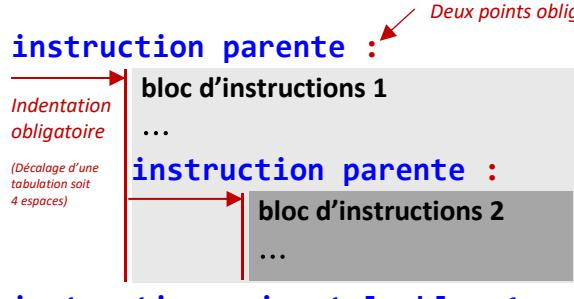


0. Structuration et indentation :



1. Instruction conditionnelle \Rightarrow if / elif / else

Le bloc d'instructions à exécuter dépend d'une (if), deux (if / else) ou plusieurs (if / elif /else) conditions à tester.

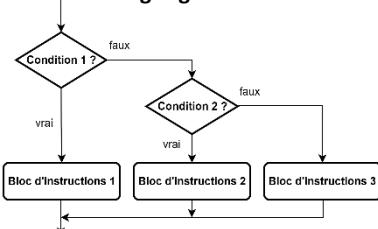
Pseudo code

```

si condition 1 vraie
alors exécuter bloc d'instructions 1
sinon si condition 2 vraie
alors exécuter bloc d'instructions 2
sinon
exécuter bloc d'instructions 3

```

Algorigramme



Code Python

```

if condition1 :
    bloc d'instructions 1
elif condition 2 :
    bloc d'instructions 2 } optionnel et répétable
else :
    bloc d'instructions 3 } optionnel

```

Exercice 1. Année bissextile

Écrire un programme indiquant si une année (rentrée au clavier) est bissextile.

Définition : Une année est bissextile si elle est :

- soit divisible par 4 mais non divisible par 100 ;
- soit divisible par 400.



Exercice 2. Location de voiture

Une agence de location de voiture propose deux formules de location au choix :

1. *Location au kilomètre*. Le prix de la location se calcule ainsi : 20 €HT / jour
 - pour les 100 premiers km : tarif 0.3 €HT du km ;
 - pour les kilomètres 101 à 1000 : tarif 0.2 €HT du km ;
 - au-delà de 1000 km : tarif 0.1 €HT du km.
2. *Forfait journalier*, pour lequel le kilométrage est illimité pour un prix fixe de 60 €HT par jour.

Dans les deux cas, il convient d'ajouter une assurance au prix de 5 €HT / jour ainsi que la TVA (20 %).

Écrire un programme qui, connaissant le nombre de jours de location et le nombre de kilomètres parcourus (données rentrées au clavier), permet de calculer le coût total de la location avec l'une ou l'autre des formules, ou de comparer les deux solutions afin de donner la plus avantageuse.

Exercice 3. Recherche dans une chaîne de caractères

Écrire un programme qui permet de savoir si une lettre est dans une chaîne de caractères. Toutes les deux devront être tapées au clavier et le programme devra afficher le résultat correspondant.

Exemples de résultats attendus :

▪ Rentrer la chaîne de caractères : <i>L'année 2020 est bissextile</i>	▪ Rentrer la chaîne de caractères : <i>L'année 2020 est bissextile</i>
▪ Rentrer un symbole : <i>a</i>	▪ Rentrer un symbole : <i>8</i>
▪ « <i>a</i> » est dans la chaîne : <i>«L'année 2020 est bissextile»</i>	▪ « <i>8</i> » n'est pas dans la chaîne : <i>«L'année 2020 est bissextile»</i>

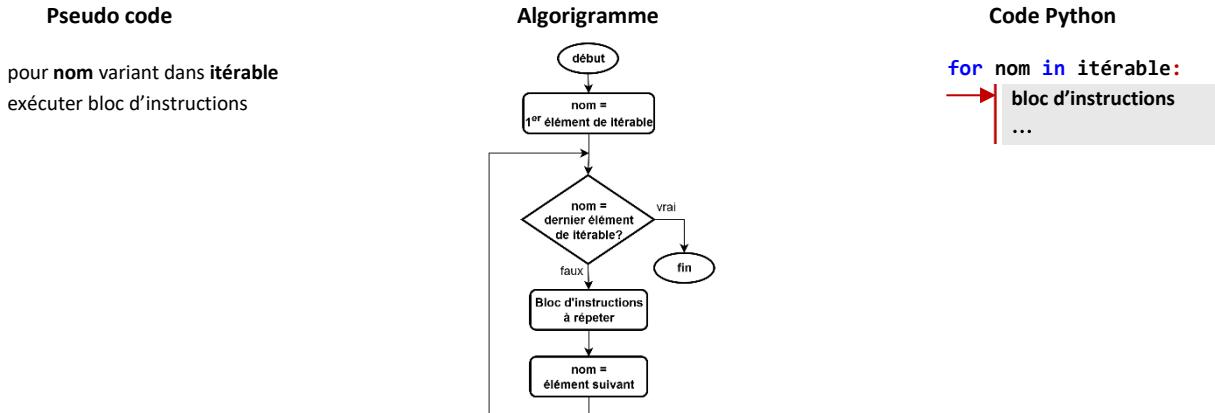
Exercice 4. Opérations sur deux entiers

Écrire un programme qui permet de saisir deux entiers et un code opération (+, -, * ou /) et qui affiche le résultat de cette opération sur ces entiers.

2. Instructions itératives :

2.a. Boucle inconditionnelle (énumérée) \Rightarrow for

Le bloc d'instructions à exécuter est répété pour chaque élément de l'itérable, désigné par un nom (nb d'itérations inconnu à l'avance).



Exercice 5. Factorielle n

Écrire un programme qui permet de calculer « n factorielle ($n!$) » (n étant un nombre entier positif).

On n'utilisera pas la fonction factorial du module math.

Rappel : $\forall n \geq 2, n! = \prod_{k=2}^n k$ avec $0! = 1$ et $1! = 1$

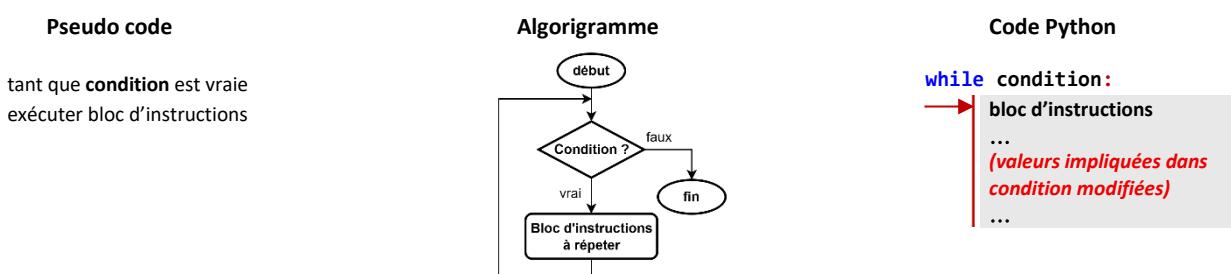
Exercice 6. Minimum, Maximum et Moyenne d'une suite de n nombres

Écrire un programme qui demande successivement n nombres à l'utilisateur (n étant donné par l'utilisateur) et qui donne le minimum, le maximum et la moyenne de ces valeurs.

Conseil : utiliser une liste.

2.b. Boucle conditionnelle \Rightarrow while

Le bloc d'instructions à exécuter est répété tant que condition est vraie (nb d'itérations inconnu à l'avance).



Exercice 7. Filtrage d'un type de nombre

Dans certains programmes, on peut demander à l'utilisateur de rentrer au clavier un type de nombre précis en utilisant la fonction « input() ».

Écrire un programme qui oblige à rentrer un entier positif et pas un flottant, des lettres ou tout autre caractère. Exemple de ce que ce programme doit renvoyer comme réponse en fonction de ce qui est tapé au clavier :

- ▶ Entrer un nombre entier positif : -15.8
-15.8 n'est pas un entier positif
- ▶ Entrer un nombre entier positif : **+@|#
**+@|# n'est pas un entier positif
- ▶ Entrer un nombre entier positif : 16.8
16.8 n'est pas un entier positif
- ▶ Entrer un nombre entier positif : 1e9
1e9 n'est pas un entier positif
- ▶ Entrer un nombre entier positif : 26
le nombre 26 est un nombre entier positif

Exercice 8. Recherche du plus petit diviseur d'un entier donné

Écrire un programme qui recherche le plus petit diviseur autre que 1 d'un entier n donné en paramètre.

Exercice 9. Nombre premier

Écrire un programme qui recherche si n est premier ou non.

Indications :

- x est premier si les restes des divisions euclidiennes $x / 2, x / 3, \dots, x / \sqrt{x}$ sont tous non nuls.
- On pourra utiliser la fonction « `sqrt` » (racine carrée) disponible dans la bibliothèque de fonctions mathématiques de Python. Pour importer cette fonction depuis la bibliothèque « `math` » de Python, écrire dans la 1^{ère} ligne de votre programme : `from math import sqrt`

Exercice 10. Recherche du PGCD de deux entiers naturels

Étant donnés deux entiers naturels a et b, écrire un programme qui calcule leur plus grand diviseur commun.

- On pourra écrire un premier programme, utilisant les propriétés suivantes du PGCD :

$$\begin{aligned} \text{PGCD}(a, b) &= \text{PGCD}(a - b, b) && \text{si } a > b \\ \text{PGCD}(a, b) &= \text{PGCD}(a, b - a) && \text{si } b > a \\ \text{PGCD}(a, a) &= a \end{aligned}$$

- On pourra écrire un deuxième programme, utilisant l'algorithme d'Euclide.

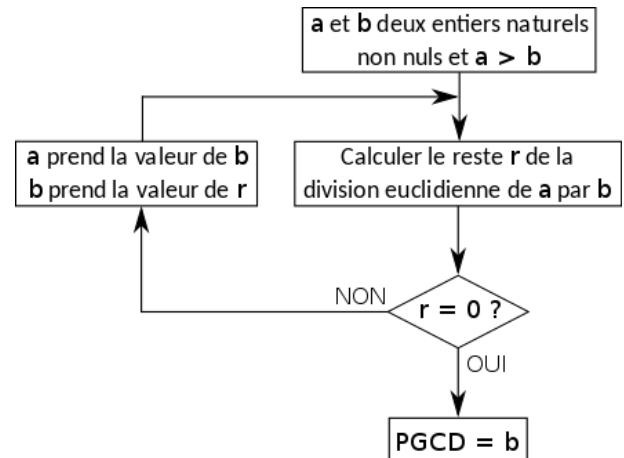
Algorithme d'Euclide :

Soit a et b deux entiers naturels, si b est non nul et inférieur à a, on peut effectuer la division euclidienne de a par b.

Il existe un couple unique d'entiers (q, r) tels que :

$$a = b \cdot q + r \quad \text{et} \quad 0 \leq r < b$$

- Si $r = 0$, alors b divise a et $\text{PGCD}(a, b) = b$.
- Si $r > 0$, alors tout diviseur commun à a et b est diviseur de r, et réciproquement, tout diviseur commun à b et r est diviseur de a.
On a donc $\text{PGCD}(a, b) = \text{PGCD}(b, r)$.



2.c. Instruction d'arrêt prématué de boucle \Rightarrow `break`

L'instruction `break` dans le bloc d'instructions d'une boucle `for` ou `while` termine la boucle prématièrement. Quoique pratique dans certains cas, cela gêne la compréhension du programme, on évitera donc son usage.