Exercice 1 : Étant donnés deux entiers a et b.

1. Écrire une fonction itérative qui calcule le pgcd de deux a et b.

   Le pgcd de a et b peut être calculé à l'aide de l'algorithme d'Euclide :
   — Si $b = 0$, alors $pgcd(a, b) = a$.
   — Sinon, $pgcd(a, b) = pgcd(b, a \mod b)$.
2. Écrire une fonction récursive qui calcule le pgcd de a et b.
3. Comparer les performances des deux fonctions pour différentes paires de valeurs de a et b de l'ordre de 2^{100}.

Exercice 2 : Étant donné n un entier positif. La suite de Fibonnaci est définie par la relation de récurrence $F(n+2) = F(n+1) + F(n)$, avec $F(0) = 0$ et $F(1) = 1$

1. Écrire une fonction de récursivité arborescente qui prend n en entrée et calcule $F(n)$
2. Écrire une deuxième fonction de récursivité terminale qui calcule $F(n)$
3. Comparer les performances des deux fonctions pour des valeurs élevées de n en analysant le temps d'exécution.

Exercice 3 : Une permutation est un arrangement ordonné d'objets. Par exemple, $[3, 1, 2, 4]$ est une permutation possible de $[1, 2, 3, 4]$.

1. Écrire une fonction récursive qui prend $n \geq 1$ en entrée et retourne la liste de toutes les permutations de $[1, 2, \ldots, n]$.
2. Écrire une fonction itérative qui prend $n \geq 1$ en entrée et retourne la liste de toutes les permutations de $[1, 2, \ldots, n]$.
3. Comparer les performances des deux fonctions pour des valeurs élevées de n, en analysant notamment le temps d'exécution et l'utilisation des ressources.

Exercice 4 : Soit a_n la suite définie par $a_1 = 1$, et de manière récursive pour $n \geq 1$:

$$\begin{cases} a_{2n} = 2a_n \\ a_{2n+1} = a_n - 3a_{n+1} \end{cases}$$

Les six premiers termes de la suite sont $1, 2, -5, 4, 17, -10$.

Soit $S(n) = \sum_{k=1}^{n} a_k$.

1. Écrire une fonction récursive qui retourne a_n pour un n donné.
2. Écrire une fonction récursive qui vous permettant de calculer $S(n)$. Ne calculez pas $S(2^{2025})$.
3. Écrire une fonction qui vous permet de calculer de grandes valeurs de n notamment $S(2^{2025})$.

Exercice 5 : Un nombre est dit palindromique (comme 27372) s'il reste identique lorsque ses chiffres sont inversés. En d'autres termes, il présente une symétrie réflexive par rapport à un axe vertical. Un nombre de Lychrel est un nombre naturel qui ne peut pas former un palindrome par le processus itératif consistant à inverser ses chiffres puis à additionner le nombre obtenu avec l'original. Ce processus est parfois appelé « algorithme 196 ». Par exemple :

— Le nombre 47 devient palindromique après une seule itération :

$$47 + 74 = 121$$

— Le nombre 49 devient palindromique après deux itérations :

$$49 + 94 = 143, \quad 143 + 341 = 484$$

— Le nombre 59 devient palindromique après trois itérations :

$$59 + 95 = 154, \quad 154 + 451 = 605, \quad 605 + 506 = 1111$$

Étant donné n un entier positif.

1. Écrire une fonction itérative qui retourne le nombre d'itérations nécessaires pour que n devienne un nombre palindromique en suivant le processus de l'algorithme 196.
2. Écrire une fonction récursive qui retourne le nombre d'itérations nécessaires pour que n devienne un nombre palindromique en suivant le processus de l'algorithme 196.
3. Combien d'itérations sont nécessaires pour que 196 devienne palindromique ?