

Contrôle Continu L1 Python Avril 2025



Vous avez le droit de consulter [mon site web](#) ou la [documentation Python](#) ainsi que vos exercices de TP au **format papier**. Il est interdit de faire appel à des outils comme ChatGPT. Tout programme qui ne s'exécute pas correctement (erreur, exception) est compté pour 0.



Vous devez rendre quatre fichiers qui correspondent chacun à une question. Si vous ne répondez pas à une question ne rendez pas le fichier correspondant.

Exercice 1 - 5 pts -

Ecrire un programme **chaine.py** qui étant données deux chaînes appelées **chaine** et **motif** permet d'afficher à quels indices de **chaine** on peut trouver **motif**. Par exemple, avec "CoucouRouCouCou" et "coU" on doit afficher sur le terminal :

```
chaine en minuscule = "coucouroucoucou"
motif en minuscule = "cou"
cou trouvé à la position 0
cou trouvé à la position 3
cou trouvé à la position 9
cou trouvé à la position 12
Le motif a été trouvé 4 fois
```

Il faut donc commencer par convertir les chaînes initiales en minuscules. Ces chaînes doivent être définies dans la fonction **main**.

Exercice 2 - 5 pts -

Soit la fonction suivante :

$$f(x) = (x^2 + x \times \sin(x))/x!$$

définie sur l'intervalle entier $[0, 9]$ et où $x!$ représente la factorielle de x .

1. Ecrire un programme **derive.py** qui pour tout entier $x \in [0, 9]$ donne les valeurs de x , de $f(x)$ ainsi que la valeur de sa dérivée en x .

Vous devez afficher à l'écran les nombres flottants **avec 5 chiffres après la virgule** :

x	f(x)	f'(x)
0	0.00000	0.00000
1	1.84147	3.38177
2	2.90930	2.03850
3	1.57056	0.52853
4	0.54053	0.19286
5	0.16838	0.08716
6	0.04767	0.02428
7	0.01063	0.00396
8	0.00178	0.00039
9	0.00023	0.00003

Exercice 3 - 5 pts -

Ecrire un programme **data.py** qui réalise l'affichage graphique de données avec matplotlib.

1. en utilisant **numpy** créer un tableau nommé **x** de 200 valeurs comprises entre 1 et 5 :

```
x = [1. 1.0201005 1.04020101 ... 4.9798995 5.]
```

2. en utilisant **numpy**, créer un tableau **delta** de 200 valeurs comprises entre -1 et 1 de manière aléatoire
3. créer un tableau **y** qui résulte de l'application de $f(x) = 3 \times x - 2$, auquel on aura ajouté **delta**
4. calculer la pente de la droite a et l'ordonnée à l'origine b en utilisant les formules suivantes et **afficher** les valeurs de a et b :

$$a = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}$$

$$b = \bar{y} - a \times \bar{x}$$

où $\bar{x}$ et $\bar{y}$ sont respectivement la moyenne de x et de y .

5. dessiner (voir Fig. 1 ci-après) sur le même graphique, en utilisant matplotlib, le nuage de points correspondant aux vecteurs x et y ainsi que la droite $y' = a \times x' + b$. Le choix des couleurs n'est pas important mais on doit pouvoir distinguer le nuage de point et la droite y' .

Exercice 4 - 5 pts -

On considère le **cryptarithme** suivant KILO + KILO = TONNE pour lequel chaque lettre doit être remplacée par un chiffre entre 0 et 9 de manière à ce que le calcul obtenu soit correct. On doit également vérifier que toutes les lettres ont une valeur différente et il faut prendre en compte les retenues éventuelles.

Ecrire un programme **crypte.py** qui résoud ce problème et donne toutes les solutions.

1. dans la fonction **main** définir une chaîne **chn** dont le contenu est KILO+KILO=TONNE

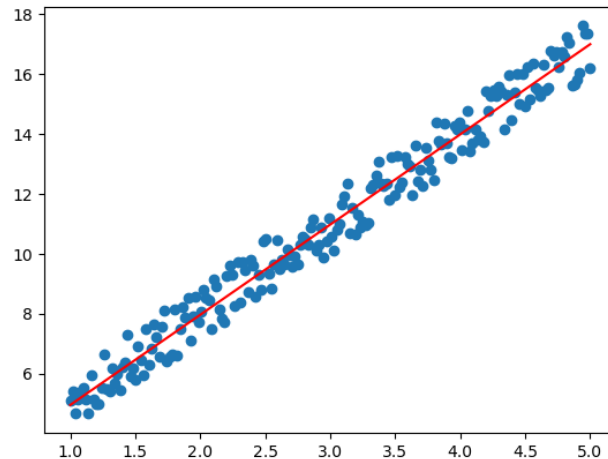


FIGURE 1 – Exemple de résultat attendu

2. à partir de cette chaîne créer un ensemble **ens** des lettres qui la composent
3. à partir de **ens** créer un dictionnaire appelé **dico** dont les clés sont les lettres de **ens** et les valeurs sont des entiers initialisés à 0
4. créer une fonction **tous_différents(d)** qui prend en paramètre un dictionnaire et vérifie que chaque lettre possède une valeur différente des autres lettres et donc retourne True dans ce cas, sinon on retournera False
5. créer une fonction **formule_valide(d)** qui prend en paramètres un dictionnaire et vérifie que la formule KILO + KILO = TONNE est valide
6. créer une fonction récursive **cherche_solutions(d,lettres)** qui prend en paramètres un dictionnaire (lettre,entier) et un ensemble (ou une liste de lettres). Cette fonction est récursive et doit permettre de rechercher une solution et l'afficher.
7. à partir de la fonction **main** appelez **cherche_solutions** avec les paramètres : **dico** et **ens** ou alors les paramètres **dico** et **list(ens)**

Vous devriez obtenir et afficher 4 solutions.