Analyse 2
Suites

Exercice 1.
On s'intéresse à la suite de terme général $u_n = 1 + \dfrac{1}{n+1}$ pour $n \in \mathbb{N}$.
Calculer les 10 premiers termes de cette suite.

Exercice 2 (*).
On considère la suite définie par $u_n = \frac{n^{100}}{100^n}$ pour $n \geq 1$. Calculer les 10 premiers termes de la suite. Quelle est sa limite lorsque $n \to \infty$? Vérifier avec Sage.

Exercice 3.
Un point important lorsque l'on étudie un algorithme est sa complexité, c'est-à-dire une estimation de la quantité de ressources (temps ou espace) qui vont être nécessaires à l'exécution de cet algorithme.
Etudions l'algorithme suivant qui calcule la somme des n premiers entiers :

Algorithme 1 : Somme des n premiers entiers

	coût	fois
Entrée : n : entier		
Sortie : s : entier, la somme des n premiers entiers		
$s \leftarrow 0$	c_1	1
pour *i allant de 1 à n* **faire**		
$s \leftarrow s + i$	c_2	n
retourner s		

L'instruction $s \leftarrow 0$ a un coût c_1 et est réalisée une seule fois.
L'instruction $s \leftarrow s + i$ a un coût c_2 et est réalisée n fois.
Ainsi, si on note $T(n)$ le coût de cet algorithme pour le calcul de la somme des n premiers entiers, on a :

$$T(n) = 1 * c_1 + n * c_2.$$

Supposons par exemple que $c_1 = 1$ et $c_2 = 2$.

- Calculer les 20 premières valeurs de $T(n)$.
- Quel semble être le comportement de la suite $T(n)$ (croissante, décroissante, stationnaire) ?
- On souhaite maintenant voir graphiquement le comportement de cette suite.
  Pour cela, il faut représenter les points de coordonnées $(n, T(n))$.
  Créer une liste de listes `L`, qui contient les coordonnées des 20 premiers points $(n, T(n))$ et tracer ces points par la commande `list_plot(L)`.
- Calculer à la main l'expression de $T(n+1)$ en fonction de $T(n)$: la suite $T(n)$ est donc aussi définie par récurrence.

Une suite est *récurrente* (ou *définie par récurrence*) si son terme général est exprimé en fonction des termes précédents.

Exercice 4 (Suite de Fibonacci).
La suite de Fibonacci est définie par :

$$u_0 = u_1 = 1, \quad \text{et} \quad u_n = u_{n-1} + u_{n-2} \text{ pour } n \geq 2.$$

Cette suite représente par exemple l'évolution du nombre de couples de lapins chaque mois, si on place un couple de lapins dans un milieu clos.

- Programmer le calcul du 10e terme de cette suite.
- Modifier le programme pour ne pas faire d'appels récursifs, en utilisant une approche itérative.
- Comparer le temps mis par chacun des programmes pour calculer le 30e terme de cette suite.

Exercice 5 (Suite de Syracuse).
La suite de Syracuse est définie par la donnée de $u_1 \in \mathbb{N}^*$ et pour tout entier naturel n par la relation de récurrence suivante :

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

- Programmer le calcul des 11 premiers termes de la suite de Syracuse avec $u_1 = 6$.
- Une conjecture énonce que, pour toute valeur initiale $u_1 \in \mathbb{N}^*$, il existe un rang N tel que $u_N = 1$. Ecrire une fonction qui renvoie la plus petite valeur de N. Tester votre fonction pour plusieurs conditions initiales.

La suite (u_n) est une *suite arithmétique* s'il existe un réel r tel que :

$$\forall n \in \mathbb{N}, u_{n+1} = u_n + r.$$

Le nombre r est alors appelé la raison de la suite (u_n).

Exercice 6 (Un exemple de suite récurrente : suite arithmétique).
Dans un processeur, la quantité de chaleur dégagée est essentiellement liée à la quantité de transistors et à la taille de la gravure. Pour un ordinateur donné, une quantité fixe de chaleur est liée aux autres composants, notons-là c_0. Si r est la chaleur dégagée par un transistor, la quantité de chaleur pour n transistors, c_n vérifie : $c_{n+1} = c_n + r$.

- Supposons que $c_0 = 10$, $r = 1.5$, calculer les 10 premières valeurs de c_n.
  Cependant, un processeur contient des millions de transistors. Il faudrait pouvoir calculer c_n de manière explicite, sans calculer les $n - 1$ termes précédents.
- Calculer à la main l'expression de c_n en fonction de n et de r uniquement. Vérifier le calcul sur les valeurs calculées précédemment.
- Que vaut $c_{10000000}$?

La suite (u_n) est une *suite géométrique* s'il existe un réel q tel que :

$$\forall n \in \mathbb{N}, u_{n+1} = q * u_n.$$

Le nombre q est alors appelé la raison de la suite (u_n).

Exercice 7 (Un exemple de suite récurrente : suite géométrique).
La loi de Moore (proposée par G. Moore en 1975) prédit que le nombre de transistors double tous les deux ans. En 1971, l'Intel 4004 comportait 2300 transistors.
Etudier et représenter sur un graphique le nombre de transistors prédits par la loi de Moore jusqu'au début des années 2000 (à partir de cette date, la loi de Moore n'est plus totalement vérifiée).

Exercice 8 (Un exemple de suite récurrente définie par l'intermédiaire d'une fonction).
On considère la suite définie par :

$$u_0 = 1.3, \quad u_{n+1} = f(u_n) = \left| u_n^2 - \frac{1}{4} \right| \text{ pour } n \geq 0.$$

Calculer les 4 premières valeurs de la suite.
Représenter la suite en reliant les points avec la commande `line` grâce à la liste :

$$[[u_0, 0], [u_0, u_1], [u_1, u_1], [u_1, u_2], [u_2, u_2], [u_2, u_3], [u_3, u_3]].$$

Tracer également la courbe représentative de la fonction f, ainsi que la première bissectrice sur $[-1, 2]$.
Reprendre la question avec $u_0 = -0.4$ et $u_0 = 1.1$.