

# Exercices supplémentaires fonctions

Christophe Viroulaud

Première - NSI

**Lang 07**

# Sommaire

1. Exercice 1

2. Exercice 2

3. Exercice 3

Exercice 1

Exercice 2

Exercice 3

# Exercice 1

Exercice 1

Exercice 2

Exercice 3

Écrire une fonction `volume(long: int, larg: int, haut: int) → int` qui calcule le volume d'un pavé.

## Avant de regarder la correction



- ▶ Prendre le temps de réfléchir,
- ▶ Analyser les messages d'erreur,
- ▶ Demander au professeur.

Exercice 1

Exercice 2

Exercice 3

Exercice 1

Exercice 2

Exercice 3

```
1 def volume(long: int, larg: int, haut: int) -> int:
2     """
3     renvoie le volume d'un pavé
4
5     Args:
6         long (int): longueur
7         larg (int): largeur
8         haut (int): hauteur
9
10    Returns:
11        int: volume
12    """
13    return long*larg*haut
```

# Sommaire

1. Exercice 1

2. Exercice 2

3. Exercice 3

Exercice 1

Exercice 2

Exercice 3

## Exercice 2

Exercice 1

Exercice 2

Exercice 3

Un entier naturel est dit parfait lorsqu'il est égal à la somme de tous ses diviseurs propres autres que lui-même. Par exemple 6 est parfait car  $6 = 1 + 2 + 3$ .

Écrire la fonction `est_parfait(n: int) → bool` qui vérifie si `n` est un nombre parfait.

## Avant de regarder la correction



- ▶ Prendre le temps de réfléchir,
- ▶ Analyser les messages d'erreur,
- ▶ Demander au professeur.

Exercice 1

Exercice 2

Exercice 3

Exercice 1

Exercice 2

Exercice 3

```
1 def est_parfait(n: int) -> bool:
2     """
3     vérifie si n est parfait
4     """
5     somme = 0
6     for i in range(1, n):
7         if n % i == 0: # i est multiple de n
8             somme += i
9     # la somme des multiples == n
10    return somme == n
```

# Sommaire

1. Exercice 1
2. Exercice 2
3. Exercice 3

Exercice 1

Exercice 2

Exercice 3

## Exercice 3

La conjecture de COLLATZ est un énoncé mathématique dont on ne sait pas encore s'il est vrai. Prendre un nombre entier positif, et lui appliquer lui le traitement suivant :

- ▶ s'il est pair, le diviser par 2
- ▶ sinon, le multiplier par 3 et ajouter 1

On obtient alors un nouveau nombre, sur lequel la procédure est répétée.

La conjecture s'énonce ainsi : **quel que soit l'entier choisi au départ, on finira par obtenir 1.**

Écrire une fonction `collatz(n: int) → bool` qui renvoie `True` si le processus s'achève, c'est à dire si on finit par atteindre 1.

Exercice 1

Exercice 2

Exercice 3

## Avant de regarder la correction



- ▶ Prendre le temps de réfléchir,
- ▶ Analyser les messages d'erreur,
- ▶ Demander au professeur.

Exercice 1

Exercice 2

Exercice 3

Exercice 1

Exercice 2

Exercice 3

```
1 def collatz(n: int) -> bool:
2     """
3     renvoie True si la procédure termine
4     Si la boucle ne se termine jamais, la
5     conjecture
6     est fausse.
7     """
8     while n != 1:
9         if n % 2 == 0: # pair
10            n = n//2
11        else:
12            n = 3*n+1
13    return True
```