
Algorithmique

Chapitre 0- Une introduction à la programmation

Introduction

- **Programmation** = déterminer la **démarche** permettant d'obtenir, à l'aide d'un ordinateur, la **solution à un problème** donné.
 - ⇒ *On parle d'analyse* ou encore de *recherche d'un algorithme*
- Utiliser directement un langage implique que les aspects fondamentaux de l'analyse sont masqués par :
 - Les contraintes spécifiques au langage (=, :=, ...)
 - Ses particularités techniques (objet, ...)
 - Sa complexité (Allocation mémoire, ...)

Introduction

- Objectif de ce cours :
 - ⇒ **Fournir les éléments de bases** intervenant en programmation **quel que soit le langage employé**
 - ⇒ Variable
 - ⇒ Type
 - ⇒ Instructions d'affectation
 - ⇒ Instruction de lecture ou d'écriture
 - ⇒ Structures

Plan

- 1 - A la découverte de l'ordinateur
- 2 - La notion de variable
- 3 - L'instruction d'affectation
- 4 - Communiquer avec votre programme
- 5 – Une première structure conditionnelle
- 6 – Une première structure répétitive : répéter
... tant que
- 7 - Compter

Plan

- 8 - La boucle avec compteur
- 9 - Accumuler
- 10 - L'itération
- 11 – Procédures et fonctions
- 12 - les tableaux à une dimension
- 13 - Les boucles imbriquées
- 14 - Les tableaux à deux dimensions

Algorithmique

Chapitre I - Ordinateur et Programmation

Introduction

- Présence importante de l'ordinateur dans notre environnement :
 - Automobile (ordinateurs de bords)
 - Robots (automatisme)
 - Jeux vidéo
 - Routage (GSM)
 - ...
- Rôle fondamental de la programmation qui se cache derrière ces tâches

Introduction

- Objectif de ce chapitre :
 - Comprendre le rôle de l'ordinateur dans la vie actuelle
 - Comprendre succinctement comment fonctionne un ordinateur

1 - Le rôle de l'ordinateur

1.1 - La multiplicité des applications

- Des applications nombreuses :
 - Jeux vidéo, Synthèse d'images
 - Consultation de bases de données
 - Simulations physiques
 - Suivi médical des patients dans un hôpital
 - Traitement de texte
 - ...

1 - Le rôle de l'ordinateur

1.2 - Le programme : source de diversité

- Comment effectuer des tâches aussi variées?
 - ⇔ Mettre en mémoire un programme puis l'exécuter
- L'ordinateur possède un nombre limité d'opérations élémentaires (**instructions**) qu'il est capable d'exécuter très rapidement et sans erreurs

1 - Le rôle de l'ordinateur

1.2 - ...

- **Programme** = ensemble d'instruction qui spécifient :
 - Les opérations élémentaires à exécuter
 - La façon dont elles s'enchaînent
- Possibilité de répéter des parties du programme selon des conditions données.

1 - Le rôle de l'ordinateur

1.3 - ...

- Exemple 1 : Programme de calcul d'une moyenne de notes
 - Pour s'exécuter, ce programme nécessite qu'on lui fournisse des notes dont on cherche la moyenne
 - ⇒ **Informations données** ou **données**
 - En retour, le programme fournit la moyenne cherchée
 - ⇒ **Résultats**

1 - Le rôle de l'ordinateur

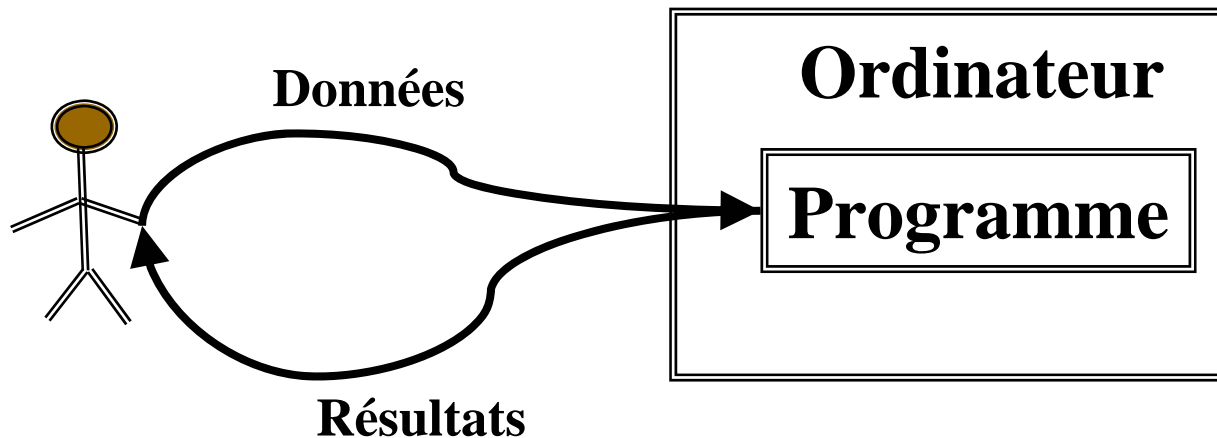
1.3 - Les données du programme, les résultats

- Exemple 2 : Un programme de gestion de paye
 - Données :
 - Nom des employés,
 - Situation de famille
 - Numéros S/S
 - Nombre d'heures supplémentaires
 - ...
 - Résultats :
 - Un bulletin de paye individuel calculé en fonction des données transmises
 - Stockage des informations pour utilisation ultérieure

1 - Le rôle de l'ordinateur

1.4 - Communication ou archivage ?

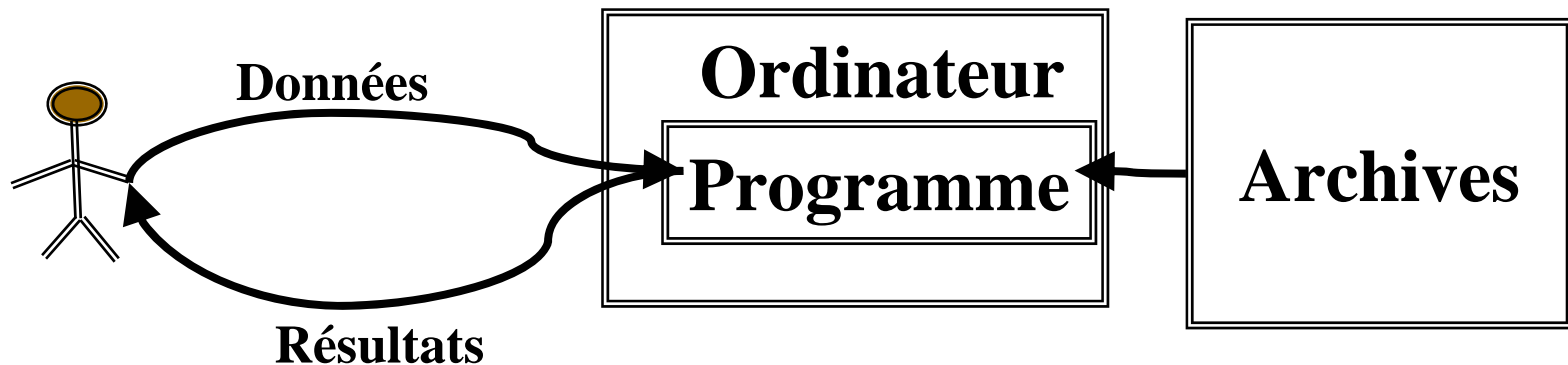
- D'où proviennent les données ?
- Que deviennent les résultats ?
 - Cas 1 : Fournir les données manuellement



1 - Le rôle de l'ordinateur

1.4 - ...

- Cas 2 : Faire appel à des informations déjà archivées



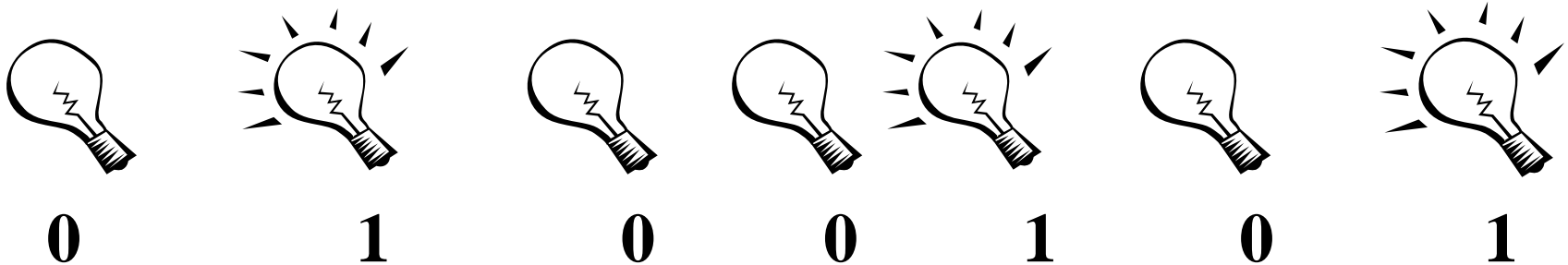
Rmq : :

- Possibilité de stocker le résultat en archive
- Peuvent aussi devenir des données pour un autre programme

2 - Pour donner une forme à l'information: La notion de codage

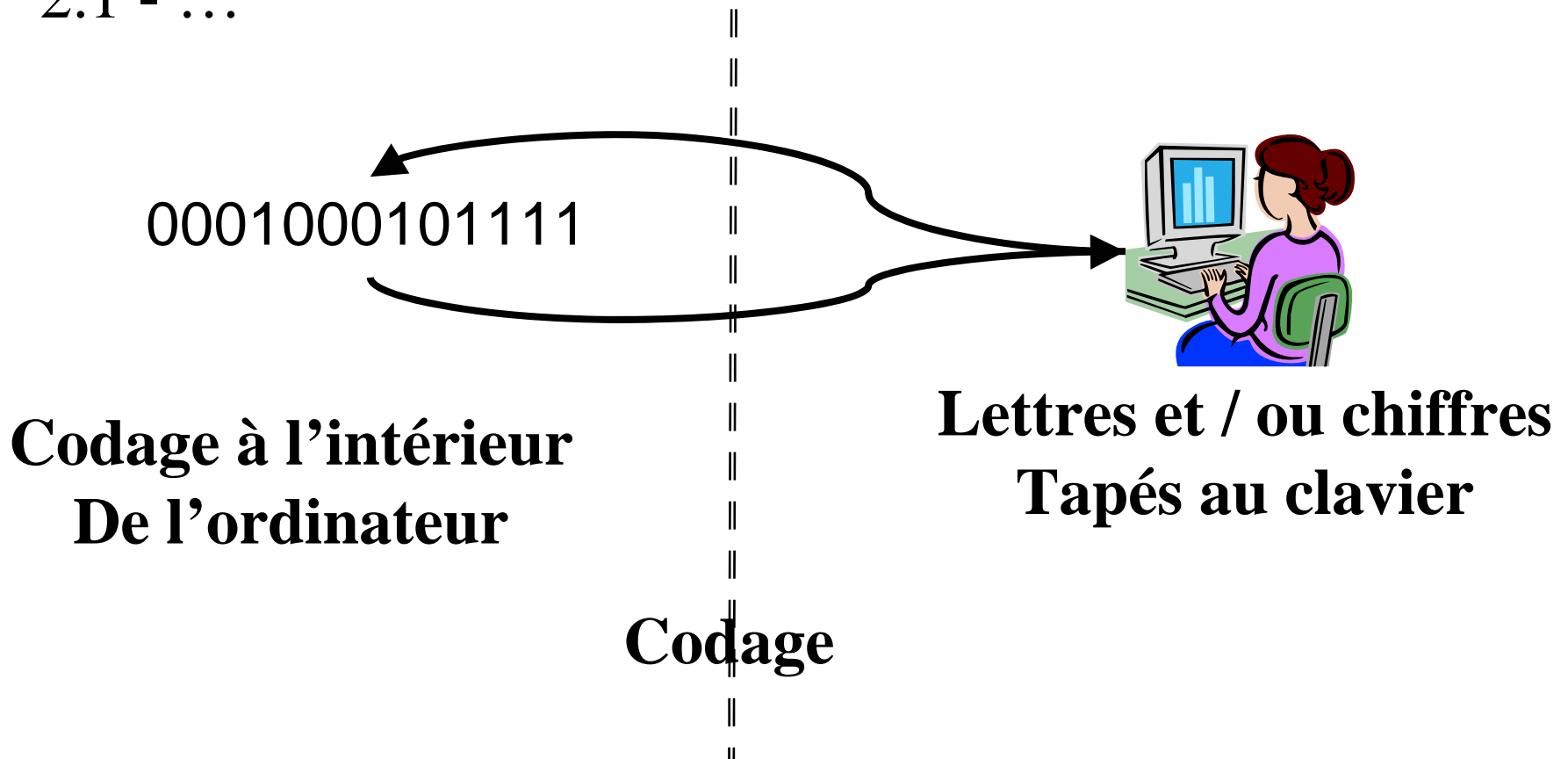
2.1 - L'ordinateur code l'information

- L'ordinateur ne peut traiter que des informations sous forme binaire
- **Codage binaire** : codage de l'information à l'aide d'une suite de 0 et de 1



2 - Pour donner une forme à l'information: La notion de codage

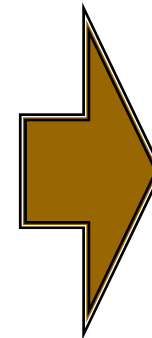
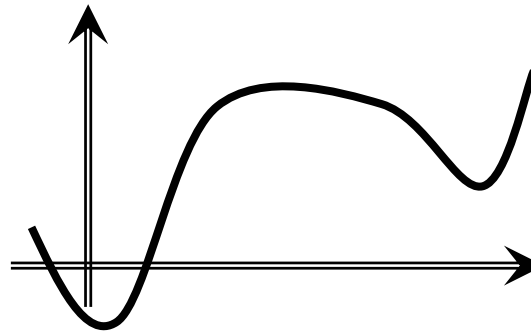
2.1 - ...



2 - Pour donner une forme à l'information: La notion de codage

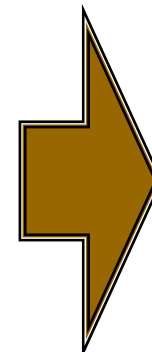
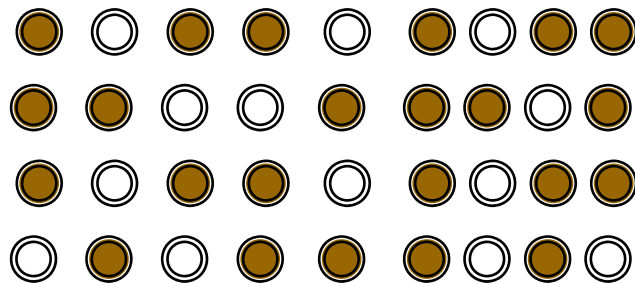
2.1 - ...

1048
25.32
Solde au 15/12/83



**Forme
compréhensible
par l'homme**

Codage



**Forme
utilisable par
l'ordinateur**

2 - Pour donner une forme à l'information: La notion de codage

2.2 - Il n'est pas le seul ...

- Exemple : 13, treize, vous avez dit XIII
 - La même quantité apparaît sous trois « formes différentes »
 - 13 : S'exprime avec deux symboles, chacun choisi parmi les chiffres 0 à 9
 - Deux **positions** d'un code à 10 **moments** (0, 1, 2, ...9)
 - treize : s'exprime avec six symboles (lettres), choisis parmi les 26 lettres de l'alphabet
 - Six positions d'un code à 26 **moments**
 - XIII :
 - 4 **positions** d'un code à 7 **moments** (chiffres romains I, V, X, L, C, M, D)

2 - Pour donner une forme à l'information: La notion de codage

2.2 - ...

- Code binaire :
 - ⇒ Code à **2 moments**
 - ⇒ Chaque position porte le nom de « **bit** »

2 - Pour donner une forme à l'information: La notion de codage

2.3 - mais ...

- ❑ Le codage binaire est moins « naturel », non seulement pour l'homme mais également pour l'ordinateur
- ❑ Exemple : que représente 00101001 ?
 - ⇒ Ecriture en binaire du nombre 41
 - ⇒ Ces 8 bits peuvent très bien représenter une lettre ou un code instruction de programme, ...

2 - Pour donner une forme à l'information: La notion de codage

2.3 - ...

- En conclusion, il est nécessaire de savoir à quoi correspond la suite de 0 et de 1 vue précédemment pour savoir ce qu'elle signifie
⇒ Notion de **type**

3 - Comment fonctionne l'ordinateur

3.1 - A chacun son rôle

■ Un ordinateur :

- **Traite** l'information grâce à un programme qu'il **mémore**
- **Communique** et **archive** des informations
- Est composé de trois constituants :
 - 1) La mémoire centrale
 - 2) L'unité centrale
 - 3) Les périphériques

3 - Comment fonctionne l'ordinateur

3.1 - A chacun son rôle

1) La mémoire centrale :

- Permet de mémoriser les programmes le temps nécessaire à leur exécution
- Stockages d'informations temporaires manipulées par les programmes
 - données après leur introduction
 - Résultats avant leur communication
 - Informations intermédiaires
 - ...

3 - Comment fonctionne l'ordinateur

3.1 - A chacun son rôle

2) L'unité centrale

- Partie active de l'ordinateur
- Chargée de prélever une à une chaque instruction du programme situé en mémoire centrale et de l'exécuter
- Trois types d'instructions :
 - Celles qui agissent sur des informations situées en mémoire centrale
 - Celles qui assurent la communication ou l'archivage de l'information
 - Celles qui agissent sur le déroulement du programme

3 - Comment fonctionne l'ordinateur

3.1 - ...

3) Les périphériques

- ❑ Désignent tous les appareils susceptibles d'échanger des informations avec la mémoire centrale
- ❑ Sont de deux sortes :
 - Ceux qui assurent la communication entre l'homme et l'ordinateur (clavier, écrans, imprimantes, souris, tablette graphique...)
 - Ceux qui assurent l'archivage d'information (disques, bandes, disquettes, clés USB, DVD, ...)

3 - Comment fonctionne l'ordinateur

3.2 - La mémoire centrale

- ❑ Elle est composée par des éléments qui peuvent prendre deux états distincts : **les bits**
- ❑ Un **Octet** = groupe de 8 bits
- ❑ Le **Mot** = terme plus général qui correspond à un ensemble de bits (dépend de l'ordinateur) : 8, 16, 32, ...
- Nous considérerons dans ce cours que la mémoire centrale est formée de mots sans en préciser la taille

3 - Comment fonctionne l'ordinateur

3.2 - ...

- Chaque mot de la mémoire est repéré par un numéro qu'on appelle son **Adresse**
- Un dispositif, associé à cette mémoire permet :
 - Soit d'aller chercher en mémoire un mot d'adresse donnée (aller recopier le contenu)
 - Soit d'aller ranger une information donnée dans un mot d'adresse donné (on remplace alors l'ancienne information figurant à cette adresse par la nouvelle).

3 - Comment fonctionne l'ordinateur

3.3 - L'unité centrale

- Sait exécuter très rapidement un certain nombre d'opérations très simples, telles que :
 - Opérations arithmétiques : Addition, soustraction, multiplication ou division de nombres codés dans des mots en mémoire centrale
 - Opérations logiques : Comparaison de valeurs contenues dans deux mots
 - Communication avec un périphérique d'une information élémentaire

3 - Comment fonctionne l'ordinateur

3.3 - ...

- Chaque instruction du programme doit préciser :
 - La nature du code à réaliser
 - Numéro (code en binaire) appelé **code opération**
 - Les adresses (ou l'adresse) des informations sur lesquelles doit porter l'information

3 - Comment fonctionne l'ordinateur

3.4 - Les périphériques

- Servent à échanger de l'information avec la mémoire centrale
- Deux catégories :
 - Communication
 - Archivage

3 - Comment fonctionne l'ordinateur

3.4 - ...

a) Les périphériques de communication

- Clavier
- Ecran
- Souris
- Imprimantes

b) Les périphériques d'archivages

- Mémoire centrale
 - a un coût élevé
 - est généralement « volatile »
- Les périphériques d'archivages permettent :
 - Des mémoires permanentes
 - Des coûts beaucoup plus faible (mais accès plus lent) -> de l'ordre de 1 à 1000.

3 - Comment fonctionne l'ordinateur

3.4 – b) ...

- Deux types de périphériques d'archivages
 - Le disque (ou disquette)
 - permet l'accès quasi-direct à une information
 - La bande (ou la cassette) magnétique
 - n'autorise qu'un accès séquentiel
 - Coût beaucoup moins élevé

4 - Comment parler à l'ordinateur

4.1 - Dans son langage ou dans le notre ?

- L'ordinateur ne sait exécuter qu'un nombre limité d'opérations élémentaires dictées par des instructions de programmes, codées en binaire
 - ⇒ **L'ordinateur ne peut comprendre que le langage machine**

4 - Comment parler à l'ordinateur

4.1 - ...

- Il est possible d'utiliser des « outils » qui permettront de traduire un langage simple vers le langage machine
- Il existe différents types de langage :
 - Langage machine, ou langage symbolique
 - Les langages dits « évolués »

4 - Comment parler à l'ordinateur

4.2 - En langage symbolique (langage orienté machine)

- **0101010011011010** pourrait signifier (en langage machine) additionner (**0101**) les valeurs comprises dans les mots d'adresses **010011** et **011010**
⇔ Add A, B
- Il faut disposer d'un programme capable de convertir :
 - Add ⇔ 0101
 - A ⇔ 010011
 - B ⇔ 011010

4 - Comment parler à l'ordinateur

4.3 - En langage évolué

- Intéressant de définir des langages :
 - Utilisables sur n'importe quel ordinateur
 - Plus tourné vers le problème à résoudre que vers les caractéristiques particulières à l'architecture de la machine
 - ⇒ Apparition de langages dits « évolués »
- Exemples de langages
 - Fortran (Domaine scientifique)
 - Cobol (Gestion)
 - Pascal, ADA, C, C#, C++, Visual Basic, JAVA...

4 - Comment parler à l'ordinateur

4.4 - De toutes façons, il faut traduire...

- ❑ Quel que soit le langage évolué, il faut, par un programme le traduire en langage machine
- ❑ Existence de deux techniques différentes :
 - **Compilation** : consiste à traduire dans un premier temps l'ensemble du programme (écrit en langage évolué) en langage machine. Dans un deuxième temps, ce programme est exécuté
 - **Interprétation** : consiste à traduire chaque instruction du programme avant de l'exécuter

5- Qu'est-ce que la programmation?

- ❑ Plusieurs façon de dire la même chose

La plupart des langages utilisent les mêmes concepts fondamentaux :

- ❑ Imposés par la nature même de l'ordinateur (variables, ...)
 - ❑ Liés aux structures qui se révèlent utiles dans un programme (choix, répétition, ...)
- ❑ Ne pas confondre : connaître un langage et savoir programmer

5- Qu'est-ce que la programmation?

Analyse du problème à résoudre et **transcription**
dans un langage de programmation

Recherche d'un algorithme

Traduction dans un langage de programmation

5- Qu'est-ce que la programmation?

- Objectifs de ce cours d'algorithmique
 - Mettre en place un langage simple permettant :
 - D'exprimer les notions fondamentales des langages de programmation
 - De ne pas avoir à se soucier de certaines contraintes techniques propre à un langage de programmation (« ; », ...)
 - Utiliser ce langage algorithmique pour résoudre des problèmes