Langage C

Cours 1 - Introduction

#include<stdio.h>

INSTITUT d'OPTIQUE GRADUATE SCHOOL ParisTech
LEnsE Laboratoire d'Enseignement Expérimental

Sylvie LEBRUN / Charles BOURASSIN / Julien VILLEMEJANE

L'ÉQUIPE PÉDAGOGIQUE

RESPONSABLE : SYLVIE LEBRUN

INTERVENANTS PERMANENTS

- Julien VILLEMEJANE
- Xavier DELEN

INTERVENANTS

- Mathys Thiers (doc 1A)
- Valentin Guichard (doc 1A)
- Amal ZIDI (doc 2A)
- Marc CHAMMAS (doc 3A)
- George Crisan (doc 1A)
- Maha BOUHADIDA (ingé.)

OBJECTIFS

PROGRAMMATION / LANGAGE C

- Introduction au langage C par l'exemple
- Langage de base
- Réalisation d'un mini-projet autour des images

SYSTÈME À MICROPROCESSEUR

- Fonctionnement d'un système à microprocesseur
- Programmation bas niveau
- Gestion de la mémoire
- Vers des langages plus évolués

DEROULEMENT

SÉANCE 1

- Bases, prise en main de Code:Blocks

SÉANCES 2-3-4-5

- **Notions de base**
  - Tableaux 1D
  - Fonctions
  - Gestion de fichiers
  - Chaînes de caractères

SÉANCE 12

- **Présentation orale**

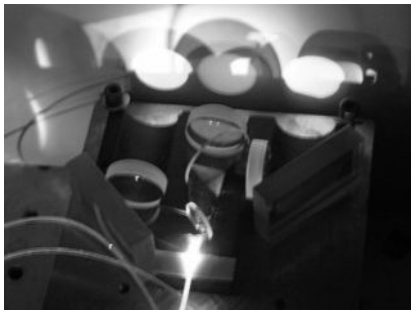
SÉANCES 6-7

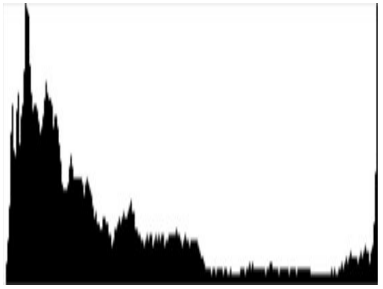
- **Notions avancées**
  - Pointeurs
  - Allocation dynamique
- **Presentation mini-projet**

SÉANCES 8-9-10-11

- **Projet en groupe**

A PARTIR DE LA SÉANCE 8





MINI-PROJET / TRAITEMENT D'IMAGES

- **Travail en groupe**

OBJECTIF PRINCIPAL

- **CALCULER L'HISTOGRAMME**

OBJECTIFS INTERMÉDIAIRES

- Ouvrir le fichier ASCII en lecture
- Afficher l'en-tête du fichier
- Ré-afficher l'image en console
- Créer un tableau **histogramme**

- **AFFICHER L'HISTOGRAMME**

- Créer un fichier PGM
- Ecrire l'en-tête du fichier
- Créer une image DAMIER
  - N x N cases de K pixels chacune
- Créer une image à partir d'un tableau 1D (type histogramme)

EVALUATION

PROJET

- **Présentation**
- **Evaluation en séance**

CRITÈRES

- Lisibilité du code : commentaires, indentation, variables...
- Explications claires
- Qualité du code : algorithme efficace
- Réutilisation possible du code : modularité

EXAMEN écrit Mercredi 11 janvier 2023

- **QCM**
- **Problème**

GENERALITES

CRÉÉ EN 1970

- Dennis Ritchie
- Brian Kernighan

UNIX®

MULTIPLATEFORME

- Windows, Mac, Linux
- Compilateurs disponibles pour de nombreux processeurs et microcontrôleurs

LANGAGE INTERMÉDIAIRE

- **Haut niveau :** compréhensible par l'être humain
- **Bas niveau :** suffisamment proche de la machine (fonction bas niveau)

POUR QUOI FAIRE ?

MANIPULATION DE DONNÉES

- Stockées dans des espaces mémoires
- Typées / Adressables par octet (1o = 8 bits)

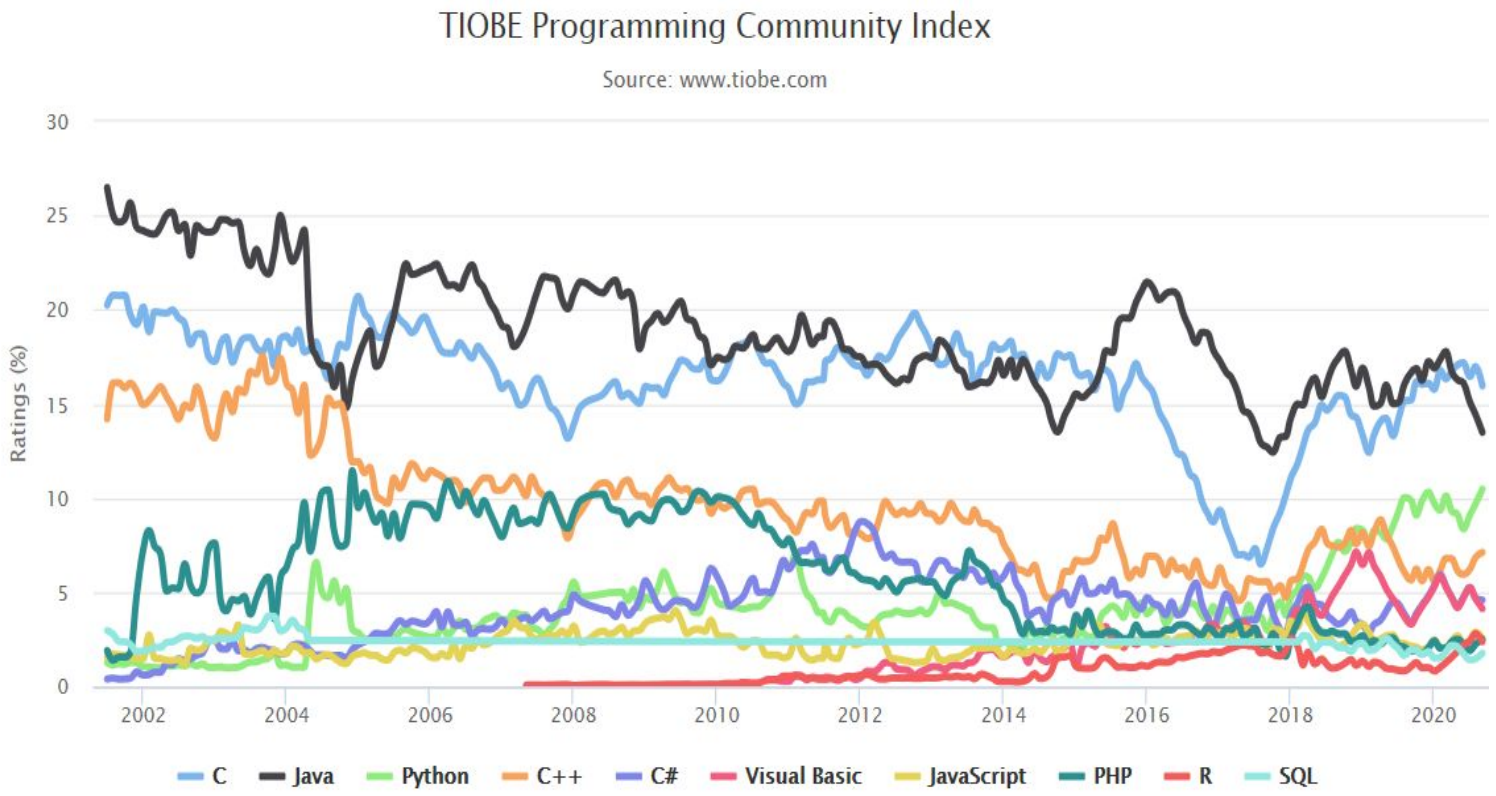
INTÉRÊTS

- Rapidité d'exécution / Code compilé
- Protection intellectuelle du code source

UTILISATION

- Traitement de données
- Flux de données importants (vidéo...)
- Systèmes embarqués / grande réactivité

UN LANGAGE RÉPANDU



DÉVELOPPEMENT LOGICIEL

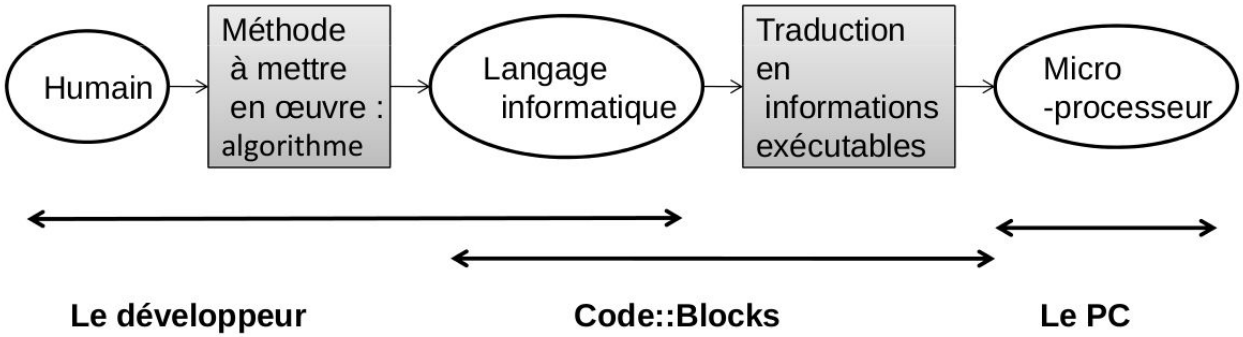
EN 2 PHASES

RÉSOLUTION

- Compréhension du problème
- Ébauche de solution
- Liste des variables nécessaires
- Solution détaillée / Algorithme
- Essais pas à pas avec des données

MISE EN OEUVRE

- Traduction dans un langage
- Ecriture du programme
- Correction syntaxique
- Essais avec des données



Sylvie LEBRUN / Charles BOURASSIN / Julien VILLEMEJANE

LANGAGE COMPILÉ / PROCEDURAL

CODE SOURCE

- Syntaxe particulière / Typage de données
- Modularité / Réutilisation grâce à des fonctions/procédures / Bibliothèques existantes

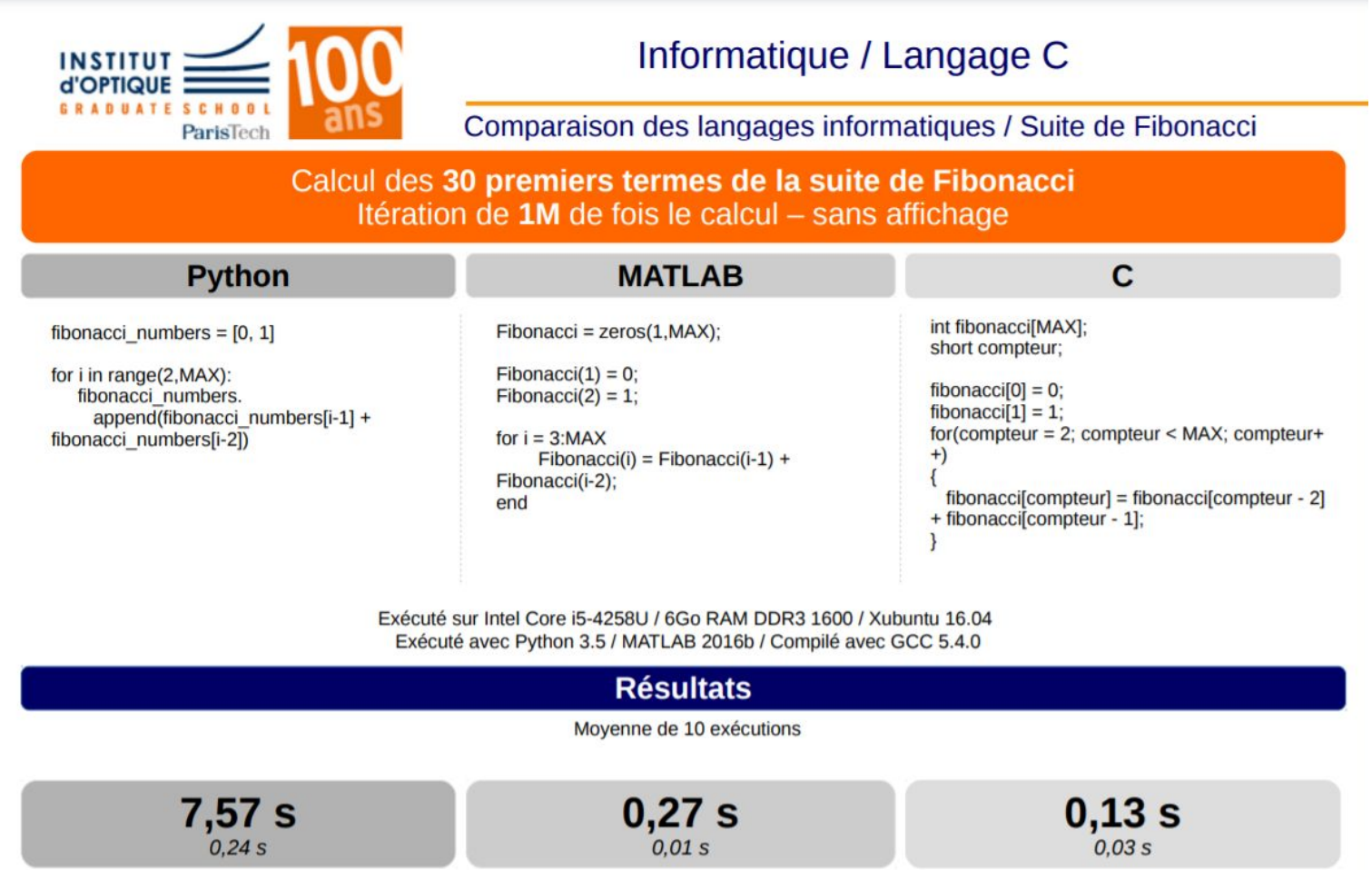
COMPILATION

- Nécessite un compilateur par type de calculateur
- Difficilement portable / C Standardisé
- Protection intellectuelle du code source

EXÉCUTION

- Rapidité d'exécution

Comparaison des langages informatiques / Suite de Fibonacci



Sylvie LEBRUN / Charles BOURASSIN / Julien VILLEMEJANE

DURANT LES SÉANCES...

Ecriture du code → **Compilation** → **Exécution**

```
int main()
{
    int j;
    for (j=0;j<10;j++) ...
}
```

Mémoire vive