

République Algérienne Démocratique et Populaire
Université Constantine 1 des frères mentouri
Département de génie de transport

INTRODUCTION A L'ALGORITHMIQUE

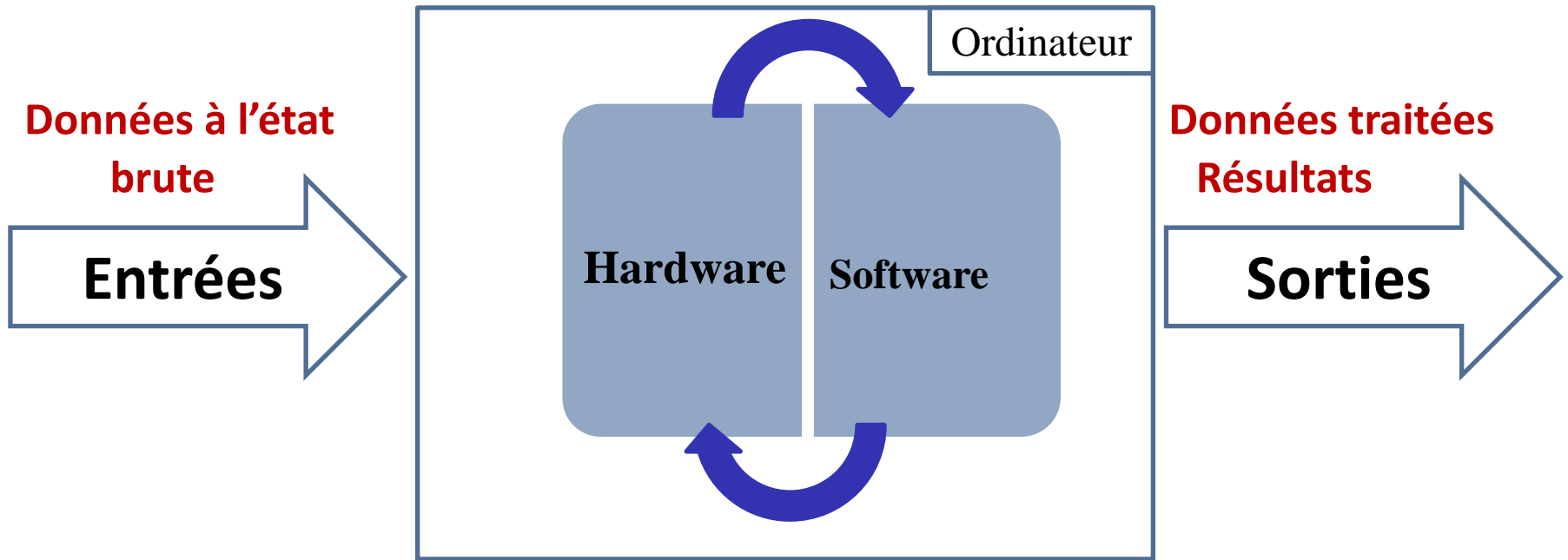
Licence 1 TL

Dr. S. BOUZOUBIA

Rappel

Informatique?

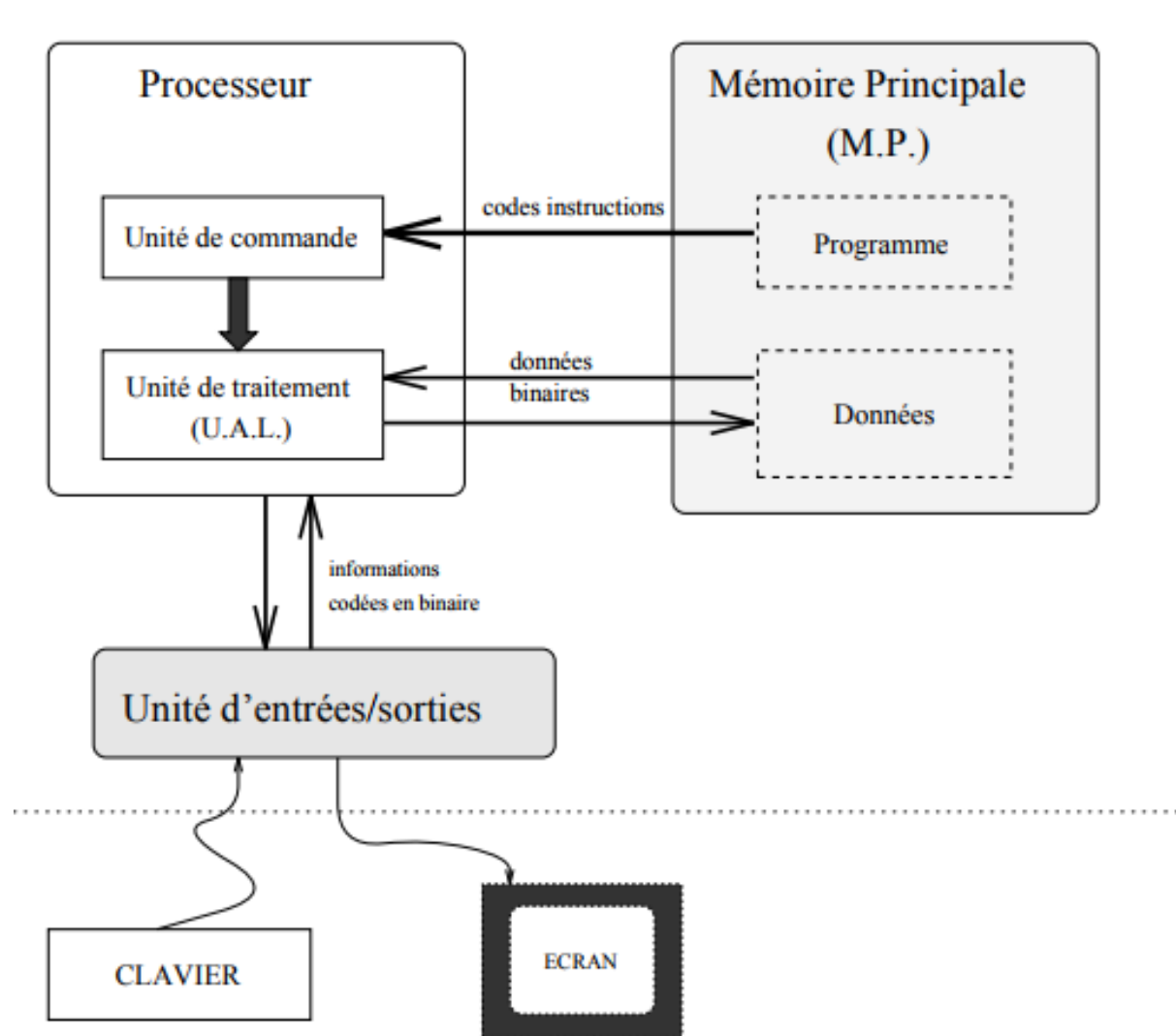
- Informatique = **Information** + Autom**atique**
C'est le traitement de l'information par la machine.



Traitement de l'information par ordinateur (programme= actions = instructions)

Rappel

Structure de la machine



Rappel

Représentation de l'information

- Octet = **Unité de la mesure de la mémoire.** = Byte
- 1 octet = 8 bit tel que Bit = soit **0** soit **1** (**B**inary **D**igit).
- La mémoire $\longrightarrow 2^n$ tel que **n** est supérieur ou égal à 0.

Les unités

1 Kilo octet = 10^3 Octet = 2^{10} Octet.

1 Mégaoctet = 2^{20} Octet.

1 Gigaoctet = 2^{30} Octet.

1 Terraoctet = 2^{40} Octet.

1 Pétaoctet = 2^{50} Octet.

1 Exa octet = 2^{60} Octet.

1 Zetta octet = 2^{70} Octet.

Introduction

QU'EST-CE QUE L'ALGORUTHMIQUE ?

- L'algorithmique est un terme d'origine arabe, « **El Khawarzmi** ».
- Avez-vous déjà ouvert un livre de recettes de cuisine ?
- Avez vous déjà déchiffré un mode d'emploi pour faire fonctionner quelle que chose ?
- Si oui, sans le savoir, vous avez déjà exécuté des algorithmes.

Introduction

QU'EST-CE QUE L'ALGORUTHMIQUE ?

- Avez-vous déjà indiqué un chemin à un touriste égaré ?
 - Avez vous fait chercher un objet à quelqu'un par téléphone ? Ecrit des étapes stipulant comment procéder à la résolution d'une équation ?
 - Si oui, vous avez déjà fabriqué – et fait exécuter – des algorithmes.
- l'algorithmique n'est pas un savoir réservé à quelques personnes, mais une aptitude partagée par la totalité de l'humanité.

Introduction

QU'EST-CE QUE L'ALGORITHMIQUE ?

- Un algorithme, c'est une suite d'instructions, qui une fois exécutée correctement, conduit à un résultat donné
- Si l'algorithme est juste, le résultat est le résultat voulu, et le touriste se retrouve là où il voulait aller dans notre exemple. Si l'algorithme est faux, le résultat est, disons, aléatoire ou erroné.
- Définition incomplète de l'algorithme. Que doit-on ajouter ?

Introduction

QU'EST-CE QUE L'ALGORUTHMIQUE ?

- Pour fonctionner, un algorithme doit donc contenir uniquement des instructions **compréhensibles par celui qui devra l'exécuter**.

C'est d'ailleurs l'un des points délicats pour les rédacteurs de modes d'emploi : les références culturelles, ou lexicales, des utilisateurs, étant variables, un même mode d'emploi peut être très clair pour certains et parfaitement abscons pour d'autres.

En informatique, il n'y a pas ce problème : les choses auxquelles on doit donner des instructions sont **les ordinateurs**, et ceux-ci ont le même langage et la même compréhension .

Introduction

- les ordinateurs, quels qu'ils soient, ne sont fondamentalement capables de comprendre que quatre catégories d'ordres (en programmation, on n'emploiera pas le terme d'ordre, mais plutôt celui d'instructions). Ces quatre familles d'instructions sont :
 - l'affectation de variables
 - la lecture / écriture
 - les tests
 - les boucles
- Un algorithme informatique se ramène donc toujours au bout du compte à la combinaison de ces quatre petites briques de base.

- Pourquoi apprendre l'algorithmique pour apprendre à programmer ?
l'algorithmique exprime les instructions résolvant un problème donné **indépendamment** des particularités de tel ou tel langage.

Algorithme = plan de résolution du problème.

- Apprendre l'algorithmique, c'est apprendre à manier la structure logique d'un programme informatique.

Apprendre l'algorithmique de manière séparée, c'est donc sérier les difficultés pour mieux les vaincre

AVEC QUELLES CONVENTIONS ECRIT-ON UN ALGORITHME ?

- plusieurs types de notations ont représenté des algorithmes.
 1. Il y a eu une représentation graphique, avec des carrés, des losanges, etc. qu'on appelait des **organigrammes**. Aujourd'hui, cette représentation est quasiment abandonnée.
 2. On utilise généralement une série de conventions appelée « **pseudo-code** », qui ressemble à un langage de programmation authentique dont on aurait évacué la plupart des problèmes de syntaxe. Ces caractéristiques:
 - Purement conventionnel
 - Aucune machine n'est censée le reconnaître
 - Susceptible de varier légèrement d'un livre à un autre

AVEC QUELLES CONVENTIONS ECRIT-ON UN ALGORITHME ?

- Le squelette d'un algorithme

Algorithme : nom

déclaration

Constante

Variable

Début

Fin

En tête

Zone déclarative

Corps de l'algorithme

AVEC QUELLES CONVENTIONS ECRIT-ON UN ALGORITHME ?

- Le squelette d'un algorithme

Algorithme : nom

} En tête

déclaration

Constante

Variable

} Zone déclarative

Début

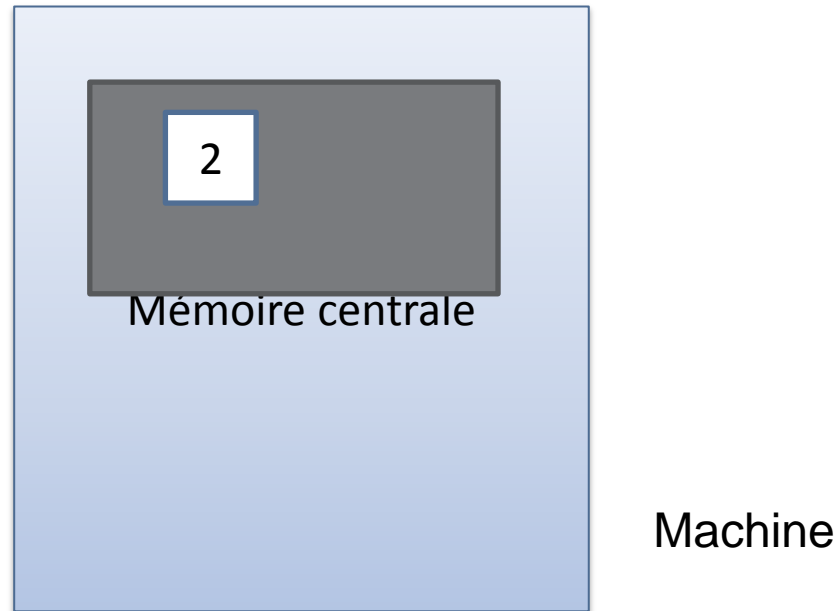
} Corps de l'algorithme

Fin

Les constantes

Exemple $d = 2$;

La machine va réserver un emplacement physique en mémoire, la machine lui donne le nom d et met l'information 2 dans cet emplacement.



Les variables

A QUOI SERVENT LES VARIABLES ?

Dans un programme informatique, on va avoir en permanence besoin de stocker provisoirement des valeurs. Il peut s'agir de données issues du disque dur, fournies par l'utilisateur. Il peut aussi s'agir de résultats obtenus par le programme, intermédiaires ou définitifs.

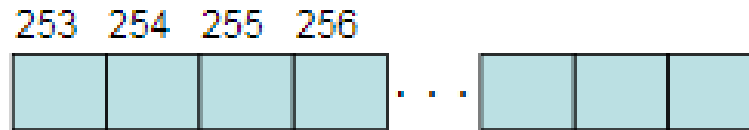
Ces données peuvent être de plusieurs types : elles peuvent être des nombres, du texte, etc.

dès que l'on a besoin de stocker une information au cours d'un programme, on utilise une variable.

Les variables

est un emplacement en mémoire principale destiné à recevoir une donnée. Cette zone reçoit une valeur qui peut ensuite être réutilisée.

La mémoire de votre ordinateur est comparable à des cases alignées une à une. Ces emplacements sont numérotés séquentiellement; il s'agit d'adresse en mémoire.



Déclaration des variables

- Pour identifier une variable, on utilise un identificateur pour désigner le nom de cette variable.
- L'identificateur doit indiquer le rôle joué par cette variable; il faut éviter d'utiliser des noms qui n'évoquent rien. Exemple: Rayon_du_cercle au lieu de x.
- **But:** cela vous épargnera de devoir connaître l'adresse réelle en mémoire de celle-ci.
- Un identificateur est composé de lettres (minuscule + majuscule) et de chiffres. Le premier caractère doit être une lettre. Le symbole « _ » est considéré comme une lettre.

Exemple: Variable_entiere, entier1, mot_en_francais sont valides mais 1er_entier, nom.2, nom de variable, deuxième_entier et a-b ne le sont pas

Déclaration des variables

- Les variables utiles dans un programme doivent être déclarées.
- Vous devez spécifier le nom de la variable (identificateur) deux points son type ensuite point vėrgule.

Syntaxe :

```
nom_d_une_variable : type_de_la_variable ;
```

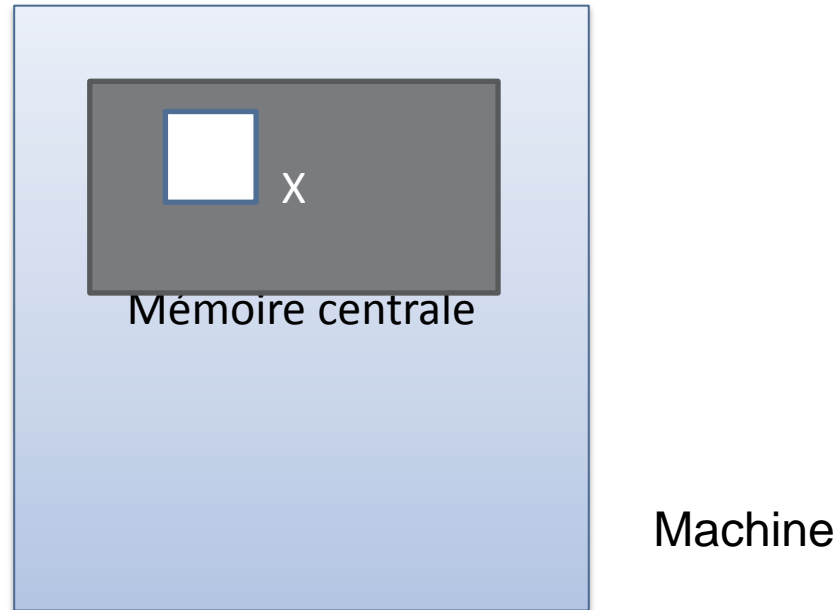
Exemple :

```
entier1: entier; // entier1 est une variable entière.
```

Comment va se comporter la machine ?

- La machine va allouer l'emplacement physique
- Elle va lui donner le nom de la variable.

exemple x: entier;



Types numériques classiques des variables

- Si l'on réserve un octet pour coder un nombre, alors on ne pourra coder que $2^8 = 256$ valeurs différentes. Cela peut signifier par exemple les nombres entiers de 1 à 256, ou de 0 à 255, ou de -127 à $+128$... Si l'on réserve deux octets, on a droit à 65 536 valeurs ; avec trois octets, 16 777 216, etc.
- le type de codage (autrement dit, le type de variable) choisi pour un nombre va déterminer :
 1. Les valeurs maximales et minimales des nombres pouvant être stockés dans la variable.
 2. La précision de ces nombres (dans le cas de nombres décimaux).

Types numériques classiques des variables

Type Numérique	Plage
Byte (octet)	0 à 255
Entier simple	-32 768 à 32 767
Entier long	-2 147 483 648 à 2 147 483 647
Réel simple	-3,40x10 ³⁸ à -1,40x10 ⁴⁵ pour les valeurs négatives 1,40x10 ⁻⁴⁵ à 3,40x10 ³⁸ pour les valeurs positives
Réel double	1,79x10 ³⁰⁸ à -4,94x10 ⁻³²⁴ pour les valeurs négatives 4,94x10 ⁻³²⁴ à 1,79x10 ³⁰⁸ pour les valeurs positives

- Pourquoi ne pas déclarer toutes les variables numériques en réel double ?

Il ne s'agit pas que de résoudre le problème mais d'optimiser le programme

Type alphanumérique des variables

- **Type alphanumérique** (également appelé type **caractère**, type **chaîne de caractère** ou en anglais, le type **string**).
- Dans une variable de ce type, on stocke des caractères, qu'il s'agit de lettres, de signes de ponctuation, d'espaces, ou même de chiffres. Le nombre maximal de caractères pouvant être stockés dans une seule variable string dépend du langage utilisé.
- Un groupe de caractères qu'il soit ou non stocké dans une variable est appelé chaîne de caractères.
- En pseudo-code, une chaîne de caractères est toujours notée entre guillemets.

Type Booléen des variables

- **Le type booléen** : on y stocke uniquement les valeurs logiques VRAI et FAUX.
- On peut représenter ces notions abstraites de VRAI et de FAUX par tout ce qu'on veut : de l'anglais (TRUE et FALSE) ou des nombres (0 et 1).
- Ce qui compte, c'est de comprendre que le type booléen est très économique en termes de place mémoire occupée, puisque pour stocker une telle information binaire, un seul bit suffit.

AVEC QUELLES CONVENTIONS ECRIT-ON UN ALGORITHME ?

- Le squelette d'un algorithme

Algorithme : nom

déclaration

Constante

Variable

Début

Fin

En tête


Zone déclarative


Corps de l'algorithme

Instructions: Affectation

L'affectation permet d'affecter une valeur à une variable. Elle est symbolisée en algorithmique par " \leftarrow ". Elle est spécifique aux variables seulement. Syntaxe: $x \leftarrow \text{expression}$.

- expression peut être une variable
- expression peut être une constante
- expression peut être une expression arithmétique.
- expression peut être une expression logique.

 On appelle l'affectation $x \leftarrow x+1$ incrémentement. Et l'affectation $x \leftarrow x-1$ décrémentement.

 Une instruction d'affectation ne modifie que ce qui est situé à gauche de la flèche.