III

Initiation à l'Ingénierie Informatique

PLATYPUS

Introduction du sujet

Programmer

Note importante

Les informations contenues dans ce support sont données sous réserve de révisions ultérieures.

Plan de la séance

- Programmer
- La machine, le programmeur, l'algorithme, le programme
- Programmer, compiler, exécuter un programme
- Ce que vous allez devoir apprendre pour devenir programmeur
- → Bascule sur série « comment ça va se passer cette année »

Piloter une machine

- Piloter = contrôler le comportement d'une machine
  - Lui donner des instructions (contrôles)
  - Lui fournir des informations (données)
  - Pour lui déléguer la réalisation de ces instructions
- Exemples évidents
  - Véhicules
  - Calculatrices
  - Machines industrielles
  - Ordinateurs personnels

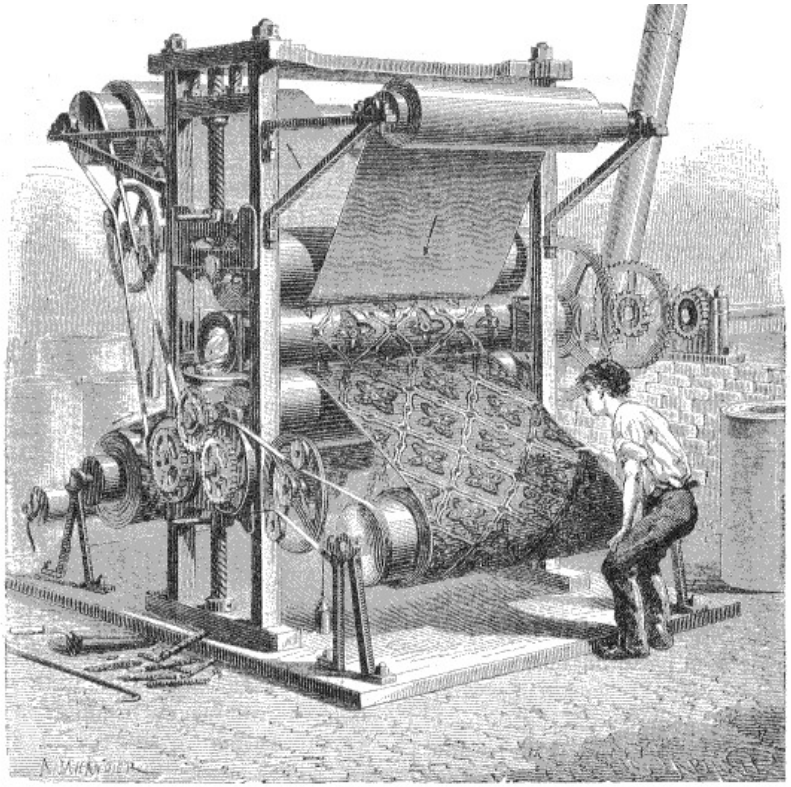
Programmer une machine

- Programmer = pré-définir un scénario de contrôle
  - Définir une séquence d'instructions (le programme)
  - A laquelle peut se référer la machine
  - Pour enchaîner les opérations qui lui sont déléguées..
  - En d'autre termes : pour lui déléguer une opération complexe
- Exemples évidents
  - Définir le comportement d'un pilote automatique
  - Programmer une calculatrice pour effectuer un calcul complexe
  - Programmer un robot industriel pour le spécialiser sur une tâche
  - Programmer un ordinateur pour qu'il délivre des applications / services

La Pascaline

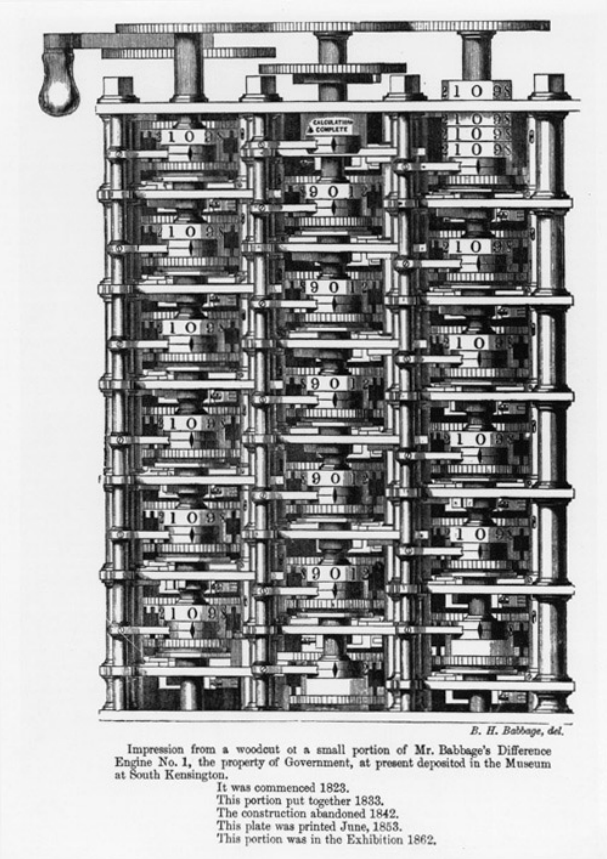
- Pascaline
  - La première calculatrice
  - Inventée par Pascal, à 22 ans
- But
  - Aider son père comptable..
  - .. à faire des calculs
- Contexte
  - 1645
  - Richelieu, etc
- Pas programmable
  - Mais opérant sur commande

Le métier Jacquard



- Métier à tisser
  - De la machine à usage unique..
  - .. à la machine programmable
- But
  - Moduler les motifs
  - Industrialiser la production
- Contexte
  - Lyon, ville de la soie, 1801
  - Cause de la révolte des Canuts
- Ancêtre du robot
  - Automatisation : crochets guidés
  - Programme : carte perforée

La machine différentielle de Babbage

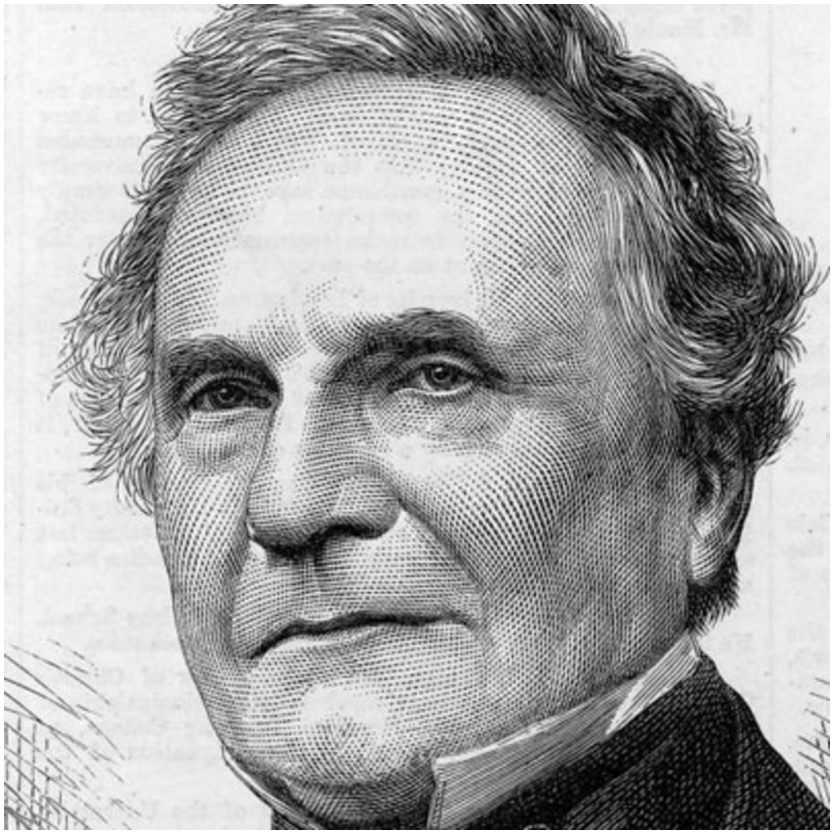


B. H. Babbage, del.

Impression from a woodcut of a small portion of Mr. Babbage's Difference Engine No. I, the property of Government, at present deposited in the Museum at South Kensington.
It was commenced 1823.
This portion put together 1833.
The construction abandoned 1842.
This plate was printed June, 1853.
This portion was in the Exhibition 1862.

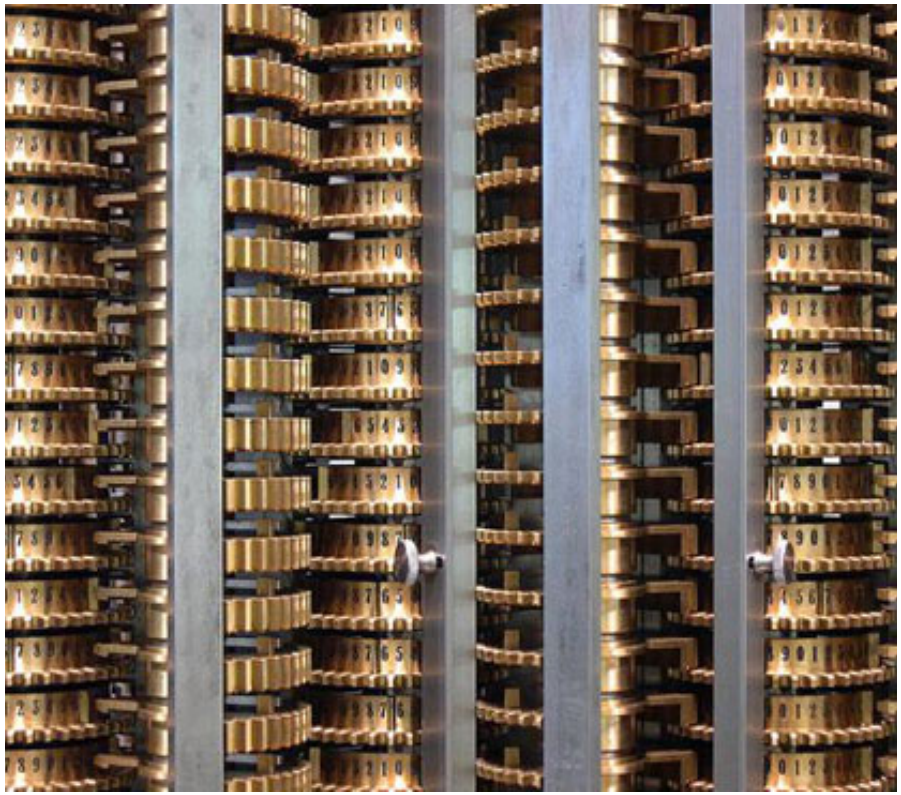
- Machine différentielle
  - Une nouvelle Pascaline
- But
  - Etablir des tables de calcul..
  - .. sans erreurs
- Contexte
  - Charles Babbage : mathématicien
  - Casseur du code de Vigenère
  - Commande du gouv. Britanique
  - Réalisée autour de 1830

La machine analytique de Babbage



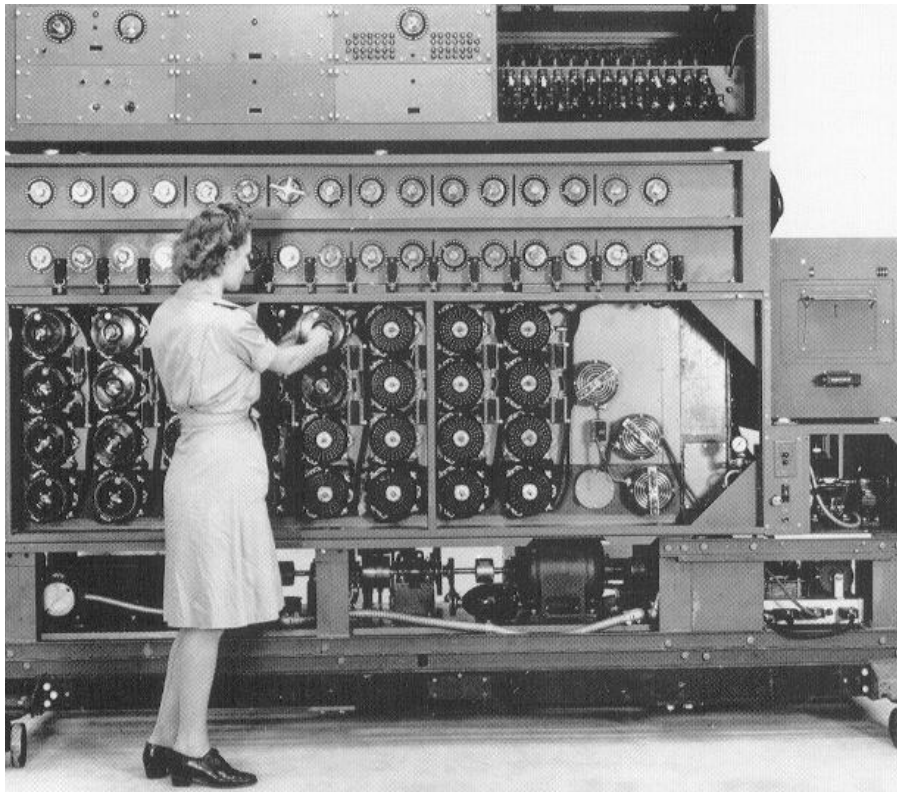
- Machine analytique
  - 1er ordinateur
- But
  - Intégrer le principe de Jacquard..
  - .. à la machine différentielle..
  - .. pour la rendre programmable
- Contexte
  - Conçue à partir de 1834
  - Le gouv. Britannique ne suit pas
  - Réalisation amorcée par son fils seulement en 1900
  - Le premier ordinateur aurait pu exister avec un siècle d'avance

Architecture de la machine de Babbage



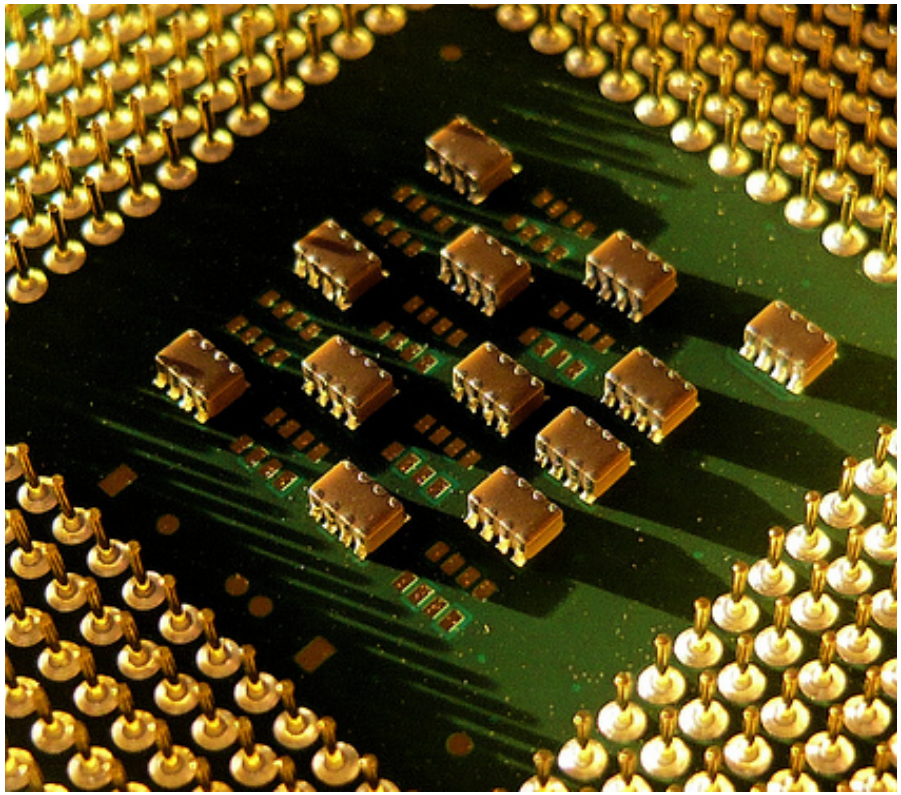
- Moulin
  - En charge d'opérer les calculs
  - Ancêtre du micro-processeur
- Magasin
  - En charge de stocker les données
  - Ancêtre de la mémoire
- Impression (sortie)
  - En charge de fournir les résultats
  - Ancêtre de l'écran
- Programmation : 2 cartes
  - Carte d'instructions (programme)
  - Carte de données (entrée)

Bombe électromécanique de Turing



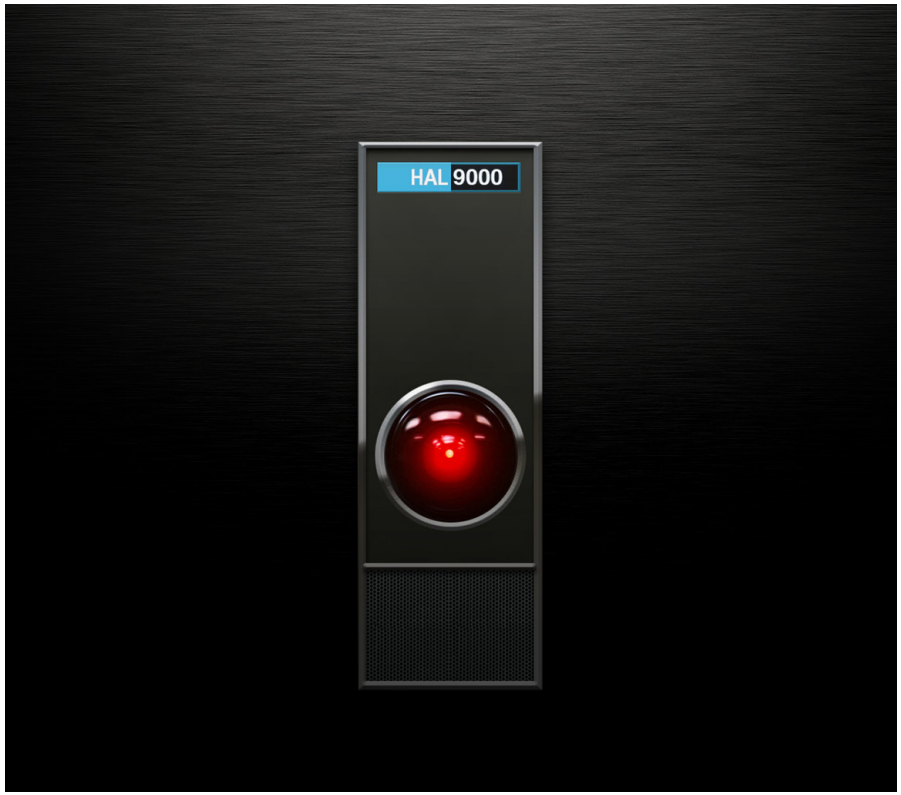
- Bombe de Tunring
  - Machine à cryptanalyse
- Contexte
  - Seconde guerre mondiale
  - Alan Turing : logicien
- But
  - Casser le code Enigma des nazis
- Enjeu
  - La rapidité (performance)
- Pas vraiment programmable

Ere moderne



- Ere moderne
  - Ouverte dans les années 60
  - Le calculateur programmable devient un bien de consommation
- Miniaturisation
  - Micro-processeur
  - Ordinateur personnel (PC)
  - Loi de Moore (Intel)
- Démocratisation
  - Consommation de masse
  - Processeurs & programmes partout
  - De nombreux langages pour les programmer

Hal 9000



- Hal 900
  - Machine intelligente
- Contexte
  - 2010 l'Odyssée de l'espace
  - Conçue par le Pr Chandra
- But
  - Plus besoin de la programmer
  - Elle s'adapte aux circonstances
- Plus tant de la science-fiction
  - Nous en sommes à l'ère du :
    - Big data
    - et du Machine learning

Ada Lovelace

- Ada Lovelace
  - A pretty name
  - Nièce de Lord Byron
  - Le premier programmeur
  - A conçu les cartes de la machine de Babbage
  - A donné son nom à un langage de programmation
  - Et on en reste pourtant à seulement 25% d'ingénieurs informatique féminins

Le programmeur moderne

Notion d'algorithme

- Qu'est ce qu'un algorithme ?
  - Une stratégie de résolution de problèmes
- Son principe : effectuer une opération complexe
  - Analyser un problème en entrée
  - Pour produire une solution en sortie
- Sa structure
  - Représentations du problème et de la solution : structures des données
  - Méthode systématique pour analyser et produire : schéma de traitement
- Equifinalité
  - Pour un résoudre un problème, il peut y avoir plusieurs stratégies
  - Elles ne se valent pas toutes => enjeu fondamental : la performance !
  - On en reparle de manière plus approfondie l'année prochaine

Algorithme et langage algorithmique

Algorithm 4: L'algorithme de Bellman-Ford

```
début
    Initialise-source(G, s);
    pour i ← 1to|V| − 1 faire
        pour chaque arc (u, v) ∈ E faire
            RELAXER(u, v, w);
        fin
    fin
    pour chaque arc (u, v) ∈ E faire
        si d[v] < d[u] + w(u, v) alors
            return FALSE ;
        fin
    fin
    return TRUE ;
fin
```

- Langage algorithmique
  - Aussi appelé pseudo-code
  - Sert à formaliser un algorithme
- Algorithme
  - Comment résoudre un problème
  - Sans faire d'hypothèse..
  - .. sur la machine qui va opérer
  - objectif d'intelligibilité pour l'humain
- Programme exécutable (ou 'exécutable')
  - Traduction de l'algorithme..
  - .. dans une langue..
  - .. que sait interpréter la machine

Langage interprétable par la machine

- Langage machine
  - Interprété par le micro-processeur
  - Pas loin des cartes perforées
- Programmer en langage machine ?
  - Oui c'est possible, en théorie :
  - Il suffit d'éditer un fichier en binaire
  - Pensez à Ada et ses cartes perforées
  - Vous n'avez vraiment pas envie
- Langage de programmation
  - Intelligible par l'humain..
  - ...enfin, le geek, pas tous les humains...
  - mais pas directement par la machine

Compilé vs. interprété

- Comment faire le lien entre les deux ?
  - Le programme en langage geek (le langage de programmation)
  - Le programme en langage machine
- Deux possibilités
  - Compilation – traduction de l'un vers l'autre avant l'exécution
  - Interprétation – traduction de l'un vers l'autre pendant l'exécution
- Encore des programmes
  - Pour compiler, on utilise un programme spécialisé : un compilateur
  - Pour interpréter : un interpréteur
  - Solution intermédiaire : machine virtuelle (encore un programme)

Exemples, portabilité

- Compilé
  - Le C est compilé
  - Chaque processeur parle un langage machine différent des autres
  - Il faut recompiler le programme sur les différents types de machines
  - Ex. un exécutable pour PC n'est pas exécutable sur un MAC et vice et versa
- Interprété
  - Le JavaScript est interprété par le navigateur Web
  - Chrome, Firefox, IE, Opera parlent la même langue, le JavaScript
  - Le programme n'a pas donc pas besoin d'être compilé pour être exécuté
- Write Once, Run Anywhere ©
  - Le Java est compilé vers un langage machine virtuel, le bytecode
  - Sur chaque machine physique, on installe un interpréteur léger : la VM
  - L'exécutable bytecode est interprétable par la VM sur toutes les machines

Devenir programmeur

- Vous voilà sur les trace d'Ada Lovelace
- Ce qu'il va falloir apprendre à faire
  - A analyser des problèmes et concevoir des solutions (algorithmes)
  - A traduire ces solutions en langage JavaScript (programmes)
  - A exécuter ces traductions en langage machine (exécutables)
- Ca peut paraître simple, mais en pratique
  - Il faudra faire beaucoup d'aller et retours entre ces étapes..
  - .. Bugs, problèmes de performance, mise au point fonctionnelle, etc

Devenir programmeur

- Concevoir des algorithmes
  - Cela demande méthode, rigueur et entraînement
  - C'est ce que vous ferez en TD
- Traduire en JS et mettre au point votre programme
  - Cela demande de maîtriser une nouvelle langue
  - Avec son lexique, sa syntaxe, sa sémantique
  - C'est ce que vous ferez en Cours, TD et TP
  - Vous aurez un cours dédié HTML/CSS/JS avec Grégoire Puget
- Exécuter (interpréter avec un navigateur)
  - La console JS est votre amie
  - Elle vous dira tout ce qui ne va pas : à vous de comprendre ce qu'elle vous dit
  - Elle vous accompagnera pendant quelques nuits blanches

La boîte à outils du programmeur

- Un éditeur de texte
  - N'importe lequel fait l'affaire
  - Mais certains sont plus adaptés que d'autres
  - Par exemple Sublime Text
- Des outils de mise au point
  - Console JavaScript
  - Etc

N'oubliez pas de structurer votre démarche

En langage algorithmique

Algorithme 1 : $PGCD\,(a : entier, b : entier) : entier$

Données : a et b deux entiers naturels
Résultat : le $PGCD$ de a et b
début
  si $b = 0$ **alors retourner** a
  sinon retourner $PGCD\,(b, a \bmod b)$
fin

Traduction, par ex. en JS

```
function gcd(a, b)
{
   if (b == 0) return a;
   else return gcd(b, a % b);
   // ou // return b ? gcd(b, a % b) : a;
}
```

Elaborer
(c'est la partie créative)

Spécifier
(langage algorithmique)

Implémenter
(langage de programmation)