

## Chapitre 1 : Introduction Générale à l'Informatique

Contenu du chapitre :

Brève historique de l'informatique

Introduction au langage algorithmique et le langage C

### I. Brève historique de l'Informatique

1. **Origine du mot Informatique** : Le mot Informatique fut créé par Philippe Dreyfus (*néologiste* de la langue française) en 1962, à partir des mots "information" et "automatique".

(Anglais : *Computer Science, Informatics*).

### 2. Informatique :

- L'informatique est un domaine d'activité scientifique, technique, et industriel concernant le traitement automatique de l'information numérique par l'exécution de programmes effectuant des tâches spécifiques.

(Wikipédia)

- L'informatique désigne l'automatisation du traitement de l'information par un système, concret (machine) ou abstrait. En général, l'informatique désigne l'ensemble des sciences et techniques en rapport avec le traitement de l'information.

(<https://www.techno-science.net/definition/162.html>)

3. **Bref historique sur l'informatique** : Dans ce qui suit, nous rapportons des événements et/ou inventions ayant contribué à l'avènement de l'informatique, directement ou indirectement :

**Origine du mot calcul** : Le mot calcul vient du latin *calculus* (« caillou »). Il est dit que les bergers comptabilisaient leurs moutons avec des cailloux dans un pot à l'entrée et à la sortie de la bergerie. (Wikipédia).

**Le boulier chinois (13<sup>ième</sup> Siècle)** : Le boulier a été inventé en Chine au XIII<sup>ème</sup> siècle, en s'inspirant des modes de calcul des *Babyloniens*, des *Indous*, et des *romains*. Les opérations + - / \* sont réalisables sur cet instrument.

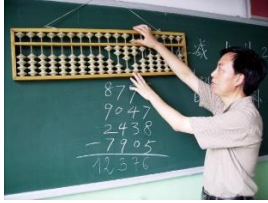


Illustration du boulier chinois pour les opérations arithmétiques (<https://iremi.univ-reunion.fr/spip.php?article292>)

### Le Système Décimal Positionnel basé sur les chiffres arabes :

**Al-Khawarizmi (780-800)** Introduisit ce système comme alternative au système numéral romain, ainsi que la science de l’algèbre (Al-Jabr). Ces deux outils et d’autres travaux de Al-Khawarizmi ont beaucoup contribué à l’avancement des mathématiques et des autres sciences utilisant les calculs.



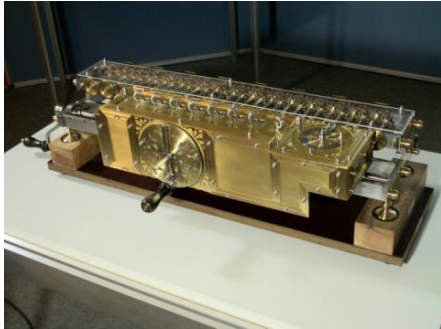
AlKhawarizmi (780 et 800) (<https://www.pourlascience.fr/>)

**La Pascaline 1642-1644** : Blaise Pascal (1623-1662) inventa une machine à calculer (« la Pascaline ») capable d'additionner et de soustraire des nombres de huit chiffres.



La Machine Pascaline de Pascal (Didier Muller 1.1)

**La machine de Leibnitz (1646-1716)** : Vers 1706, Leibnitz inventa une machine capable de réaliser les opérations de base :  $+ - * /$ .



Réplique de *machine de Leibnitz*, construite à l'université de Dresde entre 1998 et 200. (<https://journals.openedition.org/bibnum/551>)

**Origine du mot Algorithme : Lady Ada Lovelace (fille de Lord Byron)** : Le terme *algorithme* provient du nom du mathématicien arabo-persan du IX<sup>e</sup> siècle, Al Khwarizmi. Ce serait la mathématicienne *Lady Ada Lovelace fille de Lord Byron* qui utilisa la première le mot *algorithme* à partir d'*algorisme*. Le langage informatique Ada (1980) a été ainsi nommé en hommage à cette femme, considérée comme étant « le premier informaticien » de l'Histoire. Son portrait est aussi sur les hologrammes des produits Microsoft.



Lady Ada Lovelace (Blog TIC-Informatique - La Grande Histoire de l'Informatique)

**Historique des algorithmes** : L'utilisation des algorithmes est profonde dans l'Histoire. Ainsi, il existe des algorithmes, certains utilisés jusqu'à aujourd'hui, qui remontent aux temps très anciens : *Babyloniens* (1600 av. JC), *Egyptiens*, *Indiens*, etc. Le plus connu des algorithmes est l'algorithme d'Euclide (Grèce, 300 av. JC) pour calculer le pgcd de deux nombres entiers.

**La machine de Alan Turing (20<sup>ème</sup> Siècle)** : Le mathématicien Anglais Alan Turing proposa le premier modèle de calcul (dite, *Machine de Turing*, 1936). Cette machine est considérée comme étant aussi un modèle des ordinateurs, des algorithmes, des langages de programmation et des programmes. Elle est considérée alors comme le point de départ de l'informatique.

**Les premiers ordinateurs** : (Wikipédia)

**1946** : ENIAC (Presper Eckert et John William Mauchely).

**1948** : SSEM (IBM).

**1952** : IBM 701

**1956** : Ordinateurs de deuxième génération

**1963** : Ordinateurs de troisième génération

**1971+** : Ordinateurs de quatrième génération (Microprocesseur)

**Structure générale d'un ordinateur** : (Wikipedia)

Dans le modèle le plus connu de l'ordinateur, le modèle de Von Neumann, un ordinateur est constitué en général des éléments suivants :

- a) **Une unité centrale (Processeur)**, composée elle-même de 2 éléments :
- b) **L'unité de commande ou unité de contrôle (UC)** : Cette unité est chargée du «séquençage» des opérations de base en agissant sur un registre appelé **Compteur Ordinal (CO)**. Ceci assure alors la logique du programme (*séquence naturelle, saut en avant ou saut en arrière*, lors de l'exécution d'un programme).
- c) **Unité arithmétique et logique (UAL) ou Unité de traitement** : Son rôle est d'effectuer les opérations de base (*Affectation, Lecture, Ecriture, opérations arithmétiques et/ou logiques*). Cette unité est dotée de registres accumulateurs pour stocker les résultats intermédiaires des calculs arithmétiques et/ou logiques.
- d) **La mémoire** : la mémoire contient à la fois les données et le programme qui indiquera à l'unité de contrôle quels sont les calculs à faire sur ces données. La mémoire se divise entre

mémoire volatile ou vive ou **RAM** (*Random Acces Memory*) et mémoire permanente ou **ROM** (*Read Only Memory*). La RAM contiendra les programmes et données en cours de fonctionnement alors que la ROM contient les programmes et données de base de la machine.

- e) **Les unités d'entrées et les unités de sorties** : Ces éléments sont chargés de l'échange (communication) du processeur avec les **périphériques** : Disque, Flash, CD, Clavier, Souris, etc.

La Figure 1.1 illustre cette structure.

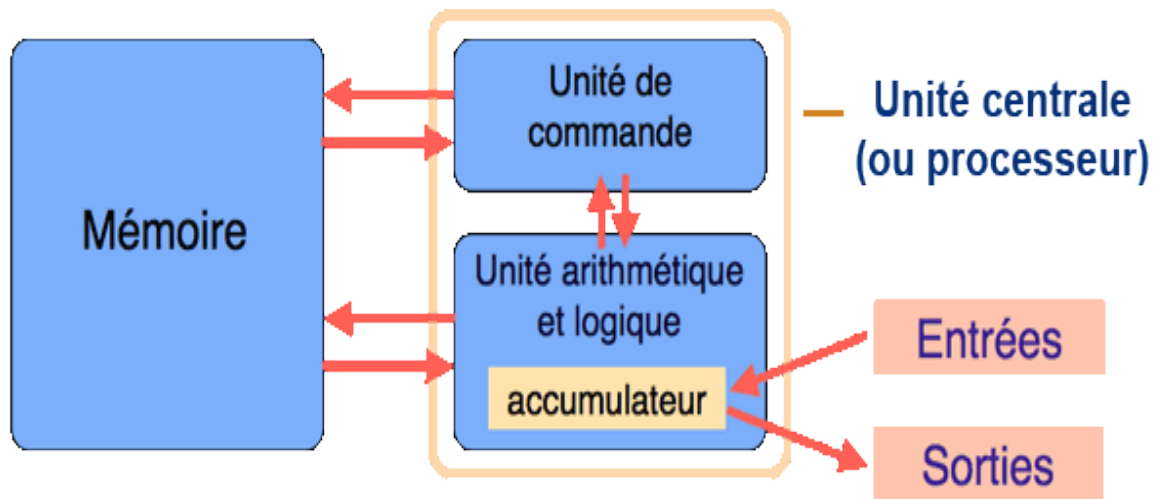


Figure 1.1. Le modèle originel de Von Neumann des ordinateurs. (<https://interstices.info/le-modele-darchitecture-de-von-neumann/>)

f) **Langages de programmation célèbres, historiques et actuels** : (Wikipédia) :

([https://fr.wikipedia.org/wiki/Histoire\\_des\\_langages\\_de\\_programmation](https://fr.wikipedia.org/wiki/Histoire_des_langages_de_programmation))

Période	Langages
1940-1950	Langages machines
1951-1960	COBOL, Lisp, FORTRAN
1970+	Pascal, C
1971-1980	SQL, Prolog
1981-1990	Ada, C++, Eiffel, Perl, Matlab, DBaseIII, CAML
1991-2000	C#, PHP, Java, JavaScript, ANSI Common Lisp, Python, R, Haskell, Ruby, ADA95, OCAML
2000+	Rust, Ceylon, ULM

#### 4. Introduction à l'algorithmique

##### 4.1 Définition de l'Informatique :

*Formellement, l'Informatique* est la science et technologie qui s'intéresse à la *résolution des problèmes* par l'outil de l'ordinateur. Pour cela, soit :

- a. Nous utilisons des *programmes* et *logiciels* existants sur ordinateur. Dans ce cas, nous disons que nous sommes des *utilisateurs* (des programmes et logiciels préexistants sur ordinateur). Ou :
- b. Nous résolvons les problèmes nous-même par l'outil de la *programmation*. Dans ce dernier cas :
  - nous utilisons une *méthodologie d'analyse du problème posé (P)* pour dégager une *solution (S)* (dite *algorithme*), puis,
  - nous traduisons l'algorithme *S* dans un *langage de programmation* (pour ce cours : Langage C). Cette étape nous donne un *programme* exécutable sur ordinateur

après sa *compilation* par le **compilateur** (dans ce cours : *Compilateur du langage C*).

- Nous exécutons ensuite le programme fourni par le compilateur sur ordinateur en lui fournissant les données (D).
- Nous obtenons alors les résultats R.

Nous donnons alors les précisions suivantes :

- La phase de détermination de la solution (algorithme) est dite : **Algorithmique**
- La phase de traduction de l'algorithme en un programme exécutable sur ordinateur est dite : **Programmation**. Le programme obtenu est dit **programme source**.
- Ces deux phases sont évidemment suivies par la phase de compilation du programme grâce à un **compilateur** (**Définition** : un **compilateur** est un programme qui transforme le code source d'un programme en un **code exécutable** sur ordinateur).
- Ces phases sont suivies par la *phase d'exécution, qui comprend aussi la **correction** des éventuelles **erreurs** existantes dans le programme (source)*.
- Ces phases ensemble constituent ce que l'on appelle : **Méthodologie de résolution des problèmes par l'outil informatique** (ordinateur + programmation).
- La Figure 1.2 illustre les phases de résolution d'un problème par ordinateur. Sur cette Figure, nous insistons sur deux éléments essentiels à la méthodologie : **Les données D** et les **résultats R**.

Nous avons ajouté aussi des commentaires à droite pour chaque phase. Ces phases seront expliquées dans ce qui suit.

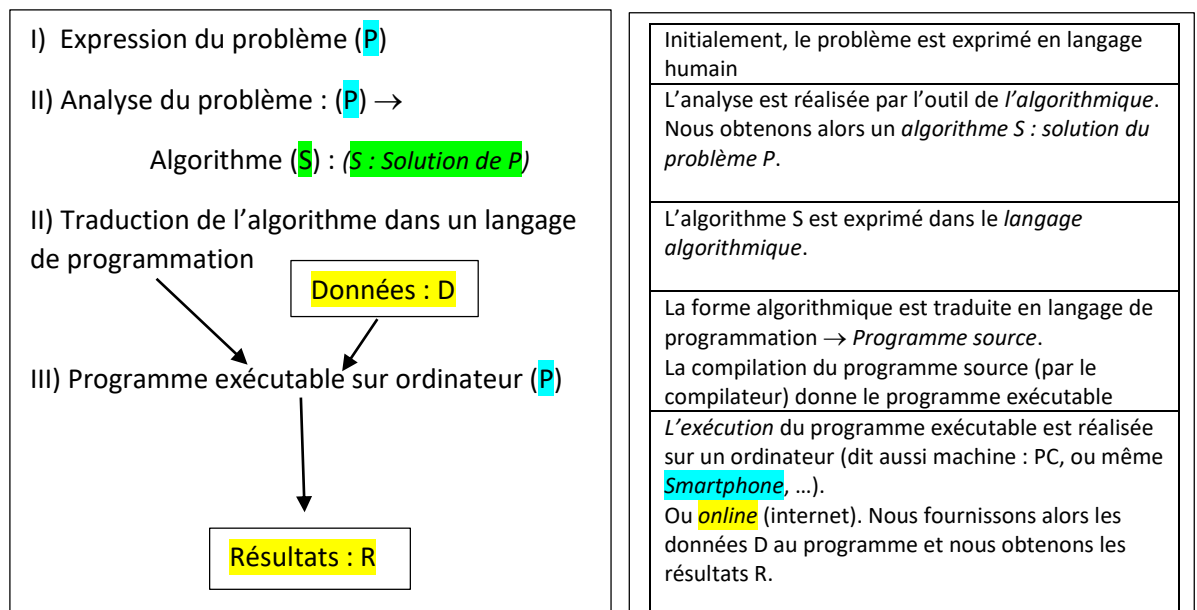


Figure 1.2 Résumé de la méthodologie de résolution des problèmes par ordinateur

#### 4.2 Notion d'algorithmique :

L'algorithmique est la branche de l'informatique qui s'intéresse à la *confection et à l'analyse des algorithmes*, solutions des problèmes posés. La confection consiste en la création (invention) des algorithmes, l'analyse consiste en leurs évaluations, pour déterminer les meilleurs d'entre eux, relativement à un problème donné.

##### a. Définition d'un algorithme :

Un algorithme est une suite d'actions (ou pas) organisées logiquement pour résoudre un problème.

##### b. Propriétés d'un algorithme :

Un algorithme doit être :

- **Claire et non ambigu.**

- **Universel (Général)** : Il doit tenir compte de tous les cas possibles (Il traite le cas général et les cas particuliers).

- **Fini** : Il doit toujours s'arrêter au bout d'un temps fini.



- **Structuré** : un algorithme doit être composé de différentes parties facilement identifiables.
- Il est **indépendant** des langages de programmation et des matériels informatiques (\*).
- **Déterministe** : Dans des conditions d'exécution similaires (avec des données identiques), il fournit toujours le même résultat (\*\*)

Nous exprimons alors les pas d'un algorithme selon les modes d'expression suivants :

- Expression d'un algorithme par narration السرد (i.e. : langage humain)
- Expression d'un algorithme par des *diagrammes* (*organigrammes*, *Flowchart* ou *Bloc Diagram* en anglais)
- Expression d'un algorithme par le *Langage Algorithmique*
- Expression d'un algorithme par un langage de programmation (dans ce cours, le langage C).

Tous ces modes, seront développés dans les cours suivants.

## 5. Brève introduction au langage algorithmique et au langage C

### 5.1 Le langage algorithmique :

Le langage algorithmique exprime de façon simple les programmes informatiques, mais de manière proche des langages de programmation. Ce langage sera traité en détail dans le chapitre 2.

Ici, nous présentons deux exemples simples d'algorithmes exprimés dans le langage algorithmique.

Ces deux exemples seront par la suite traduits en langage C.

**Exemple1** : Calcul de la somme, différence et produit de deux nombres entiers, en respectant la méthodologie de la programmation.

#### a) Définition du problème : C'est quoi le problème ?

Données  $x, y$  : entier { Ce qu'on donne à l'algorithme }  
 Résultats : som, diff, prod : entier { Ce que l'algorithme nous renvoie }

#### b) Analyse : {Comment aller des données vers les résultats : Solution}

Som  $\leftarrow x+y$   
 Diff  $\leftarrow x-y$   
 Prod  $\leftarrow x*y$

#### c) Forme algorithmique : {écriture de la solution dans le langage algorithmique}

**Algorithme A**

**Données  $x, y$  : entier**

**Résultats : som, diff, prod : entier**

**Début**

**Ecrire('Entrez x :');**

**Lire(x)**

**Ecrire('Entrez y :');**

**Lire(y)**

**som  $\leftarrow x+y$ ;**

**diff  $\leftarrow x-y$ ;**

**prod  $\leftarrow x*y$ ;**

**Ecrire(' Résultats : somme=', som, ' différence=', diff, ' produit=', prod) ;**

**Fin A**

#### d) Forme en langage C (point suivant)

**5.2 Langage C :**

Le langage C est un langage très utilisé par beaucoup de programmeurs. Il a été développé par l'informaticien Dennis Ritchie à partir d'un langage appelé B. En 1973, le langage C atteint un niveau de performance pour être utilisé pour l'écriture du système UNIX.

Le langage C (*impératif*), possède une version évoluée de type orienté objets : C++.

Le langage C sera présenté dans le chapitre 2.

Dans ce chapitre, nous présentons l'exemple cité dans la partie langage algorithmique, exprimé dans le langage C.

**Exemple 1 :**

```

#include<stdio.h>           // header (entête) pour lire (scanf) et écrire (printf)
main()
{
    int a,b, som, diff, prod; // déclaration des données

    printf("Entrez x= ");    // Lecture des données x et y
    scanf("%d", &a);
    printf("Entrez y= ");
    scanf("%d", &b);        // remarquez le symbole & (accès par adresse. (*))

    som = a + b;           // Réalisation des opérations de l'algorithme
    diff= a-b ;
    prod= a*b;

    printf("somme = %d\n", som); // Ecriture des résultats
    printf("difference=%d\n", diff); // remarquez : pas de symbole &
    printf("produit= %d",prod);    // ici, accès aux variables par valeur. (*)
}

```

**Références utilisées ou suggérées :**

- 1/ Algorithmique Structures de données, Florent Hivert Mél, Université Paris Sud.
- 2/ Notes de cours et recueil d'exercices corrigés Algorithmique et Structure de Données 1, Mohamed Amine NEMMICH, Département d'Informatique, USTO – Oran
- 3/ Algorithmique et Programmation en Langage C, Djelloul Benchiha, Editions Universitaires Européennes, 2020.
- 4/ Data Structures Using c, Second Edition, Reema Thareja, Department of Computer Science, Shyama Prasad Mukherjee College for Women, University of Delhi.

**IDE + Compilateur recommandés : Dev C++**

**C online sites:**

[https://www.onlinegdb.com/online\\_c\\_compiler](https://www.onlinegdb.com/online_c_compiler)

<https://onecompiler.com/c>

<https://ide.geeksforgeeks.org/online-c-compiler>

<https://www.mycompiler.io/fr/new/c>

<https://www.scaler.com/topics/c/online-c-compiler/>

<https://www.programiz.com/c-programming/online-compiler/>

<https://www.codechef.com/ide>