binaire(0)
'0'
>>> binaire(77)
'1001101'
```

```
"""
Author: Pascal Padilla
Source: correction de l'exercice 1 du sujet 15 des épreuves pratiques NSI 2022

Remarque: une version plus 'pythonesque' du code consiste à remplacer
'for i in range(len(tab)):' par ¿ 'for val in tab:'
et du coup 'tab[i]' devient ¿ 'val'.
"""


from doctest import testmod


def nb_repetitions(elt, tab: list) -> int:
    """Renvoie le nombre de fois où l'élément 'elt' apparaît
    dans le tableau 'tab'.capitalize()

    Args:
        elt (_type_): élément à dénombrer
        tab (list): tableau contenant (ou pas) l'élément

    Returns:
        int: nombre d'occurrences de 'elt'.

    Tests et exemples:
    >>> nb_repetitions(5,[2, 5, 3, 5, 6, 9, 5])
    3
    >>> nb_repetitions('A',[ 'B', 'A', 'B', 'A', 'R'])
    2
    >>> nb_repetitions(12,[1, '! ', 7, 21, 36, 44])
    0
    """
    # boucle FOR
    # invariant de boucle:
    #   * somme est égale au nombre d'occurrence de 'elt'
    #   dans la partie tab[0 .. i-1] du tableau
    # condition d'arrêt
    #   * i vaut n - 1 (avec n = longueur de 'tab')
    somme = 0
    for i in range(len(tab)):
        if tab[i] == elt:
            somme = somme + 1

    # variante pythonesque de la boucle:
    # somme = 0
    # for val in tab:
    #     if val == elt:
    #         somme = somme + 1

    return somme


# exemples de l'énoncé
assert nb_repetitions(5,[2, 5, 3, 5, 6, 9, 5]) == 3
assert nb_repetitions('A',[ 'B', 'A', 'B', 'A', 'R']) == 2
assert nb_repetitions(12,[1, '! ', 7, 21, 36, 44]) == 0


# tests de l'énoncé avec la bibliothèque doctest
testmod()
```

```python
"""
Author: Pascal Padilla
Source: correction de l'exercice 2 du sujet 15 des épreuves pratiques NSI 2022
"""


def binaire(a):
    # initialise la chaîne de caractère finale
    # avec le dernier caractère : le reste de la
    # division euclidienne du nombre 'a' par 2
    # (qui vaut '0' si 'a' est paire et '1' sinon)
    bin_a = str(a % 2)

    # met à jour 'a' avec le quotient de la division
    # ce quotient devient le nouveau nombre à diviser
    a = a // 2

    # condition d'arrêt de la boucle : le nouveau nombre à diviser
    # (c'est-à-dire le quotient de la division précédente)
    # est nul. C'est donc la fin des divisions successives
    while a != 0 :
        # la chaîne de caractère est mise à jour
        # pour cela on utilise l'opérateur '+' qui permet
        # de concaténer des chaînes
        # on écrit le reste de la division (converti en
        # chaîne de caractère) avant ce qui était
        # déjà écrit dans la chaîne
        bin_a = str(a % 2) + bin_a

        # mise à jour du quotient pour la prochaine division
        a = a // 2

    # renvoie de la chaîne complète
    return bin_a


# Tests de l'énoncé
assert binaire(0) == '0'
assert binaire(77) == '1001101'
```