

MATHIEU NEBRA

APPRENEZ À PROGRAMMER EN C

ENFIN UN LIVRE POUR LES DÉBUTANTS !



Issu du célèbre

Site du Zéro

www.siteduzero.com



www.siteduzero.com




Sauf mention contraire, le contenu de cet ouvrage est publié sous la licence :
Creative Commons BY-NC-SA 2.0

La copie de cet ouvrage est autorisée sous réserve du respect des conditions de la licence
Texte complet de la licence disponible sur : <http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>

Simple IT 2010 - ISBN : 978-2-9535278-0-3

Chapitre 1

Vous avez dit programmer ?

Difficulté : 

Vous avez déjà entendu parler de programmation et nul doute que si vous avez ce livre entre les mains, c'est parce que vous voulez « enfin » comprendre comment ça fonctionne.

Mais programmer en langage C... ça veut dire quoi ? Est-ce que c'est bien pour commencer ? Est-ce que vous avez le niveau pour programmer ? Est-ce qu'on peut tout faire avec ?

Ce chapitre a pour but de répondre à toutes ces questions apparemment bêtes et pourtant très importantes. Grâce à ces questions simples, vous saurez à la fin de ce premier chapitre ce qui vous attend. C'est quand même mieux de savoir à quoi sert ce que vous allez apprendre, vous ne trouvez pas ?



Programmer, c'est quoi ?

On commence par la question la plus simple qui soit, la plus basique de toutes les questions basiques. Si vous avez l'impression de déjà savoir tout ça, je vous conseille de lire quand même, ça ne peut pas vous faire de mal ! Je pars de zéro pour ce cours, donc je vais devoir répondre à la question :



Que signifie le mot « programmer » ?

Programmer signifie réaliser des « programmes informatiques ». Les programmes demandent à l'ordinateur d'effectuer des actions.

Votre ordinateur est rempli de programmes en tous genres :

- la calculatrice est un programme ;
- votre traitement de texte est un programme ;
- votre logiciel de « chat » est un programme ;
- les jeux vidéo sont des programmes (cf. fig. 1.1, le célèbre jeu Half-Life 2).



FIGURE 1.1 – Le jeu Half-Life 2 a été programmé en C++, un langage voisin du C

En bref, les programmes sont partout et permettent de faire a priori tout et n'importe quoi sur un ordinateur. Vous pouvez inventer un logiciel de cryptage révolutionnaire si ça vous chante, ou réaliser un jeu de combat en 3D sur Internet, peu importe. Votre ordinateur peut tout faire (sauf le café, mais j'y travaille).

Attention ! Je n'ai pas dit que réaliser un jeu vidéo se faisait en claquant des doigts. J'ai simplement dit que tout cela était possible, mais soyez sûrs que ça demande beaucoup de travail.

Comme vous débutez, nous n'allons pas commencer en réalisant un jeu 3D. Ce serait suicidaire. Nous allons devoir passer par des programmes très simples. Une des premières choses que nous verrons est *comment afficher un message à l'écran*. Oui, je sais, ça n'a rien de transcendant, mais rien que ça croyez-moi, ce n'est pas aussi facile que ça en a l'air.

Ça impressionne moins les amis, mais on va bien devoir passer par là. Petit à petit, vous apprendrez suffisamment de choses pour commencer à réaliser des programmes de plus en plus complexes. Le but de ce cours est que vous soyez capables de vous en sortir dans n'importe quel programme écrit en C.

Mais tenez, au fait, vous savez ce que c'est vous, ce fameux « langage C » ?

Programmer, dans quel langage ?

Votre ordinateur est une machine bizarre, c'est le moins que l'on puisse dire. On ne peut s'adresser à lui qu'en lui envoyant des 0 et des 1. Ainsi, si je traduis « Fais le calcul $3 + 5$ » en langage informatique, ça pourrait donner quelque chose comme¹ :

0010110110010011010011110

Ce que vous voyez là, c'est le langage informatique de votre ordinateur, appelé **langage binaire** (retenez bien ce mot!). Votre ordinateur ne connaît que ce langage-là et, comme vous pouvez le constater, c'est absolument incompréhensible.

Donc voilà notre premier vrai problème :



Comment parler à l'ordinateur plus simplement qu'en binaire avec des 0 et des 1 ?

Votre ordinateur ne parle pas l'anglais et encore moins le français. Pourtant, il est inconcevable d'écrire un programme en langage binaire. Même les informaticiens les plus fous ne le font pas, c'est vous dire !

Eh bien l'idée que les informaticiens ont eue, c'est d'inventer de nouveaux langages qui seraient ensuite traduits en binaire pour l'ordinateur. Le plus dur à faire, c'est de réaliser le programme qui fait la « traduction ». Heureusement, ce programme a déjà été écrit par des informaticiens et nous n'aurons pas à le refaire (ouf!). On va au contraire s'en servir pour écrire des phrases comme : « *Fais le calcul $3 + 5$* » qui seront traduites par le programme de « traduction » en quelque chose comme : « 0010110110010011010011110 ».

Le schéma 1.2 résume ce que je viens de vous expliquer.

1. J'invente, je ne connais quand même pas la traduction informatique par cœur.

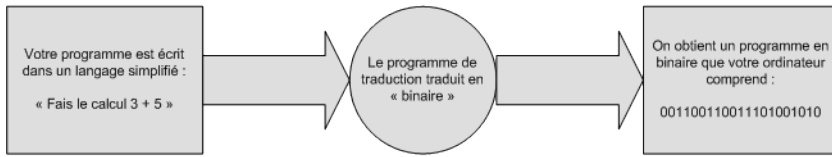


FIGURE 1.2 – Schéma (très simplifié) de réalisation d'un programme

Un peu de vocabulaire

Là j'ai parlé avec des mots simples, mais il faut savoir qu'en informatique il existe un mot pour chacune de ces choses-là. Tout au long de ce cours, vous allez d'ailleurs apprendre à utiliser un vocabulaire approprié. Non seulement vous aurez l'air de savoir de quoi vous parlez, mais si un jour (et ça arrivera) vous devez parler à un autre programmeur, vous saurez vous faire comprendre².

Reprenons le schéma que l'on vient de voir. La première case est « Votre programme est écrit dans un langage simplifié ». Ce fameux « langage simplifié » est appelé en fait **langage de haut niveau**. Il existe plusieurs niveaux de langages. Plus un langage est haut niveau, plus il est proche de votre vraie langue (comme le français). Un langage de haut niveau est donc facile à utiliser, mais cela a aussi quelques petits défauts comme nous le verrons plus tard.

Il existe de nombreux langages de plus ou moins haut niveau en informatique dans lesquels vous pouvez écrire vos programmes. En voici quelques-uns par exemple :

- le C ;
- le C++ ;
- Java ;
- Visual Basic ;
- Delphi ;
- etc.

Notez que je ne les ai pas classés par « niveau de langage », n'allez donc pas vous imaginer que le premier de la liste est plus facile que le dernier ou l'inverse. Ce sont juste quelques exemples³.

Certains de ces langages sont plus haut niveau que d'autres (donc en théorie un peu plus faciles à utiliser). On va voir notamment un peu plus loin ce qui différencie le langage C du langage C++.

Un autre mot de vocabulaire à retenir est **code source**. Ce qu'on appelle le code source, c'est tout simplement le code de votre programme écrit dans un langage de haut niveau. C'est donc vous qui écrivez le code source, qui sera ensuite traduit en

2. Certes, les gens autour de vous vous regarderont comme si vous étiez des extra-terrestres, mais ça il ne faudra pas y faire attention !

3. D'avance désolé pour tous les autres langages qui existent, mais faire une liste complète serait vraiment trop long !

binaire.

Venons-en justement au « programme de traduction » qui traduit notre langage de haut niveau (comme le C ou le C++) en binaire. Ce programme a un nom : on l'appelle **le compilateur**. La traduction, elle, s'appelle **la compilation**.

Très important : il existe un compilateur différent pour chaque langage de haut niveau. C'est d'ailleurs tout à fait logique : les langages étant différents, on ne traduit pas le C++ de la même manière qu'on traduit le Delphi.



Vous verrez par la suite que même pour le langage C il existe plusieurs compilateurs différents ! Il y a le compilateur écrit par Microsoft, le compilateur GNU, etc. On verra tout cela dans le chapitre suivant. Heureusement, ces compilateurs-là sont quasiment identiques (même s'il y a parfois quelques « légères » différences que nous apprendrons à reconnaître).

Enfin, le programme binaire créé par le compilateur est appelé **l'exécutable**. C'est d'ailleurs pour cette raison que les programmes (tout du moins sous Windows) ont l'extension « .exe » comme EXEcutable.

Reprenons notre schéma précédent, et utilisons cette fois des vrais mots tordus d'informaticien (fig. 1.3).

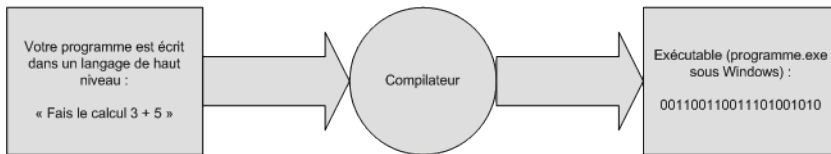


FIGURE 1.3 – Le même schéma, avec le bon vocabulaire

Pourquoi choisir d'apprendre le C ?

Comme je vous l'ai dit plus haut, il existe de très nombreux langages de haut niveau. Doit-on commencer par l'un d'entre eux en particulier ? Grande question.

Pourtant, il faut bien faire un choix, commencer la programmation à un moment ou à un autre. Et là, vous avez en fait le choix entre :

- **un langage très haut niveau** : c'est facile à utiliser, plutôt « grand public ». Parmi eux, on compte Python, Ruby, Visual Basic et bien d'autres. Ces langages permettent d'écrire des programmes plus rapidement, en règle générale. Ils nécessitent toutefois d'être accompagnés de fichiers pour qu'ils puissent s'exécuter (comme un interpréteur) ;
- **un langage un peu plus bas niveau** (mais pas trop quand même!) : ils sont peut-être un peu plus difficiles certes, mais avec un langage comme le C, vous allez en apprendre beaucoup plus sur la programmation et sur le fonctionnement de votre

ordinateur. Vous serez ensuite largement capables d'apprendre un autre langage de programmation si vous le désirez. Vous serez donc plus autonomes. Par ailleurs, le C est un langage très populaire. Il est utilisé pour programmer une grande partie des logiciels que vous connaissez.

Enfin, le langage C est un des langages les plus connus et les plus utilisés qui existent. Il est très fréquent qu'il soit enseigné lors d'études supérieures en informatique.

Voilà les raisons qui m'incitent à vous apprendre le langage C plutôt qu'un autre⁴. Je ne dis pas qu'il *faut* commencer par ça, mais je vous dis plutôt que c'est un bon choix qui va vous donner de solides connaissances.

Je vais supposer tout au long de ce cours que c'est votre premier langage de programmation, que vous n'avez jamais fait de programmation avant. Si par hasard, vous avez déjà un peu programmé, ça ne pourra pas vous faire de mal de reprendre à zéro.



Il y a quelque chose que je ne comprends pas... Quelle est la différence entre le langage « C » et cet autre langage dont on parle, le langage « C++ » ?

Le langage C et le langage C++ sont très similaires. Ils sont tous les deux toujours très utilisés. Pour bien comprendre comment ils sont nés, il faut faire un peu d'histoire.

- Au tout début, à l'époque où les ordinateurs pesaient des tonnes et faisaient la taille de votre maison, on a commencé à inventer un langage de programmation appelé **l'Algol**.
- Les choses évoluant, on a créé un nouveau langage appelé le **CPL**, qui évolua lui-même en **BCPL**, qui prit ensuite le nom de **langage B**.
- Puis un beau jour, on en est arrivé à créer un autre langage encore, qu'on a appelé... le **langage C**. Ce langage, s'il a subi quelques modifications, reste encore un des plus utilisés aujourd'hui.
- Un peu plus tard, on a proposé d'ajouter des choses au langage C. Une sorte d'amélioration si vous voulez. Ce nouveau langage, que l'on a appelé « C++ », est entièrement basé sur le C. Le **langage C++** n'est en fait rien d'autre que le langage C avec des ajouts permettant de programmer d'une façon différente.



Qu'il n'y ait pas de malentendus : le langage C++ n'est pas « meilleur » que le langage C, il permet juste de programmer différemment. Disons aussi qu'il permet au final de programmer un peu plus efficacement et de mieux hiérarchiser le code de son programme. Malgré tout, il ressemble beaucoup au C. Si vous voulez passer au C++ par la suite, cela vous sera facile.

Ce n'est PAS parce que le C++ est une « évolution » du C qu'il faut absolument faire du C++ pour réaliser des programmes. Le langage C n'est pas un « vieux langage oublié » : au contraire, il est encore très utilisé aujourd'hui. Il est à la base des plus grands systèmes d'exploitation tels Unix (et donc Linux et Mac OS) ou Windows.

4. On pourrait citer d'autres raisons : certains langages de programmation sont plus destinés au Web (comme PHP) qu'à la réalisation de programmes informatiques.

Retenez donc : le C et le C++ ne sont pas des langages concurrents, on peut faire autant de choses avec l'un qu'avec l'autre. Ce sont juste deux manières de programmer assez différentes.

Programmer, c'est dur ?

Voilà une question qui doit bien vous torturer l'esprit. Alors : faut-il être un super-mathématicien qui a fait 10 ans d'études supérieures pour pouvoir commencer la programmation ?

La réponse, que je vous rassure, est non. Non, un super-niveau en maths n'est pas nécessaire. En fait tout ce que vous avez besoin de connaître, ce sont les quatre opérations de base :

- l'addition ;
- la soustraction ;
- la multiplication ;
- la division.

Ce n'est pas trop intimidant, avouez ! Je vous expliquerai dans un prochain chapitre comment l'ordinateur réalise ces opérations de base dans vos programmes.

Bref, niveau maths, il n'y a pas de difficulté insurmontable. En fait, tout dépend du programme que vous allez réaliser : si vous devez faire un logiciel de cryptage, alors oui, il vous faudra connaître des choses en maths. Si vous devez faire un programme qui fait de la 3D, oui, il vous faudra quelques connaissances en géométrie de l'espace.

Chaque cas est particulier. Mais pour apprendre le langage C lui-même, vous n'avez pas besoin de connaissances pointues en quoi que ce soit.



Mais alors, où est le piège ? Où est la difficulté ?

Il faut savoir comment un ordinateur fonctionne pour comprendre ce qu'on fait en C. De ce point de vue-là, rassurez-vous, je vous apprendrai tout au fur et à mesure.

Notez qu'un programmeur a aussi certaines qualités comme :

- **la patience** : un programme ne marche jamais du premier coup, il faut savoir persévérer !
- **le sens de la logique** : pas besoin d'être forts en maths certes, mais ça ne vous empêchera pas d'avoir à réfléchir. Désolé pour ceux qui pensaient que ça allait tomber tout cuit sans effort !
- **le calme** : non, on ne tape pas sur son ordinateur avec un marteau. Ce n'est pas ça qui fera marcher votre programme.

En bref, et pour faire simple, il n'y a pas de véritables connaissances requises pour programmer. Un nul en maths peut s'en sortir sans problème, le tout est d'avoir la patience de réfléchir. Il y en a d'ailleurs beaucoup qui découvrent qu'ils adorent ça !

En résumé

- Pour réaliser des programmes informatiques, on doit écrire dans un **langage** que l'ordinateur « comprend ».
- Il existe de nombreux langages informatiques que l'on peut classer par niveau. Les langages dits de « haut niveau » sont parfois plus faciles à maîtriser au détriment souvent d'une perte de performances dans le programme final.
- Le **langage C** que nous allons étudier dans ce livre est considéré comme étant de bas niveau. C'est un des langages de programmation les plus célèbres et les plus utilisés au monde.
- Le **code source** est une série d'instructions écrites dans un langage informatique.
- Le **compilateur** est un programme qui transforme votre code source en **code binaire**, qui peut alors être exécuté par votre processeur. Les **.exe** que l'on connaît sont des programmes binaires, il n'y a plus de code source à l'intérieur.
- La programmation ne requiert pas en elle-même de connaissances mathématiques poussées⁵ ; néanmoins, il est nécessaire d'avoir un bon sens de la logique et d'être méthodique.

5. Sauf dans quelques cas précis où votre application doit faire appel à des formules mathématiques, comme c'est le cas des logiciels de cryptage.