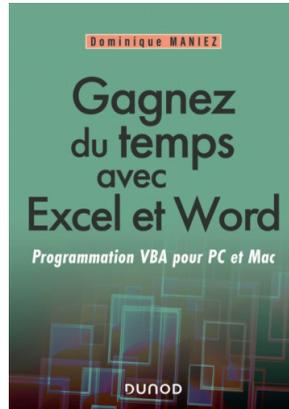


**APPRENDRE À
PROGRAMMER**
EN 10 SEMAINES CHRONO

Du même auteur



Conception et réalisation de la maquette intérieure : Maud Warg

© Dunod, 2019, 2023

11 rue Paul Bert, 92240 Malakoff

www.dunod.com

ISBN 978-2-10-085203-1

Toute représentation ou reproduction intégrale ou partielle faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause est illicite selon le Code de la propriété intellectuelle [Art. L 122-4] et constitue une contrefaçon réprimée par le Code pénal.

Seules sont autorisées [Art. L 122-5] les copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective, ainsi que les analyses et courtes citations justifiées par le caractère critique, pédagogique ou d'information de l'œuvre à laquelle elles sont incorporées, sous réserve, toutefois, du respect des dispositions des articles L 122-10 et L 122-12 du même Code, relatives à la reproduction par reprographie.

Dominique Maniez

**APPRENDRE À
PROGRAMMER**
EN 10 SEMAINES CHRONO

Une méthode visuelle pour tous

DUNOD

TABLE DES MATIÈRES

AVANT DE COMMENCER

p. 7

SEMAINE 1

VASTE PROGRAMME !

Découvrir les notions de programme et de langage de programmation

p. 10

SEMAINE 2

FAIRE BLOC

Prendre en main Blockly et la programmation visuelle par blocs

p. 28

SEMAINE 3

C'EST VARIABLE...

Définir et utiliser les variables

p. 42

SEMAINE 4

ÇA TOURNE EN BOUCLE !

Répéter des instructions grâce aux boucles

p. 74

SEMAINE 5

AVEC DES SI...

Complexifier ses programmes avec les tests conditionnels

p. 94





SEMAINE 6

J'EN AI TOUTE UNE LISTE !

Exploiter un grand nombre d'informations avec les listes

p. 116

SEMAINE 7

EN FONCTION DE

Utiliser les fonctions pour exécuter des tâches récurrentes

p. 148

SEMAINE 8

BIEN PROGRAMMER

S'organiser pour produire des programmes clairs et sans erreurs

p. 168

SEMAINE 9

DE L'IMAGE AUX MOTS

Passer de la programmation visuelle aux langages de programmation classiques

p. 188

SEMAINE 10

DÉVELOPPEMENT D'APPLICATIONS MOBILES

Utiliser App Inventor 2 pour créer des applis mobiles

p. 212

POUR ALLER PLUS LOIN

p. 249

INDEX

p. 252

AVANT DE COMMENCER

POURQUOI APPRENDRE À PROGRAMMER?

Dans l'avant-propos de la première édition de ce livre¹, je dressais la liste de toutes les raisons qui peuvent pousser un adulte à apprendre à programmer et vous pouvez constater qu'elles sont nombreuses :

- ▶ Aujourd'hui, tout le monde apprend à coder !
- ▶ Apprendre à programmer est très formateur pour l'esprit
- ▶ Apprendre à coder permet de trouver du travail
- ▶ Programmer fait gagner du temps
- ▶ Coder permet de mieux utiliser un ordinateur
- ▶ Il vaut mieux programmer qu'être programmé !

Dans cette deuxième édition, après avoir formé pendant six ans plus d'un millier d'étudiants de sciences humaines à l'initiation à la programmation, je reste persuadé qu'il est très important d'apprendre le codage quand on n'a pas eu la chance de suivre des cours d'informatique au collège ou au lycée. Les nouvelles technologies ont envahi notre quotidien² et l'apprentissage de la programmation facilite la compréhension des enjeux de la transformation de notre société qui repose de plus en plus sur des dispositifs techniques.

La deuxième certitude est que tout le monde peut programmer et qu'il n'y a nul besoin d'être un génie des mathématiques. Cet ouvrage aurait d'ailleurs pu s'intituler *Apprendre à programmer quand on n'est pas matheux*, afin de bien souligner qu'il s'adresse à des gens qui ne sont pas spécialement doués dans cette discipline. Bien évidemment, si vous parcourez le livre de Monsieur Cormen³, vous constaterez que sur le millier de pages de cet ouvrage de référence, il y a vraiment beaucoup d'équations. De la même manière, le terme « algorithmique » provient du nom d'un mathématicien et il faudrait être de bien mauvaise foi pour prétendre que les mathématiques et l'informatique n'ont aucun lien. Pour autant, faut-il avoir lu les 1760 pages de la seizième édition du *Bon usage* de Maurice Grevisse pour avoir le droit d'écrire dans notre belle langue ? Faut-il être un champion de la syntaxe pour goûter le plaisir d'aligner les mots afin de constituer des phrases ? Des millions de francophones accèdent quotidiennement aux joies

¹ Que vous pouvez retrouver à www.progbloc.fr/projet-editorial/

² Au moment où j'écris ces lignes, on ne parle plus que de ChatGPT et des dérivés de l'intelligence artificielle.

³ *Introduction à l'algorithmique*, publié chez Dunod.

de l'écriture sans même connaître l'existence de l'excellente grammaire de Monsieur Grevisse et ils ne s'en portent pas plus mal. Par cette analogie, je veux ici affirmer que l'on peut se faire plaisir à programmer même si l'on ne comprend rien aux mathématiques.

À QUI EST DESTINÉ CET OUVRAGE ?

Ce livre est destiné à tous ceux et à toutes celles qui n'ont aucune expérience de la programmation et qui veulent s'initier aux joies du codage. Comme l'informatique est désormais enseignée au collège et au lycée, je pense bien entendu à tous les parents et grands-parents qui aimeraient discuter avec leurs enfants et petits-enfants de ce qu'ils font en cours d'informatique au collège ou au lycée. J'ai en tête également toutes les personnes qui n'ont pas eu la chance de bénéficier d'un enseignement de la programmation au cours de leur scolarité et qui se rendent bien compte que quelques notions de codage ne seraient pas superflues dans leur pratique professionnelle quotidienne. Je vise aussi toutes les personnes qui utilisent un smartphone ou une tablette et qui aimeraient bien pouvoir créer leurs propres applications. Le dernier chapitre de cet ouvrage présente deux applications utilisant les techniques enseignées dans ce livre qui vont vous permettre de créer facilement des programmes tournant sur votre smartphone ou votre tablette.

Si par commodité cet ouvrage est rédigé au masculin générique, je lutte activement contre les stéréotypes de genres et je regrette amèrement la faible proportion de codeuses chez les informaticiens⁴. Alors, Mesdames, soyez donc parfaitement assurées que ce livre s'adresse totalement à vous !

Je souhaite enfin insister lourdement sur ce qui, au fil des années, est devenu pour moi un véritable credo : toute personne motivée, quelle que soit son âge et ses compétences en mathématiques, peut apprendre à programmer. Il faut donc déculpabiliser et désinhiber tous ceux et toutes celles qui ont envie d'apprendre à coder en leur disant que les seules qualités requises pour ce faire sont la rigueur, l'imagination, la persévérance et la logique.

ORGANISATION DE CET OUVRAGE

Ce livre est composé de dix chapitres qui permettent de découvrir progressivement les principaux concepts de la programmation et il est conseillé de lire les chapitres dans l'ordre où ils apparaissent. Comme toute science, l'informatique a son vocabulaire spécialisé et j'ai attaché un soin tout particulier à définir tous les termes techniques la première fois où ils sont employés. J'ai pris le parti d'utiliser un outil de programmation visuelle intitulé Blockly⁵ car il s'agit d'une application à la fois simple et puissante. À la fin de chaque chapitre, des exercices pratiques sont proposés et vous trouverez les corrigés en ligne.

⁴ www.scoop.it/topic/femmes-informatique

⁵ Qui est proche de Scratch, le logiciel sélectionné pour l'apprentissage de la programmation au collège

PRÉREQUIS POUR L'UTILISATION DE CET OUVRAGE

Comme vous pourrez le constater, les prérequis pour pouvoir profiter pleinement du contenu de ce livre sont excessivement simples : vous devez disposer d'un ordinateur ainsi que d'une connexion à Internet, et savoir utiliser un navigateur. Si vous n'avez pas d'ordinateur, vous pouvez employer une tablette ou un smartphone tournant sous le système Android, même si cela est un peu moins pratique.

COMPLÉMENTS ÉLECTRONIQUES EN LIGNE

Cet ouvrage s'accompagne de compléments électroniques qui sont disponibles sur le site www.progbloc.fr. Outre la correction des exercices proposés à la fin de chaque chapitre, vous trouverez sur ce site des informations complémentaires ainsi que toutes les URL mentionnées dans ce livre. Le téléchargement de ces ressources est vivement recommandé.

Vous trouverez également sur ce site les fichiers nécessaires à l'installation de Blockly sur votre ordinateur. Dans la première édition de ce livre, j'ai souvent critiqué la piètre traduction des messages de Blockly et j'ai fini par me souvenir que j'étais également un peu traducteur si bien que j'ai modifié les messages litigieux et simplifié l'interface utilisateur de Blockly. Si vous voulez bien profiter de ce livre, il est donc vivement conseillé de télécharger la version de Blockly que je propose sur mon site afin de l'installer sur votre machine.

COMMUNICATION AVEC L'AUTEUR

En tant qu'auteur, c'est toujours un réel plaisir d'échanger avec ses lecteurs et ses lectrices ; c'est donc avec joie que je réponds aux sollicitations de celles et ceux qui veulent me poser des questions sur cet ouvrage. Comme cela est indiqué dans la rubrique À propos du site www.progbloc.fr, vous pouvez m'envoyer un courriel à dm@progbloc.fr, ou encore mieux, commenter un des articles du site, si bien que chaque visiteur pourra profiter de ma réponse à votre message.

Je souhaite à toutes et à tous un très bon apprentissage de la programmation !

Dominique Maniez
Lyon, mars 2023



VASTE PROGRAMME !

SEMAINE 1

Tout le monde peut apprendre à programmer. Nous considérons même que la programmation est une chose naturelle que nous accomplissons sans nous en rendre compte, un peu comme Monsieur Jourdain faisait de la prose sans le savoir. Le vocabulaire de la programmation a d'ailleurs envahi notre vie quotidienne et il ne se passe pas une semaine sans qu'un homme politique ne nous dise que nous devons changer de logiciel. De la même manière, avant de savoir écrire des programmes informatiques, vous avez sans doute déjà programmé un magnéto, un réveil ou bien votre four. Afin de prouver notre thèse, nous allons commencer notre tour d'horizon de l'apprentissage du codage en vous proposant un programme que vous allez adorer.

C'EST QUOI AU JUSTE PROGRAMMER?

Le terme « programme » a de nombreux sens que l'on retrouve dans les expressions « programme télé », « programme politique » ou bien encore « programme scolaire ». On note ces mêmes significations dans le mot « programmation » avec les expressions « programmation musicale » ou « programmation des vacances ». Bien entendu, ce que nous entendons par programmation dans ce livre a un sens différent, mais on retrouve quand même les idées d'organisation et de planification.

Pour mieux se représenter le concept de programme, nous allons prendre un exemple que chacun peut aisément comprendre et que je vous invite à exécuter :

Ingrédients

- ▶ 400 g de chocolat noir à 55 %
- ▶ 400 g de compote de pommes
- ▶ 100 g de sucre de canne roux
- ▶ 6 œufs
- ▶ 100 g de farine de riz
- ▶ 125 g de noix de pécan

Si vous prenez du chocolat à 70 % de cacao, vous pouvez augmenter la dose de sucre.

Si vos œufs sont très gros, vous pouvez n'en utiliser que cinq.

Accomplissez dans l'ordre les instructions suivantes :

1. Faire fondre le chocolat au bain-marie.

Dans un grand bol, mélanger (au fouet ou à l'aide d'un robot) les œufs et le sucre jusqu'à obtenir un liquide homogène.

2. Ajouter petit à petit la farine tout en continuant à mélanger.
3. Attendre que le chocolat soit tiède et l'incorporer au mélange tout en continuant à fouetter.
4. Incorporer la compote de pommes tout en continuant à fouetter.
5. Concasser les noix de pécan et les incorporer dans la préparation.
6. Répartir la préparation dans un plat ou des ramequins allant au four.
7. Enfourner dans un four préchauffé à 180 degrés.
8. Laisser cuire de 20 à 35 minutes en fonction de la taille et de la hauteur des moules.

Les personnes intolérantes au gluten et au lactose constateront avec plaisir que ce programme de brownie est parfaitement compatible avec leur régime et qu'elles peuvent donc l'exécuter autant de fois que nécessaire.



UN PREMIER PROGRAMME BIEN ALLÉCHANT!

Comme vous l'aurez compris, nous considérons qu'une recette de cuisine est un très bon exemple de programme et qu'il existe des analogies fortes entre un programme informatique et l'exécution en séquence des instructions d'une recette. La plupart des concepts que nous allons étudier dans ce livre, comme les boucles (répétition d'une opération) et les tests conditionnels (phrase commençant par un si), figurent d'ailleurs explicitement dans notre exemple de programme.

Tentons à présent de donner une définition de la programmation afin d'être certain que nous parlons tous de la même chose. Nous définirons tout d'abord la programmation comme l'art d'écrire des programmes. Bien entendu, le terme « programme » est ici pris au sens informatique où il a une signification bien précise et on peut, par exemple, reprendre la définition du *Grand Robert* :

« Ensemble ordonné des opérations nécessaires et suffisantes pour obtenir un résultat. »

Cette définition introduit la notion importante de **résultat** ; on ne programme pas pour le plaisir de programmer, mais on programme toujours pour aboutir à un résultat. Un programmeur a un but, une idée bien précise du résultat qu'il veut obtenir et une partie importante des problèmes que l'on rencontre quand on programme vient du fait que l'on n'aura pas défini clairement le résultat auquel on veut parvenir.

Dans l'exemple de programme que nous avons donné précédemment, le résultat est clairement défini : un brownie pour douze personnes que les intolérants au gluten et au lactose pourront consommer sans modération.

L'autre élément important de la définition est que les opérations sont ordonnées. Cela peut paraître évident, mais il est préférable de le rappeler : l'ordre dans lequel les opérations sont effectuées est capital. Tous ceux qui cuisinent savent cela et il en va de même en informatique. Le fait d'exécuter des opérations dans un sens plutôt qu'un autre pourra, dans de très nombreux cas, faire échouer le programme qui n'aboutira donc pas au résultat escompté.

Le dernier aspect de la définition évoque les opérations nécessaires et suffisantes pour parvenir au résultat. On peut en conclure qu'un programme qui comporterait des opérations inutiles à la réalisation du but recherché ne serait pas un bon programme. Comme dans un discours où les mots inutiles ne servent à rien, sinon à embrouiller l'auditoire, les lignes de code d'un programme qui n'ont pas à y figurer vont au mieux ralentir l'exécution du programme et au pire l'empêcher d'accomplir le résultat recherché.

On peut aussi déduire de cette définition qu'il n'existe peut-être pas un ensemble unique d'opérations pour arriver à un résultat. De la même manière qu'il existe en général plusieurs chemins pour rallier un point A à un point B, il peut y avoir plusieurs solutions à un problème informatique. Certains chemins sont plus courts que d'autres, ou bien plus rapides, mais tous permettent d'arriver à la même destination. Certains programmes seront aussi plus rapides que d'autres pour arriver au même résultat, et la différence

de temps d'exécution peut être importante entre deux programmes équivalents en termes de fonctionnalités. On privilégiera en général la version la plus rapide et l'efficacité d'un programme sera un des critères d'appréciation de sa qualité. Un programme qui résoudrait un problème correctement, mais mettrait trop longtemps à s'exécuter pourrait d'ailleurs voir son intérêt totalement anéanti¹. Dans certains programmes, le temps d'exécution est primordial et fait partie du résultat à obtenir ; on parle alors de systèmes temps réel et c'est notamment le cas pour les systèmes de pilotage automatique en aéronautique.

Il y a également une chose importante qui est plus ou moins induite par la définition que nous avons citée : un programme doit marcher dans tous les cas possibles. Prenons l'exemple d'un programme qui convertit des degrés Fahrenheit en degrés Celsius : ce programme doit fonctionner pour toutes les valeurs qu'il aura à convertir, qu'elles soient très petites, négatives ou bien très grandes. A contrario, un programme qui marcherait neuf fois sur dix ne serait pas un programme qui fonctionne correctement, et cela même si la proportion de réussite est très proche de 100 %. Sur ce plan-là, l'informaticien est binaire et il considère qu'un programme marche ou ne marche pas ; il ne peut pas marcher un peu ou à moitié, même si l'on a tous fait l'expérience de programmes commerciaux qui ne marchaient pas si bien que cela...

On pourra en définitive adopter la définition suivante d'un programme :

« Suite ordonnée d'instructions permettant de résoudre efficacement et à tous les coups un problème clairement défini. »

Dans ces conditions, notre recette de brownie correspond bien à la définition que nous venons de donner d'un programme et on peut conclure que l'on n'a pas besoin d'un ordinateur pour programmer et que les programmes ne se limitent pas au domaine de l'informatique.

Si vous n'êtes pas très porté sur la cuisine et que notre exemple ne vous dit pas grand-chose, vous pouvez penser au montage d'un meuble en kit ; cette opération s'apparente également à la réalisation d'un programme et quiconque a tenté l'expérience sait que l'ordre de montage des éléments est primordial si l'on veut aboutir au résultat recherché. Si vous commencez à réfléchir à d'autres opérations de la vie quotidienne, vous vous rendrez alors vite compte qu'il existe de nombreuses activités où l'on doit reproduire en séquence toute une série d'actions afin d'aboutir à un résultat. Prendre son petit-déjeuner le matin ou bien se laver les dents sont en général des activités qui sont parfaitement codifiées et que vous accomplissez tous les jours sans vous poser de questions. Pourtant, au sens informatique du terme, il s'agit de programmes que vous exécutez. Programmer consiste à écrire le scénario complet de ces activités pour arriver à un résultat toujours

¹Dans sa passionnante Histoire des codes secrets, Simon Singh raconte les exploits du célèbre et génial mathématicien anglais, Alan Turing, qui était chargé de créer un programme pour casser le code de la machine à chiffrer allemande, Enigma. Les premières versions de son programme fonctionnaient, mais elles étaient trop lentes pour trouver le code secret qui changeait tous les jours...

identique ; dans le cas du petit-déjeuner, le but est d'ingérer des aliments qui apporteront suffisamment d'énergie pour vous permettre de tenir le coup jusqu'au repas de midi. Exécuter un programme consiste à effectuer les unes après les autres les différentes étapes d'un scénario qui, dans la vie courante, n'a bien entendu pas besoin d'être écrit : prendre le tube de dentifrice, ouvrir le tube, étaler la pâte sur la brosse à dents, refermer le tube, etc. Il n'en reste pas moins que si vous aviez à expliquer par écrit toutes les opérations nécessaires pour se brosser les dents, vous seriez capable de lister de manière ordonnée toutes les instructions pour ce faire. Programmer, ce n'est donc que cela : décrire en détail des suites d'opérations pour arriver à un but.

Grâce à ces exemples extraits de la vie quotidienne, on constate facilement que la logique et les concepts de la programmation nous sont en fait très familiers. Il n'y a donc pas lieu de redouter d'apprendre la programmation car nous en possédons la plupart des mécanismes ; les seules choses qui vont changer quand vous allez programmer un ordinateur sont le but que vous allez assigner au programme et le langage qui va permettre de décrire le déroulement des opérations à exécuter.

ALGORITHMIQUE

Algorithmique est un mot difficile à prononcer et à orthographier qui fait un peu peur quand on ne sait pas vraiment de quoi il s'agit. L'algorithmique est une technique indissolublement liée à la programmation et c'est la raison pour laquelle nous allons présenter cette science.

Comme pour la programmation, commençons par examiner la définition qu'en donne *le Robert* :

« Science qui étudie l'application des algorithmes à l'informatique. »

Ce même dictionnaire définit l'algorithme en ces termes :

« Ensemble des règles opératoires propres à un calcul ou à un traitement informatique. »

La définition est complétée par « calcul, enchaînement des actions nécessaires à l'accomplissement d'une tâche » et renvoie à la notion d'automate. *Le Robert* donne l'exemple de l'algorithme d'Euclide, qui permet de trouver le plus grand commun diviseur (PGCD) de deux nombres. Par exemple, le PGCD de 20 et 30 est 10 : il s'agit du plus grand nombre capable de diviser à la fois 20 et 30. Euclide ayant écrit ses traités à une période que l'on situe habituellement aux environs de 300 av. J.-C., on peut en conclure que les algorithmes existaient bien avant l'invention de l'informatique. Les algorithmes sont liés aux mathématiques et le nom algorithme vient d'ailleurs d'al-Khwārizmī (parfois orthographié Al-Khawarizmi ou al-Khuwārizmī), qui était un savant perse ayant vécu à la fin du VIII^e et dans la première moitié du IX^e siècle. En latin médiéval, son nom a été latinisé en Algorithmus. Le terme algèbre vient également

du nom de ce grand mathématicien qui a vécu à Bagdad et à qui on doit notamment l'introduction des chiffres arabes en Occident.

Un algorithme est donc tout simplement une suite d'actions permettant d'accomplir une tâche. En d'autres termes, on peut également considérer qu'un algorithme est une méthode pour résoudre un problème en un nombre d'étapes fini. Il existe des algorithmes depuis l'Antiquité car très tôt on a eu besoin de calculer et de compter (le bétails, les impôts, etc.), mais l'invention de l'informatique dans la deuxième moitié du XX^e siècle a remis au goût du jour la notion d'algorithme.

On parle de compétences algorithmiques et on peut notamment citer le fait de savoir analyser un problème concret en le modélisant ou bien de savoir décomposer un problème en sous-problèmes.

Encore une fois, la notion d'algorithme, même si elle est née dans le domaine des mathématiques et qu'elle est omniprésente en informatique, peut avoir des implications dans d'autres domaines qui n'utilisent pas forcément des machines. Il existe, par exemple, d'innombrables algorithmes dans le domaine des jeux. Les méthodes de résolution de jeux tels que le Rubik's cube, les tours de Hanoï ou le taquin sont des algorithmes qui permettent de réussir à tous les coups ces casse-tête.

Pour illustrer notre propos, nous allons prendre l'exemple de l'algorithme du casse-tête des tours de Hanoï.

LES TOURS DE HANOÏ

Inventées au XIX^e siècle par le mathématicien Édouard Lucas, les tours de Hanoï sont un casse-tête dont la règle du jeu est très simple. Le jeu est constitué de huit disques concentriques de diamètres de plus en plus petits, qui sont empilés au départ sur un mât, le jeu comportant trois mâts.

Le but du jeu est de déplacer l'ensemble des disques sur le mât de droite et de les remettre dans leur disposition initiale (le disque le plus large en bas et le plus petit disque tout en haut de la pile) en respectant les deux règles suivantes :

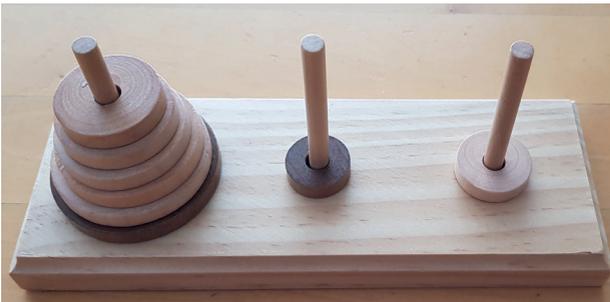
INFORMATIQUE DÉBRANCHÉE

Si vous voulez apprendre à programmer et que vous n'avez pas encore d'ordinateur, vous pouvez, en attendant d'en acquérir un, vous lancer dans l'informatique débranchée. Le mouvement éducatif de «l'informatique débranchée» (traduction de *computer science unplugged*), qui est né en Nouvelle-Zélande, consiste à proposer des activités ludiques sans ordinateur pour enseigner des concepts de l'informatique, comme les nombres binaires, le codage des informations, les algorithmes, la compression de données ou bien encore la cryptographie. Ce mouvement a donné lieu à la production d'un livre d'activités qui est disponible sur le site <http://csunplugged.org/>. Cet ouvrage, qui a été traduit en français par l'association Interstices, est disponible en deux tomes à :

<https://interstices.info/wp-content/uploads/2018/01/csunplugged2014-fr.pdf>



POSITION DE DÉPART DES TOURS DE HANOÏ



POSITION DES DISQUES APRÈS LE DEUXIÈME DÉPLACEMENT

- ▶ On ne peut déplacer qu'un seul disque à la fois
- ▶ On ne peut placer un disque que sur un emplacement vide ou sur un disque de diamètre supérieur

Si vous n'avez jamais joué à ce jeu, nous vous conseillons de commencer avec une tour de quatre ou cinq disques (vous trouverez dans les compléments électroniques de cet ouvrage un fichier PDF qui permet d'imprimer sur du papier une version de ce célèbre casse-tête) et de passer directement à la section suivante puisque nous allons, dans les lignes qui suivent, fournir un moyen de résoudre ce problème...

fr.wikipedia.org/wiki/Tours_de_Hano%C3%AF

Il existe plusieurs méthodes pour résoudre ce casse-tête (la notice Wikipédia² en détaille au moins quatre) et nous allons vous en livrer une pour vous montrer ce qu'est un algorithme. Si l'on part du principe que les trois mâts se nomment, de gauche à droite, A, B et C, la répétition des deux règles suivantes permet de résoudre le jeu :

- ▶ Prendre le disque le plus petit et le déplacer, un coup sur deux, en respectant la séquence de déplacement A vers B, B vers C, puis C vers A.
- ▶ Déplacer le seul disque autre que le plus petit disque pouvant être déplacé

Si l'on ne joue qu'avec quatre disques, numérotés de 1 à 4, 1 étant le plus petit disque et 4 le plus grand, le tableau suivant indique la liste des coups pour déplacer les quatre disques du mât de gauche vers le mât de droite. Le premier coup (B1) signifie que l'on déplace le plus petit disque sur le mât du milieu (B) et les cases en jaune foncé indiquent les déplacements du disque le plus petit.

B1	C2	C1	B3	A1
B2	B1	C4	C1	A2
A1	C3	B1	C2	C1

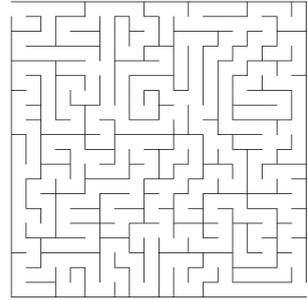
Comme tout bon algorithme qui se respecte, celui-ci permet de résoudre le casse-tête avec n'importe quel nombre de disques au départ (4, 5, 6, 7, 8, etc.).

SORTIR D'UN LABYRINTHE

Vous êtes de ceux que les labyrinthes angoissent et vous faites des cauchemars atroces où vous n'arrivez pas à trouver la sortie? Pas de panique! Un enfant de douze ans, John Pledge, a trouvé un algorithme qui vous permet de sortir à tous les coups d'un labyrinthe. Vous pouvez donc aller visiter les pyramides sans souci...

Vous trouverez le détail de cet algorithme dans cet article :

https://interstices.info/jcms/c_46065/l-algorithme-de-pledge



DIFFÉRENCES ENTRE ALGORITHME ET PROGRAMME

Maintenant que les notions d'algorithme et de programme sont plus précises dans votre esprit, vous vous demandez peut-être ce qui différencie finalement un algorithme d'un programme et si ces deux termes ne sont pas interchangeables.

Même s'il existe de grandes similitudes entre ces deux mots, ils ne désignent pas tout à fait la même chose. Un programme utilise des algorithmes et avant d'écrire un programme, il convient de déterminer les algorithmes qui permettront d'atteindre le résultat recherché. Un algorithme décrit de manière détaillée un traitement qui sera exécuté dans un programme et il est rare qu'un programme ne soit constitué que d'un seul algorithme. Un programme, même simple, peut contenir une dizaine d'algorithmes différents. Par exemple, si au cours d'un programme quelconque, on doit déterminer si un nombre est pair, on emploiera un algorithme particulier pour ce faire, mais cela ne constitue en général qu'une petite partie d'un programme.

Un programme ne se réduit pas seulement à ses algorithmes car il s'agit d'un ensemble complet qui comprend aussi ce que les informaticiens appellent le dialogue homme-machine, et que l'on nomme également **interface utilisateur**. Ainsi la conception d'un formulaire ou bien l'affichage d'informations sur la page d'un écran ou sur une imprimante font partie intégrante d'un programme, mais il n'y a pas vraiment d'algorithme sous-jacent, même s'il existe de nombreuses règles de conception pour élaborer des interfaces intuitives et conviviales.

Finalement, la plus grande différence entre un algorithme et un programme est le langage dans lequel il sera écrit. Les algorithmes sont toujours décrits dans un langage naturel (le français pour ce qui nous concerne) et si nous souhaitons écrire l'algorithme pour déterminer si un nombre est pair, on peut utiliser la formulation suivante (pour mémoire, une division euclidienne, également appelée division entière, ne fonctionne

qu'avec des entiers. Si je fais une division euclidienne de 35 par 4, le résultat est 8 et le reste est égal à 3) :

Réaliser une division euclidienne du nombre par deux. Si le reste est égal à 0, le nombre est pair, sinon il est impair.

Alors que si l'on veut coder cet algorithme dans un programme, il faudra utiliser un langage de programmation. Un algorithme est donc indépendant de tout langage de programmation. Quand on veut programmer, on doit déterminer le but du programme et le résultat à atteindre ; il faut ensuite définir les algorithmes nécessaires aux traitements à mettre en œuvre, puis on doit traduire les algorithmes dans un langage de programmation.

LES LANGAGES DE PROGRAMMATION

Un langage de programmation sert à traduire un algorithme en une suite d'instructions compréhensibles par un ordinateur, qui est une machine ne sachant fondamentalement gérer que des suites de 0 et 1, ce qu'on appelle un système binaire. Avant l'apparition des ordinateurs, on utilisait d'ailleurs des cartes perforées (pour tisser sur un métier Jacquard ou bien jouer de la musique sur un orgue de Barbarie), qui sont elles aussi des systèmes binaires (la carte est perforée ou n'est pas perforée).

POURQUOI EST-CE QU'IL EXISTE AUTANT DE LANGAGES DE PROGRAMMATION ?

Les débutants posent souvent cette question face au nombre impressionnant de langages de programmation disponibles. On pourrait déjà répondre qu'il existe aussi beaucoup de langues naturelles et que les tentatives pour imposer l'esperanto n'ont pas été très concluantes...

La première raison est que l'informatique est un domaine varié et qu'il existe des langages qui sont spécifiquement conçus pour une tâche particulière : ainsi JavaScript sert à programmer des pages Web, et il existe des langages de programmation pour gérer des animations graphiques ou bien des bases de données. Les langages de programmation sont considérés comme des outils et il est préférable d'utiliser le bon outil pour une tâche spécifique.

Les langages de programmation sont également liés aux caractéristiques du matériel que l'on trouve dans les ordinateurs et il est par conséquent normal qu'ils évoluent en fonction des performances croissantes des microprocesseurs. On notera également que certains langages n'existent que sur certaines plateformes (Windows ou Mac, par exemple).

Enfin, les codeurs sont des créatifs et ils rêvent d'avoir un outil qui leur ressemble, raison pour laquelle certains n'hésitent pas à créer leur propre langage de programmation, soit pour améliorer un langage plus ancien, soit parce qu'un nouveau besoin est apparu (le développement mobile, par exemple). Si ce nouveau langage apporte une réelle plus-value par rapport à d'autres langages et qu'il est facile à apprendre, il peut alors devenir populaire.

Il existe des modes dans les langages de programmation et certains langages sont plus en vogue que d'autres à un moment donné, mais comme dans la vraie vie, les modes passent...