



Algorithmique Avancée

Pr. Essaid SABIR

Contact : e.sabir@ensem.ac.ma

Cycle : Ingénieur

Niveau : Ing1

Année : 2017/2018



- Généralités
- Programmation informatique
- Algorithmique Classique
- Récursivité
- Algorithmes de Tri
- Complexité et Optimalité
- Structures de Données
- Arbres de Recherche
- Graphes

❑ Kézako l'informatique ?

- ❑ Ensemble de sciences et techniques en relation avec le traitement de l'information.

❑ Kézako l'ordinateur ?

- ❑ C'est une machine universelle capable de traiter automatiquement des informations. Un ordinateur est généralement composé deux parties distinctes



Généralités

L'ordinateur : Aspect matériel



■ Stockage de données :

- Disque dur, flash USB, CD-ROM/DVD-ROM, Blue Ray, ...



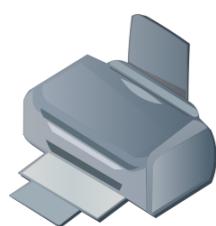
■ Entrée de données :

- Clavier, souris, écran tactile, scanner, palette graphique, ...



■ Sortie de données :

- Ecran, volumes de stockage, imprimante, ...



Généralités

L'ordinateur : Aspect matériel



■ Stockage de données :

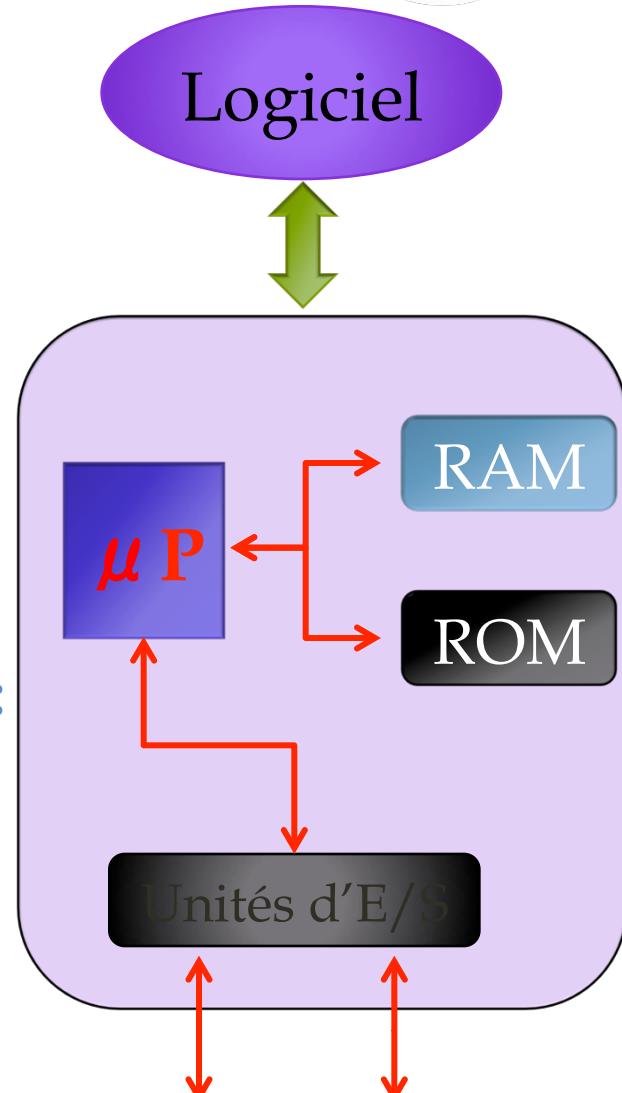
- Mémoire vive : RAM
- Mémoire morte : ROM

■ Traitement de données :

- Processeur(s)

■ Communication avec les périphériques :

- Unités d'entrée/sortie



Généralités

L'ordinateur : Aspect logiciel



- ❑ **Le Système d'Exploitation :** Liaison entre
 - ❑ Ressources Matérielles
 - ❑ Applications
 - ❑ Utilisateurs



- ❑ **Le système d'exploitation gère :**
 - ❑ La mémoire
 - ❑ Le(s) processeur(s)
 - ❑ L'ordonnancement
 - ❑ Les utilisateurs (droits)
 - ❑ Les services réseaux
 - ❑ Les fichiers



Généralités

L'ordinateur : Aspect logiciel



❑ Les Applications :

❑ Programmes élaborés pour résoudre une(des) tâche(s) spécifique(s) !

Traitement

Tableur

Retouche de photos

Gestion de base de données

Jeux

Navigateurs Web

Messagerie instantanée

Écoute de musique

GPS

Reconnaissance vocale

Bref, presque tout ce que vous imagineriez !!

La programmation

Définition



Kézako la programmation ?

- Séquences d'instructions qui permettent de donner un ensemble d'ordres à une machine (Ordinateur, Smartphone, ...).

A quoi sert-elle ?

- Développer des applications;
- Automatiser des tâches récurrentes;
- Effectuer des calculs complexes (simulations).

Exemples de langages de programmation

- C, C++, C#, JAVA, (Visual) Basic, Pascal, Delphi, Perl, PHP, ...

La programmation

Unité de base de la programmation



Kézako une instruction ?

- Les instructions sont un type de données particulières qui permettent d'agir sur des données réelles !!

Génération d'instructions

- Opérations de base
 - Addition, soustraction, multiplication, division
 - Comparaison
 - Branchement
 - Communication d'une information
- Un autre programme

La programmation

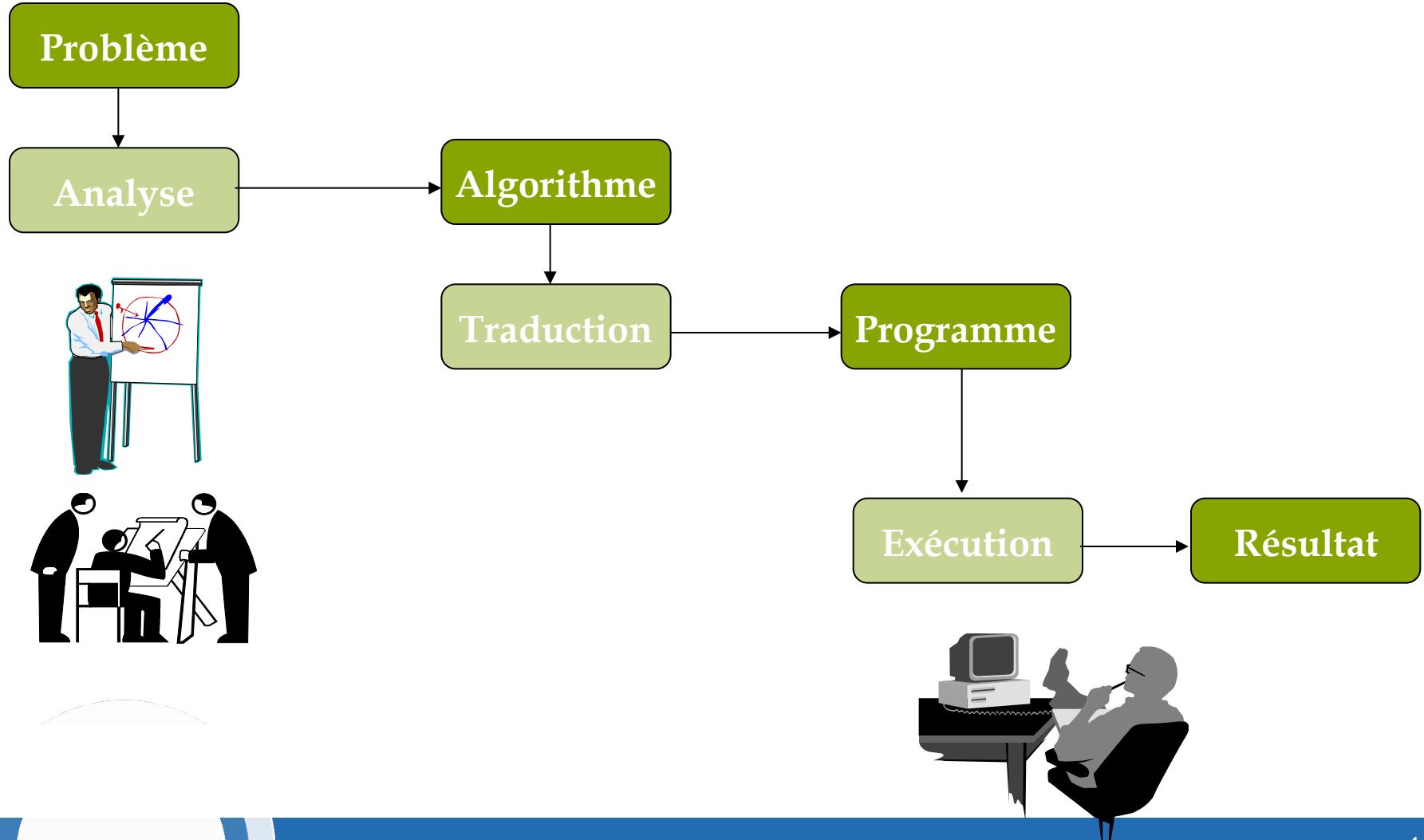
Ce qu'il faut retenir !!



- Le processeur **exécute** des instructions
- Les mémoires **stockent** des données
- Les instructions **manipulent** des données

La programmation

Démarches de la programmation



La programmation

Le cycle du développement

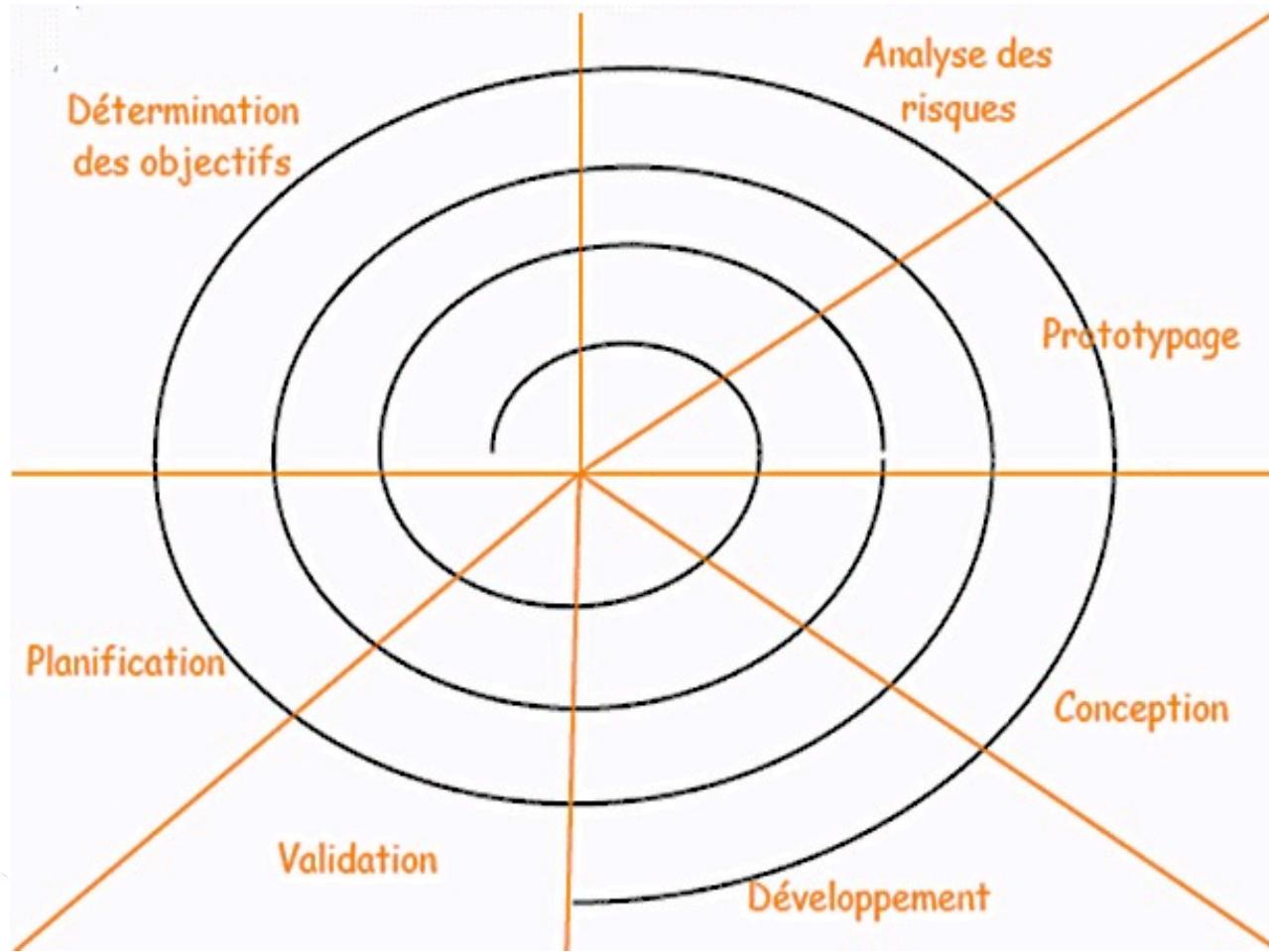


Il s'agit d'une approche partant de l'abstraction du problème jusqu'à sa réalisation concrète.

- **Spécification du problème :** Quel est le problème ?
- **Conception abstraite :** Phase de réflexion permettant de déterminer les données et les résultats ;
- **Conception algorithmique :** Définitions lexicales des éléments utilisés, et ordonnancement des traitements ;
- **Codification :** Traduction dans un ou dans plusieurs langages de programmation ;
- **Tests applicatifs :** Mise en œuvre d'un échantillon représentatif de données pour valider l'application ;
- **Maintenance :** Correction, évolution, optimisation, sécurisation,

La programmation

Le cycle du développement



La programmation

Niveaux de programmation



Langage machine

```
0101010010100111011110101010101  
01010101010101010101101010101010  
100001111100111110010100000100
```

Langage bas niveau

```
mov eax, 4 ;  
mov ebx, 1 ;  
mov eax, ebx ;
```

Langage évolué

```
printf("Entrez la valeur de x : ");  
scanf(&x);  
if (x==1){  
    printf("x vaut 1");  
}
```

La programmation

Quel langage choisir ?



Différences principales entre les langages de programmation sont :

Structure : lexique, syntaxe, sémantique ;

Famille : Langages impératifs, langages orienté-objet, langages logiques, langages fonctionnels, ...;

Mode d'exécution : Langages compilés, Langages interprétés, langages exécutés ;

Usage spécifique :

- A Internet (pages web): PHP, Perl, JAVA, ... ;
- Aux simulations : Matlab, Maple, Mathematica, ... ;
- Aux programmes nécessitant une grande vitesse : C, C++, ... ;
- Aux applications graphiques et bureautiques : VB, Delphi,

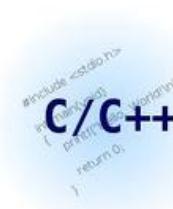
La programmation

Types de langages



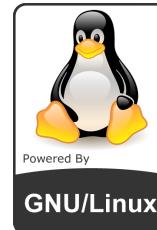
Langages compilés

- ❑ C/C++/C#, Java, VB, ...



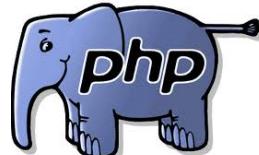
Langages interprétés

- ❑ Script Shell, Perl, Python



Langages spécialisés

- ❑ Web : PHP, ASP, VB script, Java script, ...



- ❑ Simulation : Matlab, Maple, Mathematica, ...



La programmation

Compilation V.S. Interprétation



	Avantages	Inconvénients
Langage exécuté	<ul style="list-style-type: none">• Fichier exécutable• Autonome• Rapide à l'exécution	<ul style="list-style-type: none">• Programmation très pénible• Impraticable !
Langage compilé	<ul style="list-style-type: none">• Fichier exécutable• Autonome• Rapide à l'exécution	<ul style="list-style-type: none">• Toute modification du fichier source mène à une recompilation complète
Langage interprété	<ul style="list-style-type: none">• Lisible• Modifiable plus facilement	<ul style="list-style-type: none">• Pas sécurisé• Moins rapide à l'exécution



D'où proviennent-elles ?

Que deviennent-elles ?

Archivage

- ◆ Mises en mémoire persistante

Communication

- ◆ Saisies (Entrée) ou restituées (Sortie) à l'Utilisateur
- ◆ Transmise à un nouveau programme

Toute donnée peut être repérée à l'aide de **son adresse**

La programmation

L'instruction



Opérations de base

Addition, soustraction, multiplication, division

Comparaison

Communication d'une information

Branchement

Un autre programme

Les instructions ne sont que des données particulières

La programmation

Codage de l'information



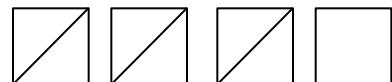
Codage de l'information...pour l'humain

XIX

19

dix-neuf

diznoef



Sens interdit



20-1

La programmation

Codage de l'information



Codage de l'information...pour l'ordinateur

Tout est codé en binaire : 0 / 1

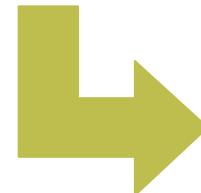
10011

Le codage binaire :

1 bit : 0 ou 1

1 octet : 8 bits

Les chiffres
Les caractères
Les instructions
Les dessins
Les sons...



Importance de connaître le type de données

La programmation

Codage de l'information



Codage de l'information...pour l'ordinateur

Unités de mesure

1Kio (kibioctet) = 1 024 octets

1Mio (mébioctet) = 1 048 576 octets

1Gio (gibioctet) = 1 073 741 824 octets

1Tio (tébioctet) = 1 099 511 627 776octets

La programmation

Codage de l'information



Codage de l'information...pour l'ordinateur

Unités de mesure

1Ko (kilo octet) = 1 000 octets

1Mo (méga octet) = 1 000 000 octets

1Go (giga octet) = 1 000 000 000 octets

1To (téra octet) = 1 000 000 000 000 octets

La programmation

Codage de l'information

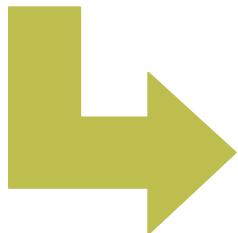


Première question...

Comment communiquer avec l'ordinateur ?

Programmerons-nous en binaire ? **NON : Utilisons un traducteur...**

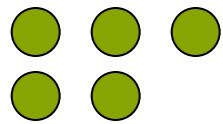
L'ordinateur a été programmé pour traduire un **programme dans un langage de programmation donné en langage machine**



- Restriction du langage naturel
- Plus le langage est élaboré, plus le travail de traduction est grand

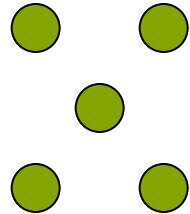
La programmation

Codage de l'information



V

101



5

Five

Cinq

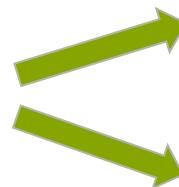
plusieurs représentations différentes du nombre 5
↳ système de numération

La programmation

Codage de l'information



Systèmes de numération

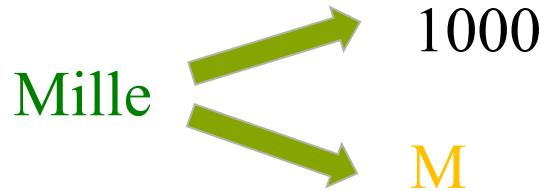


positionnels

ex.: décimal

non positionnels

ex.: romain



Un système de numération positionnel est caractérisé par

- sa base et
- par le nombre de symboles

Base = valeur attribuée au symbole 10

La programmation

Codage de l'information



■ Système décimal

base : 10

symboles : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

$$4693 = 4*10^3 + 6*10^2 + 9*10^1 + 3*10^0$$

■ Système hexadécimal

base : 10_{16} ($= 16_{10}$)

symboles : 0, 1, 2, 3, 4, 5,
6, 7, 8, 9, A, B, C, D, E, F

$$\begin{aligned} A3C_{16} &= A_{16}*10_{16}^2 + 3*10_{16}^1 + C*10_{16}^0 \\ &= 10*16^2 + 3*16^1 + 12*16^0 \end{aligned}$$

■ Système binaire

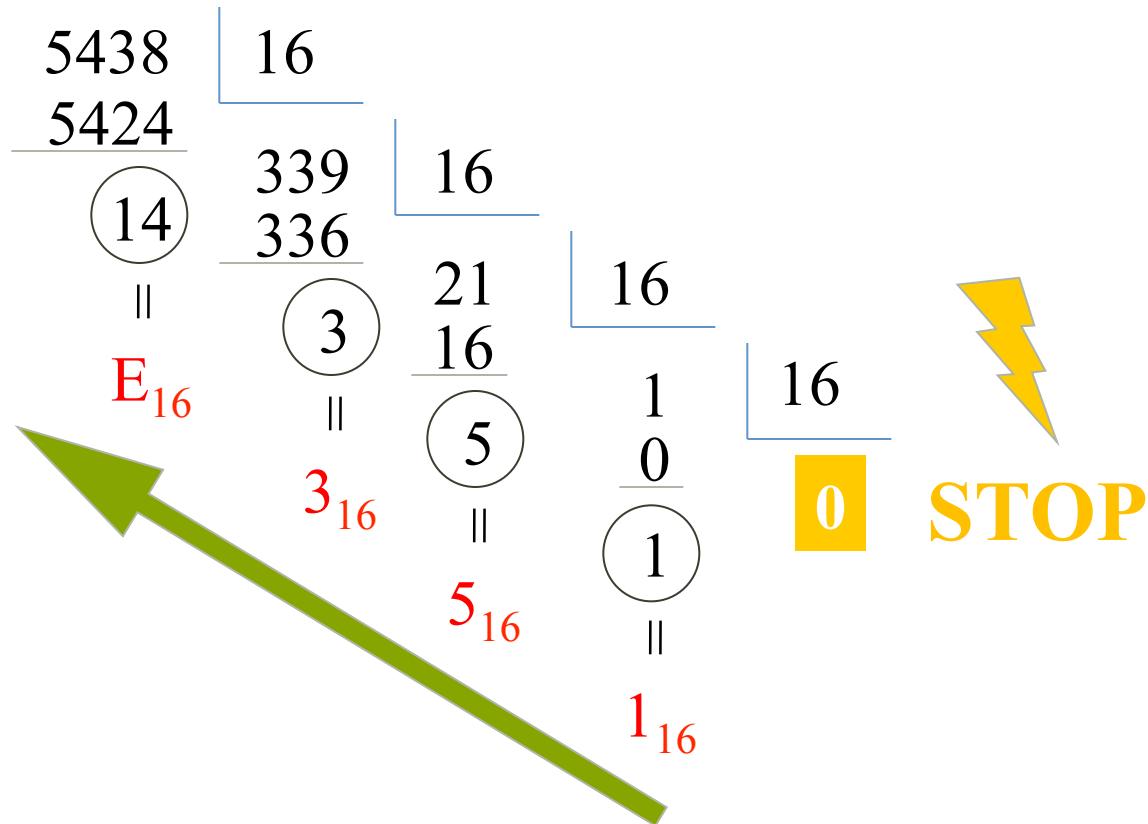
base : 10_2 ($= 2_{10}$)

symboles : 0, 1

$$\begin{aligned} 101_2 &= 1_2*10_2^2 + 0_2*10_2^1 + 1_2*10_2^0 \\ &= 1*2^2 + 0*2^1 + 1*2^0 \end{aligned}$$

La programmation

Conversion Décimal → Hexadécimal

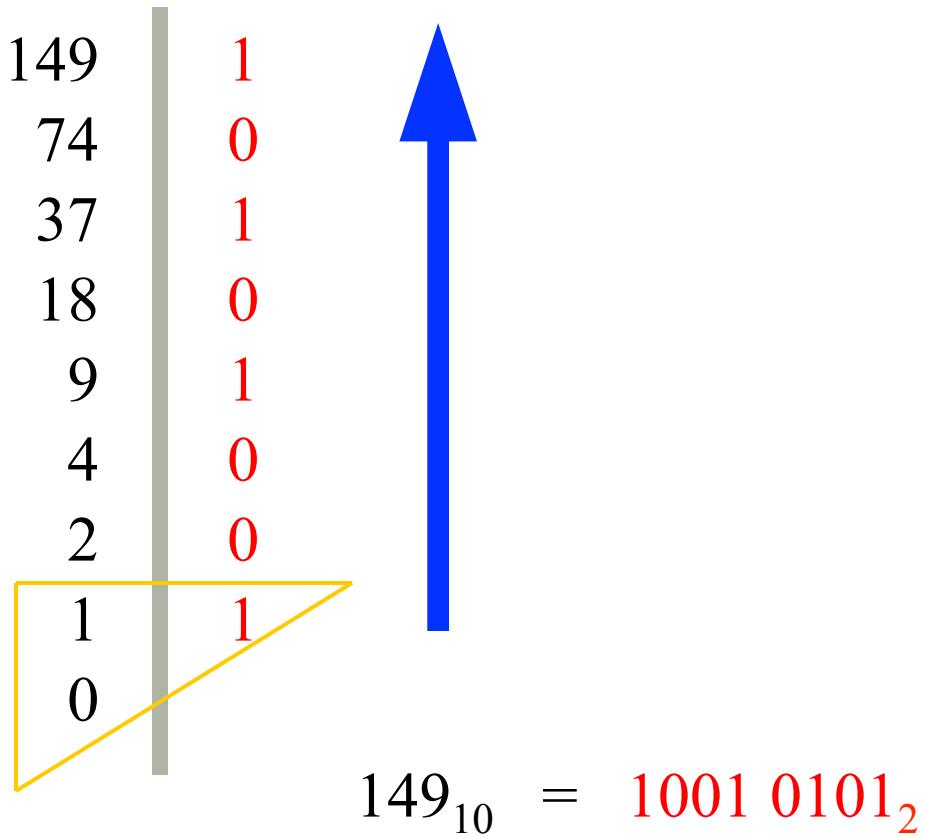


$$5438_{10} = 153E_{16}$$

Méthode des divisions successives par 16

La programmation

Conversion Décimal → Binaire



Méthode des divisions successives par 2

La programmation

Conversion Hexadécimal → Décimal



$$\begin{array}{r} 1 \quad \quad \quad 5 \quad \quad \quad 3 \quad \quad \quad E \\ *16 \quad \quad \quad | \quad \quad \quad | \quad \quad \quad | \\ \hline 16 \quad + \quad 5 \\ *16 \quad \quad \quad | \\ \hline 336 \quad + \quad 3 \\ *16 \quad \quad \quad | \\ \hline 5424 \quad + \quad 14 \quad = \quad 5438_{10} \end{array} \quad 16$$

$$153E_{16} = 5438_{10}$$

La programmation

Conversion Hexadécimal → Décimal



autre méthode

$$1 \ 5 \ 3 \ E_{16} = E * 16_{10}^0 \\ 1_{10} \qquad \qquad \qquad 14$$

$$10_{16} = 16_{10}$$

$$+ 3 * 16_{10}^1 \\ 16_{10} \qquad \qquad \qquad 48$$

$$+ 5 * 16_{10}^2 \\ 256_{10} \qquad \qquad \qquad 1280$$

$$+ 1 * 16_{10}^3 \\ 4096_{10} \qquad \qquad \qquad \underline{4096} \\ 5438_{10}$$

La programmation

Conversion Binaire → Décimal



$$\begin{array}{r} 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 2 \\ *2 \quad | \\ 2 + 0 \quad \\ \underline{*2} \quad \\ 4 + 0 \quad \\ \underline{*2} \quad \\ 8 + 1 \quad \\ \underline{*2} \quad \\ 18 + 0 \quad \\ \underline{*2} \quad \\ 36 + 1 \quad \\ \underline{*2} \quad \\ 74 + 0 \quad \\ \underline{*2} \quad \\ 148 + 1 = 149_{10} \end{array}$$

La programmation

Conversion Binaire → Décimal



autre méthode

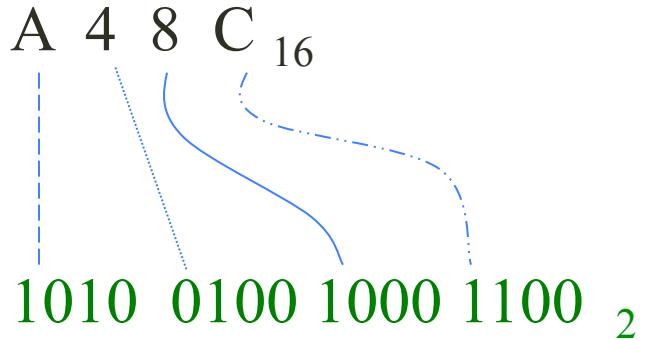
$$1011_2 = 1 * 2_{10}^0 + 1 * 2_{10}^1 + 0 * 2_{10}^2 + 1 * 2_{10}^3$$
$$1_{10} + 2_{10} + 4_{10} + 8_{10}$$
$$\underline{+ 8}$$
$$11_{10}$$

$10_2 = 2_{10}$

A curved arrow points from the binary number 1011_2 to the first term $1 * 2_{10}^0$.

La programmation

Conversion Hexadécimal → Binaire



Remplacer chaque chiffre hexadécimal par son équivalent binaire (en 4 chiffres)

Remarque : on peut supprimer les 0 en début de nombre

Exemple : 39D₁₆ = X11 1001 1101₂

La programmation

Conversion Binaire → Hexadécimal



1010 0100 1000 1100₂
A 4 8 C

$$1010010010001100_2 = \text{A}48\text{C}_{16}$$

Regrouper les chiffres binaires par 4
(en commençant par la fin du nombre)

Remplacer chaque groupe de 4 chiffres binaires
par son équivalent hexadécimal

Exemple : 0011 1001 1011₂
 3 9 B

$$1110011011_2 = 39\text{B}_{16}$$

La programmation

Exercices



Exercice 1

Représentez le nombre 248_{10} dans les bases 2, 3, 8, 9 et 16.

(Utilisez la technique des divisions successives pour les bases 2, 3 et 16.)

Exercice 2

Représentez les nombres 28_{10} , 129_{10} , 147_{10} , 255_{10} sous leur forme binaire par une autre méthode que les divisions successives.

À partir de cette représentation binaire, vous en déduirez leur représentation hexadécimale.

La programmation

Exercices



Exercice 3

1. Les nombres 11000010_2 , 10010100_2 , 11101111_2 , 10000011_2 , 10101000_2 sont-ils pairs ou impairs ?
2. Lesquels sont divisibles par 4, 8 ou 16 ?
3. Donnez le quotient et le reste d'une division entière par 2, 4 et 8 de ces nombres.

Exercice 4

Donnez les valeurs décimales, minimales et maximales, que peuvent prendre des nombres signés et non signés codés sur 4, 8, 16, 32 et n bits.

La programmation

Concepts fondamentaux



Les variables

Imposés par la nature de l'ordinateur et l'existence du compilateur

Les variables, avec leurs types et leurs adresses (notion d'affectation)

La lecture et l'écriture

Imposés par les possibilités d'actions

Les boucles

Les structures de contrôle

Liés à certains paradigmes de programmation

Structures de données

Fonction

Objet

La programmation

Concepts fondamentaux



Les variables et les structures de données

Permettent de représenter les informations à manipuler

Une seule valeur : la variable (est un nom qui sert à repérer un emplacement donné de la mémoire centrale)

Un ensemble de variables : la structure de données

Connaissent un type

Un entier, un réel, un caractère, un booléen

A chaque valeur correspondra un emplacement défini par son adresse

La programmation

Concepts fondamentaux



Les structures de contrôle

Précisent l'enchaînement des instructions d'un programme

Répétitions

Choix

Font intervenir des instructions de branchement

La programmation

Concepts fondamentaux



Les fonctions et les procédures

Donnent un nom à un ensemble d'instructions

Utilisables à volonté

Peuvent être paramétrées

Peuvent renvoyer des résultats

La programmation

Types de langages de programmation



Les langages non-structurés

Suite d'actions à exécuter séquentiellement

Abandonné (sauf pour des calculettes)

Début programme

Entrer dans Cuisine

Allumer Lumière

Ouvrir Réfrigérateur

Prendre Oeuf

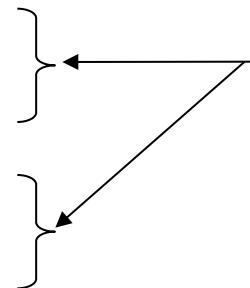
Fermer Réfrigérateur

Ouvrir Réfrigérateur

Prendre Beurre

Fermer Réfrigérateur

Fin programme



Ces actions sont
quasi-identiques

La programmation

Types de langages de programmation



Les langages structurés

Regroupement possible d'instructions (= Fonctions)

Passage de paramètres

→ Premiers pas vers la modularité

Action PrendreDansRéfrigérateur (*ingrédient*)

Ouvrir Réfrigérateur
Prendre *ingrédient*
Fermer Réfrigérateur



Fonctions à paramètres

Fin PrendreDansRéfrigérateur

Action EntrerDansCuisine ()

Entrer dans cuisine
Allumer Lumière



Fonctions sans paramètre

Fin EntrerDansCuisine

Début programme

EntrerDansCuisine
PrendreDansRéfrigérateur (oeuf)
PrendreDansRéfrigérateur(beurre)

Fin programme

La programmation

Types de langages de programmation



Les langages orientés-objet

Tout est objet avec des caractéristiques et des comportements

Extrêmement modulable

Plus facile à modifier

Meilleure réutilisation

Réalisation d'actions plus complexes

Rigueur d'analyse et de programmation

La programmation

Types de langages de programmation



Les langages orientés-objet

Classe d'objets Lumière

Caractéristiques :

60 Watts

Comportements :

Allumer
Eteindre

Fin Objets Lumière

Classe d'objets Cuisine

Caractéristiques :

15 m², Rez-de-chaussée

Comportements :

Allumer : (Lumière : Allumer)

Fin Objets Cuisine

Classe d'objets EtreHumain

Caractéristiques :

Nom, Prénom, Age

Comportements :

EntrerDans(Pièce) : (Pièce : Allumer)

PrendreDans(objet1,objet2) :

Ouvrir(objet1), Saisir(objet2), Fermer(objet1)

Ouvrir(objet)

Fermer(objet)

Saisir(objet)

Fin Objets EtreHumain

Classe d'objets Réfrigérateur

Caractéristiques :

Hauteur, Couleur

Comportements :

Allumer : (Lumière : Allumer)

Lumière : (Lumière : Eteindre)

Fin Objets Réfrigérateur

Début Programme

Etre humain : EntrerDans(cuisine)

Etre humain : Ouvrir (Réfrigérateur)

Etre humain : PrendreDans (Réfrigérateur, oeuf)

Etre humain : PrendreDans (Réfrigérateur, beurre)

Fin Programme

La programmation

Les paradigmes de programmation



Déterminent la façon de penser du programmeur par rapport à l'exécution de son programme

Interdisent certaines instructions

En autorisent d'autres

Un langage peut supporter plusieurs paradigmes mais en privilégie le plus souvent un

La programmation

Les paradigmes de programmation



La programmation impérative

Se fonde sur les instructions de base

Affectation

Branchements

Boucles

Description des opérations en terme de séquences d'instructions exécutées par l'ordinateur pour modifier l'état global du programme

Applications possibles : les recettes de cuisines, les vérifications de processus industriels



La programmation impérative

- On distingue deux types de langages impératifs. D'une part les langages machines et assembleurs. Ceci explique que les premiers langages de programmation apparus soient des langages impératifs : une instruction du langage correspond à un ensemble d'instructions du langage machine. Les structures de données et opérations sont plus complexes qu'au niveau de la machine, mais le paradigme suivi reste le même.
- Les langages procéduraux forment la seconde famille de langages impératifs. Une procédure, appelée également fonction, est une suite d'instructions devant être effectuée dans un ordre précis. On distingue parfois procédure et fonction par la caractéristique qu'une procédure ne renvoie pas de résultat.

(Wikipedia)



La programmation fonctionnelle

Ensemble de fonctions et appels de fonctions

Se fonde sur la notion de calcul

Pas de changement d'état

Pas de modification des données

Equivalent aux fonctions mathématiques

Applications possibles : calculs mathématiques

La programmation

Les paradigmes de programmation



La programmation logique

Se fonde sur un ensemble

De faits élémentaires

Des règles logiques

De conséquences

Un moteur d'inférences

Applications possibles : analyse sémantique de textes

La programmation

Les paradigmes de programmation



Et encore...

Les langages descriptifs

Les langages de script

Les langages de requêtes

Les langages de calcul formel

Les langages déclaratifs

La programmation

Les paradigmes de programmation



Quel langage allons nous utilisé ?

ATTENTION : un langage peut regrouper plusieurs paradigmes !

Pour nous, **IMPERATIF, PROCEDURAL** et **ORIENTE OBJET**

Le **langage C** mais sans se servir de la notion d'**OBJET**

La programmation

Les paradigmes de programmation



En conclusion

Plusieurs niveaux de langage

Pour les langages évolués

Compilation et Interprétation

Plusieurs paradigmes de programmation

Plusieurs familles de programmation

De nombreux langages possibles

Comment représenter les problèmes que nous voulons résoudre en fonction du paradigme que nous avons choisi ?