

Algorithmique

correction CC 1 2025 / 2026

v.pagé

Exo 1 (7 pts)

Question 1

On doit faire une fonction qui calcule la somme d'un tableau.

- Je choisis un nom pour la fonction : *somme*.
- Je dois lui passer le tableau comme argument.
- Je choisis un nom pour ce tableau *tab*.

Tout ceci me donne la première ligne du code suivant

dans ce code, je calcule la somme des éléments du tableau *tab*, que je mets dans une variable dont je choisis le nom (*s*)

Enfin, ma fonction renvoie ce qu'elle a calculé (*s*)

```
def somme(tab):  
    s = 0  
    for val in tab:  
        s = s + val  
  
    return s
```

Question 2

voyons comment on réalise l'appel de cette fonction

```
notes = [12, 14, 8, 17]  
coeffs = [2, 1, 6, 1]  
  
# si je veux calculer la somme des éléments de coeffs,  
# je fais mon appel comme suit  
  
somme_coeffs = somme(coeffs)  
print("la somme vaut",somme_coeffs)
```

Question 3

Ici, pas besoin de créer de fonction (selon l'énoncé). Je vais donc, dans le programme principal, parcourir les 2 tableaux en même temps.

Si on suppose que ce qui suit vient après le code tapé en question 2, on a déjà :

- le tableau *notes*
- le tableau *coeffs*
- la somme des coefficients dans la variable *somme_coeffs*

```
s = 0  
for i in range(len(notes)):
```

```

s = s + notes[i]*coeffs[i]

moy = s / somme_coeffs
print("la moyenne pondérée est",moy)

```

Exo 2 (8 pts)

Question 1

On veut calculer le prix suivant à partir du prix actuel, dans une fonction appelée *prix_suivant*.

- la fonction doit avoir pour nom *prix_suivant*
- on lui passe le prix actuel, sous forme d'une variable dont je choisis le nom (*prix*)
- la fonction fait son calcul à partir de *prix*
- la fonction renvoie le résultat

```

def prix_suivant(prix):
    prix_apres = prix * 1 + prix * 10 / 100 - 800

    return prix_apres

```

Question 2

Il faut, 10 fois de suite, calculer le prix suivant du prix actuel, puis considérer que ce prix suivant devient le prix actuel pour le prochain calcul...

Je vais le faire avec beaucoup de variables intermédiaires,

```

print("entrez un prix")
# je sauvegarde le prix initial
prix_init = float(input())

n_annees = 10

# mon prix actuel, au départ, est le prix initial
prix_actuel = prix_init

# je mets ce prix de départ dans un tableau recap_prix
# Il est en case 0 (pour l'année 0)
recap_prix = [prix_actuel]

for i in range(n_annees):
    # je calcule le prix suivant du prix actuel
    nouveau_prix = prix_suivant(prix_actuel)
    print ("en fin d'année",i,"le prix est de",nouveau_prix)

    # j'ajoute ce prix dans le tableau
    recap_prix.append(nouveau_prix)

    # enfin, je mets ce nouveau prix dans prix actuel
    # pour qu'au prochain tour de boucle, on travaille
    # sur ce nouveau prix
    prix_actuel = nouveau_prix

# j'ai fini ma boucle, j'affiche le tableau
print(recap_prix)

```

Question 3

Meme chose. Si j'ai pris soin de sauvegarder le prix initial dans une variable spéciale, c'est facile.

```
n = 0 # mon compteur d'années

prix_actuel = prix_initial
while prix_actuel < 2*prix_initial :
    # je calcule le prix suivant du prix actuel
    nouveau_prix = prix_suivant(prix_actuel)

    # j'incrémante mon compteur d'années
    n = n+1
    print ("en fin d'année",n,"le prix est de",nouveau_prix)
    # enfin, je mets ce nouveau prix dans prix actuel
    # pour qu'au prochain tour de boucle, on travaille
    # sur ce nouveau prix
    prix_actuel = nouveau_prix

# la boucle est finie, j'affiche le nombre d'années (n)
print("en",n,"années, le prix a doublé")
```

Voilà !