Flottants et codage BCD

Sujet n°8 — Corrigé

Question 1 *Des centimes qui s'évaporent*

```
def calcul_recettes():
    total = 0
    for restaurant in range(1000):
        for menu in range(500):
            total = total + 2.27 + 5.19 + 1.81
    return total

print(calcul_recettes())
```

On obtient 4 634 999,999 986 519 au lieu de 4 635 000. **Justification** : 2,27, 5,19 et 1,81 n'ont pas d'écriture binaire finie ; chaque flottant est une valeur approchée et les 500 000 additions accumulent ces erreurs d'arrondi. Calculer en centimes (entiers) donnerait le résultat exact.

Question 2 *Du BCD au décimal*

```
def convertir_BCD_vers_decimal(liste_quartets):
    entier = 0
    for quartet in liste_quartets:
        entier = entier * 10 + int(quartet, 2)
    return entier / 100

assert convertir_BCD_vers_decimal(['0001', '0011', '0101', '0110']) == 13.56
```

Question 3 *La correction BCD oubliée*

Sans correction, `additionner_nombres_format_BCD('27', '35')` renvoie `['0101', '1100', '0000', '0000']` : le quartet `1100` (12) n'est pas un chiffre décimal. La correction doit être appliquée à chaque quartet, juste après son addition binaire, car elle peut produire une retenue pour le quartet suivant.

```
        somme, retenue = additionner_binaire_quartets(
            liste_quartets1[index], liste_quartets2[index], retenue)
        somme, retenue = corriger_BCD(somme, retenue)
        resultat.insert(0, somme)
```

On obtient `['0110', '0010', '0000', '0000']`, soit 62,00.

Question 4 *Nombres de longueurs différentes*

`'23'` est converti en 4 quartets et `'4'` en 3 : les indices ne correspondent plus et l'addition lève une `IndexError`. On complète la liste la plus courte par des quartets `'0000'` à gauche.

```
def aligner_quartets(q1, q2):
    while len(q1) < len(q2):
        q1 = ['0000'] + q1
    while len(q2) < len(q1):
        q2 = ['0000'] + q2
    return q1, q2
```

23 + 4 donne alors `['0010', '0111', '0000', '0000']` (27,00) et 0,23 + 1,75 donne 1,98.