

# Chapitre 0

## Algorithmique

« Un algorithme est une suite finie de règles à appliquer dans un ordre déterminé à un nombre fini de données pour arriver, en un nombre fini d'étapes, à un certain résultat et cela indépendamment des données. »

Encyclopaedia Universalis

### I. Notion d'algorithme

Au collège, nous avons rencontré des algorithmes (algorithmes opératoires, algorithme des différences, algorithme d'Euclide, algorithmes de construction en géométrie). L'objectif de cette année sera de formaliser ces algorithmes en langage naturel.

#### Exemples :

- Dans la vie de tous les jours : je m'habille, je remplis mon sac, automates (machines industrielles, digicode, carte bancaire...)
- En mathématiques : opérations posées (multiplication, soustraction, division euclidienne), arithmétique (algorithme d'Euclide, des soustractions), algèbre (résolution équation du premier degré), géométrie (programme de construction, tracé de courbes).

### II. Langage de programmation

Afin de simplifier leurs formulations, des **langages de programmation** ont été mis en place. Il en existe beaucoup (un algorithme exprimé dans l'un de ces langages est un **programme**) mais le principe reste le même et est fondé sur l'utilisation de différentes **données** et **instructions**.

#### 1) Variables

Les algorithmes utilisent différents objets :

- nombres (pour effectuer des calculs),
- chaînes de caractères (pour afficher, classer,...),
- listes (pour stocker, étudier et comparer différents nombres ou chaînes ,...)
- ,...

C'est la première chose à faire : **déclarer** au **compilateur** (le lecteur de l'algorithme) combien de variables il va utiliser et signaler pour chacun de quel type il s'agit.

#### Exemple :

- **n** est un *nombre* (nom et type)
- **paris** est une *liste* (nom et type)
- **13** est une *chaîne de caractères* (nom et type)

Il va alors falloir attribuer à la variable un contenu.

## 2) Instructions

### Affectation

Une **affectation** est une instruction qui consiste à « remplir » une variable.

#### Exemple :

- **n** prend la valeur 56,2
- **paris** prend la valeur {2 ; 5 ; 8 ; 2}
- **13** prend la valeur *PORTO*

### Séquence

Une **séquence** est une suite d'**instruction** (l'ordre peut donc être important) qui permet de modifier les valeurs.

Les instructions dépendent bien entendu du type de la variable :

- pour les nombres : opérations, calculs,... ;
- pour les chaînes de caractères : longueur, tri,..
- pour les listes ou tableaux : taille, longueur,...

#### Exemple :

Division euclidienne :

on regarde combien de fois on rentre le diviseur ...  
puis on soustrait ...  
puis on abaisse le chiffre ...

### Structures

- structure alternative

Les instructions à effectuer dépendent parfois d'une **condition** :

#### Exemple :

Dans le cas de la réciproque du théorème de Pythagore : **si**  $AB^2 + AC^2 = BC^2$  **alors...** **sinon ...**  
Il faut donc mettre en place des **tests**.

- structure itérative

De même les instructions doivent parfois s'effectuer en **boucle**.

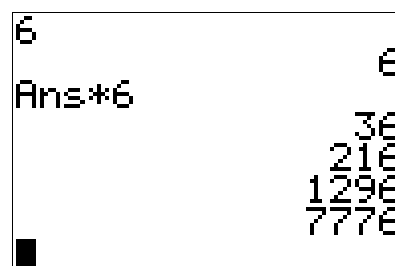
Le plus simple étant lorsque l'on répète la même instruction un certain nombre de fois déterminé à l'avance.

- pour **i allant de 1 jusqu'à n** (on répète *n* fois).

#### Remarque :

les calculatrices utilisées au collège et au lycée permettent d'effectuer des itérations de façon simple avec la touche **ANS**.

Exemple : Calculer  $6^5$



○ On ne sait pas toujours combien de boucles vont devoir s'effectuer :  
**tant que** le test est négatif on continue ... (on parle de **boucle conditionnée**)

**Exemple :**

pour le PGCD tant que reste  $\neq 0$ , on continue

### 3) **Entrée et sortie**

- Il reste évidemment à préciser que dans la plupart des cas il faudra que le compilateur lise des données (c'est l'utilisateur qui devra donner l'information, la plupart du temps à partir du clavier).

On parle de lecture (ou **entrée**) de données.

**Exemple :**

Pour l'algorithme de calcul du PGCD de deux nombres, il faut entrer les 2 nombres dans le programme.

- Et il est également intéressant d'obtenir un résultat : le compilateur écrit des données.

On parle d'écriture (ou **sortie**) de données.

**Exemple :**

Pour l'algorithme de calcul du PGCD de deux nombres, le programme doit donner le résultat (le PGCD!).

**Remarque :**

Les logiciels de géométrie dynamique permettent également de  
« programmer » des constructions.

On peut ainsi créer des « **Macros** ».

**Exemple :** Créer une macro pour la construction du cercle circonscrit

