

# CHAPITRE 1 :

## INITIATION À LA PROGRAMMATION

### I- INTRODUCTION AUX ALGORITHMES

#### **I-1) Définitions :**

- Un algorithme est une méthode structurée de résolution d'un problème. Il a pour but de décrire le travail qu'on veut que l'ordinateur effectue à notre place. Un bon algorithme ne doit dépendre ni du langage dans lequel il est traduit, ni de la machine sur laquelle il va être exécuté.
- L'algorithmique ou algorithmie est l'étude des techniques de conception des algorithmes. Ainsi, dans le cadre de notre cours, nous allons concevoir des algorithmes orientés **Pascal**.
- Programme : traduction d'un algorithme dans un langage de programmation.
- Un langage de programmation est un ensemble de règles s'appliquant à un vocabulaire particulier dont le but est de traduire des algorithmes en programmes.
- Une structure de données est un objet permettant de garder des informations pendant et après les traitements. On l'appelle aussi variable.
- Implémentation : validation d'un programme à l'issue de plusieurs tests.
- Le code source : texte original d'un programme informatique ou d'un script.
- Portabilité : aptitude d'un programme à être utilisé sur des machines différentes.

#### **I-2) Objectifs :**

Les objectifs de cet enseignement sont :

- Rendre l'étudiant capable de raisonner comme un ordinateur, c'est-à-dire savoir décortiquer un problème complexe en suite finie d'opérations simples (instructions).

- Savoir concevoir de bons algorithmes et faciliter l'apprentissage de nouveaux langages. Un bon algorithme est un algorithme qui occupe moins d'espace dans la mémoire et qui permet d'obtenir rapidement les résultats. Le fait d'améliorer la qualité d'un algorithme s'appelle l'**optimisation**.

### **I-3) Intérêts des algorithmes :**

Nous pratiquons plusieurs algorithmes dans la vie de tous les jours : en allant à l'école le matin, en préparant un repas, en organisant une fête, etc.

Cependant, la maîtrise de la conception des algorithmes informatiques permet, grâce à un langage compris par l'ordinateur (C, PASCAL, PHP, BASIC...), de demander à celui-ci de réaliser un traitement spécifique. Le résultat de ce traitement est appelé **programme**.

Un Programme n'est pas en lui-même un élément de résolution d'un problème. Il traduit juste un algorithme pour produire des valeurs cherchées à partir des valeurs connues.

**NB :** On utilise les programmes pour exploiter la formidable capacité de calcul de l'ordinateur.

### **I-4) Les 4 étapes de construction d'un programme :**

**Étape 1 :** ici on décrit clairement le problème à résoudre pour savoir s'il peut être solutionné par l'ordinateur. (e.g : si le problème est de faire la lessive, la réponse est non)

**Étape 2 :** ici on échafaude un modèle de résolution du problème. Ce modèle est en général une relation existant entre les grandeurs connues et les grandeurs cherchées. (Exemple : si le problème est de calculer le carré d'un nombre, la relation est  $C=N*N$ )

**Étape 3 :** dans cette étape il est question de choisir les structures de données qu'on va utiliser ainsi que les algorithmes à mettre en œuvre. Ces algorithmes devront contenir la description détaillée des actions à exécuter pour résoudre le problème. (Exemple : pour concevoir un algorithme qui fait la somme de deux nombres, il faut : 1<sup>o</sup>) recueillir les valeurs de ces deux nombres 2<sup>o</sup>) faire leur somme 3<sup>o</sup>) afficher le résultat)

**Étape 4 :** c'est le choix d'un langage de programmation pour l'implémentation finale. Ce choix est fait en fonction du résultat qu'on veut obtenir, car chaque langage a ses forces et ses faiblesses.

## **II- LA PROGRAMMATION**

### **II-1) Définitions :**

Comme nous l'avons vu plus haut, la programmation c'est la technique qui consiste à traduire un algorithme dans un langage compréhensible par l'ordinateur, et ensuite à l'exécuter. Il existe plusieurs sortes de programmation parmi lesquelles :

- La programmation structurée : c'est une programmation dans laquelle on utilise des structures de contrôle standardisées (boucles, conditions...). C'est le cas du langage **Turbo Pascal** que nous allons étudier.
- La programmation événementielle : ici le programme est défini suivant les principaux événements produits par l'utilisateur (clic de souris, focus, sélection de texte...). Les langages qui admettent ce type de programmation sont par exemple le **JavaScript** ou encore le **Visual Basic**.
- La programmation orientée objet (POO) : elle consiste à modéliser informatiquement un ensemble d'éléments du monde réel en un ensemble d'entités informatiques. Ces entités informatiques sont alors appelées **objets**. Exemple de langage : **UML**.
- La programmation fonctionnelle : ici tous les programmes sont des fonctions.
- La programmation visuelle : ici l'utilisateur réagit essentiellement au moyen de la souris. Ce type de programmation n'est pas très utilisé pour les gros projets.

### **II-2) Les différents types de langages :**

Nous venons de voir quelques types de programmations. Toutefois, il est utile de remarquer qu'un langage de programmation peut appartenir à plusieurs de ces types en même temps. C'est par exemple le cas du C++ qui permet en même temps de faire des programmations structurées ainsi que des programmations orientées objets.

Ceci dit, il existe deux grandes catégories de langages de programmation : les langages **compilés** et les langages **interprétés**. Lorsqu'un algorithme est traduit dans un langage de programmation pour être exécuté par l'ordinateur, cela peut donc se faire de 2 manières :

- 1- La compilation consiste à vérifier tout le contenu du code source avant de lancer l'exécution du programme. S'il y a une seule erreur de syntaxe, l'implémentation est

annulée et aucune partie du programme n'est exécutée. Si par contre la compilation se déroule avec succès, un nouveau fichier (exécutable) est généré.

2- L'interprétation consiste à interpréter chaque ligne du code source une à une, et à l'afficher au fur et à mesure. S'il y a erreur sur une ligne, l'interpréteur l'ignore et continue le traitement avec le reste de la page. Ce type de programme est très répandu dans les scripts de conception de pages web (HTML, JavaScript...)

**NB :** pour qu'un programme soit exécuté sur un ordinateur, il faudrait que cet ordinateur possède l'environnement permettant de compiler ce programme, ou alors de l'interpréter. Dans certains cas il sera donc nécessaire d'installer un **compilateur** ou un **interpréteur**.

### **II-3) Le langage machine :**

En fait, pendant l'exécution d'un programme, chaque instruction décrite dans l'algorithme représente en réalité une action à effectuer par le processeur. Hors le seul langage que comprenne le processeur c'est le **langage machine**.

Le langage machine est un codage complexe de 0 et de 1 incompréhensibles par l'être humain. C'est pourquoi des langages plus évolués ont été mis au point afin de faciliter la conception de programmes sophistiqués.

Cependant, quel que soit le langage évolué utilisé, il est d'abord traduit en langage machine avant d'être exécuté par le processeur.

Ainsi, plus un langage est proche du langage machine, plus il est dit « de bas niveau ». Plus il s'en éloigne, plus il est dit « de haut niveau ». L'**Assembleur** est un exemple de langage de bas niveau, et le **PASCAL** est un exemple de langage de haut niveau.

## **TRAVAUX DIRIGÉS**

1. En ignorant tous les langages cités dans ce chapitre, faire des recherches et citer 3 langages de programmation compilés ainsi que 3 langages de programmation interprétés.
2. La plupart des navigateurs possèdent des environnements permettant d'interpréter les scripts du web. Citer donc 4 navigateurs en dehors d'Internet Explorer.