INFORMATIQUE PSI P2B

Année 2023/24

TP n°1

Fondamentaux en Python

Table des matières

1 Exercices faciles 2

2 Exercices de difficulté modérée 2

3 Exercices plus avancés 4

Vincent ROHART

http://vrohart.e-monsite.com/

1 Exercices faciles

Exercice 1 Calculer $1^3 + 2^3 + 3^3 + \ldots + 500^3$ et vérifier que cela vaut $(1 + 2 + 3 + \ldots + 500)^2$.
Rappel. Pour tester si deux entiers sont égaux, on utilise `==`.

Exercice 2 *(Tracés de courbes ou de surfaces).*

1. Tracé en rouge la courbe représentative de $x \mapsto \sin(\frac{1}{x})$ sur l'intervalle $]0, \pi]$, dans un repère orthonormal. On affichera une grille, on mettra un titre au graphique ainsi qu'une légende.
2. Avec les mêmes options, tracer sur un même graphique les supports des arcs paramétrés de Lissajous
$$t \longmapsto (\cos(pt), \sin(qt))$$
quand $(p, q) \in [\![1, 7]\!]^2$ est tel que $\text{pgcd}(p, q) = 1$.
3. Pour tracer une courbe définie par une équation, par exemple $x^2 + y^2 + xy = 0$, voici une documentation donnée par le concours Centrale-Supélec à
https://www.concours-centrale-supelec.fr/CentraleSupelec/MultiY/C2015/Python-plot.pdf
En s'en inspirant, tracer, sur un même graphique, les courbes d'équation $x^3+3xy^2-15x-12y = k$ où k décrit $[-60, 60]$ avec un pas de 1.
4. À l'aide de la même documentation, tracer la surface représentative de la fonction $f : \mathbb{R}^2 \to \mathbb{R}$ définie par $f(x, y) = x^3 + 3xy^2 - 15x - 12y$ et faire le lien avec la question précédente.

2 Exercices de difficulté modérée

Exercice 3 *(Nombres d'Armstrong).* Un *nombre d'Armstrong* est un entier naturel non nul qui est égal à la somme des cubes des chiffres qui composent son écriture en base 10. Par exemple 153 est un nombre d'Armstrong car $153 = 1^3 + 5^3 + 3^3$.
Trouver tous les nombres d'Armstrong inférieurs à 10 000.
Rappel. `str`(153) *renvoie la chaîne de caractère* `'153'`

Exercice 4 *(Disco* vs. *hard rock).* On rappelle qu'un *carré parfait* est un entier s'écrivant n^2 avec $n \in \mathbb{N}$.
On appelle *nombre disco* tout carré parfait dont l'écriture en base 10 est ABBA, avec $A \neq B$ et $A \neq 0$.
On appelle *nombre hard rock* tout carré parfait dont l'écriture en base 10 est ACDC avec A, C, D distincts et $A \neq 0$.
Déterminer tous les nombres disco et tous les nombres hard rock.
Rappel. `[k**2 for k in range(5)]` *renvoie* `[0, 1, 4, 9, 16]`.

Exercice 5 *(Nombres apocalyptiques, d'après J.-M. Saint-Jalm).* Un *nombre apocalyptique* est un entier de la forme 2^n avec $n \in \mathbb{N}$ qui comporte au moins une séquence 666 dans son écriture décimale.
Trouver le plus petit nombre apocalyptique.

17 Καὶ ἵνα μή τις δύναται ἀγοράσαι ἢ πωλῆσαι, εἰ μὴ ὁ ἔχων τὸ χάραγμα, τὸ ὄνομα τοῦ θηρίου ἢ τὸν ἀριθμὸν τοῦ ὀνόματος αὐτοῦ.

18 Ὧδε ἡ σοφία ἐστίν. Ὁ ἔχων νοῦν ψηφισάτω τὸν ἀριθμὸν τοῦ θηρίου : ἀριθμὸς γὰρ ἀνθρώπου ἐστίν, καὶ ὁ ἀριθμὸς αὐτοῦ **ἑξακόσιοι ἑξήκοντα ἕξ**.

Jean, Apocalypse 13 :17-18.

Exercice 6 *(Attracteur de Sierpiński).* Voici un algorithme permettant de tracer un ensemble de points du plan.

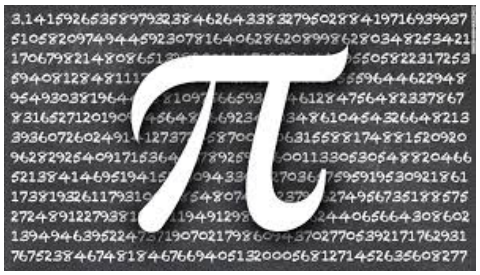
- Construire un triangle équilatéral ABC, de centre O.
- Placer un point M_0 arbitraire à l'intérieur de ABC.
- Soit $n \in \mathbb{N}$. Si le point M_n est placé, alors
  - choisir au hasard l'un des sommets S du triangle ABC.
  - placer le point M_{n+1} comme étant le milieu du segment $[M_nS]$.

Tracer les points M_n quand $n \in [\![0, 2000]\!]$.

Exercice 7 *(Approximation de π).* Il est bien connu des écoliers que $\frac{22}{7}$ est une approximation convenable de π.

1. Parmi toutes les fractions $\frac{a}{b}$ telles que $(a, b) \in [\![1, 99]\!]^2$ et $\text{pgcd}(a, b) = 1$, montrer qu'effectivement $\frac{22}{7}$ est la meilleure approximation de π.
2. Déterminer la meilleure approximation de π parmi les fractions de la forme $\frac{a}{b}$ telles que $(a, b) \in [\![100, 999]\!]^2$ et $\text{pgcd}(a, b) = 1$.
3. Au V$^{\text{e}}$ siècle ap. J.-C., le Chinois Zu Chongzhi a montré que cette dernière approximation est la meilleure parmi toutes les fractions dont le numérateur et dénominateur ont au plus 5 chiffres. Démontrer le résultat de Zu Chongzhi.

Fonctions utiles : `gcd` et `pi` du module `math`.



Exercice 8 *(La suite de Syracuse).* Programmer (avec la commande `def`) la célèbre suite $(u_n)_{n\in\mathbb{N}}$ définie par un premier terme $u_0 \in \mathbb{N}$ de votre choix, et pour tout $n \in \mathbb{N}$,

$$u_{n+1} = \begin{cases} \frac{1}{2}u_n & \text{si } u_n \text{ est pair,} \\ 3u_n + 1 & \text{sinon.} \end{cases}$$

On fera en sorte que u_n soit toujours du type `int`*, et jamais du type* `float`.

1. Observer sur quelques exemples que quelle que soit la valeur de u_0 choisie, il existe $n_0 \in \mathbb{N}$ tel que $u_{n_0} = 1$.

2. Trouver le plus petit entier n_0 convenant si $u_0 = 15$: cet entier s'appelle *la longueur du vol de 15*. On le note $v(15)$.
3. Tracer un graphique représentant les points de coordonnées $(n, v(n))$ quand n parcourt $[\![1, 100]\!]$.

Remarque. Le nom de *Syracuse* n'a pas de rapport avec la ville de Sicile où est né Archimède. C'est le nom de l'université américaine (État de New-York) où cette suite (et sa redoutable difficulté) a été introduite par le mathématicien L. Collatz, dans les années 50.

3 Exercices plus avancés

Exercice 9 *(L'énigme de Bérangère).* Bérangère (professeur d'EPS et néanmoins amie) me dit : « tiens le matheux, je parie que tu ne peux même pas trouver 26 en utilisant 2, 3, 4, 5 et les quatre opérations $+$, $-$, $\times$ et $\div$, mais sans utiliser deux fois le même truc [nombre ou opération, je traduis] mais en utilisant vraiment les quatre chiffres ! »

Donner tort à cette médisante sportive en écrivant un programme `Python` qui :
— trouve le résultat sous une forme de chaîne de caractères, par exemple :

$$\texttt{'(((2 + 3) } \times \texttt{ 4) - 5)'}$$

(qui n'est évidemment pas la solution),
— trouve toutes les solutions possibles, en convenant que `(((3 + 2)` $\times$ `4 ) - 5)` et `(((2 + 3)` $\times$ `4 ) - 5)` désignent la même solution.
— renvoie en plus le nombre de solutions au problème.

Clouer définitivement le bec de Bérangère en calculant le temps de calcul.

Exercice 10 Un *cryptarithme* (du grec κρυπτός : caché et ἀριθμός : nombre) est une devinette calculatoire du type

$$\text{MORE} + \text{SEND} = \text{MONEY}$$

où chaque lettre désigne un chiffre (entre 0 et 9) de sorte que l'opération ainsi créée soit exacte. Ce problème admet une unique solution : 1085 + 9567 = 10652 (donc M = 1, O = 0, Y = 2, etc.)

Créer un programme `Python` qui résout le cryptarithme suivant :

$$\text{UN} + \text{UN} + \text{NEUF} = \text{ONZE}$$

On calculera le temps de calcul en utilisant la fonction `clock()` qu'on peut charger en début de programme grâce à l'instruction : `from time import clock`.

Exercice 11 La série $\sum \frac{(-1)^{n+1}}{n}$ (définie sur $\mathbb{N}^*$) est convergente et il est classique de démontrer que sa somme vaut $\ln(2)$. En revanche, on peut montrer, et l'objet de cet exercice est de comprendre que si l'on change l'ordre de ses termes, on peut la faire converger vers n'importe quoi d'autre.

1. De quelle nature sont les séries $\sum \frac{-1}{2n}$ et $\sum \frac{1}{2n+1}$?
2. Soit $L \in \mathbb{R}$ quelconque. Expliquer comment construire une série $\sum a_n$ dont les termes sont exactement les mêmes que $\sum \frac{(-1)^{n+1}}{n}$, mais qui converge vers L.
3. Notons $u_n = \frac{(-1)^{n+1}}{n}$. On a ainsi créé une bijection $\sigma : \mathbb{N}^* \to \mathbb{N}^*$ qui « réordonne les termes de $\sum u_n$ », c'est-à-dire telle que $a_n = u_{\sigma(n)}$ pour tout $n \in \mathbb{N}^*$. Écrire une fonction `Python` prenant L et n en paramètre et renvoyant la valeur $\sigma(n)$.