

ALGORITHME ET INSTRUCTIONS DE BASES

1- Qu'est-ce qu'un algorithme ?

Un algorithme est une suite finie et non-ambiguë d'instructions, qui une fois exécutée correctement, conduit à un résultat donné.

un algorithme est une suite d'instructions permettant de donner la réponse à un problème.

Le mot « algorithme » vient du nom du [mathématicien Al Khuwarizmi](#) (latinisé au Moyen Âge en *Algoritmi*), qui, au [IX^e siècle](#) écrivit [le premier ouvrage systématique](#) sur la solution des [équations linéaires](#) et [quadratiques](#).

Exemples :

- ✓ Les manuels d'utilisation actuels sont essentiellement des recueils d'algorithmes : des instructions sont données afin de faire fonctionner tel ou tel appareil.

- ✓ les recettes de cuisine : une recette de cuisine comporte trois étapes :

1. Réunir les ingrédients (les entrées)
2. Préparer (l' algorithme)
3. Déguster (la sortie)

- La préparation consiste à exécuter une suite d'instructions : par exemple, plonger les tomates dans une casserole d'eau bouillante pendant quelques instants avant de les peler.

- En comparant avec les algorithmes de mathématiques, on pourrait dire que les ingrédients de la recette sont les **entrées** du processus auxquelles on applique l'algorithme (la **préparation**) pour obtenir, en **sortie**, un plat que l'on dégustera avec plaisir (ou pas !)

2- Construction d'un algorithme

Un algorithme doit être lisible et compréhensible par plusieurs personnes. Il doit donc suivre des règles précises, il est composé d'un **entête** et d'un **corps** :

➤ **l'entête spécifique :**

- ✓ le nom de l'algorithme (**Nom** :)
- ✓ son utilité (**Rôle** :)
- ✓ les données "en entrée", c'est-à-dire les éléments qui sont indispensables à son bon fonctionnement (**Entrée** :)
- ✓ les données "en sortie", c'est-à-dire les éléments calculés, produits, par l'algorithme (**Sortie** :)
- ✓ les données locales à l'algorithmique qui lui sont indispensables (**Déclaration** :)

➤ **le corps, qui est composé :**

- ✓ du mot clef **début**
- ✓ d'une suite d'**instructions** indentées
- ✓ du mot clef **fin**

Exemple : addition de deux nombres :

Nom : addition

Rôle : additionner deux entiers et mémoriser le résultat

Entrée : a,b :Entier

Sortie : c :Entier

Déclaration :

Début :

c ← a+b

Fin

3- Les variables :

Une **variable** est un emplacement mémoire qui sert à stocker la valeur d'une donnée.

Son contenu peut changer au cours d'un programme (d'où le nom de variable)

Chaque emplacement mémoire a un numéro qui permet d'y faire référence de façon unique : c'est l'adresse mémoire de cette cellule.

Une variable doit être **déclarée** avant d'être utilisée, elle doit être caractérisée par :

- un **identifiant** : nom attribué à la variable
- une **valeur** qui peut évoluer au cours du programme
- un **type** qui indique l'ensemble des valeurs que peut prendre la variable (entier, réel, booléen, caractère, chaîne de caractères, ...)

4- L'affectation :

Après sa déclaration, une variable est une case « vide » : son contenu est indéfini.

L'affectation est l'opération qui donne un contenu à la variable.

En langage algorithmique, on utilisera de préférence $a \leftarrow 2 + 3$.

En Python, l'affectation est présentée par $=$. A ne pas confondre avec le test d'égalité qui est $==$.

Identifiant_de_la_variable \leftarrow nouvelle_valeur

5- Les instructions d'entrées et de sorties :

Les instructions de lecture et d'écriture permettent à la machine de communiquer avec l'utilisateur

✓ La **lecture** permet d'entrer des données à partir du clavier

- En pseudo-code, on note: **lire (var)**

la machine met la valeur entrée au clavier dans la zone mémoire nommée var

Remarque: Le programme s'arrête lorsqu'il rencontre une instruction Lire et ne se poursuit qu'après la saisie de l'entrée attendue par le clavier et de la touche Entrée (cette touche signale la fin de l'entrée)

Conseil: Avant de lire une variable, il est fortement conseillé d'écrire des messages à l'écran, afin de prévenir l'utilisateur de ce qu'il doit frapper

✓ **L'écriture** permet d'afficher des résultats à l'écran (ou de les écrire dans un fichier)

- En pseudo-code, on note: **écrire (liste d'expressions)**

la machine affiche les valeurs des expressions décrite dans la liste.

Ces instructions peuvent être des variables ayant des valeurs, des nombres ou des commentaires sous forme de chaînes de caractères.

Exemple :

Écrire un algorithme qui demande un nombre entier à l'utilisateur, puis qui calcule et affiche le carré de ce nombre.

Nom : Calcul_du_Carre

Rôle : calculer le carré d'un entier

Entrée : A : entier

Sortie : B :entier

Début

écrire("entrer la valeur de A ")

lire(A)

$B \leftarrow A * A$

écrire("le carré de ", A, "est :", B)

Fin