**Université de Badji Mokhtar – Annaba - Faculté de Médecine**
**Module : Informatique pour les étudiants de la 1ère année**
**Enseignante : Dr GUESSOUM S.**

# Cours n° 1 : Introduction à l'Algorithmique

## 1.1 Quelques définitions

- *Une instruction* : est un ordre donné par le programmeur à l'ordinateur dans un programme pour faire une action spécifique.

- *Un langage de programmation* : est un ensemble de vocabulaire et de règles grammaticales compris par l'ordinateur (exp : Pascal, Delphi, Java), et avec lequel nous pouvons réaliser des programmes exécutables par l'ordinateur.

- *Un programme* : une suite ordonnée d'instructions écrites dans un langage de programmation compréhensible par l'ordinateur dans le but d'être exécutée pour donner des résultats. Un programme a toujours un objectif de résoudre un problème donné.

- *Un algorithme* : est une suite logique d'instructions élémentaires bien ordonnée écrite dans un langage naturel (exp : en français), et il constitue le passage obligatoire pour un programmeur avant d'arriver à son programme.

- *Un programmeur* : c'est quelqu'un qui a appris à faire des programmes, généralement un informaticien. L'apprentissage de la programmation passe obligatoirement par l'apprentissage de l'algorithmique.

- *Un utilisateur* : c'est quelqu'un qui utilise des programmes faits par des programmeurs. Il est en possession des données qu'il passera au programme pour attendre en retour des résultats.

## 1.2 Les étapes de programmation

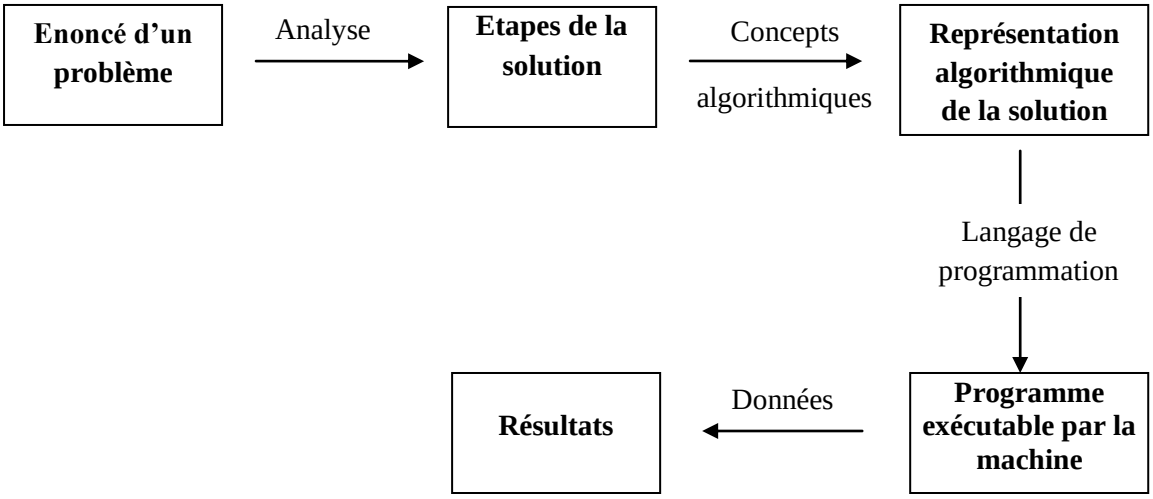
Programmer démarre toujours par établir une description détaillée d'un problème donné. Le schéma ci-dessous montre les différentes étapes par lesquelles nous devons passer pour réaliser un programme informatique exécutable par la machine et obtenir ses résultats de manière automatique.

- *L'analyse* sert à déterminer des différentes étapes à suivre pour obtenir une solution « manuelle » qui peut être décrite par un schéma, un paragraphe, une liste des grandes lignes à suivre… Ces différentes étapes constituent les éléments de base de notre programme.

- *Les concepts algorithmiques* servent à donner une autre forme aux étapes de la solution obtenue dans l'étape précédente, plus proche de la forme d'un programme tout en travaillant en langage naturel.

- *Le langage de programmation* sert à traduire l'algorithme obtenu dans l'étape précédente à un programme compréhensible et exécutable par la machine.

- *L'exécution* en dernier lieu, et pour avoir des résultats, il faudrait passer les données au programme.




Lorsque le programmeur estime que les résultats du programme sont faux, il doit revoir toutes les étapes du schéma (en marche arrière) pour détecter l'origine de son erreur.

## 1.3 Les concepts algorithmiques

Un algorithme est un programme écrit en langage naturel alors qu'un programme proprement dit est écrit en langage de programmation. Le passage d'un algorithme à un programme n'est en fait qu'une histoire de traduction. Et pour simplifier ce passage, nous allons adopter la même syntaxe du langage « pascal » à nos algorithmes.

### 1.3.1 Structure générale d'un algorithme

**Algo <nom de l'algo> ;** {entête de l'algo}

**Déclaration** des constantes et des variables ; {utilisées dans le bloc d'instruction}

**Début** {début du corps de l'algo}

Suite des instructions ;

**Fin.** {fin du corps de l'algo}

- ***Entête d'un algorithme***

Un algorithme commence toujours par un en-tête qui se compose du mot '**algo'** suivi du **nom de l'algorithme** attribué par le programmeur lui-même. Ce nom est une chaine de caractères qui peut être significative ou non (exp : algo calcul ; algo surface ; algo abc5). Mais il est toujours préférable que ce nom soit le plus significatif possible, c'est-à-dire qu'il reflète l'objectif de l'algorithme, pour permettre une distinction facile entre nos algorithmes. L'entête se termine par **un point-virgule** qui joue le rôle de séparateur dans les algorithmes. Ici il sépare l'en-tête de la déclaration.

- ***Déclaration des variables et des constantes***

Dans un algorithme nous manipulons des variables et des constantes. La différence entre ces deux est que les premières peuvent changer de valeurs pendant l'exécution du programme alors que les constantes gardent la même valeur tout au long de l'exécution. Déclarer des constantes ou des variables sous-entends citer leurs identificateurs (noms) dans la partie déclaration de l'algorithme.

L'objectif de la déclaration est la *réservation des cases mémoires* au niveau de la mémoire centrale. Ces cases vont servir pendant l'exécution du programme pour contenir les valeurs des constantes et des variables. Toutes les constantes et toutes les variables utilisées dans le bloc d'instructions doivent être déclarées, le cas échéant, elles ne peuvent être utilisées par le programme.

La déclaration des constantes commence par le mot '**Const**', suivi du **nom de la constante** suivi de **sa valeur** suivi d'un **point-virgule**. Dans le cas où le programme utilise plusieurs constantes, celles-ci doivent être séparées par des points-virgules et le mot 'Const' sera porté une seule fois au début de la déclaration.

```
Exp 1 : const A =13 ;

Exp 2 :  const A= 13,5 ;
                 B= 9 ;
                 Pi = 3,14 ;
```

La déclaration des variables commence par le mot « **Var** » suivi des noms des variables suivis de leurs types. Le type peut être : entier, réel ou logique.

```
Exp  :

Var    x, y, z5 : réel ;
       nombre : entier ;
       trouver : logique ;
```

- Une variable de type « réel » peut accepter des valeurs réelles (avec virgule). Exp : x=21 ; y = 34.5 ; z5=0.
- Une variable de type « entier » accepte uniquement des valeurs entières (sans virgule). Exp nombre=15.
- Une variable de type « logique » accepte uniquement les deux valeurs : vrai et faux. Exp : trouver=vrai.

Quand des variables du même type sont citées dans une liste elles sont séparées par des virgules.

***Remarque*** : dans la déclaration, lorsque l'algorithme/programme comporte des constantes, elles doivent être toujours déclarées avant les variables.

- ***Le bloc d'instructions***

C'est le corps de l'algorithme. Il commence toujours par le mot '**Début**' et se termine obligatoirement par le mot '**Fin**'. Entre les deux nous mettons toutes les instructions de l'algorithme nécessaires pour résoudre notre problème. L'ordre des instructions doit être bien étudié pour que l'algorithme donne le résultat attendu. Les instructions doivent impérativement être séparées par un point-virgule.

***Remarque*** : Nos programmes pascal vont obéir à cette même structure algorithmique (en-tête, déclaration, bloc d'instructions). La traduction d'un algorithme à un programme pascal consiste à traduire les mots clés algorithmiques vers leurs équivalents en pascal. Le tableau suivant donne la traduction des mots clés vus dans ce 1er cours.

| **Algorithme** | **Pascal** |
|---|---|
| Algo | Program |
| Const | Const |
| Var | Var |
| Entier | Integer |
| Réel | Real |
| Logique | Boolean |
| Début | Begin |
| Fin | End |