

# Technologie Informatique

TIC – technologies de l'information et de la communication

XX siècle – l'informatique – conception, l'exploitation, évolution des ordinateurs

XXI – association à la technologie - IT

## IT – *information technology*

regroupent les techniques (les outils et les méthodes) utilisées dans le traitement et la transmission des informations, principalement de

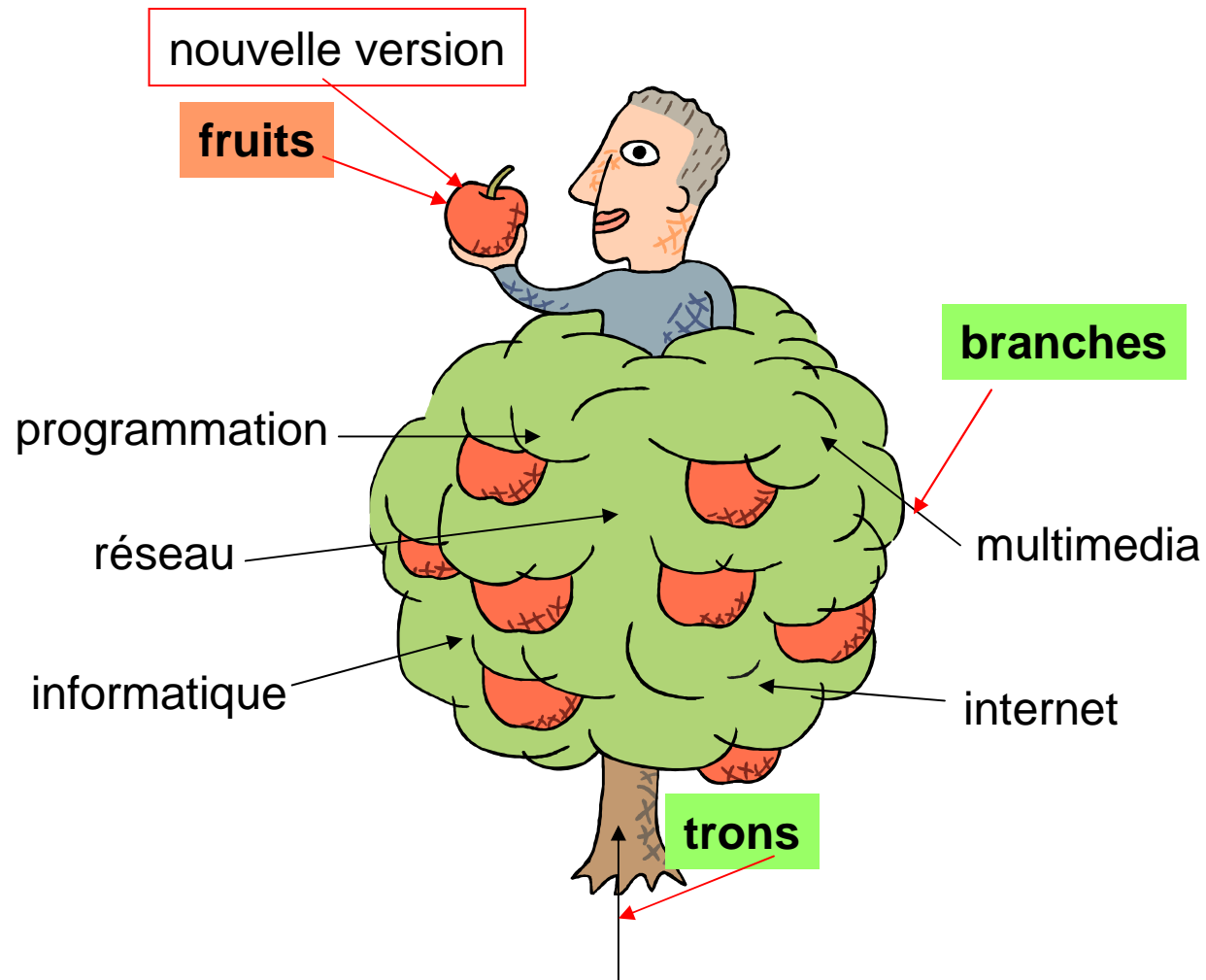
- l'informatique,
- l'internet et
- télécommunications.

Association  
l'informatique à la  
technologie de  
l'information



- monde digitale
- l'information
- la multimédialité
- les inforoutes
- la réalité virtuelle

# technologie de l'information



**Mathématique, communication, théorie de l'information....**

# TIC – technologies de l'information et de la communication

regroupent un ensemble de ressources nécessaires pour manipuler de l'information et particulièrement les ordinateurs, programmes et réseaux nécessaires pour la convertir, la stocker, la gérer, la transmettre et la retrouver.

On peut regrouper les TIC par secteurs suivants :

- L'équipement informatique
- La microélectronique
- Les télécommunications
- Le multimédia
- Le commerce électronique
- ....





- Naissance de l'ordinateur : 1945  
J.von Neumann

EDVAC (Electronic Discrete Variable Automatic Computer)

*enregistrement le programme en mémoire*  
*Instructions et données en mémoire*

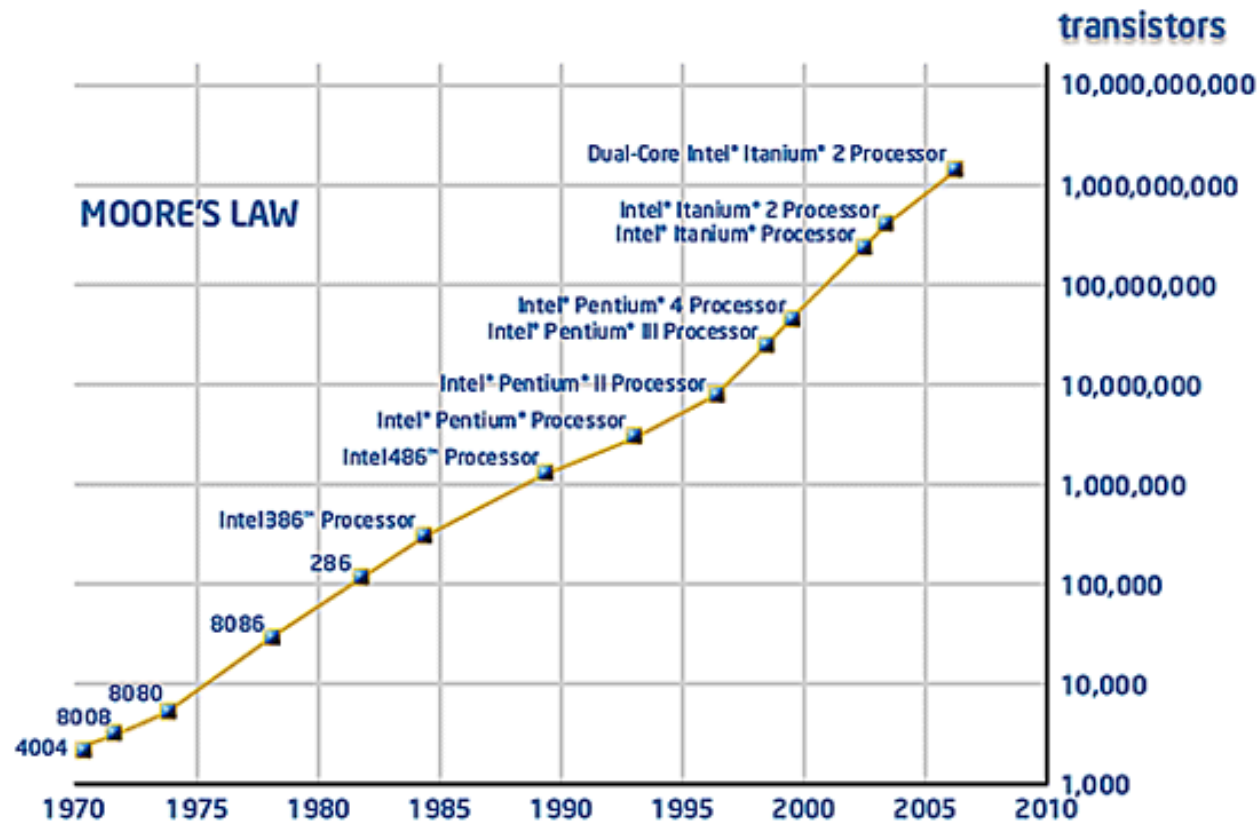
L'ordinateur est né – Machine de von Neumann

**Caractéristiques de l'ordinateur selon von Neumann**

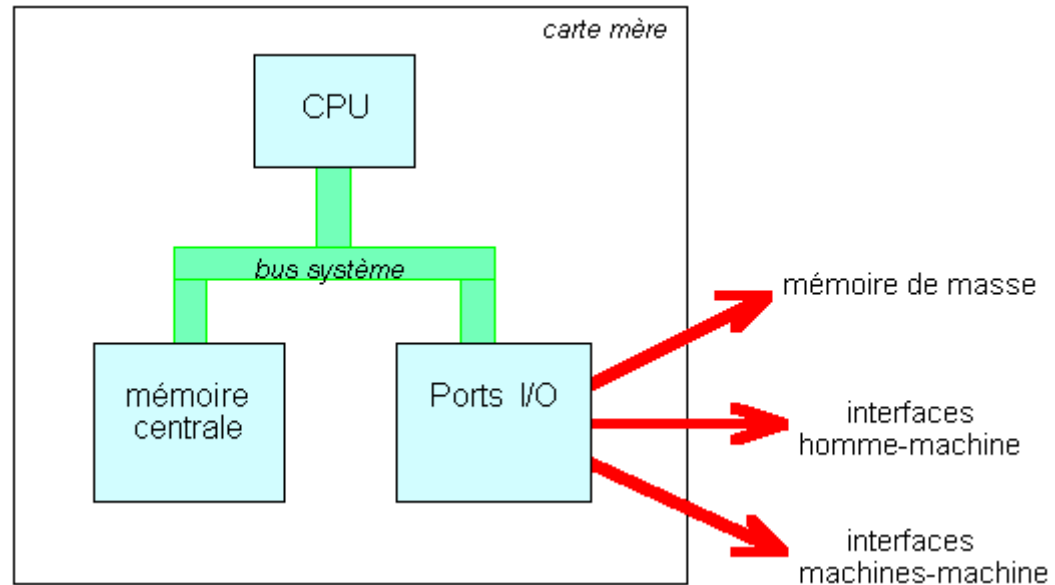
- Une machine universelle contrôlée par programme
- les instructions du programme sont codées sous forme numérique
- les instructions sont enregistrées en mémoire
- le programme peut modifier ses propres instructions, exécutées en séquence
- les instructions permettant les ruptures de séquence

# l'évolution de l'ordinateur - la loi de Moore (1965)

Une croissance exponentielle permettant d'intégrer, sur une surface donnée, deux fois plus de transistors tous les deux ans...



## l'organisation générale d'un ordinateur



**CPU** (Central Processing Unit) - commande tout le système

### **Mémoire centrale**

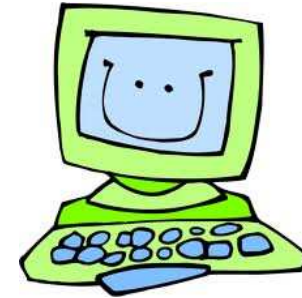
- la *mémoire vive* (*RAM* Random Access Memory)
- la *mémoire morte* (*ROM*, Read Only Memory)

**Les interfaces** - permettent à la carte mère de dialoguer avec des dispositifs externes nommés "*périphériques*,"

- la mémoire de masse - permet de stocker des informations (depuis la mémoire centrale): disques durs, bandes, CD, DVD ...
- les interfaces homme machine: le clavier et l'écran, la souris, l'imprimante, le joystick, le scanner, le micro et la carte son, la caméra numérique
- les interfaces machine - machine : modem, carte réseau,...

# La numération binaire et ses liens avec l'informatique

Les ordinateurs – dispositifs électroniques fonctionnant sur la principe de création, transmission et conversion d'impulses électriques



Les informations traitées par l'ordinateur (*nombres, instructions, images, sons...*) sont toujours représentées, à la base, sous forme élémentaire – binaire (0 ou 1)

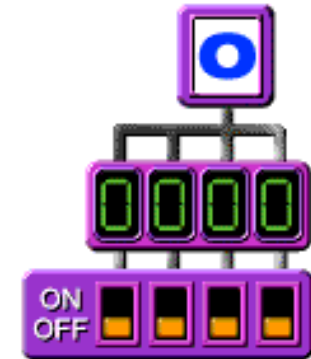
## Informatique : bit 0-1 (binary digit)

**la plus petite unité d'information manipulable par un ordinateur**

- physiquement représenté par une impulsion unique sur un circuit,
- par une petite zone d'une surface de disque, capable de stocker 0 ou 1

## Avantages du code binaire

- simple (0 ou 1, vrai ou faux ...)
- facile à réaliser techniquement à l'aide de bistables (systèmes à deux états d'équilibre)
- opérations (+ et \*) simples  
La table de multiplication en binaire est très simple
- grande utilité pour les machines modernes



*dès 1930, C.Shannon montre qu'une machine, à l'aide de contacteurs fermés (vrai) ou ouverts (faux), pourrait accomplir des opérations logiques.*

o **Inconvénients** du code binaire : difficultés dues au maniement de très grands nombres





# Theorie d'information

$$I(x) = \log_b(1/P(x))$$

nous utiliserons le logarithme en base 2 - **bit** - *binary information unit*

La quantité d'information dans un message est inversement proportionnelle à la probabilité d'apparition de ce message."

**L'unité d'information que nous utiliserons est le bit:**

Un bit est égal à la quantité d'information fournie par le choix d'une alternative parmi deux équiprobables.

$$H = -\log_2 p = 1 \text{ si } p = 1/2$$

(qui émerge dans la réponse « oui » ou « non » 0 ou 1)

## Mesure d'information

- bit – (*binary unit*), chiffre binaire: 0 ou 1

*nombres, instructions, images, séquences d'images animées, sons, etc., toujours représentées sous forme binaire*

- ensemble 8 bits = 1 octet - élément d'information code sur 8 bits

Dans un nombre binaire, la valeur d'un bit, appelée **poids**, dépend de la position du bit en partant de la droite

Nombre binaire	1	1	1	1	1	1	1	1
Poids	$2^7=128$	$2^6=64$	$2^5=32$	$2^4=16$	$2^3=8$	$2^2=4$	$2^1=2$	$2^0=1$

**difference B, b et o:**

B=byte, KB=kilobyte, MB, GB, ...

o = octet, Ko, Mo, Go

b=bits, Kb=kilobits, Mb-mégabits ....

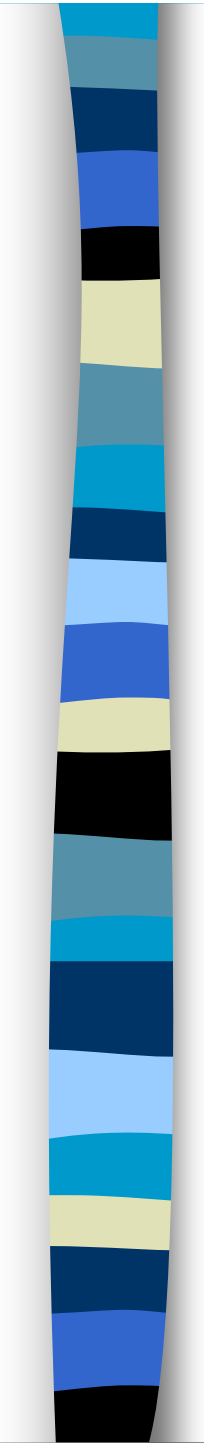
$$\text{Kilo} = 10^3 \approx 2^{10} = 1024$$

$$\text{Méga} = 10^6 \approx 2^{20} = 1\,048\,576$$

$$\text{Giga} = 10^9 \approx 2^{30} = 1\,073\,740\,824$$

$$\text{Tera} = 10^{12} \approx 2^{40} = 1\,099\,511\,627\,776$$

$$\text{Peta} = 10^{15} \approx 2^{50} = 1\,125\,899\,906\,842\,624$$

- 
- **En décimal**, avec  $n$  chiffres, on obtient  $10^n$  combinaisons possibles, i.e. on peut compter de 0 à  $10^n-1$ .
  - Exemple : Avec 3 chiffres, on a  $10^3 = 1000$  combinaisons possibles et on peut compter de 000 à 999.
  - 
  - **En binaire**, avec  $n$  bits, on obtient  $2^n$  combinaisons possibles, i.e. on peut compter de 0 à  $2^n-1$
  - Exemple : avec 8 bits, on a  $2^8 = 256$  combinaisons possibles et on peut compter de 00000000 à 11111111, i.e. de 0 à 255.

# Données numériques et non numériques

Notre but:

- comment les caractères et les nombres entiers positifs et négatifs sont représentés dans la mémoire d'un ordinateur
- comment on effectue les opérations arithmétiques avec des entiers binaires

Chaque caractère possède son équivalent en code numérique.  
Le code ASCII de base représentait les caractères sur 7 bits (c'est-à-dire 128 caractères possibles, de 0 à 127)



# 1. Le codage des informations non numériques

**Code ASCII** (*American Standard Code for Information Interchange* - traduisez « Code Américain Standard pour l'Echange d'Informations »).

- Les codes 0 à 31 - *caractères de contrôle* car ils permettent de faire des actions telles que p.ex. retour à la ligne (CR)
- Les codes 65 à 90 représentent les majuscules
- Les codes 97 à 122 représentent les minuscules

7 bits ->128 caractères :

26 lettres majuscules A - Z

26 lettres minuscule a - z

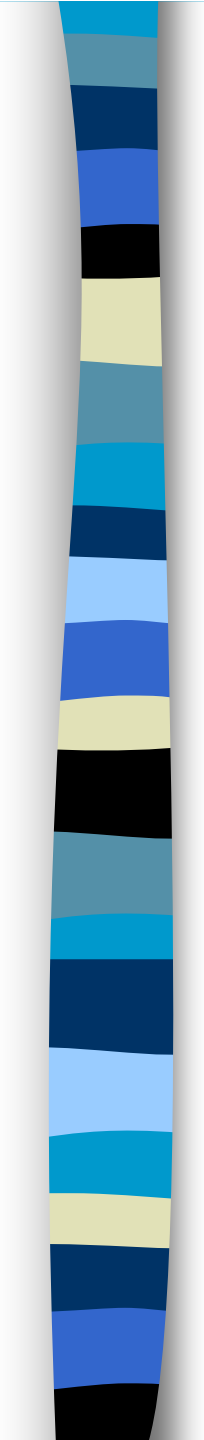
10 chiffres 0 à 9

33 caractères de ponctuation

sp, ! " # \$ % & ' ( ) \* + , - . / < = > ? @ [ ] ^ \_ `  
{ | } ~

33 caractères de contrôle :

null, etx, bel, bs, ht, lf, vt, ff, cr, ...,  
del



**Code ASCII Etendue** - étendu à 8 bits (un octet) pour pouvoir coder plus de caractères.

Aussi connu sous le nom de ISO-8859-1 ou ISO latin 1.

Définit les caractères ASCII 128 à 256

8 bits -> 256 caractères

caractères internationaux

caractères semi-graphiques

**Ex.**

**A – 01000001**

**L – 01001100**

**A**                    **L**                    **A**  
**01000001**    **01001100**    **01000001**



caractère	code ASCII	code hexadécimal
:	58	3A
;	59	3B
<	60	3C
=	61	3D
>	62	3E
?	63	3F
@	64	40
A	65	41
B	66	42
C	67	43
D	68	44
E	69	45
F	70	46
G	71	47
H	72	48
I	73	49
J	74	4A
K	75	4B
L	76	4C
M	77	4D
N	78	4E
O	79	4F
P	80	50
Q	81	51
R	82	52
S	83	53
T	84	54

## Table des caractères ASCII

qq exemples



## Unicode

- permet de représenter n'importe quel caractère par un code sur 16 bits, indépendamment de tout système d'exploitation ou langage de programmation.
- il regroupe ainsi la quasi-totalité des alphabets existants (arabe, arménien, cyrillique, grec, hébreu, latin, ...) et est compatible avec le code ASCII.
- dernière version Unicode Standard Version 6.2.0.  
<http://www.unicode.org/standard/standard.html>

## Le codage Uuencode

- (contraction de *Unix-to-Unix encode*) est un algorithme permettant de convertir des données binaires codées sur 8 bits en un format de codage sur 7 bits
- il a été prévu pour convertir les fichiers binaires dans un format texte compatible avec la messagerie et ne comportant pas de caractères risquant d'être convertis, ce qui rendrait illisible la pièce jointe.





## 2. Le codage des informations numériques

### Systèmes de numération

**Une base de numération est un système de représentation des nombres**

**N - nombre**

**p - base**

Représentation pondérée d'un nombre N dans une base p :

$$N = a_n p^n + a_{n-1} p^{n-1} + \dots + a_1 p + a_0$$

où  $a_n = 0, 1, \dots, p-1$

Les bases p les plus usitées sont :

p = 10, décimal

p = 2, binaire

p = 16, hexadécimal

p = 8, octal

Exemple :

$$1542_{10} = 1 \times 10^3 + 5 \times 10^2 + 4 \times 10^1 + 2 \times 10^0$$



## 2.1. Le codage des entiers naturels

La taille du plus grand nombre codable sur :

- 1 octet : 0-255
- 2 octets : 0-65 535
- 4 octets : 0-4 294 967 295

**Le codage des entiers naturels est des plus simples : il suffit de convertir le nombre décimal en binaire et vice-versa.**

### Changement de base (binaire, octale, hexadécimal)

**l'algorithme** pour convertir des nombres en base 10 en base 2,8,16 :

- on effectue une division entière du nombre à convertir par la base de destination et à conserver le reste
- on réitère le processus en utilisant le quotient obtenu
- on s'arrête lorsque le quotient est  $\leq$  à la base de destination
- le résultat est constitué par le dernier quotient suivi des restes.

# Représentation d'un nombre dans un ordinateur

## Base de représentation binaire

### opérations arithmétiques sur des entiers dans base binaire

#### L'addition

les retenues se font dès que la somme est supérieure à 1

$$0+0 = 0 \text{ r} = 0$$

$$0+1 = 1 \text{ r} = 0$$

$$1+1 = 0 \text{ r} = 1$$

$$\begin{array}{r} 00011010 \\ +00001011 \\ \hline 00100101 \end{array}$$

#### La multiplication

$$0 \times 0 = 0$$

$$0 \times 1 = 0$$

$$1 \times 0 = 0$$

$$1 \times 1 = 1$$

$$\begin{array}{r} 1 \ 1011 \\ \times 1101 \\ \hline 11011 \\ 00000 \\ 11011 \\ 11011 \\ \hline 10101111 \end{array}$$

## 2.2 Données numériques. Le codage des entiers relatifs

### Notation signe et grandeur

**un entier relatif positif ou nul** sera représenté en binaire comme un entier naturel, à la seule différence que le bit de poids fort représente le signe.

Il faut réserver un bit pour coder le signe (0 pour positifs et 1 pour négatifs).

Par exemple, avec la notation **signe et grandeur** sur 8 bits, on aurait :

$$+5 = 0000\ 0101$$

$$-5 = 1000\ 0101$$



Il faut que les règles d'addition soient conservées. L'astuce consiste à utiliser un codage que l'on appelle *complément à deux*.



## Entiers négatifs

✓ Complément à 1 sur 8 bits :

Le complément à 1 est obtenu en inversant tous les bits du nombre :

$$+5 = 0000\ 0101$$

$$-5 = 1111\ 1010$$

Dans le cas du complément à 1  $+5 + (-5) = 1111\ 1111 = -0$

✓ Complément à 2 :

Le complément à 2 est obtenu en additionnant 1 au complément à 1 :

$$+5 = 0000\ 0101$$

$$-5 = 1111\ 1011 \quad (1111\ 1010 + 1)$$

Dans le cas du complément à 2,  $+5 + (-5) = 0000\ 0000$

✓ Les micro-ordinateurs actuels utilisent tous le complément à 2 sur 8, 16, 32... bits pour représenter les nombres négatifs.



## Le codage des entiers négatifs

Pour les nombres dont le bit le plus significatif est 1, il y a deux interprétations possibles.

Par exemple, sur 8 bits,

1001 0000 peut représenter

+144 si on le considère comme un nombre non signé

-112 si on le considère comme un nombre en complément à 2.

C'est lors de la déclaration d'une variable en mémoire qu'on détermine si le processeur doit la traiter comme signée ou non signée.

`-char a;` // a est considéré comme signé en complément à 2

`-unsigned char b;` // a est considéré comme non signé



## 2.3 Représentation d'un nombre réel

Ex.  $23,45686443 \cdot 10^{22}$

La norme *IEEE* définit la façon de coder un nombre réel.  
Cette norme se propose de coder le nombre sur 32 bits et définit trois composantes :

- le signe est représenté par un seul bit, le bit de poids fort (celui le plus à gauche)
- l'exposant est codé sur les 8 bits consécutifs au signe
- la mantisse (les bits situés après la virgule) sur les 23 bits restants

Ainsi le codage se fait sous la forme suivante :

**Seeeeeemmeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee**

le **s** représente le bit relatif au signe

les **e** représentent les bits relatifs à l'exposant

les **m** représentent les bits relatifs à la mantisse

## Système octal-hexadécimal

la base octale - les chiffres de 0 à 7

la base hexadécimale - les chiffres de 0 à 9  
et les lettres de A(=10) à F(=15)

Conversion binaire → octale (hexadécimal)

Conversion octale (hexadécimal) → binaire

$$\text{ex. } 3A5_h = 0011\ 1010\ 0101_b$$

Conversion octale (hexadécimal) → décimal

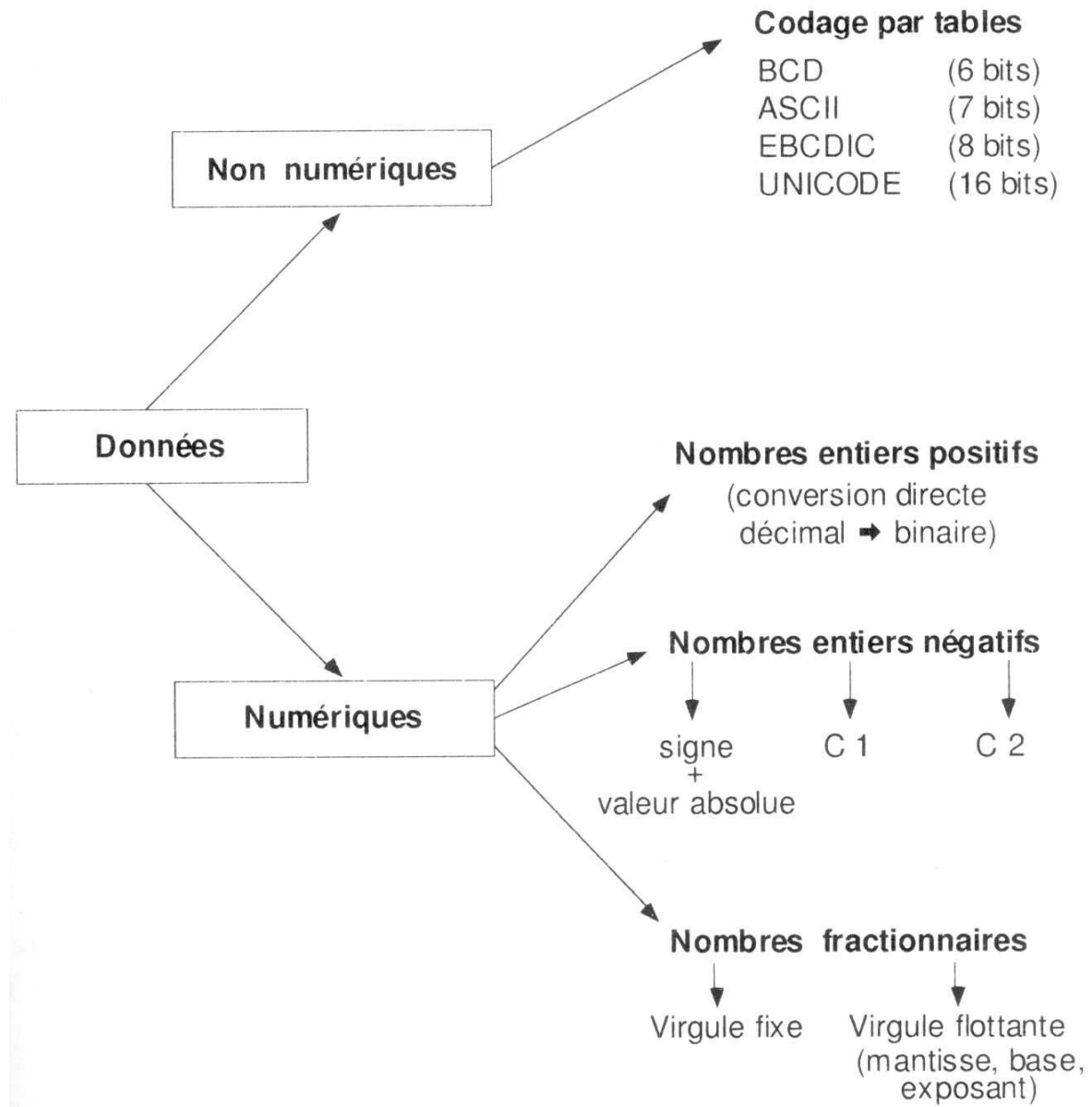
$$\text{ex. } 3A5_h \text{ vaut } 3 \times 16^2 + 10 \times 16^1 + 5 \times 16^0 = 3 \times 256 + 160 + 5 = 933_d$$

H	D	B
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111





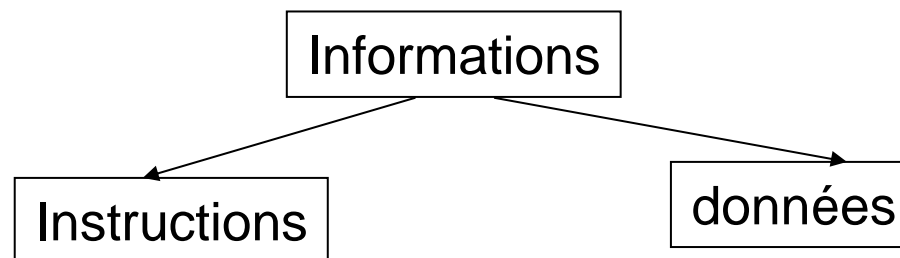
<b>O</b>	<b>H</b>	<b>D</b>	<b>B</b>
0	0	0	0000
1	1	1	0001
2	2	2	0010
3	3	3	0011
4	4	4	0100
5	5	5	0101
6	6	6	0110
7	7	7	0111
10	8	8	1000
11	9	9	1001
12	A	10	1010
13	B	11	1011
14	C	12	1100
15	D	13	1101
16	E	14	1110
17	F	15	1111

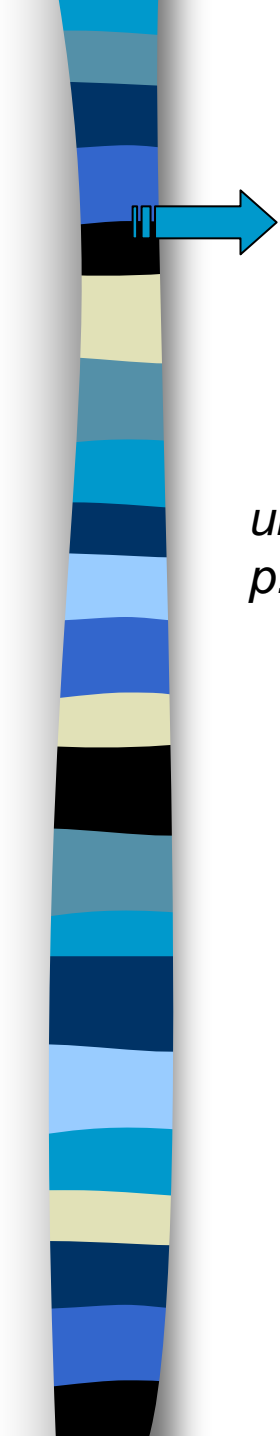


# Représentation interne des informations

Informations traitées par l'ordinateur:

- instructions (les opérations)
- données (nécessaires à l'exécution des programmes)





*def.* Les représentations physiques des abstractions que nous définissons dans l'appréhension du réel perçu sont les données et la signification que nous leur donnons représente l'information que l'on peut en déduire.

*une **donnée** est une description élémentaire, souvent codée –  
p.ex. d'une objet, d'une transaction d'affaire, d'un événement, etc*

## Types de données

logiques : „oui” ou „non”

alphanumériques : (lettres+nombre)

numériques

graphiques

musiques.....



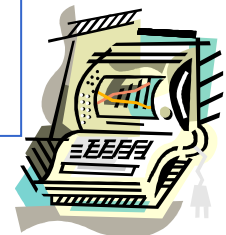
# données

On distingue les données numériques et non numériques



Informations a entrée  
– chaînes de caractères  
alphanumériques

Ordinateur  
- langage binaire



Codage des informations  
Langages de codification

# Comment donne-t-on des ordres à un ordinateur ?

Grâce à un *programme*



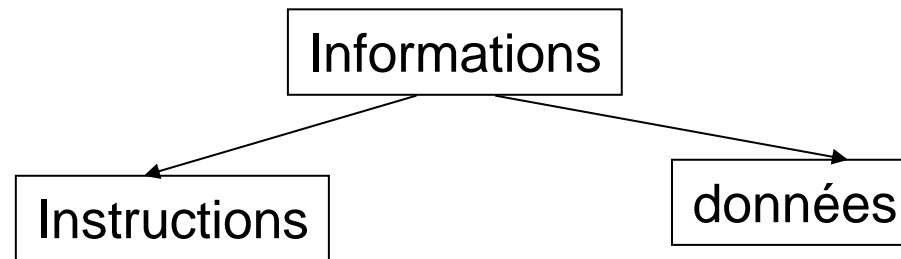
## programme

- une séquence d'instructions décrivant les actions que doit exécuter l'ordinateur, écrites dans un langage compréhensible par l'ordinateur
- les informations utilisées par ce programme sont les données

## algorithme

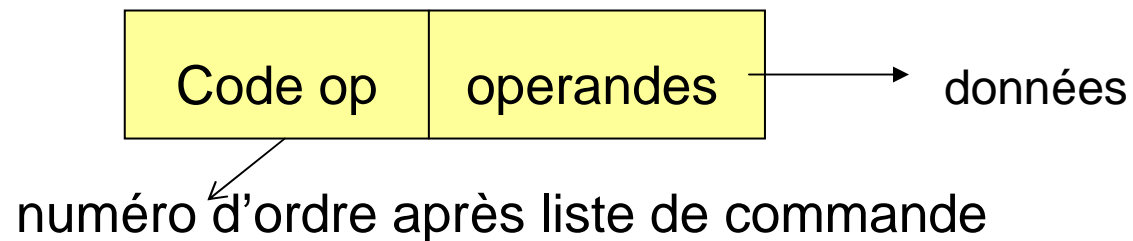
- une succession d'actions (instructions) destinées à résoudre un problème en un nombre fini d'opérations

# instructions



composée de :

le code de l'opération à effectuer – doit subir un décodage  
les opérandes impliqués dans l'opération



# langages de programmation

- langage *haut niveau*
  - langage *bas niveau*
- } langage externe

# langages de programmation

- langage machine (binaire pur)
- langage assembleur (codes mnémonique, ex.ADD, SUB...)
- langage évolués (Fortran, Pascal, C, C++, Java...)



Langage humains

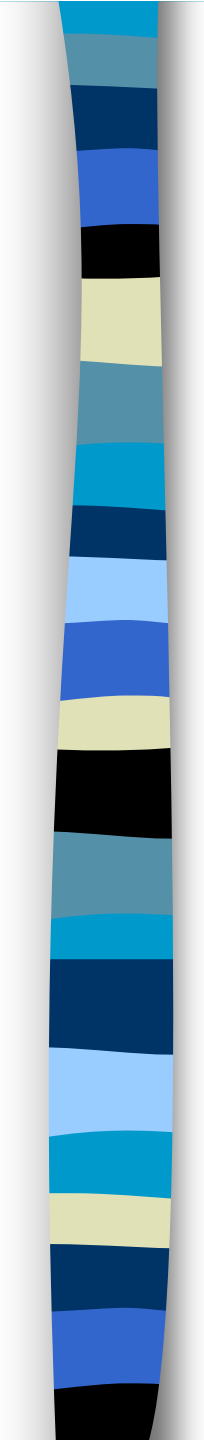
Langage évolués

Langage orientés machine



langage machine





1. Programmer rédige un programme  
- programme source

2. Programme après traduction  
- programme objet

## Traducteurs

assembleurs

↓  
langages source -  
langage assembleur  
(une variante  
symbolique du langage  
machine)

compilateurs

↓  
langages source est un  
langage évolué

**Assembleur ou compilateur assure la transcription d'un programme source en un programme objet**