

## Chapitre 2 : Algorithme séquentiel simple

0. Notion de langage et langage algorithmique
  1. Parties d'un algorithme
  2. Les données : variables et constantes
  3. Types de données
  4. Opérations de base
  5. Instructions de base
    - Affectations
    - Instructions d'entrée sorties
5. Construction d'un algorithme simple
6. Représentation d'un algorithme par un organigramme
7. Traduction en langage C

### Définition

**L'algorithme est la description précise de la méthode de résolution d'un problème sous forme d'instructions simples. L'algorithme est caractérisé par :**

- 1) Identification des informations intervenant dans un programme**
- 2) Abstraction : modélisation, formalisation et représentation de ces informations**
- 3) Énumération des opérations à appliquer sur ces informations.**
- 4) Définition de l'ordre des opérations à respecter.**

### Exemple :

On désire obtenir une quantité de capacité de 4l(litres) d'eau, sachant qu'on dispose seulement de deux seaux (récipients) un de 3l et un autre de 5l.

**Un algorithme de résolution de ce problème :**

1 Les informations du problème sont : les seaux 3l et 5l le but : obtenir 4l d'eau

2 Les opérations dans l'ordre :

- Remplir le seau de 3l.
- Le verser dans celui de 5l.
- Remplir le seau de 3l.
- Compléter le seau de 5l avec le contenu de 3l.
- Vider le seau de 5l.
- Mettre le reste du seau de 3l (1litre) dans le seau de 5l.
- Remplir le seau de 3l.
- Le verser dans celui de 5l.
- Voir le contenu de 5l.

*En **résumé** l'algorithme est une séquence finie d'actions permettant de résoudre un problème donné en adoptant les étapes de résolution d'un problème suivantes :*

1. Comprendre l'énoncé du problème,
2. Identification et représentation des entités intervenant:
  - Les données nécessaires,
  - Les données résultantes,
  - Les données intermédiaires
- 3 La démarche à suivre pour arriver aux résultats en partant d'un ensemble de données.
4. Elaboration et écriture d'un algorithme.

Les premiers algorithmes sont destinés à résoudre certains problèmes mathématiques simples, par exemple multiplier ou diviser des nombres. Comme l'exemple de calcul de la moyenne des 10 valeurs. Ils étaient appliqués manuellement avant l'invention des ordinateurs.

### I.4.2 – Traduction de l'algorithme en un programme :

Un **programme** est la liste d'instructions destinées à être exécutées par un ordinateur. Une instruction est l'action que l'ordinateur peut réaliser.

Un **langage de programmation** comme Pascal, C, ... permet de traduire nos algorithmes en des instructions réalisables (appelés aussi codes source) sur ordinateur. Le **code source** est un programme exprimé dans un langage de programmation compréhensible par un être humain, et ne pouvant pas être exécuté directement par l'ordinateur qu'après sa compilation. Chaque langage de programmation est doté d'un compilateur ou d'un interpréteur.

Le compilateur ou l'interpréteur est un **logiciel** qui permet de générer un code exécutable pour chaque code source.

#### Note :

L'étape de l'exécution est l'étape de l'exploitation du programme qui produit le résultat de la solution du problème. L'exécution est réalisée par la machine (l'ordinateur) sans ou avec la coopération de l'utilisateur du programme.

#### I.4.2.1 L'algorithme :

Informally, an algorithm is any well-defined computational procedure that takes some value, or set of values, as input and produces some value, or set of values, as output. An algorithm is thus a sequence of computational steps that transform the input into the output [cormen].

Rappelons que l'algorithme est une suite finie de règles et d'opérations à appliquer dans un ordre déterminé sur un ensemble fini de données pour obtenir en un nombre fini d'étapes à un ou plusieurs résultats.

Il est nécessaire de définir et de décrire parfaitement les données d'entrée et les résultats de sortie. Il est possible de décomposer le passage de ces données vers ces résultats en une suite d'opérations élémentaires.

Exercice :

Donner l'algorithme qui permet de calculer la surface d'un cercle.

Solution :

1 Analyse du problème

La représentation des informations :

- Rayon : rayon du cercle sa valeur est de type réel.
- Pi: est une constante nécessaire pour le calcul de la surface et de type : réel.
- Surface: constitue le but, la surface constitue le résultat désiré et cette surface est de type: réel.

2 Les opérations

Pour obtenir la surface il suffit de multiplier le carré du rayon par Pi.

#### **Résumé de la méthodologie :**

Il s'agit donc de comprendre la nature du problème en précisant les données (en entrée), les résultats (en sortie), et le procédé qui permet l'obtention du résultat à partir des données.

### I.4.2.2 L'expression d'un algorithme

Un algorithme peut être exprimé dans un langage informel, en texte libre (i.e. description des différentes étapes en langue naturelle comme le français), ou en un organigramme (diagramme représentant les étapes), mais préférablement en pseudo-code (version simplifiée d'un langage informatique) en adoptant un langage algorithmique.

#### Structure d'un algorithme :

il s'agit de la forme générale ou du squelette d'un algorithme

Un algorithme est structuré(composé) de 3 parties :

**Entête** : est la partie où est défini le nom de l'algorithme.

**Environnement** : est la partie où sont citées les déclarations des objets (lister les entités variables, constantes et type).

**Actions** : est la partie où sont citées les opérations.

#### La structure d'un algorithme et syntaxe en pseudo-code:

Entête	{	<b>Algorithme</b> nom_algorithme ;
Environnement	{	<b>const</b> nom_constant1=valeur1 ; nom_constant2=valeur2 ; ..... <b>Type</b> ....  <b>var</b> nom_var1 : type ; nom_var2 : type ; .....
Actions	{	<b>début</b> action1 ; action2 ;  <b>fin.</b>

#### **Définitions :**

##### **Environnement :**

Appelé aussi partie déclaration composée d'Ensembles d'objets ou chaque objet appartient à une seule famille qui va déterminer son type.

##### **Type :**

C'est un attribut qui indique la nature d'un objet et surtout sur le genre d'actions qu'on peut lui appliquer.

##### **Var :**

Dans cette partie on définit des noms et des représentations des données et des objets nécessaires dans un algorithme.

##### **Note :**

Action appelée aussi Instruction. Dans cette partie où sont exprimées les opérations. Les détails seront donnés dans le chapitre suivant.

### I.4.2.3 L'algorithmique et le langage de programmation C

Ici nous donnons une idée sur ce que peut être un algorithme et son programme en C. Pour le problème de l'exercice de calcul de la surface du cercle : l'algorithme a la forme de gauche et le programme C est représenté dans la partie droite.

#### Algorithme

```
Algorithme Surface ;
Const  Pi=3.14;
Var

    surf: réel;
    ray: réel;
Début
    Lire (ray) ;
    Surf ← ray*ray*Pi ;
    Ecrire (surf) ;
Fin.
```

#### Programme C

```
#include <stdio.h>
#define Pi 3.14
main()
{
    float surf;
    float ray;

    scanf("%d",&ray) ;
    surf= ray*ray*pi ;
    printf ("%d",surf) ;
}
```

Les instructions ( lire ,ecrire , ← , scanf et printf ) seront détaillées dans le chapitre suivant. Dans ce chapitre on verra à quoi sert chacune de ces instructions dans le but d'introduire et d'anticiper les notions de base de la machine de Van Newman.

#### **Note :**

**lire** permet l'entrée de valeur du clavier vers la mémoire.

**Ecrire** permet d'afficher une valeur sur écran.

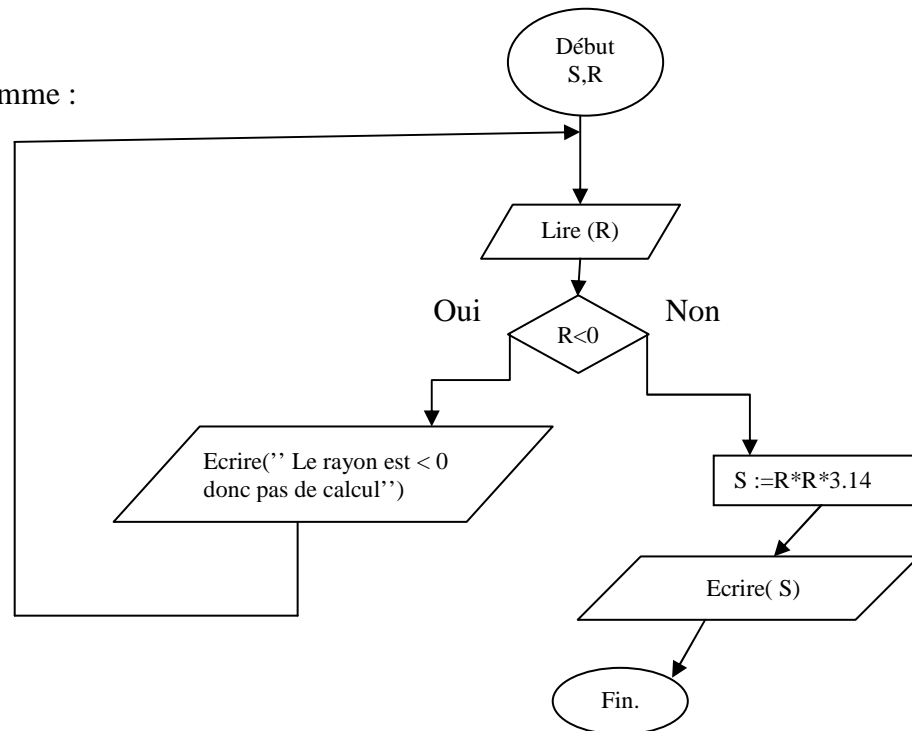
**←** Instruction(action) d'affectation permet d'attribuer une valeur à la mémoire qui est représentée par '=' en C.

### I.4.3 L'algorithme une représentation par organigramme

Lorsqu' on élabore un algorithme simple et de taille petite il est possible de le représenter par un graphe (schéma). Ce graphe présente un avantage d'exprimer un algorithme visuellement accessible à la compréhension seulement au regard. Et un inconvénient de ne pouvoir représenter de grands algorithmes. En tant que débutant il est intéressant d'avoir une visualisation de l'algorithme car il permet de mieux distinguer les différentes parties de l'algorithme et surtout pour comprendre la nécessité des différentes structures de contrôles : les tests, les choix, et les boucles.

**Exemple :**

Soit l'organigramme :



Que fait cet organigramme ?

Quel est le rôle des différents symboles : flèches, cercles et rectangles ?

Qu'exprime le losange ?

Quelles sont les différents types d'instructions.

**Exercices :**

- 1) Ecrire un algorithme qui détermine la valeur de deux nombres connaissant leur somme et leur différence.
- 2) Donner un Organigramme qui permet de résoudre une équation du premier degré.