

BACCALAURÉAT

SESSION 2026

Épreuve de l'enseignement de spécialité

NUMÉRIQUE et SCIENCES INFORMATIQUES

Partie pratique

Classe Terminale de la voie générale

Sujet n°8

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.**

Cette situation d'évaluation comporte ce document ainsi que des fichiers de codes et de données présents sur l'ordinateur à la disposition du candidat. Le candidat doit restituer ce document avant de sortir de la salle d'examen. Le candidat doit agir en autonomie et faire preuve d'initiative tout au long de l'épreuve.

En cas de difficulté, le candidat peut solliciter l'examineur afin de lui permettre de continuer la tâche. Des moments privilégiés pour solliciter l'examineur sont indiqués dans le document sous la forme d'appels professeur.

L'examineur peut intervenir à tout moment, s'il le juge utile.

En informatique, utiliser des nombres flottants pour manipuler des valeurs monétaires est une erreur de conception classique. Les ordinateurs utilisant le système binaire (base 2), certains nombres décimaux comme 0.1 ne peuvent pas être représentés de manière exacte et génèrent une infinité de décimales dans leur représentation binaire flottante ($0.00011001100110011\dots$).

Lors de calculs financiers, ces erreurs d'arrondi s'accumulent et faussent les bilans comptables.

Question 1

La chaîne de restauration "RESTO NSI" comprend 1 000 restaurants qui délivrent chacun 500 menus par jour. Un menu est composé d'une entrée à 2.27 €, d'un plat à 5.19 € et d'un dessert à 1.81 €.

Écrire, dans le fichier `addition_BCD.py`, une fonction `calcul_recettes()` qui additionne le prix de chaque menu vendu dans la journée en utilisant une boucle.

Afficher le résultat de cette fonction. Sachant que la valeur théorique exacte est de 4 635 000 €, justifier le comportement observé.



Appeler le professeur pour lui présenter votre réponse ou en cas de difficulté.

Historiquement, pour pallier ce problème dans les calculatrices et les systèmes financiers, on utilise le système BCD (Binary Coded Decimal). Le principe est de coder chaque chiffre du nombre décimal séparément sur 4 bits (un quartet).

La virgule n'étant pas codée, on utilise la convention monétaire "virgule implicite deux rangs avant la fin". Il faut donc au minimum 3 quartets pour représenter une somme.

Montant en euros	Représentation BCD (listes de chaînes)
59.00	['0101', '1001', '0000', '0000']
1.75	['0001', '0111', '0101']
0.23	['0000', '0010', '0011']

Le fichier `addition_BCD.py` contient des fonctions permettant de manipuler ces données.

Question 2

Écrire la fonction `convertir_BCD_vers_decimal(liste_quartets)` qui prend en paramètre une liste de chaînes de caractères représentant des quartets BCD, et renvoie la valeur décimale correspondante (de type float).

Ajouter une assertion pour vérifier que `convertir_BCD_vers_decimal(['0001', '0011', '0101', '0110'])` renvoie bien la valeur 13.56.



Appeler le professeur pour lui présenter votre réponse ou en cas de difficulté.

Afin de réaliser une addition de deux nombres donnés en BCD, on additionne les nombres quartet par quartet, de droite à gauche.

Si le résultat d'une addition binaire de quartets est supérieur ou égal à 10 (soit '1010' en binaire) ou s'il génère une retenue, le format BCD n'est plus valide. Il faut alors appliquer une correction en ajoutant 6 (soit '0110') à ce quartet et propager la retenue.

Question 3

La fonction `additionner_nombres_format_BCD(a, b)` fournie dans le fichier réalise cette addition. L'évaluation de l'appel `additionner_nombres_format_BCD('27', '35')` devrait correspondre à 62, mais la liste renvoyée est fausse.

Analyser le code fourni. Identifier l'oubli de l'étape de correction dans l'algorithme, puis insérer un appel à la fonction `corriger_BCD` (déjà fournie) au bon endroit pour résoudre ce problème. Refaire le test pour valider la réparation.



Appeler le professeur pour lui présenter votre réponse ou en cas de difficulté.

Question 4

Tester maintenant l'addition de 23 et de 4 avec votre code et décrire ce que vous observez. Modifier la fonction `aligner_quartets(q1, q2)` pour qu'elle ajoute des quartets '0000' au début du nombre le plus court jusqu'à ce que les deux listes aient la même longueur et effectuer à nouveau des tests.



Appeler le professeur pour lui présenter votre réponse ou en cas de difficulté.

Description du dossier

Le dossier fourni au candidat sur l'ordinateur comporte les éléments suivants :

- Une version PDF de l'énoncé ;
- Un code source `addition_BCD.py`.