BACCALAURÉAT

SESSION 2024

Épreuve de l'enseignement de spécialité

NUMÉRIQUE et SCIENCES INFORMATIQUES

Partie pratique

Classe Terminale de la voie générale

Sujet n°29

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 `moyenne(notes)` qui renvoie la moyenne pondérée des résultats contenus dans le tableau `notes`, non vide, donné en paramètre. Ce tableau contient des couples `(note, coefficient)` dans lesquels :

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

Ainsi l'expression `moyenne([(15.0,2),(9.0,1),(12.0,3)])` devra renvoyer `12.5` comme résultat du calcul suivant :

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

EXERCICE 2 (10 points)

On cherche à déterminer les valeurs du triangle de Pascal (Figure 1).

Dans le triangle de Pascal, chaque ligne commence et se termine par le nombre 1. Comme l'illustre la Figure 2, on additionne deux valeurs successives d'une ligne pour obtenir la valeur qui se situe sous la deuxième valeur.

```
1
1   1
1   2   1
1   3   3   1
1   4   6   4   1
1   5   10  10  5   1
```

Figure 1 : triangle de Pascal

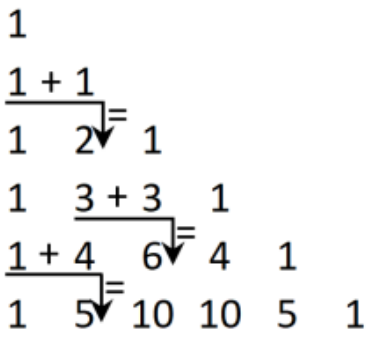

Figure 2 : méthode de calcul

Compléter les fonctions `ligne_suivante` et `pascal` ci-dessous. La fonction `ligne_suivante` prend en paramètre une liste d'entiers `ligne` correspondant à une ligne du triangle de Pascal et renvoie la liste correspondant à la ligne suivante du triangle de Pascal. La fonction `pascal` prend en paramètre un entier n et l'utilise pour construire le triangle de Pascal ayant n+1 lignes sous la forme d'une liste de listes.

```
def ligne_suivante(ligne):
    '''Renvoie la ligne suivant ligne du triangle de Pascal'''
    ligne_suiv = [...]
    for i in range(...):
        ligne_suiv.append(...)
    ligne_suiv.append(...)
    return ligne_suiv

def pascal(n):
    '''Renvoie le triangle de Pascal de hauteur n'''
    triangle = [ [1] ]
    for k in range(...):
        ligne_k = ...
        triangle.append(ligne_k)
    return triangle
```

Exemples :

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