BACCALAUREAT

SESSION 2022

Épreuve de l'enseignement de spécialité

NUMERIQUE et SCIENCES INFORMATIQUES

Partie pratique

Classe Terminale de la voie générale

Sujet n°2

DUREE DE L'EPREUVE : 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 (4 points)

Soit le couple `(note,coefficient)`:

- `note` est un nombre de type flottant (*`float`) compris entre 0 et 20 ;
- `coefficient` est un nombre entier positif.

Les résultats aux évaluations d'un élève sont regroupés dans une liste composée de couples `(note,coefficient)`.

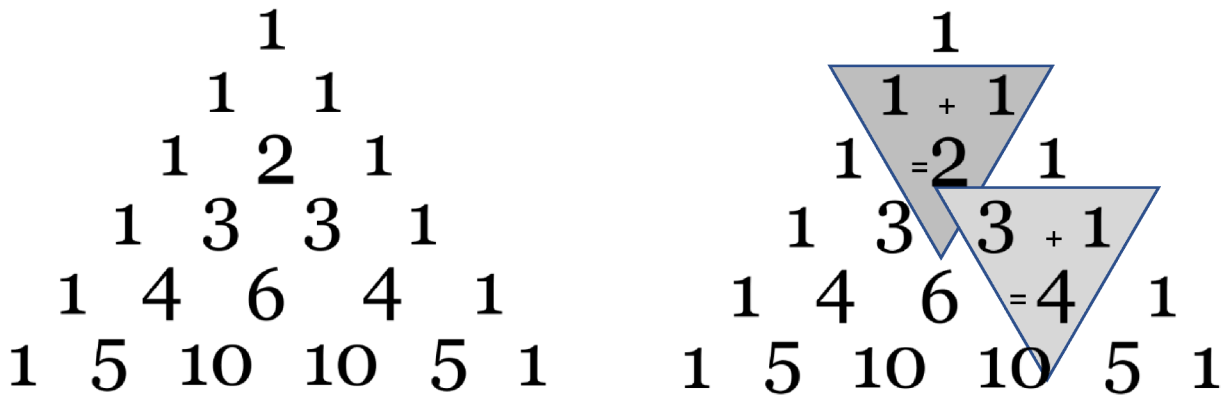
Écrire une fonction `moyenne` qui renvoie la moyenne pondérée de cette liste donnée en paramètre.

Par exemple, l'expression `moyenne([(15,2),(9,1),(12,3)])` devra renvoyer le résultat du calcul suivant :

$$\frac{2 \times 15 + 1 \times 9 + 3 \times 12}{2 + 1 + 3} = 12{,}5$$

EXERCICE 2 (4 points)

On cherche à déterminer les valeurs du triangle de Pascal. Dans ce tableau de forme triangulaire, chaque ligne commence et se termine par le nombre 1. Par ailleurs, la valeur qui occupe une case située à l'intérieur du tableau s'obtient en ajoutant les valeurs des deux cases situées juste au-dessus, comme l'indique la figure suivante :



Compléter la fonction `pascal` ci-après. Elle doit renvoyer une liste correspondant au triangle de Pascal de la ligne 1 à la ligne n où `n` est un nombre entier supérieur ou égal à 2 (le tableau sera contenu dans la variable `C`). La variable `Ck` doit, quant à elle, contenir, à l'étape numéro k, la k-ième ligne du tableau.

```
def pascal(n):
    C= [[1]]
    for k in range(1,...):
        Ck = [...]
        for i in range(1,k):
            Ck.append(C[...][i-1]+C[...][...] )
        Ck.append(...)
        C.append(Ck)
    return C
```

Pour n = 4, voici ce que l'on devra obtenir :

```
>> pascal(4)
[[1], [1, 1], [1, 2, 1], [1, 3, 3, 1], [1, 4, 6, 4, 1]]
```

Et pour n = 5, voici ce que l'on devra obtenir :

```
>> pascal(5)
[[1], [1, 1], [1, 2, 1], [1, 3, 3, 1], [1, 4, 6, 4, 1], [1, 5, 10,
10, 5, 1]]
```

```python
from doctest import testmod


def moyenne(tab: list) -> float:
    """ Calcul une moyenne pondérée.

    Args:
        tab (list): tableau de couples de notes
        de la forme (note: int, coefficient: int)

    Returns:
        float: moyenne pondérée

    Tests et Exemples:
    >>> moyenne([(15, 2), (9, 1), (12, 3)])
    12.5
    >>> moyenne([(10, 2)])
    10.0
    """
    somme = 0
    coef_total = 0

    n_notes = len(tab)
    for i in range(n_notes):
        couple = tab[i]
        note = couple[0]
        coefficient = couple[1]

        somme = somme + note*coefficient
        coef_total = coef_total + coefficient

    return somme / coef_total


assert moyenne([(15,2),(9,1),(12,3)]) == 12.5
assert moyenne([(10, 2)]) == 10

testmod()
```

```python
from doctest import testmod


def pascal(n):
    """ Générer une liste correspondant au triangle de Pascal.

    Args:
        n (int): hauteur du triangle de Pascal (moins 1...)

    Returns:
        list: tableau contenant les listes des coefficients, ligne par ligne

    Tests et Exemples:
    >>> pascal(4)
    [[1], [1, 1], [1, 2, 1], [1, 3, 3, 1], [1, 4, 6, 4, 1]]
    >>> pascal(5)
    [[1], [1, 1], [1, 2, 1], [1, 3, 3, 1], [1, 4, 6, 4, 1], [1, 5, 10, 10, 5, 1]]
    """
    C = [[1]]
    for k in range(1, n + 1):
        # on doit ajouter n lignes au tableau C
        Ck = [1]
        for i in range(1, k):
            # on ajoute k coefficients à la ligne Ck
            # le coef à ajouter est égal à la somme des deux
            #   coefficients situés au dessus de lui
            Ck.append(C[k-1][i-1] + C[k-1][i])

        # on termine la ligne Ck
        #   en ajoutant la dernière valeur qui vaut toujours 1
        Ck.append(1)
        # on ajoute la ligne Ck au tableau C
        C.append(Ck)
    return C


assert pascal(4) == [[1], [1, 1], [1, 2, 1], [1, 3, 3, 1], [1, 4, 6, 4, 1]]
assert pascal(5) == [[1], [1, 1], [1, 2, 1], [1, 3, 3, 1], [1, 4, 6, 4, 1], [1, 5, 1
0,
10, 5, 1]]
testmod()
```