

# Éléments d'architecture de l'ordinateur

## Chapitre 01

ISN

# Plan

## Langages et ordinateurs

Vocabulaire

Langage machine

Traduction, interprétation

Matériel et logiciel

## Quelques étapes de l'évolution de l'architecture des ordinateurs

Les calculateurs mécaniques (1642 – 1945)

Les tubes à vide (1945 – 1955)

Les transistors (1955 – 1965)

Les circuits intégrés (1965 – 1980)

Les ordinateurs personnels (1980 – ?)

# Plan

## Langages et ordinateurs

Vocabulaire

Langage machine

Traduction, interprétation

Matériel et logiciel

## Quelques étapes de l'évolution de l'architecture des ordinateurs

Les calculateurs mécaniques (1642 – 1945)

Les tubes à vide (1945 – 1955)

Les transistors (1955 – 1965)

Les circuits intégrés (1965 – 1980)

Les ordinateurs personnels (1980 – ?)

# Vocabulaire

- ▶ Un **ordinateur** est une machine capable d'exécuter des actions à partir d'instructions préalablement définies.
- ▶ La suite des instructions décrivant la façon dont l'ordinateur doit effectuer un certain travail est appelé **programme**.
- ▶ Un **algorithme** est une suite finie et non ambiguë d'opérations ou d'instructions permettant de résoudre un problème.
- ▶ Un programme est généralement constitué de plusieurs algorithmes exprimés dans un **langage de programmation**.

## Remarque

Différents programmes peuvent implémenter des algorithmes différents pour résoudre le même problème.

# Pourquoi ne pas utiliser un langage humain ?

Les langages humains sont :

- ▶ **Complexes** : il nous faut de longues années d'étude avant d'être capables de maîtriser notre langue maternelle
- ▶ **Ambigüs** : de nombreux mots changent de sens en fonction du contexte dans lequel ils sont utilisés
- ▶ **Évolutifs** : les langages évoluent au cours du temps avec les sociétés qui les utilisent et

## Exemple

Example is taken from the instructions with a shipment of ballistic missiles from the British Admiralty :

*"It is necessary for technical reasons that these warheads be stored upside down, that is, with the top at the bottom and the bottom at the top. In order that there be no doubt as to which is the bottom and which is the top, for storage purposes, it will be seen that the bottom of each warhead has been labeled 'TOP'."*

# Plan

## Langages et ordinateurs

Vocabulaire

Langage machine

Traduction, interprétation

Matériel et logiciel

## Quelques étapes de l'évolution de l'architecture des ordinateurs

Les calculateurs mécaniques (1642 – 1945)

Les tubes à vide (1945 – 1955)

Les transistors (1955 – 1965)

Les circuits intégrés (1965 – 1980)

Les ordinateurs personnels (1980 – ?)

# Langage machine I

- ▶ Les circuits électroniques de chaque ordinateur ne peuvent reconnaître et exécuter directement qu'un nombre très limité d'instructions.

**Exemple :**

- ▶ Additionner deux nombres ;
- ▶ Vérifier si un nombre est égal à zéro ;
- ▶ Changer d'emplacement une donnée, dans la mémoire.

- ▶ L'ensemble des instructions exécutables **directement** par un ordinateur forme un langage appelé **langage machine**.

▶ [Wikipedia](#)

## Remarque

Lorsqu'on conçoit un nouvel ordinateur, il est nécessaire de choisir l'ensemble des instructions qui formeront son langage machine. En général, on cherche à ce que ces instructions soient les plus simples possibles, compte tenu des performances attendues de la machine, **afin de réduire la complexité**, et donc le coût, des circuits électroniques nécessaires.

# Langage machine II

- ▶ Caractéristiques du langage machine qui le rendent pénible et fastidieux à utiliser :
  - ▶ Il est **numérique**. La séquence 249 75 42 pourrait par exemple signifier « Stocker le caractère K à l'adresse 42, dans la mémoire » ;
  - ▶ Les opérations de base sont tellement limitées qu'il faut en combiner un très grand nombre pour résoudre le moindre problème ;
  - ▶ Deux modèles différents d'ordinateurs n'ont pas le même langage machine.

Pour résoudre ce problème, on définit un nouveau jeu d'instructions, plus pratique à utiliser pour le programmeur.

On obtient ainsi de nouveaux langages dits de « haut niveau. »<sup>1 2</sup>

▶ Wikipedia

---

1. La vision de l'architecture d'un ordinateur présentée ici est extrêmement simpliste.

2. « de plus haut niveau » fait référence au modèle en couches de l'architecture d'un ordinateur : plus on s'élève, plus on s'éloigne du langage machine.

## Langage machine — exemple

Instructions en langage machine (les 0 et les 1 ont été remplacés par une écriture hexadécimale, 4 fois plus condensée) indiquant à l'ordinateur d'écrire « Hello World ! » à l'écran.

...

```
48 83 ec 08 e8 73 00 00 00 e8 02 01 00 00 e8 ed
01 00 00 48 83 c4 08 c3 ff 35 ba 04 20 00 ff 25
bc 04 20 00 0f 1f 40 00 ff 25 ba 04 20 00 68 00
00 00 00 e9 e0 ff ff ff 25 b2 04 20 00 68 01
31 ed 49 89 d1 5e 48 89 e2 48 83 e4 f0 50 54 49
c7 c0 80 05 40 00 48 c7 c1 f0 04 40 00 48 c7 c7
eb b0 06 60 00 48 c1 fb 03 48 83 eb 01 48 39 d8
73 24 66 0f 1f 44 00 00 48 83 c0 01 48 89 05 25
48 89 6c 24 d8 4c 89 64 24 e0 48 8d 2d 9b 01 20
00 4c 8d 25 94 01 20 00 4c 89 6c 24 e8 4c 89 74
24 f0 4c 89 7c 24 f8 48 89 5c 24 d0 48 83 ec 38
4c 29 e5 41 89 fd 49 89 f6 48 c1 fd 03 49 89 d7
```

...

# Langages d'assemblage — exemple

```
1      .file    "test.c"
2      .section   .rodata
3.LC0:
4      .string  "Hello World!\n"
5      .text
6.globl main
7      .type    main, @function
8main:
9.LFB0:
10     .cfi_startproc
11     pushq   %rbp
12     .cfi_def_cfa_offset 16
13     .cfi_offset 6, -16
14     movq    %rsp, %rbp
```

```
1      .cfi_def_cfa_register 6
2      movl    $13, %edx
3      movl    $.LC0, %esi
4      movl    $1, %edi
5      call    write
6      movl    $0, %eax
7      leave
8      .cfi_def_cfa 7, 8
9      ret
10     .cfi_endproc
11.LFE0:
12     .size    main, .-main
13     .ident   "GCC: (GNU)"
14     .section .note.GNU-stack,
15                 "",@progbits
```

# Langages de haut niveau — exemples en C et python

## En C

```
1 #include<stdio.h>
2
3 int main() {
4     printf("hello, world");
5     return 0;
6 }
```

## En python

```
1 #!/bin/env python
2 # -*- coding: utf8 -*-
3
4 print("hello, world")
```

## Remarque

Les langages de « haut niveau » ne sont pas numériques mais se rapprochent des langues parlées par les humains.

# Plan

## Langages et ordinateurs

Vocabulaire

Langage machine

Traduction, interprétation

Matériel et logiciel

## Quelques étapes de l'évolution de l'architecture des ordinateurs

Les calculateurs mécaniques (1642 – 1945)

Les tubes à vide (1945 – 1955)

Les transistors (1955 – 1965)

Les circuits intégrés (1965 – 1980)

Les ordinateurs personnels (1980 – ?)

# Traduction, interprétation

## À garder à l'esprit

Un ordinateur ne peut exécuter que des instructions écrites dans son langage machine.

Lorsqu'un programme est écrit dans un langage de « haut niveau », il est nécessaire de le transformer en langage machine. Deux techniques existent :

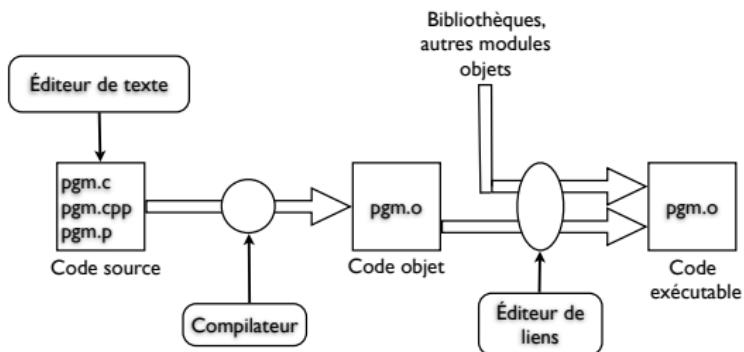
- ▶ la traduction.

Le programme qui traduit un langage de « haut niveau » s'appelle un compilateur.

- ▶ l'interprétation

Le programme qui interprète un langage s'appelle ...un interpréteur.

# Traduction I



1. Un compilateur (associé à un éditeur de liens) transforme le code source écrit en langage évolué en un fichier contenant du code exécutable formé de codes numériques propres à la machine. **Le code source est analysé dans son ensemble avant cette traduction.**
2. Le programme peut-être lancé à tout moment, une fois la compilation réalisée.

# Traduction II

## Avantages :

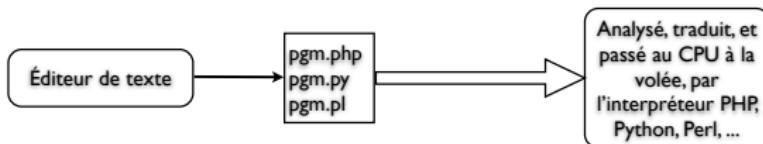
- ▶ Le code engendré est directement exécuté par la machine physique.
- ▶ Le compilateur peut optimiser le code lors de la phase de compilation.
- ▶ Le compilateur peut effectuer un certain nombre de vérifications lors de la phase de compilation (en particulier la cohérence des **types**).
- ▶ Le programme est plus « portable » que dans le cas du langage machine : en cas d'utilisation d'une nouvelle architecture, il « suffit » de compiler à nouveau le code source.

# Traduction III

## Inconvénients :

- ▶ Il n'est pas toujours facile de relier une erreur d'exécution au code source.
- ▶ Toute modification du code source impose de compiler à nouveau le programme (processus qui peut s'avérer être très long).
- ▶ Toute utilisation sur une nouvelle architecture (structure de l'ordinateur et système d'exploitation) impose de compiler à nouveau le programme.

# Interprétation I



Un interpréteur **analyse** le code source d'un programme, le **traduit** en langage machine et fait exécuter le code par l'ordinateur.

La grande différence avec un compilateur est qu'il effectue ces tâches **au fur et à mesure** :

1. Il lit une instruction et l'analyse.
2. Il traduit l'instruction en langage machine (si l'analyse n'a révélé aucune erreur).
3. Il fait exécuter le code machine.

# Interprétation II

## Avantages :

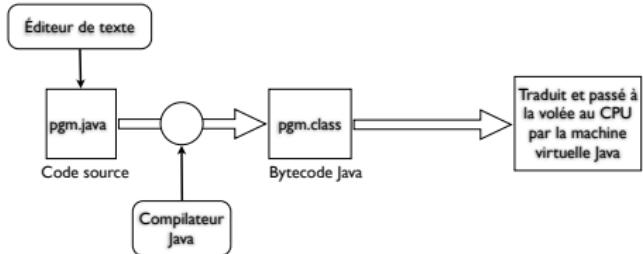
- ▶ Le programme peut fonctionner sans modification du code source sur plusieurs architectures.
- ▶ Le développement d'un logiciel prend généralement beaucoup moins de temps (on évite la phase de compilation).
- ▶ Le lien entre instruction et exécution étant plus direct, il peut être plus facile de relier une erreur d'exécution au texte source.
- ▶ Il est généralement plus facile « d'étudier » le fonctionnement du programme.

# Interprétation III

## Inconvénients :

- ▶ L'interpréteur simulant le fonctionnement de l'ordinateur, les programmes sont généralement beaucoup moins rapides (x10 à x100) que les programmes compilés.
- ▶ Il est nécessaire d'installer l'interpréteur adapté à la plateforme pour pouvoir exécuter le programme (sous réserve qu'il existe).

# Exécution mixte



La programmation en Java utilise les deux techniques précédentes :

1. Un fichier source en Java est tout d'abord compilé, mais le compilateur, au lieu de générer des instructions directement exécutables par le processeur utilisé, produit un code (**Bytecode**) spécifique à une **machine virtuelle**<sup>3</sup>
2. Un interpréteur (jre pour Java Runtime Environment), spécifique à chaque architecture, interprète par la suite ce code en langage machine<sup>4</sup>.

3. C'est le langage machine de cette machine virtuelle.

4. Le Bytecode est proche du langage d'assemblage, cette interprétation est relativement rapide.

# Plan

## Langages et ordinateurs

Vocabulaire

Langage machine

Traduction, interprétation

Matériel et logiciel

## Quelques étapes de l'évolution de l'architecture des ordinateurs

Les calculateurs mécaniques (1642 – 1945)

Les tubes à vide (1945 – 1955)

Les transistors (1955 – 1965)

Les circuits intégrés (1965 – 1980)

Les ordinateurs personnels (1980 – ?)

# Matériel et logiciel

## À méditer...

- ▶ Matériel et logiciel sont logiquement équivalents.
- ▶ Ce qui est matériel aujourd’hui pourra être logiciel demain, et réciproquement.
- ▶ Ce qui est matériel chez l’un peut être logiciel chez l’autre.

# Plan

## Langages et ordinateurs

Vocabulaire

Langage machine

Traduction, interprétation

Matériel et logiciel

## Quelques étapes de l'évolution de l'architecture des ordinateurs

Les calculateurs mécaniques (1642 – 1945)

Les tubes à vide (1945 – 1955)

Les transistors (1955 – 1965)

Les circuits intégrés (1965 – 1980)

Les ordinateurs personnels (1980 – ?)

# Blaise Pascal (1646 – 1716)



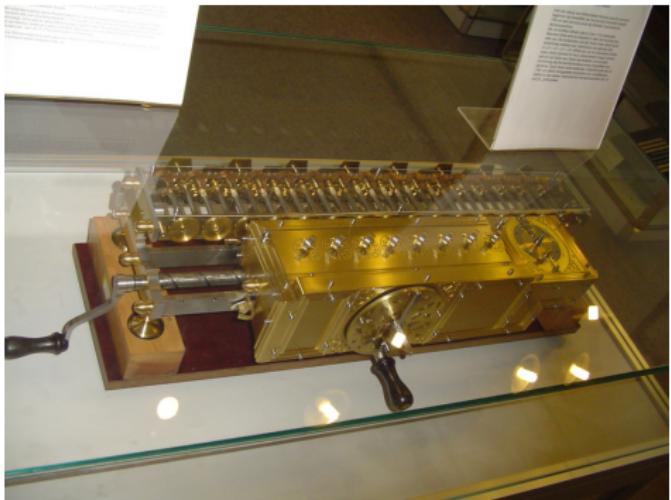
- ▶ Agé de 19 ans, Pascal<sup>5</sup> a construit une machine entièrement mécanique, à base d'engrenages et actionnée à la main, destinée à aider son père, intendant des finances en haute Normandie.
- ▶ Cette machine effectuait les additions et les soustractions.

► [Wikipedia](#)

---

5. C'est en son honneur que Niklaus Wirth a appelé Pascal un des langages de programmation qu'il a inventé.

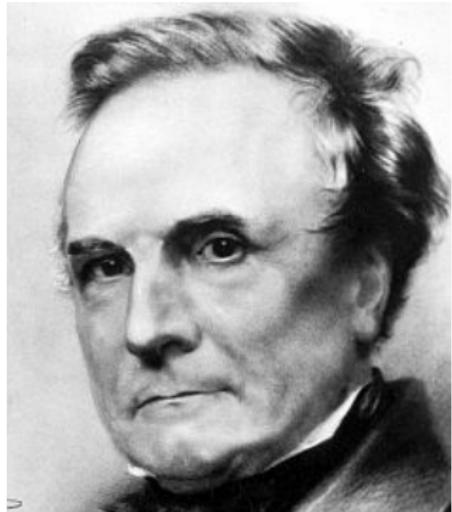
# Gottfried Wilhelm Leibniz (1623 – 1662)



- ▶ Leibniz a ajouté la multiplication et la division à la machine de Pascal.

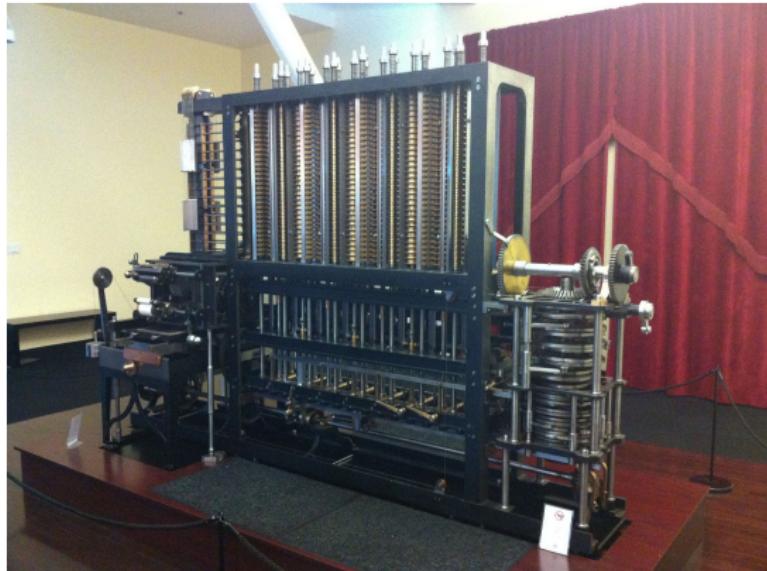
► Wikipedia

# Charles Babbage (1792 – 1871) I



► Wikipedia

# Charles Babbage (1792 – 1871) II



# Charles Babbage (1792 – 1871) III

## Machine à différences

- ▶ Cette machine effectuait les additions et les soustractions et était destinée à établir des tables numériques pour la navigation en mer.
- ▶ Cette machine n'exécutait qu'un seul programme consistant à « calculer les polynômes en utilisant le calcul différentiel ! ».
- ▶ **Originalité** : pour retourner les résultats de ses calculs la machine gravait un plateau de cuivre à l'aide d'un timbre en acier, préfigurant ainsi les médias non réinscriptibles comme les cartes perforées ou les premiers disques optiques.

# Charles Babbage (1792 – 1871) IV

## Machine analytique

- ▶ Cette machine comportait quatre parties :
  - ▶ le magasin (la mémoire)
  - ▶ le moulin (l'unité de calcul)
  - ▶ l'entrée (le lecteur de cartes perforées<sup>6</sup>)
  - ▶ la sortie (perforation ou impression)
- ▶ Le magasin disposait de 1 000 mots de 50 chiffres décimaux que l'on pouvait utiliser pour stocker variables et résultats.
- ▶ Le moulin prenait les opérandes provenant du magasin, en faisait l'addition, la soustraction, la multiplication ou la division et renvoyait le résultat vers le magasin<sup>7</sup>.
- ▶ Certaines instructions pouvaient commander de faire le test d'un nombre et provoquer ainsi un branchement conditionnel suivant que ce nombre était positif ou négatif !

---

6. Babbage a eu l'idée d'utiliser des cartes du métier Jacquard [▶ Wikipedia](#).

7. Rappel : tout était mécanique !

## Charles Babbage (1815 – 1852) V

Babbage ne put jamais mettre vraiment au point son matériel. La technologie du XIX<sup>ème</sup> siècle n'était pas capable de fournir les milliers de pièces, d'engrenages et de roues dentées avec une qualité d'usinage suffisamment bonne.

Babbage est considéré comme l'un des principaux précurseurs de l'informatique.

# Ada Lovelace (1792 – 1871)

- ▶ La machine analytique étant programmable à l'aide d'un langage d'assemblage très simple, Augusta Ada Lovelace, fille du poète Lord Byron, mathématicienne et collaboratrice de Babbage, eut l'idée d'écrire le premier programme informatique : le calcul à la machine des nombres de Bernouilli.



Ada Lovelace est considérée par les informaticiens comme la première programmeuse de l'histoire.

C'est en son honneur que le ministère américain de la Défense a appelé Ada un langage de programmation conçu par une équipe de Bull.

▶ Wikipedia

# Plan

## Langages et ordinateurs

Vocabulaire

Langage machine

Traduction, interprétation

Matériel et logiciel

## Quelques étapes de l'évolution de l'architecture des ordinateurs

Les calculateurs mécaniques (1642 – 1945)

**Les tubes à vide (1945 – 1955)**

Les transistors (1955 – 1965)

Les circuits intégrés (1965 – 1980)

Les ordinateurs personnels (1980 – ?)

# Alan Turing (1912 – 1954)

- ▶ Il est l'auteur, en 1936, d'un article de logique mathématique qui est devenu plus tard un texte fondateur de la science informatique.
- ▶ Il a imaginé une machine théorique, la « machine de Turing », afin de donner une définition précise au concept d'algorithme.
- ▶ Durant la Seconde Guerre mondiale, il a joué un rôle majeur dans les recherches sur les cryptographies générées par la machine **Enigma**, utilisée par les nazis. Ces recherches conduirent à la construction du premier ordinateur électronique : **COLOSSUS** (resté secret militaire pendant 30 ans) (1943).

▶ Wikipedia

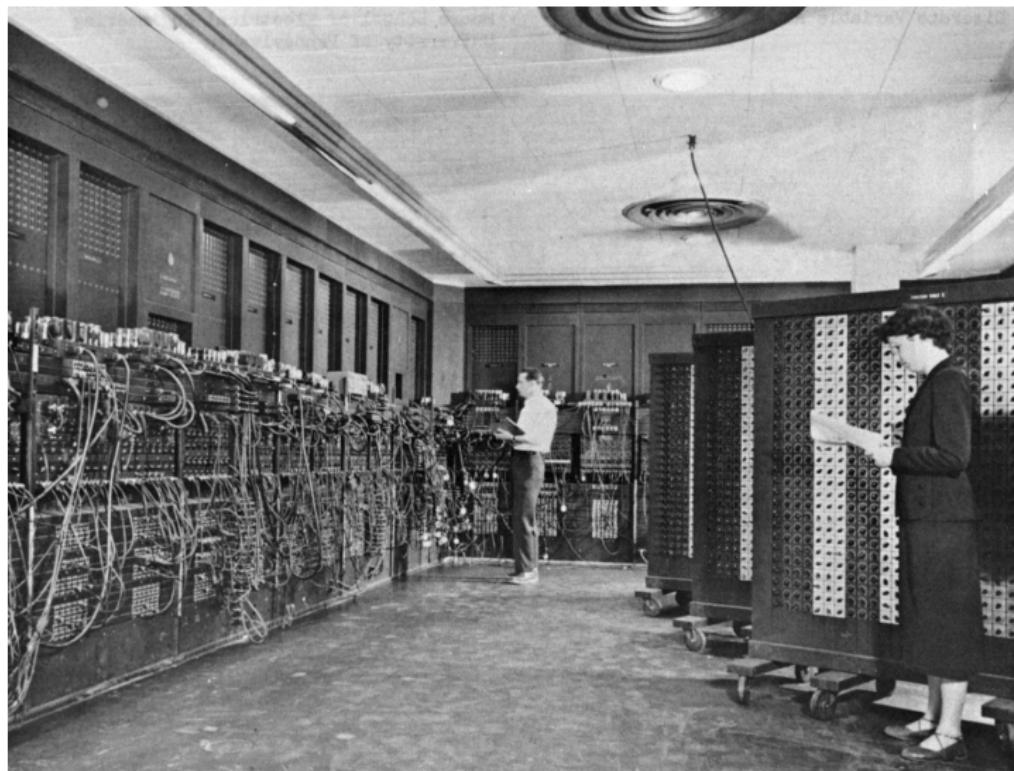


# ENIAC (Electronic Numerical Integrator Analyser and Computer) I

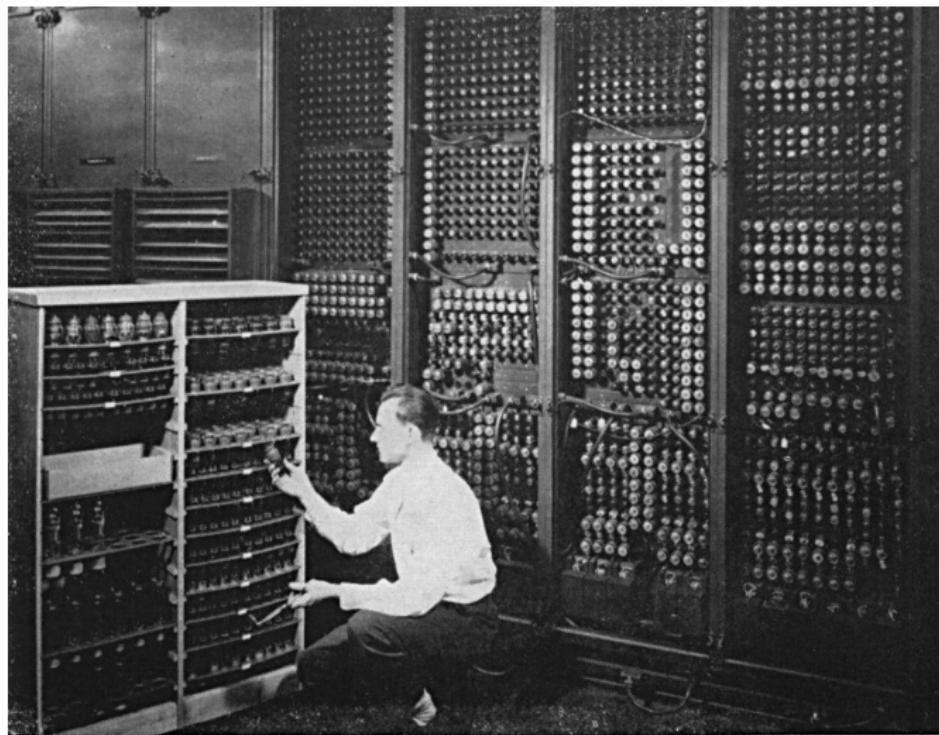
- ▶ L'ENIAC fut le premier ordinateur entièrement électronique construit pour être Turing-complet (1946). Il pouvait être reprogrammé pour résoudre, en principe, tous les problèmes calculatoires.
- ▶ L'ENIAC fut construit par John William Mauchly, professeur de physique, aidé par un de ses étudiants en thèse, J. Presper Eckert.
- ▶ L'ENIAC machine comportait 18 000 tubes à vide et 1 500 relais. Elle pesait 30 tonnes et consommait 140 kW.
- ▶ L'ENIAC disposait de 20 registres de 10 **chiffres décimaux** et on la programmait en manipulant quelques 6 000 commutateurs interconnectés par une forêt de câbles.

► [Wikipedia](#)

# ENIAC (Electronic Numerical Integrator Analyser and Computer) II



# ENIAC (Electronic Numerical Integrator Analyser and Computer) III



Replacing a bad tube meant checking among ENIAC's 19,000 possibilities.

# ENIAC (Electronic Numerical Integrator Analyser and Computer) IV

Moyen employé	Vitesse de multiplication de nombres de 10 chiffres	Calcul d'une trajectoire d'une table de tir
Homme à la main, ou machine de Babbage	5 min	2,6 jours
Homme avec calculateur de bureau	10 à 15 secondes	12 heures
Harvard Mark I (électromécanique)	3 secondes	2 heures
Model 5 (électromécanique)	2 secondes	40 minutes
Analyseur différentiel (analogique)	1 seconde	20 minutes
Harvard Mark II (électromécanique)	0,4 s	15 minutes
ENIAC (électronique)	0,001 s	3 secondes

# John von Neumann (1903 – 1957)

- ▶ Il comprit que la programmation des ordinateurs dotés d'un très grand nombre de commutateurs était trop lente, fastidieuse et rigide.
- ▶ Il proposa de représenter le programme sous forme numérique et de le stocker dans la mémoire, au même titre que les données.
- ▶ Il proposa de remplacer l'arithmétique décimale série de l'ENIAC, dans laquelle chaque chiffre est représenté par dix tubes à vide (1 allumé et 9 éteints) par une arithmétique binaire parallèle.

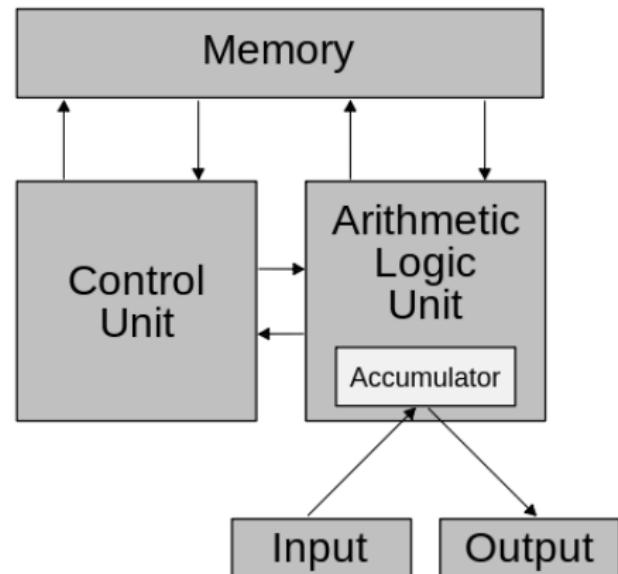


Von Neumann a donné son nom à l'architecture de von Neumann utilisée dans la quasi-totalité des ordinateurs modernes.

# Machine de von Neumann

EDSAC : premier ordinateur à programme enregistré

Mémoire : 4096 mots de 40 bits, donc deux instructions de 20 bits ou un entier signé de 39 bits.



# Grace Hopper (1906 – 1992)



À partir de 1957, Grace Hopper défend l'idée qu'un programme devrait pouvoir être écrit dans un langage proche de l'anglais plutôt que d'être calqué sur le langage machine, comme l'assembleur. De cette idée naît le langage COBOL en 1959.

[► Wikipedia](#)

# Bug informatique

Expression popularisée suite à la panne d'un ordinateur Mark II due à un papillon de nuit, de l'anglais moth, pris dans un relais.

L'insecte, « bug » en anglais, fut enlevé avec soin et placé dans le journal de bord avec la mention « **first actual case of bug being found** » (premier cas réel de découverte d'insecte).

9/9	
0800	Auton started stoped - auton ✓
1000	13° C 03W MP-MC 033 PRO-Z convol Relays 6-2 in 033 failed special speed test in relay Relays changed
1100	Started Cosine Tape (Sine check)
1525	Started Multi Adder Test.
1545	 Relay #70 Panel F (moth) in relay.
1600	First actual case of bug being found. auton startd.
1700	closed down.

# Plan

## Langages et ordinateurs

Vocabulaire

Langage machine

Traduction, interprétation

Matériel et logiciel

## Quelques étapes de l'évolution de l'architecture des ordinateurs

Les calculateurs mécaniques (1642 – 1945)

Les tubes à vide (1945 – 1955)

Les transistors (1955 – 1965)

Les circuits intégrés (1965 – 1980)

Les ordinateurs personnels (1980 – ?)

# Plan

## Langages et ordinateurs

Vocabulaire

Langage machine

Traduction, interprétation

Matériel et logiciel

## Quelques étapes de l'évolution de l'architecture des ordinateurs

Les calculateurs mécaniques (1642 – 1945)

Les tubes à vide (1945 – 1955)

Les transistors (1955 – 1965)

Les circuits intégrés (1965 – 1980)

Les ordinateurs personnels (1980 – ?)

# Plan

## Langages et ordinateurs

Vocabulaire

Langage machine

Traduction, interprétation

Matériel et logiciel

## Quelques étapes de l'évolution de l'architecture des ordinateurs

Les calculateurs mécaniques (1642 – 1945)

Les tubes à vide (1945 – 1955)

Les transistors (1955 – 1965)

Les circuits intégrés (1965 – 1980)

Les ordinateurs personnels (1980 – ?)