Algorithmique

- Les algorithmes dans l'histoire
- Des définitions d'un algorithme
- Algorithme et langages de programmation
- Les structures de bases de l'algorithmique
- Un premier algorithme
- AlgoBox
- Une galerie de « portraits »

1

Algorithmique

- **Les algorithmes ne sont pas nés avec l'informatique** :

  - ✓ L'algorithme d'Euclide pour calculer le PGCD de deux entiers est vieux de plus de 2000 ans !
  - ✓ Des descriptions précises d'algorithmes sont présents dans la Chine ancienne.
    (Par exemple, pour extraire des racines carrées à partir de divisions effectuées sur une « surface à calculer »)

- **L' origine du mot « algorithme »** est lié au nom du savant arabe du IX^{e} siècle : Al-Khwarizmi.

Ce savant arabe a publié plusieurs méthodes pour le calcul effectif de racines d'une équation du second degré et grâce à lui les chiffres arabes ont pu se diffuser en occident.

Algorithmique

- **Des définitions plus « modernes » d'un algorithme:**

  ✓ Une définition simple :
  « Un ensemble d'instructions pour résoudre un problème. »

  ✓ Une définition plus complète à partir de 5 propriétés : (selon Knuth)

  ❖ **Finitude** : Un algorithme doit toujours se terminer après un nombre fini d'étapes.

  ❖ **Précision** : Chaque étape d'un algorithme doit être définie précisément; les actions à transposer doivent être spécifiées rigoureusement et sans ambiguïté pour chaque cas.

  ❖ **Entrées** : Quantités, prises dans un ensemble d'objets spécifié, qui sont données à l'algorithme avant qu'il ne commence.

  ❖ **Sorties** : Quantités qui ont une relation spécifiée avec les entrées.

  ❖ **Rendement** : Toutes les opérations que l'algorithme doit accomplir doivent être suffisamment élémentaires pour pouvoir être en principe réalisées dans une durée finie par un homme utilisant du papier et un crayon.

des définitions d'un algorithme 3

Algorithmique

- **Algorithme et langage de programmation**

  ➢ Un algorithme est rédigé dans un pseudo-langage (en Français).

  Il peut ensuite être implémenté dans un langage de programmation donné.

  En quoi a-t-on besoin d'un langage spécial, distinct des langages de programmation compréhensibles par les ordinateurs ?

  Parce que l'algorithmique exprime les instructions résolvant un problème donné indépendamment des particularités de tel ou tel langage.
  Pour prendre une image, si un programme était une dissertation, l'algorithmique serait le plan, une fois mis de côté la rédaction et l'orthographe. Or, vous savez qu'il vaut mieux faire d'abord le plan et rédiger ensuite que l'inverse…

  Apprendre l'algorithmique, c'est apprendre à manier la structure logique d'un programme informatique.

Algorithmique

Les 3 étapes d'un algorithme

- Les entrées (ou la déclaration et la saisie des données)
- Le traitement
- Les sorties (ou l'affichage / l'impression des données transformées)

• **Les entrées**

Il s'agit de repérer les données nécessaires à la résolution du problème.
Ces données peuvent être numériques, ou sous forme de textes (on dit souvent chaines de caractères), ou de type logique (deux valeurs possibles, vrai ou faux), ou enfin de type graphique (des points).

Dans cette phase peut aussi figurer ce qu'on appelle l'entrée des données, qui peut se manifester par la saisie de caractères ou de nombres sur le clavier, ou la lecture de la position du pointeur de la souris, ou encore par la lecture d'un fichier contenant ces nombres ou caractères.

Algorithmique

Les 3 étapes d'un algorithme

- Les entrées (ou la déclaration et la saisie des données)
- Le traitement
- Les sorties (ou l'affichage / l'impression des données transformées)

• **Le traitement**

Il s'agit de déterminer toutes les étapes des traitements à faire et donc des "**instructions**" à donner pour une exécution automatique.

Si ces instructions s'exécutent en séquence, on parle d'algorithme **séquentiel**.
Si les opérations s'exécutent sur plusieurs processeurs en parallèle, on parle d'algorithme **parallèle**. Si les taches s'exécutent sur un réseau de processeurs on parle d'algorithme **réparti** ou **distribué**.
Nous ne traiterons ici que des algorithmes séquentiels.

Algorithmique

Les 3 étapes d'un algorithme

- Les entrées (ou la déclaration et la saisie des données)
- Le traitement
- Les sorties (ou l'affichage / l'impression des données transformées)

• **Les sorties**

Les résultats obtenus peuvent être affichés sur l'écran, ou imprimés sur papier, ou bien encore conservés dans un fichier.

Algorithmique

Les 3 étapes d'un algorithme

- Les entrées (ou la déclaration et la saisie des données)
- Le traitement
- Les sorties (ou l'affichage / l'impression des données transformées)

- **Un exemple simple**

On se donne deux points A et B du plan. **a. Tracer le cercle de centre A passant par B.** **b. Tracer le cercle de centre B passant par A.** **c. Nommer C et D les points d'intersection de ces cercles.** **Construire le polygone ADBC.**	**Cet algorithme décrit la construction d'un losange dont une diagonale est [AB].** ***Les entrées* sont les points A et B.** ***Le traitement* de la construction est décrit dans les phases a. b. et c.** ***La sortie* est le polygone ADBC.**

Algorithmique

Les 3 briques de base pour écrire les instructions du traitement

1. Affectation de variables
2. Structure alternative (Si … Alors …. Sinon)
3. Structures répétitives (« Boucles » Pour et Tant Que)

Algorithmique

1. Affectation de variables

Les données de l'algorithme peuvent être stockées dans des variables ou « mémoires ».
Ces données sont représentées par un nom (un **identificateur)**.

Les identificateurs sont des suites de lettres et chiffres (sans espaces) qui doivent être choisies judicieusement pour que l'algorithme soit immédiatement lisible et interprétable.

Les données peuvent avoir un type :

- ✓ numériques
- ✓ chaines de caractères
- ✓ booléen (Vrai ou Faux)
- ✓ listes (numériques ou de chaînes de caractères)

Algorithmique

1. Affectation de variables

Exemples d'affectation d'une variable :

A ← 2 : La variable nommée A prend pour valeur 2 quel que soit sa valeur précédente

Compteur ← Compteur + 1 : La variable nommée Compteur prend pour valeur la valeur courante de Compteur + 1
(Attention ! : ce n'est pas une équation !)

C ← 2
D ← 3
E ← C + D : à la fin des 3 instructions E a pour valeur 5.

Algorithmique

1. Affectation de variables

Exercice : proposer un algorithme qui permute les contenus de deux variables numériques.

Réponse :

```
Temp ← A
A ← B
B ← Temp
```

On utilise une variable temporaire nommée ici Temp

Algorithmique

2. Structure alternative

Si *condition* **alors**
    traitement 1
Sinon
    traitement 2
FinSi

Remarque :L'évaluation de la condition est un booléen (qui a pour valeur Vrai ou Faux).

Exemple :

Si N > 10 **alors**
    Afficher « Le nombre » + N + « est strictement supérieur à 10 »
Sinon
    Afficher « Le nombre » + N + « est inférieur ou égal à 10 »
FinSi

Algorithmique

3. Structures répétitives : boucle Pour

Pour I = 1 **à** N **Faire**
  traitement
I Suivant

On effectue les instructions nommées « traitement» N fois.

Exemple : Que fait cet algorithme ?

Pour I = 1 **à** 10 **Faire**
  Afficher I*I
I Suivant

Réponse : Cet algorithme affiche tous les carrés des entiers naturels compris entre 1 et 10.

Algorithmique

3. Structures répétitives : boucle « Tant que »

Tant que condition **Faire**
  traitement
FinTantQue

On effectue les instructions nommées « traitement» tant que la condition est vérifiée.

Exemple : Que fait cet algorithme ?

```
I ← 1
Tant que  I < 100 Faire
    Afficher I
  I ← I + 2
FinTantQue
```

Réponse : Cet algorithme affiche tous les entiers naturels impairs compris entre 1 et 100.

Algorithmique

3. Structures répétitives : boucle « Répéter … Jusqu'à …. »

Répéter
traitement
Jusqu' à Condition

On effectue les instructions nommées « traitement» jusqu' à ce que la condition ne soit plus vérifiée.

Algorithmique

La suite de Syracuse

Présentation du problème :

La suite de Syracuse d'un nombre entier N est définie par récurrence, de la manière suivante :

- $u_0 = N$
- Pour tout entier n naturel :
  - ✓ si u_n est pair alors $u_{n+1} = u_n/2$;
  - ✓ si u_n est impair alors $u_{n+1} = 3u_n + 1$

On se propose d'écrire un algorithme en « pseudo-code » qui en entrée lit l'entier N, calcule et affiche les 5 premiers termes de cette suite.

Algorithmique

La suite de Syracuse

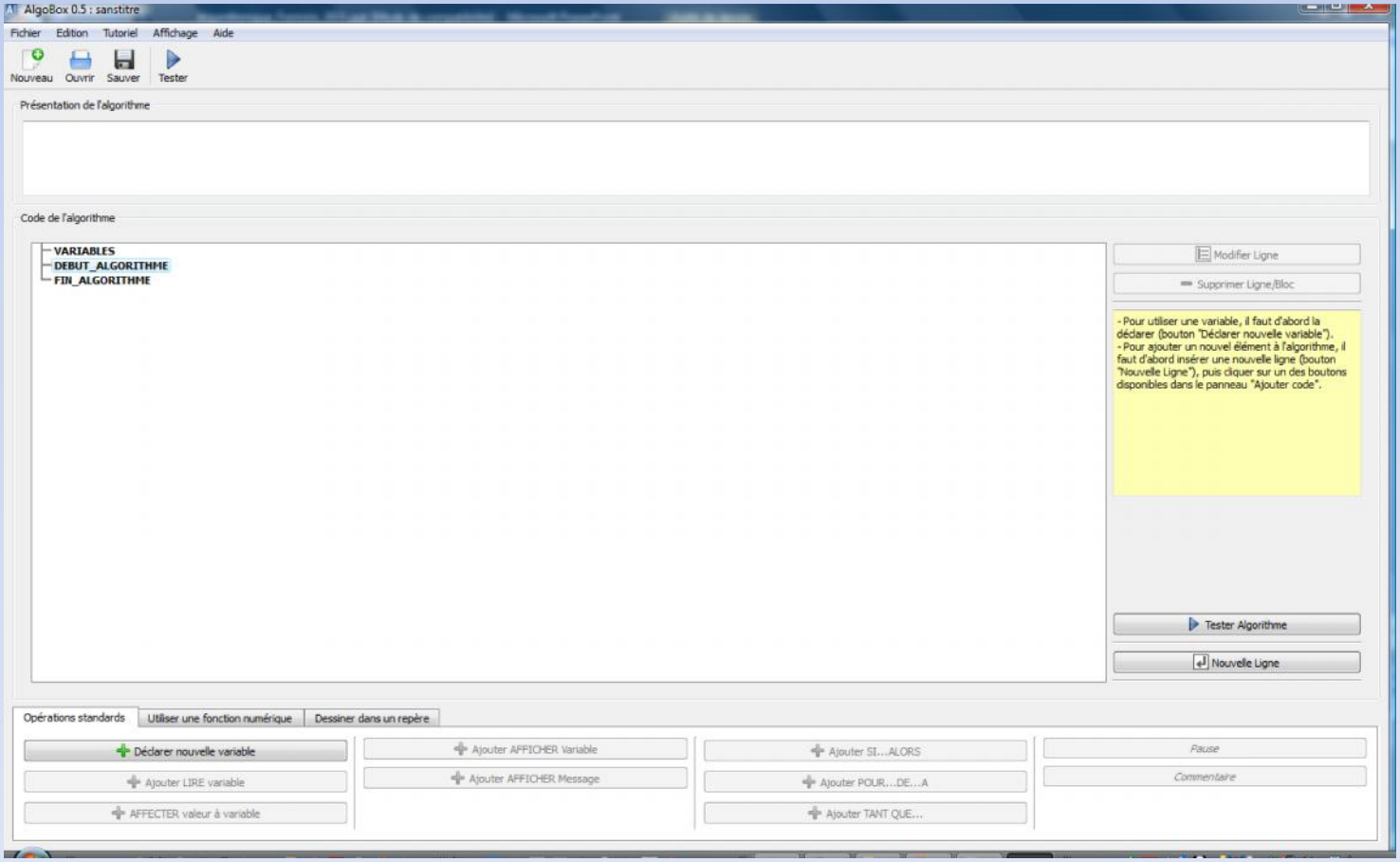
Algorithme : suite de Syracuse

Données	
N : entier	Premier terme de la suite
I : indice	Indice du terme courant de la suite u_n
Traitement	
Lire N	On demande la saisie de l'entier N
Pour I = 1 **à** 5	On effectue une boucle avec 5 itérations.
Afficher N	On affiche le terme courant de la suite.
Si N est pair **alors**	On teste la parité du terme courant de la suite.
N ← N/2	On calcule le terme suivant de la suivant selon le
Sinon	résultat du test de parité.
N ← 3*N+1	
FinSi	
FinPour	

Un premier algorithme 18

Algorithmique

La suite de Syracuse avec AlgoBox



Les premiers pas avec AlgoBox 19

Algorithmique

La suite de Syracuse avec AlgoBox

Programme AlgoBox :

```
1    VARIABLES
2      N EST_DU_TYPE NOMBRE
3      I EST_DU_TYPE NOMBRE
4    DEBUT_ALGORITHME
5      LIRE N
6      POUR I ALLANT_DE 1 A 5
7        DEBUT_POUR
8        AFFICHER N
9        SI (N % 2 == 0) ALORS
10         DEBUT_SI
11         N PREND_LA_VALEUR N/2
12         FIN_SI
13         SINON
14           DEBUT_SINON
15           N PREND_LA_VALEUR 3*N+1
16           FIN_SINON
17       FIN_POUR
18   FIN_ALGORITHME
```

Algorithmique

La suite de Syracuse avec AlgoBox

- Tester l'algorithme avec différentes valeurs de N.
- Modifier l'algorithme pour afficher les 20 premiers termes de la suite.
- Quelle conjecture peut-on émettre sur la suite de Syracuse ?

Réponse :

Quel que soit le nombre N de départ, la suite de Syracuse admet un terme qui a pour valeur 1.

Algorithmique

La suite de Syracuse avec AlgoBox

Modifier l'algorithme précédent pour s'arrêter au premier terme de la suite égal à 1 et coder le programme AlgoBox correspondant.

Algorithme : suite de Syracuse (version 2)	
Données	
N : entier	Premier terme de la suite
I : indice	Indice du terme courant de la suite u_n
Traitement	
Lire N	On demande la saisie de l'entier N
Tant que N <> 1 **Faire**	On effectue une boucle avec un test d'arrêt.
Afficher N	On affiche le terme courant de la suite.
Si N est pair **alors**	On teste la parité du terme courant de la suite.
N ← N/2	On calcule le terme suivant de la suivant selon le résultat du test de parité.
Sinon	
N ← 3*N+1	
FinSi	
FinTantQue	

Algorithmique

La suite de Syracuse avec AlgoBox

Suite de Syracuse (version 2) : programme AlgoBox

Suite_syracuse_V2 - 23.08.2010

```
1  VARIABLES
2   N EST_DU_TYPE NOMBRE
3   I EST_DU_TYPE NOMBRE
4  DEBUT_ALGORITHME
5   LIRE N
6   TANT_QUE (N != 1) FAIRE
7    DEBUT_TANT_QUE
8    AFFICHER N
9    SI (N % 2 == 0) ALORS
10     DEBUT_SI
11     N PREND_LA_VALEUR N/2
12     FIN_SI
13     SINON
14      DEBUT_SINON
15      N PREND_LA_VALEUR 3*N+1
16      FIN_SINON
17    FIN_TANT_QUE
18   AFFICHER N
19 FIN_ALGORITHME
```

Algorithmique

La suite de Syracuse avec AlgoBox

Quel est le risque encouru par le dernier algorithme ?

Réponse :

Si la conjecture de Syracuse est erronée il peut exister des entiers N tels que aucun terme de la suite ne soit égal à 1.

Dans ce cas la condition N <> 1 de la boucle Tant Que sera toujours vraie et le programme bouclera indéfiniment !

Information
A ce jour, la conjecture de Syracuse n'est pas démontrée.

Algorithmique

La suite de Syracuse avec AlgoBox

Prolongements : Etude de la longueur et de l'altitude du vol liés à la suite de Syracuse

Longueur du vol :

Pour un premier terme u_0 donné, on appelle longueur du vol de la suite l'entier l tel que $u_l = 1$ pour la première fois.

Altitude maximale du vol :

Pour un premier terme u_0 donné, on appelle altitude maximale du vol le maximum des termes de la suite .

Algorithmique

La suite de Syracuse avec AlgoBox

Prolongements : Etude de la longueur et de l'altitude du vol liés à la suite de Syracuse

Proposition d'algorithmes à implémenter avec AlgoBox :

- Calculer longueur et altitude pour un vol donné de la suite.
- Afficher graphiquement (sous forme d'histogramme) les longueurs et altitudes

de vol de la suite pour $1 \leq u0 \leq u\_max$

(avec u_max = 10 ou 100 ou 1000 ou 1000)

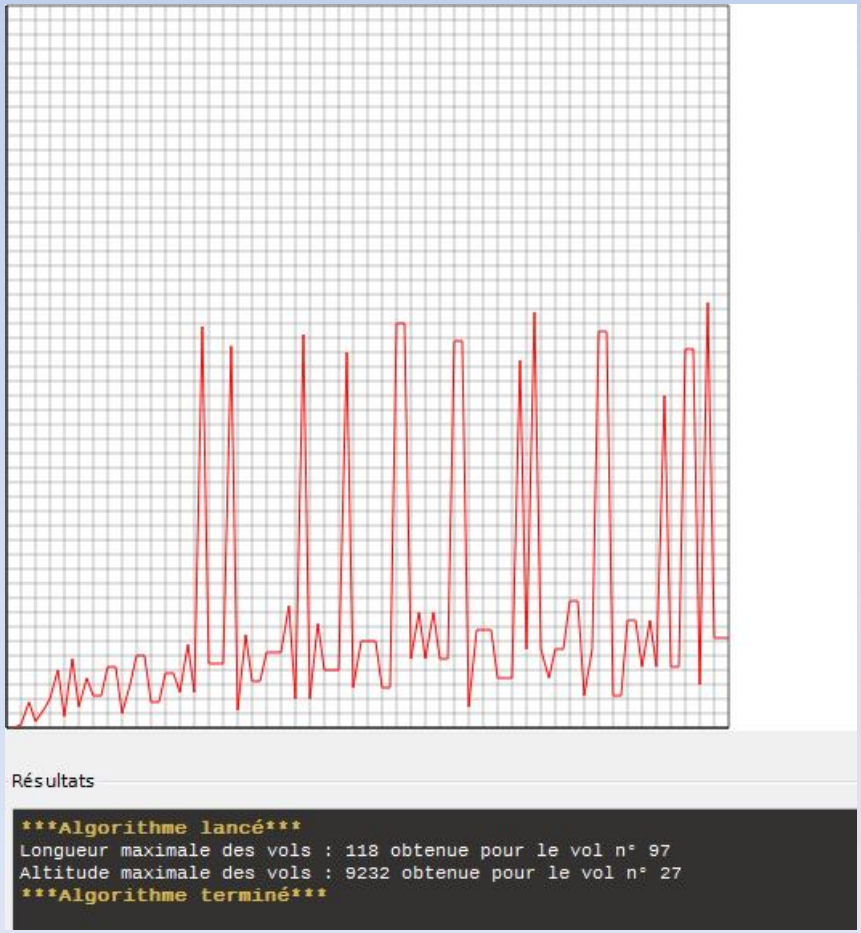
et déterminer le vol de longueur maximale et d'altitude maximale

pour $1 \leq u_0 \leq$ u_max

Algorithmique

La suite de Syracuse avec AlgoBox

Prolongements : Longueurs des vols de la suite pour $1 \leq u_0 \leq 100$

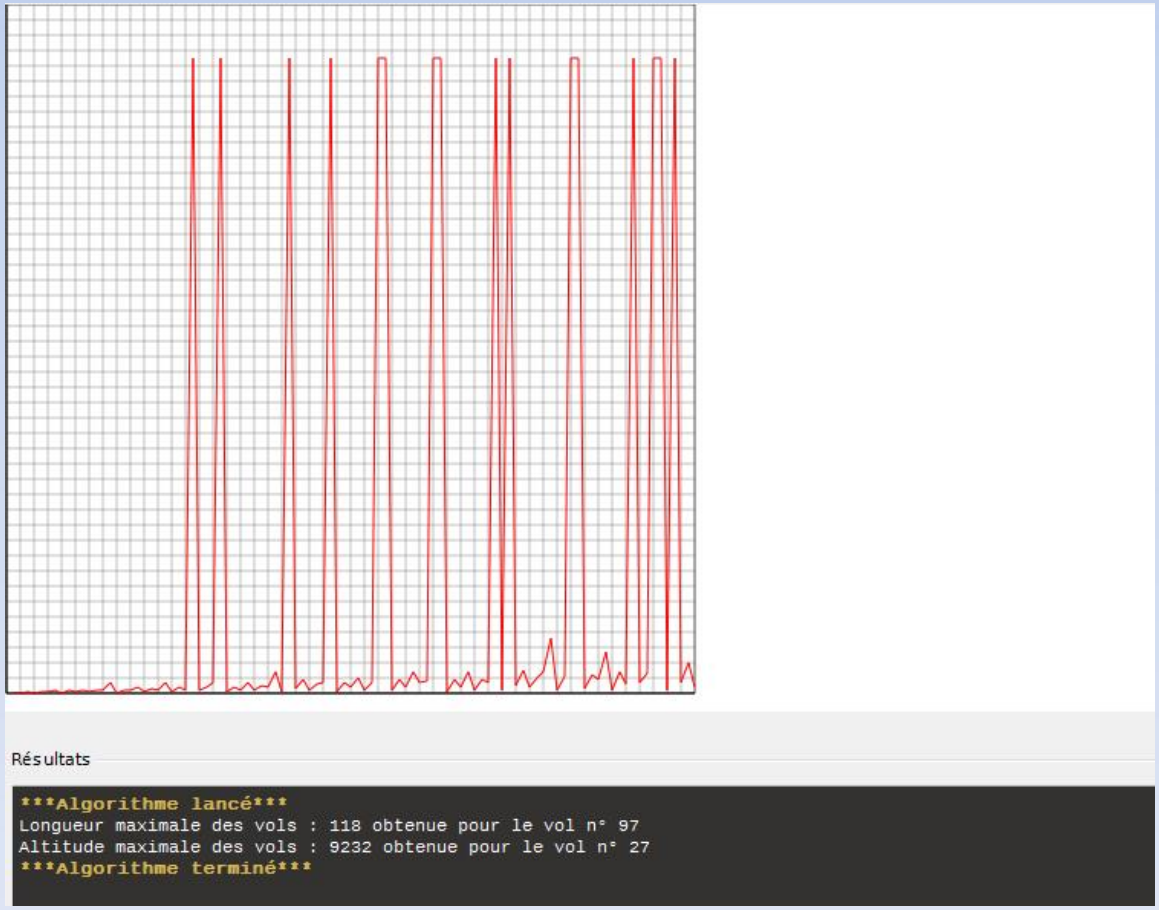


Les premiers pas avec AlgoBox 27

Algorithmique

La suite de Syracuse avec AlgoBox

Prolongements : Altitude des vols de la suite pour $1 \leq u_0 \leq 100$



Les premiers pas avec AlgoBox 28

Algorithmique

Al-Khawarizmi
(783 - 850)

Mathématicien, géographe, astrologue et astronome musulman perse dont les écrits, rédigés en langue arabe, ont permis l'introduction de l'algèbre en Europe.

Il est à l'origine des mots « algorithme » (qui n'est autre que son nom latinisé: "algoritmi" [3]) et « algèbre » (issu d'une méthode et du titre d'un de ses ouvrages) ou encore de l'utilisation des chiffres arabes

Blaise Pascal
(1623-1662)

Savant, philosophe et écrivain, il est l'inventeur d'une machine arithmétique nommée Pascaline.

Algorithmique

Charles Babbage
(1791-1871)

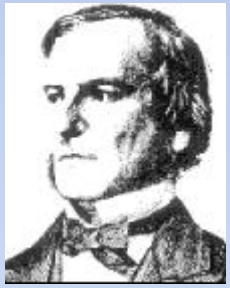
Fils unique d'un banquier, cet anglais, algébriste de talent, renoncera à une carrière prometteuse de savant pour consacrer sa vie à la construction d'un calculateur mécanique.
Il est l'inventeur de la Machine à différences puis de la Machine Analytique.

Augusta Ada King
Lady LOVELACE
(1814-1852)

Fille du poète romantique Lord George Byron et d'une mathématicienne et féministe, épouse de William King (futur comte de Lovelace), elle sera l'élève puis la collaboratrice de C. BABBAGE.
Elle est aussi à l'origine du "Principe des machines à calculer".
Un langage porte son prénom en mémoire de ses travaux.

Une galerie de portraits 30

Algorithmique



Mathématicien anglais, il publie en 1854 les Lois de la pensée. Dans ce livre, il décrit comment toute la logique peut être définie par un principe simple: le binaire.

George BOOLE
(1815-1864)

(*Prononcer noy-man*).
Il a été l'un des personnages clés des débuts de l'informatique. Il publia de nombreux articles sur l'algèbre et la mécanique quantique avant de se consacrer à la construction d'ordinateurs et à la modélisation mathématique de la réaction en chaîne de la bombe A. Ses "machines IAS" sont à l'origine de "l'Architecture Von NEUMANN", c'est à dire celle des ordinateurs tels que nous les connaissons.

John Von NEUMANN
(1903-1957)

Algorithmique

Grace Murray HOPPER
(1906 - 1992)

Cette américaine, mobilisée comme auxiliaire dans la marine américaine fut affectée aux travaux de programmation et d'exploitation de l'ENIAC. Puis, devenue une grande spécialise de la programmation des ordinateurs, elle sera l'une des principales créatrices du COBOL.

Alan TURING
(1912 - 1954)

Mathématicien anglais, maître-assistant à Cambridge dès 23 ans. Il a conçu en 1936 une machine logique capable de résoudre tous les problèmes que l'on peut formuler en termes d'algorithmes.
Pendant la guerre, il participera à la réalisation de la *Bombe*, première machine électromécanique de décryptage des messages codés avec l'Enigma Allemande.

Une galerie de portraits 32

Dennis RITCHIE
(1941)

Cet ingénieur des laboratoires Bell, est l'auteur du langage C. En 1973, avec K. THOMPSON, il réécrira dans ce nouveau langage le système d'exploitation UNIX.

Vinton G. CERF
(1943 -)

C'est l'un des pères de l'Internet. Encore étudiant de l'université de Los Angeles, il fut l'un des auteurs du protocole TCP/IP et développa avec une équipe de chercheurs les premiers outils utilisant ce mode de communication. Il est aujourd'hui président de *l'Internet Society* qui surveille les nouveaux standards d'Internet.

Algorithmique

Créateur du langage C++ basé sur le langage C mais en lui donnant une dimension de Langage Orienté Objet.

Bjarne STROUSTRUP
(1950 -)

Créateur du langage Java basé sur le langage C++. La particularité principale de Java est que les logiciels écrits dans ce langage sont très facilement portables sur plusieurs systèmes d'exploitation.

James Gosling
(1955 -)

Une galerie de portraits 34

Algorithmique

Ancien président (et fondateur avec P. ALLEN) de Microsoft.
Cette société est à l'origine du MS-DOS, de **Windows**, du **Basic-Microsoft** puis de **Visual Basic**.

Bill GATES
(1951 -)

S. WOZNIAK (*à gauche*) et S. JOBS (*à droite*).
Fondateurs de la société *Apple*.
Après son éviction d'Apple S. JOBS créera la société Next avant d'être rappelé pour redresser Apple.

Steve JOBS
(1955 - 2011)

Algorithmique

Richard STALLMAN
(1953 -)

Fondateur du projet GNU, lancé en 1984 pour développer le système d'exploitation libre GNU et donner ainsi aux utilisateurs des ordinateurs la liberté de coopérer et de contrôler les logiciels qu'ils utilisent. Il est également le créateur (entre autres) de l'éditeur Emacs et du compilateur gcc.

Linus TORVALDS
(1969 -)

Travaillant maintenant aux Etats-Unis mais Finlandais d'origine, il a construit en 1991 un nouveau système d'exploitation de type UNIX appelé Linux. Ayant choisi de le diffuser suivant le principe des logiciels libres, Linus TORVALDS ne retire aucune royaltie de son travail sur le noyau Linux. Cela n'empêche pas sa popularité de croitre de jours en jours.

Algorithmique

Larry Page
(1973 -)

Sergey Brin
(1973 -)

Créateurs du moteur de recherche Google.
Ces deux jeunes brillants nord-américains ont lancé leur moteur de recherche en 1999.
Ce mot vient du terme "googol" qui désigne un chiffre, un 1 suivi de 100 zéros, traduisant l'exhaustivité du moteur de recherche.

Mark Zuckerberg
(1984 -)

Créateur de Facebook
C'est en 2004 que la première version de Facebook voit le jour pour mettre en relation les étudiants de Harvard.