

## 1.1 Variables, structure conditionnelle et autres boucles

### 1.1.1 Structure conditionnelle, boucles et autres réjouissances

- En fonction d'une valeur entière saisie par l'utilisateur (on utilisera à cette fin la fonction `input` de Python et on prendra bien garde à convertir le résultat en entier), afficher l'un des messages suivants :
  - “Ce nombre est pair”
  - “Ce nombre est impair, mais est multiple de 3”
  - “Ce nombre n'est ni pair ni multiple de 3”
- Dénombrer le nombre de voyelles dans une phrase que l'utilisateur saisira.
- Projet Euler n°1<sup>a</sup>: Calculer la somme des entiers divisibles par 3 ou par 5 dans l'intervalle  $[0, N[$  où  $N$  sera fourni par l'utilisateur.
- Projet Euler n°16: Donner la somme des chiffres de la représentation décimale de  $2^N$  où  $N$  est un entier naturel fourni par l'utilisateur. Faire de même avec  $N!$  (Projet Euler n°20).
- La suite de Fibonacci est définie de la façon suivante

$$\begin{aligned}\mathcal{F}_1 &= 1 \\ \mathcal{F}_2 &= 1 \\ &\dots \\ \mathcal{F}_n &= \mathcal{F}_{n-2} + \mathcal{F}_{n-1}\end{aligned}$$

- Calculer le résultat de cette suite pour un ordre  $n$  donné par l'utilisateur. On essaiera en particulier de tirer profit de l'affectation parallèle des variables dans Python.
  - Projet Euler n°2: Calculer la somme des termes de Fibonacci pairs jusqu'à une valeur  $N$  donnée par l'utilisateur.
  - Projet Euler n°25: Déterminer l'indice  $n$  du premier terme  $\mathcal{F}_n$  dont la valeur contient plus de 1000 chiffres.
- La formule de Wallis permet de calculer la valeur de  $\pi$  via l'expression

$$\pi = 2 \prod_{n=1}^{\infty} \frac{4n^2}{4n^2 - 1}$$

Déterminer le nombre d'itérations  $n$  nécessaires pour que la valeur de  $\pi$  soit correcte jusqu'à la 7<sup>ème</sup> décimale.

<sup>a</sup>Le *Project Euler* est situé à l'adresse suivante: <http://projecteuler.net>. Ce site propose un grand nombre de problèmes (587 en date du 22/01/2017) aux apprentis programmeurs comme aux programmeurs chevronnés.

## 1.2 Containers et fonctions

### 1.2.1 Utilisation des *containers* de Python

1. Reprendre le Projet Euler n°1 dont l'objet est le calcul de la somme des entiers divisibles par 3 ou par 5 dans l'intervalle  $[0, N[$  **en utilisant une liste établie "en compréhension"**. On s'aidera de la fonction intégrée `sum` pour calculer la somme.
2. Projet Euler n°6 : Calculer le résultat de la formule suivante

$$\left( \sum_{k=1}^n k \right)^2 - \sum_{k=1}^n k^2$$

où  $n$  sera saisi par l'utilisateur.

3. On s'aidera des méthodes `join` et `split` de la classe `str`. Demander à l'utilisateur de saisir une chaîne de caractères puis
  - (a) renvoyer une chaîne telle qu'un lecteur l'aurait lue de droite à gauche
  - (b) renvoyer une chaîne où le premier mot devient le dernier, le second mot devient l'avant dernier, ...
  - (c) renvoyer la chaîne de caractères saisie sans les voyelles
4. Demander à l'utilisateur de saisir son nom, prénom et âge sur une seule ligne et en utilisant le caractère `,` comme séparateur de champ.
  - (a) Présenter le résultat sous la forme "Votre nom et prénom sont XXX XXX et vous êtes agé(e) de XXX" en n'oubliant pas les majuscules des majuscules aux noms propres.
  - (b) Créer ensuite un dictionnaire indexé sur le nom de la personne et conservant son âge.
5. Demander à plusieurs utilisateurs de saisir nom, prénom et âge en utilisant toujours le caractère `,` comme séparateur de champ et stocker chacune des chaînes dans une liste. On terminera la saisie en entrant le caractère `$`. Construire un dictionnaire toujours indexé sur le nom mais qui conservera le prénom et l'âge de la personne. Présenter finalement chacune des entrées du dictionnaire sous une forme intelligible.

### 1.2.2 Modularité

- Dans un fichier intitulé `euler_project.py`, reprendre les différentes solutions aux projets Euler résolus jusqu'à présent et isoler ces solutions dans des fonctions dont le nom correspondra au numéro du projet préfixé du terme `euler`. Documenter chacune de ces fonctions puis résoudre les 587 problèmes proposés<sup>b</sup>
- Exécuter le fichier `euler_project.py` dans la console `ipython` à l'aide de la commande `%run euler_project.py`. Tester les différentes fonctions puis construire un dictionnaire qui associera le numéro du projet à la fonction idoine. Demander finalement à l'utilisateur quel numéro de projet résoudre, s'assurer que le projet a une solution et proposer cette solution.

<sup>b</sup>je plaisante ! ne le faites pas