
IE Informatique et Société Numérique 1

FC, AS, EUR, AMER

Décembre 2022



Durée: 1h

Documents et calculatrice interdits

Attention : Un programme **mal indenté**, ou avec de **mauvais choix de noms de variables** sera **sanctionné** (*jusqu'à -1 point*).

« Always code as if the guy who ends up maintaining your code will be a violent psychopath who knows where you live. » *Martin Golding*

Exercice 1 Correction de code (3 pts)

(Q1.1) Le programme suivant contient **6 erreurs**. Identifiez chaque erreur en indiquant : le numéro de ligne et le problème. (*ex : ligne 5, le symbole '=' a été utilisé dans une comparaison, alors qu'il faut utiliser un '=='*.)

```
1 argent = 10 # euros
2 prix_fromage = 12 # euros/kg
3 prix_yaourt = 3 # euros/yaourt
4 prix_bonbon = 1 # euros/bonbon
5
6 if prix_fromage > 10
7     print("J'achète 200g de fromage")
8     argent = argent - prix_fromage*0,200
9 elif promo_yaourt:
10    print("J'achète 3 yaourts")
11    argent = argent - prix_yaourt * 3
12
13 While argent > prix_bonbon:
14    print("J'achète un bonbon")
15    argent = argent - prix_bonbon
16
17 print("Il me reste {argent} euros")
```

Correction:

Ligne 6 : Il manque le : à la fin de la ligne.

Ligne 8 : Il y a une virgule à la place d'un point dans 0,200.

Ligne 9 : La variable promo_yaourt n'est pas défini.

Ligne 13 : Le While a une majuscule alors qu'il ne devrait pas.

Ligne 17 : Il manque le f pour faire une f-string.

Ligne 17 : Il manque un guillemet après euros.

Exercice 2 Codage (3.5 pts)

(Q2.1) Donnez la représentation binaire du nombre entier (non signé) : 77_{10} .

Correction:

$$77_{10} = 100\ 1101_2$$

(Q2.2) Sachant qu'il s'agit de la représentation binaire d'un entier (non signé), donnez la représentation décimale de : $1001\ 0101_2$.

Correction:

$$1001\ 0101_2 = 149_{10}$$

(Q2.3) Donnez la représentation décimale de : $1100\ 0001\ 1010\ 0100\ 0000\ 0000\ 0000\ 0000_2$, codé en IEEE754 simple précision.

Correction:

$$1100\ 0001\ 1010\ 0100\ 0000\ 0000\ 0000\ 0000_2 = -20,5_{10}$$

(Explication : Signe = 1, donc nombre négatif;

Exposant = $1000\ 0011_2 = 131_{10}$, avec décalage : $e = 4$;

Mantisse = $0100\ 1000\dots$ qui encode 1,01001. Avec le décalage de l'exposant $10100,1 = 16 + 4 + 0,5 = 20,5$. D'où le résultat avec le signe.)

Exercice 3 Lecture de code (2 pts)

(Q3.1) Qu'affiche exactement l'exécution du code suivant ?

```
1 chaine = "Lucile"
2 cpt = 0
3 for i in range(len(chaine)):
4     if chaine[i] == "l":
5         cpt = cpt + 1
6         print(i)
7     else :
8         print("*")
9 print(cpt)
```

Correction:

*
*
*
*
4
*
1

Exercice 4 Pizzeria et coût de fabrication (11.5 pts)

L'objectif de cet exercice est d'aider une pizzeria à prévoir le coût de fabrication de ses pizzas. Le coût (en euros) de chaque pizza est la somme du coût de ses ingrédients. Les coûts sont définis dans le tableau suivant (les valeurs sont fantaisistes) :

| Ingrédient | tomate | fromage | jambon | champignon |
|------------|--------|---------|--------|------------|
| Coût | 2 € | 4 € | 6 € | 4 € |

Par exemple, une pizza au fromage et à la tomate coûte 6 euros à produire.

(Q4.1) En s'appuyant sur le tableau précédent, écrivez une fonction python `cout` qui prend en paramètre un ingrédient `ingr` sous forme d'une chaîne de caractères et renvoie son coût. On suppose que `ingr` est forcément l'un des quatre ingrédients ci-dessus. Par exemple, `cout("tomate")` doit renvoyer 2.

```

1 def cout(ingr):
2     if ingr == "tomate":
3         res = 2
4     elif ingr == "fromage":
5         res = 4
6     elif ingr == "jambon":
7         res = 6
8     elif ingr == "champignon":
9         res = 4
10    return res

```

```

1 def cout(ingr):
2     if ingr == "tomate":
3         res = 2
4     elif ingr == "jambon":
5         res = 6
6     else:
7         res = 4
8     return res

```

(Q4.2) **En utilisant cette fonction** `cout`, affichez le coût total d'une pizza composée de tomate, de champignon et de fromage.

```

1 print(cout("tomate") + cout("champignon") + cout("fromage"))
2 # ou
3 res = cout("tomate") + cout("champignon") + cout("fromage")
4 print(f"Cette pizza coute {res} euros")

```

(Q4.3) Chaque pizza est représentée par une liste d'ingrédients. En utilisant la fonction `cout`, écrivez une fonction `cout_pizza` qui prend en paramètre une liste d'ingrédients `liste_ingr` et renvoie le coût total de la pizza. Par exemple, `cout_pizza(["fromage", "tomate", "jambon"])` doit renvoyer 12.

```

1 def cout_pizza(liste_ingr):
2     tot = 0
3     for elem in liste_ingr:
4         tot += cout(elem)
5     return tot

```

```

1 def cout_pizza(liste_ingr):
2     tot = 0
3     for i in range(len(liste_ingr)):
4         tot += cout(liste_ingr[i])
5     return tot

```

Correction:

La version de gauche (`for elem in liste`) est la meilleure car les indices ne sont pas nécessaires.

La pizzeria souhaite étudier le coût du basilic. Ce coût augmente d'un certain **taux** chaque mois : par exemple, si le taux est de 6%, le coût c du basilic le mois suivant est augmenté de $c \times 0.06$. Ainsi, si le coût vaut 2 euros, il vaudra 2.12 euros le mois suivant. On souhaite savoir au bout de combien de mois le coût du basilic dépassera une valeur `cout_max`.

Avec les valeurs suivantes :

```

1 cout_init = 2
2 taux = 0.06
3 cout_max = 3

```

le programme doit afficher : Nombre de mois : 7

(Q4.4) Écrivez un code Python qui calcule le nombre de mois au bout duquel le coût du basilic dépassera une valeur `cout_max` en partant d'une valeur initiale `cout_init`. Vous respecterez les noms de variables indiqués dans le code précédent.

```
1 cout_init = 2
2 taux = 0.06
3 cout_max = 3
4
5 cout = cout_init
6 nb_mois = 0
7 while cout <= cout_max:
8     cout = cout * (1+taux)
9     nb_mois = nb_mois + 1
10 print(f"Nombre de mois : {nb_mois}")
```

Chaque mois, la pizzeria a noté combien a coûté le fromage. Elle a donc établi la liste `couts` des coûts du fromage sur l'année 2022. Par exemple, d'après la liste `couts` ci-dessous, le coût du fromage en janvier était de 2.7 euros. Dans la suite, le numéro du mois de janvier est 1, décembre ayant le numéro 12.

```
1 couts = [2.7, 3, 3.9, 3.1, 3.5, 3.5, 3.8, 3.8, 3, 4, 4.1, 3.9]
2 borne = 3.8
```

On veut connaître la liste des mois où le coût a dépassé de 3.80 euros, **en commençant par la fin de l'année**. Par exemple, avec les données ci-dessus, on voudrait obtenir :

```
12 : 3.9
11 : 4.1
10 : 4
3 : 3.9
```

(Q4.5) Écrivez un code Python qui affiche les numéros de mois où le coût du fromage a dépassé 3.8 euros, avec le coût correspondant. Le programme devra produire exactement l'affichage de l'exemple ci-dessus, donc en commençant par la fin de l'année.

```
1 for i in range(len(couts)-1, -1, -1):
2     if couts[i] > borne:
3         print(f"{i+1} : {couts[i]}")
```

La pizzeria dispose d'une liste de référence, `ref`, des coûts mensuels du fromage. Elle souhaite savoir quel est le premier mois de l'année (en commençant par janvier) à partir duquel le coût du fromage a augmenté de plus de 1 euro par rapport au coût de référence.

Par exemple, avec les coûts de la question précédente et les données de références suivantes :

```
1 ref = [2, 2.1, 2.3, 3, 2.3, 2.5, 2.4, 2.4, 2.5, 2.4, 2.4, 2.5]
2 ecart = 1
```

le programme doit afficher :

```
Le fromage est cher au mois numéro 3
Trop cher !
```

(Q4.6) Écrivez un code Python qui identifie et affiche le premier mois de l'année où le coût du fromage a dépassé de plus de 1 euro le coût de référence pour le même mois. Si un tel mois n'existe pas, le code doit afficher qu'aucun mois n'a été identifié. De plus la chaîne "Trop cher!" doit être affichée dans les deux cas suivants : soit l'écart dépasse 1.5 euros, soit il s'agit des mois de septembre ou novembre.¹

```
1 ecart = 1
2
3 i=0
4 while i < len(couts) and couts[i] - ref[i] <= ecart:
5     i = i + 1
6 if i < len(couts): # on en a trouve un
7     print(f"Le fromage est cher au mois no {i+1} ")
8     if i == 8 or i == 10 or couts[i] - ref[i] >= 1.5:
9         print("Trop cher !")
10 else :
11     print(f"Aucun mois identifié")
```

Exercice 5 Questions bonus (0.5 pts)

(Q5.1) D'après votre cours, citez le nom d'une personne ayant contribué à l'histoire de l'informatique.

Correction:

Ada Lovelace, Grace Hopper, John Von Neumann, etc.

1. Les mois de septembre et novembre sont difficiles pour les finances de cette pizzeria.