

Préface

Peut-on encore imaginer la vie sans informatique ?

Autant imaginer la vie avant l'invention de la roue, avant les débuts de la métallurgie, avant peut-être même celle du feu et de la céramique, avant l'écriture, avant l'agriculture. Certes dans ces temps très anciens, les hommes et les femmes vivaient déjà, s'abritaient dans des cavernes, se déplaçaient, se nourrissaient, procréaient, élevaient des enfants puisque nous sommes là, leurs descendants. Nous savons aussi qu'ils soignaient leurs malades, enterraient leurs morts et qu'ils pensaient, qu'ils sculptaient le bois et la pierre, qu'ils peignaient sur les parois de leurs cavernes des scènes de leur vie quotidienne et couvraient de graffitis, qui devaient avoir un sens, les rochers du Mercantour ou du Val Camonica.

C'est parce qu'elle s'est immiscée dans toutes les activités essentielles de la vie que l'informatique a pris en soixante ans la place prépondérante qu'elle a désormais. Des journalistes français et américains se sont amusés à essayer de vivre pendant une certaine période sans acheter ou utiliser de produit ou d'appareil dont une partie ait été fabriquée en Chine et ils ont eu bien du mal à y parvenir. Essayez donc de faire la même chose en remplaçant la Chine par l'informatique, essayez de vivre sans acheter de produit dont la production ne soit partiellement informatisée, sans utiliser d'appareil qui ne contienne une puce électronique. Vous n'y arriverez pas : l'idée de vous retirer dans un coin perdu pour y cultiver votre jardin ne marchera pas, car il faudra vous passer d'électricité dont la distribution fait appel à des programmes informatiques sophistiqués, il faudra vous passer aussi d'outils métalliques : haut-fourneaux, laminoirs et chaînes de production d'objets en fer, étant contrôlés par des programmes informatiques. Et je crains que vous ne deviez vous passer aussi de semences dont la sélection, le tri, la distribution font sûrement appel à des techniques informatiques. Il va vous falloir revenir au temps de la cueillette et vous pourrez peut-être élever quelques animaux, une vache et de la volaille, mais vous ne pourrez ni les soigner ni vous soigner vous-même avec quelque médicament que ce soit, vous ne pourrez plus communiquer avec vos semblables que de vive voix, vous ne pourrez plus vous déplacer qu'à pied ou à cheval, encore qu'un cheval doive être ferré.

Cela paraît très étonnant, qu'une discipline qui a envahi la quasi-totalité des activités humaines ne figure pas parmi les grands domaines de la connaissance auxquels les écoles, collèges et lycées s'efforcent d'initier nos enfants, et pourtant cela l'est.

Mais le but de cette préface n'est pas de m'étonner de ce phénomène, ni d'en chercher les causes, il est d'abord de me réjouir que le programme d'informatique dans les classes préparatoires aux grandes écoles (CPGE) ait été récemment considérablement étendu, ce qui a motivé l'écriture du manuel auquel est destinée cette préface. Futurs ingénieurs, les élèves de ces classes ont plus que tous les autres besoin de connaître les bases de l'informatique à laquelle on est sûr qu'ils seront confrontés plus tard, dont on est sûr qu'ils auront besoin de se servir, et il vaut mieux qu'ils sachent s'en servir bien, à bon escient et en comprenant vraiment ce qu'ils font. La base, ce sont les deux notions d'algorithmes et de programmes. Les algorithmes ce

sont les méthodes, les recettes, les façons d'obtenir un certain résultat en enchaînant une suite d'opérations, le mot qui les désigne date du treizième siècle mais ils existent depuis que les hommes ont entrepris de faire quelque chose, débiter un silex ou chasser le mammoth. Babyloniens et grecs en ont déjà développé pour calculer sur des nombres il y a plus de deux mille ans. Les programmes ce sont les suites d'instructions à donner à une machine, un être humain, ou plus généralement un groupe composé d'êtres humains et de machines pour qu'il ou elle effectue un algorithme, c'est à dire réalise bien, dans le bon ordre les opérations prévues par l'algorithme. Les premiers programmes enregistrés sur un support permettant d'automatiser l'effectuation de l'algorithme semblent avoir été ceux des orgues mécaniques au seizième siècle, suivis de peu au dix-septième par ceux des métiers à tisser. Automatiser cela signifie qu'on donne le programme à la machine, qu'on le « lance » et que l'exécution se déroule sans autre intervention humaine.

L'informatique est vraiment née au début des années soixante quand algorithmes et programmes sont devenus des objets d'étude, c'est à dire quand on a commencé à développer des méthodes pour les concevoir, les écrire, les analyser, les transformer et aussi les accumuler. On a commencé simultanément à revoir, et à reprendre, à la lumière de ces méthodes nouvelles les innombrables algorithmes et programmes existants pour en améliorer les performances, en inventer de nouveaux, détecter dans les opérations assez complexes impliquant des opérateurs humains et des machines tout ce qui était automatisable c'est à dire que l'on pouvait confier à une machine.

Je suis venu, moi-même, à l'informatique entre 1960 et 1970, après des études de mathématiques, comme d'autres venaient de la physique, de l'astronomie ou d'ailleurs, puisque l'informatique n'existait pas et n'était pas enseignée. Et j'ai vécu, comme tous les gens de mon âge qui entraient en informatique, une succession de miracles qui allaient de la découverte des langages de programmation que l'on disait alors de haut niveau, Algol, Fortran, Lisp... à l'apparition d'algorithmes inattendus, déconcertants, défiant l'intuition, au moins l'intuition mathématique, pour résoudre des problèmes très simples, celui de Dijkstra pour chercher des plus courts chemins dans des graphes, celui de Morris et Pratt pour chercher les occurrences d'un motif dans un texte.

Très vite je me suis convaincu que les notions d'algorithmes et de programmes ne sont pas réductibles à d'autres, que ce que l'on peut appeler « algorithmique » et que ce que l'on peut appeler « programmation » sont bien des branches nouvelles de la connaissance, de la science et de la technique et j'ai développé une foi inébranlable en la fécondité de tous les efforts que l'on peut faire pour comprendre, inventer, concevoir, analyser, écrire, prouver, rendre exécutables et efficaces les algorithmes et les programmes. Rien n'est venu ébranler cette foi, au contraire, l'essor incroyable de l'informatique en tant qu'outil désormais utilisé par tous n'a fait que la raffermir. En près de quarante ans d'enseignement de l'informatique à l'Université, j'ai eu tout le loisir de constater que les notions d'algorithme et de programme ne s'assimilent pas en un jour, qu'il y faut du temps et que les mathématiques qui sont traditionnellement enseignées n'y préparent pas vraiment ou même rendent plus difficile chez les jeunes qui en ont fait le changement de mode de pensée et d'action que néces-

sitent ces deux notions. Le souci d'effectivité qui doit être constant en informatique rapproche cette discipline de la technologie, le métier d'informaticien de celui d'ingénieur, la rigueur de la syntaxe fait que la plus petite erreur, un point-virgule à la place d'une virgule, vicie tout et empêche le programme de « tourner » et ainsi la mise au point (qu'on appelle aussi debugging et débogage) c'est à dire la traque des fautes que contient une version du programme et leur correction est essentielle. On est dans une autre démarche : l'idée souvent simple d'un algorithme (souvent simple et parfois géniale) se transforme assez facilement en programme qui n'est que la traduction de l'algorithme en suite d'instructions dans le langage choisi, mais il faut encore tester celui-ci en le faisant « marcher » pour voir tout ce qui ne va pas et le corriger. De plus en plus souvent, en pratique, les tests ne suffisant pas il faut prouver que le programme fait bien ce que l'on voulait qu'il fasse, ce pourquoi il a été écrit, et qu'en se terminant il fournira bien le résultat cherché. La preuve de la correction et de la terminaison d'un programme est devenue un sujet en soi qui est assez déconcertant au premier abord.

J'étais ravi que le Ministère de l'Éducation Nationale ait décidé de donner un véritable enseignement d'informatique en CPGE, j'avais vu le programme ambitieux, surtout compte tenu du faible volume horaire imparti, mais qui contenait l'essentiel à mes yeux : une approche conceptuelle et pratique des notions fondamentales d'algorithme et de programme, prévoyant que les élèves aillent jusqu'à l'écriture et la mise au point de programmes en Python.

Mais je me demandais un peu comment tout cela allait être enseigné, sachant par expérience la difficulté pour des élèves d'assimiler ces notions très nouvelles pour eux et un peu orthogonales à la culture mathématique que l'on cherche à leur donner par ailleurs.

Et c'est là que j'ai reçu un volumineux manuscrit, que ses deux auteurs Amar Ousalah et Thierry Audibert me demandent de bien vouloir préfacer. Amar, maître de conférences d'informatique jusqu'à sa très récente retraite, a travaillé avec Jean Ichbiah et Alain Colmerauer, Thierry après avoir soutenu une thèse sur les fonctions de plusieurs variables complexes, a préféré enseigner les mathématiques en classe préparatoire, ce qu'il fait toujours au lycée Paul Cézanne d'Aix en Provence. Ils se connaissent depuis qu'ils étaient sur les bancs de la fac et ont l'habitude de travailler ensemble, Thierry s'est déjà beaucoup investi dans l'enseignement du très peu d'informatique que l'on enseignait en CPGE jusqu'à l'apparition du nouveau programme. Et je pense qu'à eux deux ils ont fait du très bon travail, qu'ils ont écrit un très bon manuel, où l'essentiel est dit, où les explications sont claires et précises, où les exercices sont judicieusement choisis et gradués. Les aspects scientifiques et les nécessités techniques sont constamment heureusement mélangés, leur démarche est bien la démarche informatique toujours constructive et ils l'ont adaptée au public bien particulier des CPGE, composé de professeurs de mathématiques qui vont enseigner et d'élèves pour qui la matière principale demeure les mathématiques.

Cet enseignement d'informatique est une gageure, l'informatique est une matière en pleine croissance et les élèves qui arrivent après le baccalauréat en CPGE n'y sont nullement préparés, on peut même dire que beaucoup qui ont manipulé des ordina-

teurs et des logiciels sans rien savoir des bases de l'informatique auront d'abord à apprendre qu'ils ne savent pas et qu'ils croyaient savoir car toute une littérature imbécile répète à l'envie que l'informatique n'a pas besoin d'être apprise et que les jeunes « ont ça dans le sang ».

Je pense sincèrement que l'ouvrage de Thierry et Amar peut contribuer au succès de cet enseignement et à persuader les élèves, nos futurs ingénieurs, que l'informatique mérite qu'on y réfléchisse même s'il n'est pas interdit de s'amuser en en faisant.

Maurice Nivat,

Professeur honoraire d'informatique à l'Université Paris 7 Denis Diderot