
Introduction

L'informatique est une discipline qui pénètre chaque jour un peu plus les autres sciences ; est devenu de plus en plus omniprésente dans tous les secteurs d'activité humaine. Cette progression nous amène à accorder plus d'importance à l'algorithmique et par conséquent à la programmation qui font l'origine de cette évolution.

L'algorithmique, est très simplement une méthode ou une façon systématique de procéder pour faire quelque chose. Il est souvent synonyme de logique et de rigueur puisque il permet finalement aux développeurs de présenter leurs idées devant une machine neutre. C'est pourquoi que les nouveaux apprenants en informatique doivent obligatoirement passer par le module de l'algorithmique avant tous les autres modules.

Dans cette perspective, nous proposons ce cours d'algorithmique qui s'adresse aux étudiants de la première année tronc commun mathématiques et informatique. Ce support est destiné à des étudiants qui ne savent pas programmer et qui n'ayant aucune familiarisation avec le développement informatique.

L'objectif typique de ce cours est de permettre à l'étudiant débutant de découvrir les notions de base en algorithmique et en programmation en langage C et par conséquent se familiariser avec les notions indispensables.

Le présent support est organisé autour de six chapitres qui constituent le cursus de la première partie de ce cours (ASD 1). Nous introduisons à travers ces chapitres les notions liées à un algorithme séquentiel simple, les structures conditionnelles, les boucles, les tableaux et les types personnalisés avec leurs traductions et leurs utilisations en langage C. A la fin de ce support l'étudiant peut trouver une collection riche d'exercices corrigés qui touchent tous le programme présenté dans ce support, les corrigés de ces exercices sont donnés en mode algorithmique et en langage C.

Chapitre

1

Introduction à l'algorithmique

Objectif

L'objectif de ce chapitre est de donner un aperçu historique sur l'évolution de l'informatique ainsi que quelques notions de base sur l'algorithmique pour pouvoir l'utiliser, tout en préservant ses caractéristiques de qualité.

Introduction

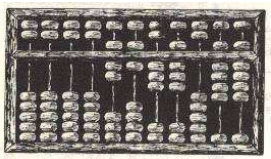



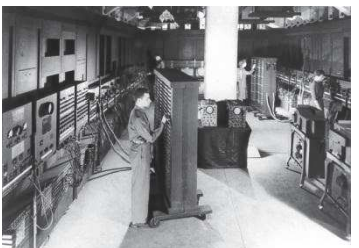
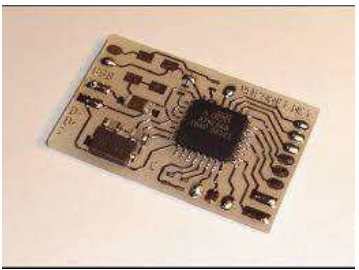
La notion d'algorithme est antérieure à l'apparition des ordinateurs. L'homme depuis longtemps essaye de déterminer des procédés suffisamment précis pour résoudre ses problèmes et pour organiser ses activités. Avec l'avènement de l'informatique, cette notion est devenue plus restrictive car un objectif supplémentaire est de pouvoir transformer un algorithme en un programme. Dans ce chapitre, nous allons faire un survol rapide sur l'évolution de l'informatique et introduire la notion algorithmique avec ses spécificités et ses facteurs de succès.

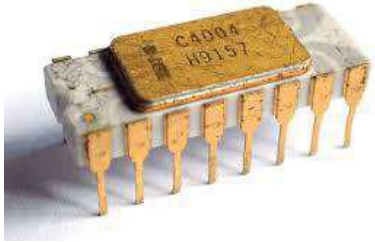




1. Qu'est-ce que l'informatique ?

Le mot informatique a été créé en 1962 par Philippe Dreyfus. Il s'agit d'une contraction des deux mots « automatique » et « information » *INFOR*mation *auto***MATIQUE**. Selon l'Académie Française, nous pouvons définir l'informatique comme la science du traitement rationnel, notamment par machines automatiques, de l'information considérée comme le support des connaissances et des communications, dans les domaines techniques, économiques et sociaux. Les anglo-saxons utilisent les termes de « computer science » ou de « data-processing » [1].

2. Brève histoire de l'informatique

L'informatique est apparue au milieu du 20ème siècle, elle a connu une évolution extrêmement rapide. L'informatique est présente dans tous les domaines de notre vie quotidienne : industrie, gestion, calculs scientifiques et techniques, enseignement, télécommunication, jeux, etc. L'histoire de l'informatique résulte de la conjonction entre des découvertes scientifiques de plusieurs disciplines techniques et sociales. La progression de l'automatique de l'électronique et des techniques de calculs a contribué fortement à la naissance et l'évolution rapide de l'informatique. Dans ce qui suit, nous passons en revue les événements scientifiques les plus importants de l'histoire de l'humanité, qui ont eu le plus grand impact sur l'apparition et l'évolution de l'informatique [2].

<p>La préhistoire : de 3000 AC à 1900</p> <ul style="list-style-type: none"> • abaqués à Babylone (3000 avant notre ère) • Formalisation du calcul : Al Khawarizmi (IXème) • XVIIème <ul style="list-style-type: none"> - Pascal : machine à additionner - Leibniz : système binaire pour le calcul • XVIIIème <ul style="list-style-type: none"> - Jacquard : métier à tisser - Babbage : machine différentielle • XIXème <ul style="list-style-type: none"> - Boole : calcul binaire et calcul logique 	<p>Abaque</p>  <p>El Khawarizmi</p>  <p>Blaise Pascal</p> 
<p>De 1900 à 1940</p> <ul style="list-style-type: none"> • Machine de Turing 	<p>Alan Turing</p> 
<p>Les années 40</p> <ul style="list-style-type: none"> • Travaux en cryptographie (décodage : Machine Enigma, Colossus) • Parallèlement, en Angleterre, en Allemagne et aux USA, construction des premiers ordinateurs : <ul style="list-style-type: none"> - Calculateur électromécanique Mark 1 (Aiken, 44) - 1946 : ENIAC Calculs balistiques (Atanasoff, Mauchly Eckert, 46) - 1944 : EDVAC, Mauchly Eckert et Von Neumann - 1948 : EDSAC, ... - Invention du transistor (Baarden, Brattain et Shockley 47) 	 <p>Ordinateur ENIAC</p>
<p>Les années 50</p> <ul style="list-style-type: none"> • Compilateurs (FORTRAN en 57) • LISP en 58 • Circuits intégrés en 59 	 <p>Circuit intégré</p>

<p>Les années 60</p> <ul style="list-style-type: none"> • Systèmes d'exploitation • Basic en 64 • Automates – Langages formels - Correction de programmes • Knuth : The Art of Computer Programming • Micro-processeurs • En 1962, création d'un réseau de communication militaire par l'US Air Force. • En 1969, création du réseau expérimental ARPANET par l'ARPA, qui est aujourd'hui considéré comme le réseau précurseur d'internet. 	 <p>Micro-processeur</p>
<p>Les années 70</p> <ul style="list-style-type: none"> • Base de Données Relationnelles • Unix et C (Thompson et Richie) • Pascal et Ada • Architecture RISC (IBM), Cray 1 en 76 • En 1972, Ray Tomlinson mit au point le courrier électronique. • 1972, le réseau ARPANET fut présenté au public • En 1976, création du protocole TCP. • En 1978, le protocole TCP fut fragmenté en deux protocoles : TCP et IP. 	 
<p>Les années 80</p> <ul style="list-style-type: none"> • Micro-ordinateur personnel (Apple – MacIntosh en 84) • NFSNet en 87 : Ancêtre d'Internet • Premiers virus en 88 • Dès 1980, création d'un système de navigation hypertexte par Tim Berners-Lee. Il développa, avec l'aide de Robert Cailliau, le logiciel Enquire permettant de naviguer. • En 1984, création du système de nommage DNS. • Fin 1990, création du protocole HTTP et du langage HTML par Tim Berners-Lee. 	 



<p>Les années 90</p> <ul style="list-style-type: none"> • Sortie du Windows • La fin de la Micro - l'ère du compatible PC 	
<p>Les années 2000</p> <ul style="list-style-type: none"> • Ressort des OrdiPhone (Smart Phone et iPad) et lancement de la programmation Mobile. 	

Tableau 1. Histoire et évolution de l'informatique

3. Introduction à l'algorithmique

Le mot « algorithme » vient de la transcription latinisée d'Al-Kwharizmi, nom d'un célèbre mathématicien arabe, et du mot grec *arithmos* qui signifie « nombre ».

3.1. Définition d'un algorithme

Un algorithme est l'ensemble de règles opératoires dont l'application permet de résoudre un problème énoncé au moyen d'un nombre fini d'opérations. Il peut être traduit grâce à un langage de programmation, en un programme exécutable par un ordinateur ou une calculatrice [4].

3.2. Définition d'un programme

Séquences d'instructions et de données enregistrées sur un support et susceptibles d'être traitées par un ordinateur [5].

3.3. Langage de programmation

Un langage de programmation est une notation conventionnelle destinée à formuler des algorithmes et produire des programmes informatiques qui les appliquent. D'une manière similaire à une langue naturelle, un langage de programmation est composé d'un alphabet, d'un vocabulaire, de règles de grammaire et de significations [6]. Le langage de programmation est l'intermédiaire entre l'humain et la machine, il permet d'écrire dans un langage proche de la machine mais intelligible par l'humain les opérations

que l'ordinateur doit effectuer. Ainsi, étant donné que le langage de programmation est destiné à l'ordinateur, il doit donc respecter une syntaxe stricte.

C'est un ensemble de vocabulaire (lettres, chiffres, symboles, caractère spéciaux,..) et un ensemble de règles (syntaxe ou grammaire) qui permet de réunir les éléments du vocabulaire pour former des phrases (ligne de programme) correctes.

3.4. Caractéristiques d'un algorithme

L'algorithme est un moyen pour le programmeur de présenter son approche du problème à d'autres personnes. En effet, un algorithme est l'énoncé dans un langage bien défini d'une suite d'opérations permettant de répondre au problème. Un algorithme doit donc être :

- **Lisible:** l'algorithme doit être compréhensible même par un non-informaticien.
- **De haut niveau:** l'algorithme doit pouvoir être traduit en n'importe quel langage de programmation, il ne doit donc pas faire appel à des notions techniques relatives à un programme particulier ou bien à un système d'exploitation donné
- **Précis:** chaque élément de l'algorithme ne doit pas porter à confusion, il est donc important de lever toute ambiguïté
- **Concis:** un algorithme ne doit pas dépasser une page. Si c'est le cas, il faut décomposer le problème en plusieurs sous-problèmes
- **Structuré:** un algorithme doit être composé de différentes parties facilement identifiables à savoir : une entête où apparaissent le nom de l'algorithme ainsi que les déclarations des objets manipulés puis le corps de l'algorithme qui commence par "Début" et se termine par "Fin. Entre ces deux mots clés se trouvent les actions ordonnées à exécuter par la machine.

3.5. Phase de résolution d'un problème informatique

La résolution des problèmes issue du monde réel consiste à appliquer un processus de traitement sur des données pour avoir des résultats, ce processus est souvent exprimé sous forme d'équations mathématiques, règles, formules, méthodes de traitements etc....

L'utilisation d'un ordinateur pour la résolution d'un problème consiste à réaliser un programme informatique qui reçoit les données en entrée et fournit les résultats en sortie.

L'élaboration d'un programme passe généralement par les étapes suivantes [3] :

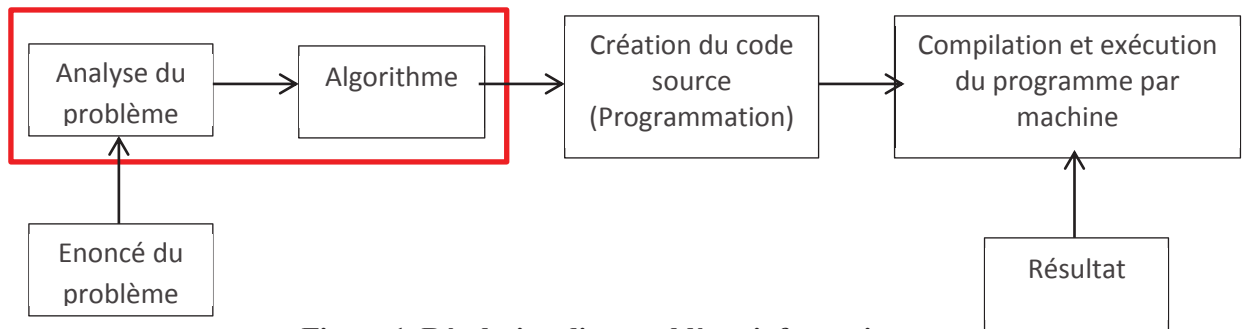


Figure 1. Résolution d'un problème informatique

3.5.1. Analyse du Pb

Selon la complexité du problème P à résoudre, cette étape peut être décomposée en trois parties :

1) Décomposer P en N sous-problèmes (P1, P2, ..., PN) de complexité inférieure. Si un ou plusieurs problèmes apparaissent encore trop complexes on les décompose à leurs tours. En fin, la solution du problème initial revient à résoudre tous les petits problèmes résultants.

2) Identifier les objets nécessaires à la formulation du modèle qui permet de définir le processus de résolution.

- Objets relatifs aux données : les entrées.
- Objets relatifs aux résultats : les sorties.
- Objets intermédiaires qui résultent de la phase de décomposition.

3) Définir les relations qui existent entre ces objets en termes de règles, de formule, d'équations mathématiques et de méthodes de traitements.

3.5.2. Algorithme

L'ensemble des objets obtenus dans la phase d'analyse et les instructions qui décrivent les étapes à suivre pour obtenir la solution est appelé *Algorithme*

3.5.3. Création et compilation du code source

Cette étape consiste à traduire les instructions de l'algorithme, développé antérieurement, dans un langage de programmation choisi par l'utilisateur.

Chaque langage de programmation contient un compilateur intégré. Le rôle de ce compilateur c'est de vérifier la conformité du code source avec la syntaxe du langage.

- Si le programme est syntaxiquement correct, le compilateur crée ce qu'on appelle programme objet, c'est un programme prêt à être exécuté.

- Si le programme contient des erreurs de syntaxe, le compilateur affiche la liste des erreurs à l'écran.

Le processeur exécute les instructions d'une manière séquentielle c'est-à-dire l'une après l'autre dans l'ordre de leurs apparition dans le programme.

Conclusion

L'informatique est une nouvelle technologie intellectuelle. Malgré la modernité de cette discipline, elle s'imposa fortement est devenue omniprésente dans tous les secteurs d'activités de notre vie quotidienne. Nous avons vu dans ce chapitre comment l'informatique à évoluer rapidement, et comment nous pouvons l'utiliser pour résoudre nos problème à travers l'algorithmique et la programmation par la suite. Dans le chapitre suivant, nous allons aborder les notions de base concernant un algorithme séquentiel simple. Ces notion constituent les ingrédients indispensables dans n'importe qu'elle algorithme.