

TIFY

COMPRENDRE L'INFORMATIQUE

l'ordinateur
les mémoires
les programmes



Toute
SPECIAL
informatique

TIFY

Une série de 4 fascicules destinés à accompagner les 12 émissions hebdomadaires de TF1 « Tify... s'il te plaît, raconte-moi une puce », soit 1 fascicule pour 3 émissions. En vente chez votre marchand de journaux, parallèlement à la diffusion des émissions.
n° 1 : Histoire de puces – Les mémoires – Les programmes / **n° 2** : Les images – Les sons – Les jeux / **n° 3** : La bureautique – Les fichiers – Les réseaux / **n° 4** : Les robots – Les systèmes experts – La prospective.

Sommaire du n° 1

Histoire de puces

L'ordinateur, c'est simple, non ? – L'ordinateur est partout – L'ordinateur, ça vous concerne – Nous vivons parmi les puces – Pourquoi s'affoler ? – Il a bon dos l'ordinateur ! – L'ordinateur n'est pas neutre

Une longue histoire...

Mega, mini, micro

... qui a fait couler beaucoup d'encre

Le test de Turing / H.L. Dreyfus

Une machine « différente »

Premiers contacts – Autopsie du MO5 – La prise péritélévision – Quelques précautions – L'ordinateur, c'est une usine !

Mais au fond, que sait faire un ordinateur ?

Les engrenages de mon enfance / S. Papert

Les périphériques

Périphériques d'entrée – Périphériques de sortie – Périphériques entrée-sortie – Tous les périphériques

Dialogue avec l'ordinateur

Le clavier – Le curseur sur tout l'écran – Un langage pour la machine

Un peu de basic

Des instructions de toutes les couleurs ! – Des couleurs et des accents – L'ordinateur a de la mémoire

L'ordinateur, c'est aussi...

Les mémoires

Mémoire humaine et mémoire informatique

11 = 1011 ? – Le goût de l'orange

Quand l'informatique est transformée en courant électrique

Piano ou trombone à coulisse ?

Des 0 et des 1 : le règne du binaire

Bit – ...du binaire dans le savoir – Tableau des correspondances caractères/codes ASCII/octets

La mémoire centrale

Un peu de basic...

Du clavier à la mémoire

Comment sont constituées les mémoires centrales : un peu d'histoire

La mémoire à travers les âges

La mémoire électronique

Capacité – Prix – Rapidité – Adressage ou comment retrouver une adresse – Jouez avec votre mémoire

Les mémoires auxiliaires

Les cassettes magnétiques – L'enregistrement magnétique – L'enregistrement sur bande – Les disques souples (ou disquettes) – Initialisation ou formatage d'une disquette – Les mémoires de l'avenir

La bibliothèque Sainte Genetière

Cartes à mémoire : des puces dans votre portefeuille

Les programmes

Qu'est-ce qu'un programme ?

L'Ordinateur / J. Prévert

Algorithmes et organigrammes

Les algorithmes d'Al Khawarizmi – L'« engin analytique » – Recherche d'une économie de l'énoncé dans la description littéraire d'une tâche / J. Perriault – L'art et la manière d'aborder votre chef de service pour lui demander une augmentation ; exposé par organigramme / G. Perec

Les langages de programmation

Le langage machine

Les langages d'assemblage

Les langages évolués

Langage... / P. Dac – Récursivité – Interpréteurs et compilateurs

Un peu de logo...

Des listes de listes

Un peu de basic...

Un programme « professionnel »

La gestion centralisée du trafic marchandises de la SNCF

L'heure du choix

Commande de la série complète

Du n° 1 au n° 4 inclus : 105 FF – 70 FB/FL – 30,5 FS.

Commande de la reliure

La reliure TIFY, qui vous permettra de conserver vos 4 numéros, sera disponible chez votre marchand de journaux dès la parution du n° 3. Elle contiendra un sommaire général et un index/lexique.

Vous pouvez également la commander dès maintenant au prix de 45 FF – 295 FB/FL – 15 FS.

Commande de numéros

Vous pouvez vous procurer l'un quelconque des numéros au prix de 29 FF en ajoutant 6,50 F de frais de port pour le 1^{er} exemplaire et 2 F pour le suivant.

Toutes ces commandes doivent être accompagnées de leur règlement par chèque bancaire, chèque postal ou mandat, libellé à l'ordre de ALP-TIFY, à l'adresse suivante :

TIFY / SERVICE ABONNEMENTS
99, rue d'Amsterdam
75385 PARIS CEDEX 08

L'EDITEUR

ALP et Cie
26, rue des Carmes, 75005 PARIS.
Fondateur : Armand Beressi.

Directrice du marketing : Frédérique Janssen, adjoint : Georges Bensoussan.
Directrice de l'édition : Dominique Aubert, adjointe : Martine Chantenay.
Directrice de la production : Monique Muller, assistée d'Hélène Caumont et de Jacky Requet.
Vente aux dépositaires : Edi 7.
Distribution N.M.P.P.

LES ÉMISSIONS DE TF1

Conception et réalisation : Francis Warin.
Coproducteurs : TF1, l'Agence de l'Informatique et le Ministère de l'Éducation Nationale (CNDP).

Producteur exécutif : TF01. Direction générale : Jean-Claude Vernier. Administration : Yannick Casanova.

Communication et marketing : Sabine Couder Hiegel Conseil.

LA PUBLICATION

Directrice de la publication : Dominique Aubert.
Conseiller scientifique et pédagogique : Jacques Perriault.

Pour l'Agence de l'Informatique : Xavier Dalloz.
Auteurs du numéro 1 : Jean-François Boudinot, Christiane Frougny, Marie Gabriel, Jacques Perriault, Marianne Raveau, Pascal Rosier.

Feuilleton « Intox et Zoé » : Images, maquillage et costumes : Serge Diakonoff – Intox : Bernard Haller – Zoé : Annie Fratellini.

Maquette, dessins et couverture : LCA (Eric Turlet, Luc Royer, Jacques Morel).
Photographe : Luc Quaglia, assisté de Mathieu Prier.

Secrétariat de rédaction : Catherine Schram.
Photocomposition : Kappa.

Photogravure : Prism'offset.

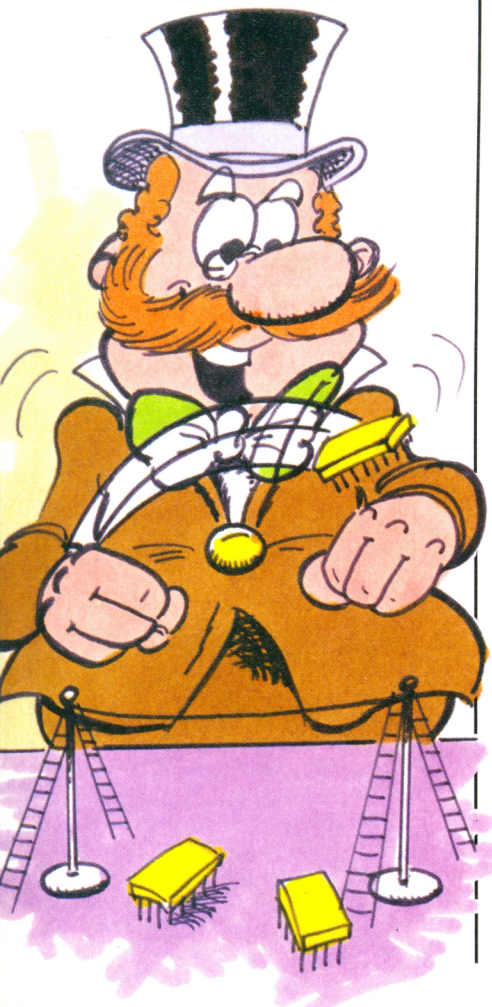
© 1984 ALP et Cie / TF01.
Dépôt légal : novembre 1984.
I.S.B.N. : 2-7365-006-7.

Crédits photos. Toutes les photos sont des photos ALP, sauf : Documents IBM p. 3, 42, 44 (en bas), 45, 57 – Rush : Philippe Boudin p. 26 (en bas), Michel Baret p. 28 (à droite) – SNCF-CAV : p. 76 (à droite), p. 77 (en haut à gauche, en bas à gauche), Michel Henri p. 26 (à droite), Bruno Vignal p. 76 (à gauche) et p. 77 (en haut à droite), Patrick Olivain p. 76 (au centre, en bas) – Photo Pictor International : p. 44 (en haut).

Les mots en gras dans le texte seront repris dans l'index/lexique livré avec la reliure.

Imprimé en Italie par IAG-Bergame.

Histoire de puces



Lorsqu'on aborde un ordinateur pour la première fois, on éprouve en général une sorte d'appréhension. Comme tout ça paraît compliqué ! Au simple coup d'œil, évidemment, qu'est-ce qu'un ordinateur ouvert sinon un entrelacs de fils, de circuits et de toutes sortes de « petits trucs » bien difficiles à identifier ? Mais, pourquoi regarder à l'intérieur ? Vous est-il déjà venu à l'esprit d'ouvrir, avant de les utiliser, votre poste de télévision ou votre téléphone, ou même de soulever le capot de votre voiture ? Si vous le faisiez, d'ailleurs, vous verriez que ce n'est pas très simple non plus. Mais évidemment le micro-ordinateur a hérité d'une mauvaise réputation : c'est le descendant de ces monstres, inapprochables par le commun des mortels, que furent les premiers gros ordinateurs, et cela a de quoi effrayer. Mais c'est une mauvaise raison pour se laisser décourager, car il doit bien y avoir quelques principes simples qui permettent de comprendre, sans trop entrer dans les détails, le fonctionnement général de cette fameuse machine.

Livrons-nous tout d'abord à un certain nombre de constatations :

- 1) l'ordinateur fonctionne à l'électricité, tout comme nos petites calettes qui marchent, elles, d'ailleurs plus généralement avec des piles. Il traite donc des **impulsions électriques**.
- 2) pour qu'il fasse un calcul, il faut que, par l'intermédiaire d'un clavier, je lui donne à combiner des informations traduites sous forme d'impulsions électriques : il y a donc, dans l'ordinateur, des **circuits** qui permettent de faire se rencontrer de telles informations. Ainsi, puisque je peux obtenir le résultat de l'opération $2+3$, il doit y avoir un circuit qui additionne.
- 3) lorsque

- 4) je tape quelque chose sur l'écran (une suite de caractères d'imprimerie, par exemple), l'ordinateur en garde la trace ; or, l'électricité, ça ne se conserve pas comme ça et on voit mal une pile suffisamment raffinée pour stocker des impulsions correspondant à des lettres, des chiffres, etc. Il y a donc dans l'ordinateur un dispositif de stockage : on l'appelle la **mémoire**.
- 4) je communique avec l'ordinateur ; il y a échange d'informations : moi, je lui en donne par le biais du clavier ou du crayon optique ; lui, grâce à son écran, me dit ce qu'il en fait. Il y a donc dans l'ordinateur des **dispositifs d'entrée et de sortie**.

En résumé, l'ordinateur est une machine qui manipule des informations, représentées sous forme d'impulsions électriques, en les faisant circuler à travers des circuits ou en les stockant dans des mémoires. Les informations lui sont communiquées grâce à un clavier et il affiche ses résultats sur un écran.

L'ordinateur, c'est simple, non ?

Au niveau des principes, oui. Mais la machine que nous connaissons aujourd'hui ne s'est pas faite en un jour, tant s'en faut ! C'est même une très longue histoire : depuis bien longtemps, en effet – allez savoir pourquoi – les hommes font des efforts énormes pour arriver à trouver des procédés de calcul automatique de plus en plus puissants. Euclide, déjà, du temps des Grecs, calculait « à la main », avec des petits cailloux (des *calculi*, qui ont donné le mot « calcul »).

Puis les Arabes comptèrent parmi eux le célèbre géomètre Al Khawarizmi dont le nom fut à l'origine du terme **algorithme**. C'était au X^e siècle après Jésus-Christ.

Plus tard, Leibniz et Pascal eurent l'idée de faire faire ces calculs par des machines... mécaniques ; et ce n'est que très récemment, lors de la seconde guerre mondiale, que l'électricité et l'électronique firent leur apparition dans ce domaine.

L'ordinateur est partout

Ou, du moins, son élément fondamental, cet ensemble de circuits qu'on désigne par le terme de **puce** – la pastille sur laquelle sont rassemblés les circuits est si minuscule qu'elle évoque inmanquablement cet animal (sans compter les pattes dont elle est entourée !). L'informatique,

notons-le, et c'est peut-être pour tenter de se raccommode avec la nature, aime le vocabulaire animal : on y parle de puces, de tortues, de souris, etc.

Des puces, on en trouve partout donc ! Des puces électroniques, évidemment : dans les montres, les calculatrices, les fours programmés... C'est pourquoi, que vous le vouliez ou non,

L'ordinateur, ça vous concerne !

vos impôts, vos contraventions, les distributeurs de billets de banque que vous utilisez, les publicités que vous recevez à domicile, tout y fait appel. Cela veut dire aussi, bien sûr, que nos noms et nos adresses figurent dans les mémoires d'ordinateurs auxquels nous n'avons jamais été présentés...



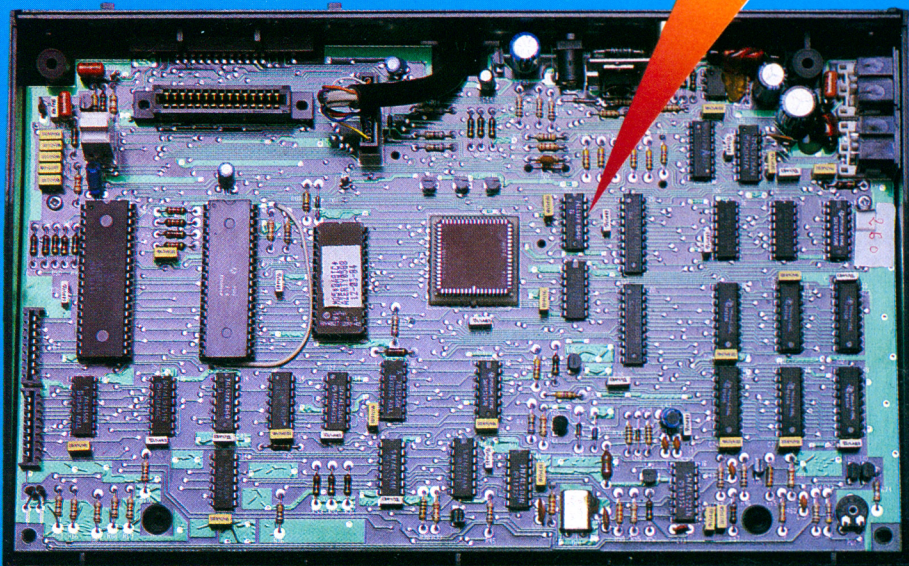
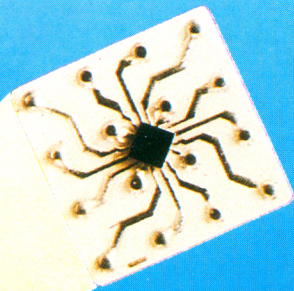
Nous vivons parmi les puces

Il faut donc nous faire à l'idée que, désormais, nous vivrons au milieu des ordinateurs et des puces. Mais pas comme des clochards, de préférence ! Notez que c'est l'une des options possibles : on se laisse envahir par les puces, on en achète tant et plus, on s'en sert (ou on fait semblant), puis on les jette. Ce sera une bonne aubaine pour ceux qui les construisent et savent les programmer ; et on verra peu à peu s'ériger un monde où certains, la puce à la main, commanderont à ceux qui seront incapables de comprendre, de critiquer ou de proposer des solutions nouvelles. Évidemment, une autre option existe : au lieu d'accepter sans savoir de quoi il s'agit, essayer d'entrer dans le secret. Tout le monde n'a pas, bien sûr, à devenir informaticien, car l'informatique n'est pas chose facile — et il faut le savoir avant de prendre la décision d'en faire son métier —, mais il est important pour tout le monde de ne pas rester inculte en la matière. Comprendre l'informatique est même une nécessité fondamentale pour les jeunes, car eux, ils veulent comprendre le monde dans lequel ils vivent. Et puis, n'est-ce pas eux

qui ont les idées neuves ? Prenez, par exemple, la machine à tester la tendreté des petits pois... Poétique, non ? Et bien ce sont des jeunes qui en ont eu l'idée : il s'agissait de construire un appareil destiné aux fabricants de conserves, lesquels sont confrontés au problème suivant : ou les petits pois sont trop durs et ils sont immangeables, ou ils sont trop mûrs et la conserve se transforme en bouillie ; la tendreté optimale est difficile à obtenir. Un tri s'impose donc, que seuls des palpeurs de petits pois pilotés par une puce étaient capables de réaliser ; il s'agissait, en effet, de palper les petits pois un par un ! Qui d'autre qu'une puce en aurait eu la patience ?

Et bien figurez-vous qu'ils en vendent de ces puces-là, et que c'est une innovation réelle, et que c'est ça l'industrie de demain ! Cela ne veut pas dire que l'informatique va résoudre toutes

les crises : ce serait aller vite en besogne. Mais comme beaucoup de pays ont fait de l'informatique un enjeu industriel, il y a quelque risque à ne pas regarder la puce en face.



Pourquoi s'affoler ?



Cette prise de conscience subite que l'ordinateur envahit progressivement notre vie s'accompagne d'ailleurs, pour beaucoup, d'une réaction fort compréhensible d'affolement, d'une crainte d'être rapidement submergé et finalement dépassé. En fait, rien ne justifie cette angoisse : l'ordinateur n'a pas fait irruption brutalement dans notre univers, c'est simplement, dans la grande ligne de notre développement technologique, un processus qui, en cours depuis une vingtaine d'années, a été accéléré, il est vrai, par l'apparition des **microprocesseurs**, lesquels ont entraîné une chute spectaculaire du coût des ordinateurs. L'idée n'est d'ailleurs pas nouvelle : il s'agit d'installer des automatismes là où il y a tâches répétitives, mécaniques en quelque sorte : une réservation de place de train ou d'avion, par exemple, se fait toujours de la même façon, alors pourquoi les employés d'agences ou de la SNCF ne se serviraient-ils pas de machines pour rendre leur travail plus facile ? Il est vrai évidemment que tout cela va de plus en plus vite, et que l'on a de plus en plus de mal à suivre le développement de ces techniques, un ordinateur chassant désormais l'autre à une vitesse surprenante...

Il a bon dos, l'ordinateur !

L'informatique suscite de nombreuses réactions, parfois extrêmement optimistes (« dans 10 ans, les machines penseront et travailleront à la place des hommes ! »), plus souvent dubitatives ou hostiles. Les erreurs imputées – à tort – à l'ordinateur ne se comptent plus, et l'informatique est responsable de tous les maux : chômage, crise économique, etc. C'est tout juste si les ordinateurs ne « détraquent pas le temps », puisque c'est bien connu : il n'y a plus de saisons ! De tout temps, les inventions nouvelles ont dû faire face à leurs détracteurs, d'autant plus dangereux que reconnus – parfois – parmi les « esprits éclairés » de

leur époque. Au siècle dernier, au premier rang des bêtes noires des réfractaires au progrès se trouve le chemin de fer. Accusées de provoquer des fluxions de poitrine et des pleurésies (François Arago, en 1836) ou des crises de nerfs et des dépressions (Jules Michelet, en 1861), les locomotives étaient plus considérées comme des « suppôts de Satan » que comme la marque du progrès !

« J'affirme sans hésiter que dans ces passages (les tunnels), les personnes sujettes à la transpiration seront incommodées, qu'elles gagneront des fluxions de poitrine, des pleurésies... » (François Arago).

« Il se pourrait que de prétendus « progrès » n'en fussent point. On s'en apercevra certainement quelque jour... Il y aurait moins d'alcooliques, premièrement, si la chimie avait fait moins de progrès. » (Ferdinand Brunetière).

« Il a été démontré par l'inutilité de mille tentatives, qu'il n'est pas possible qu'une machine, mise en mouvement par ses propres ressorts, ait assez de force pour s'élever et se soutenir en l'air. » (M. de Marles).

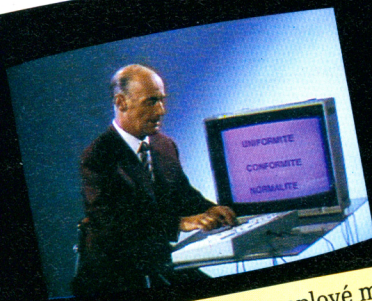
in *Dictionnaire de la Bêtise*, par Guy Bechtel et Jean-Claude Carrère.

L'ordinateur n'est pas neutre

Si personne n'est indifférent à l'égard de l'ordinateur, comme on peut l'être à l'égard d'une autre machine, c'est aussi qu'il contribue à modifier nos habitudes de vie. On le rencontre beaucoup sur les lieux de travail : terminaux bancaires, machines-outils à commande numérique, traitement de textes, etc. Certains vont jusqu'à parler de son « intelligence » : « est-il capable de penser et, un jour, pensera-t-il mieux que nous ? »

On n'en est guère là pourtant, car, à nous, il nous reste, entre autres, l'imagination !

Mais c'est vrai que l'ordinateur nous surprend parfois par ses découvertes : par exemple, trouver du pétrole là où personne ne pensait qu'il pouvait y en avoir ! L'explication n'est pas compliquée : c'est tout simplement qu'on le fait raisonner, très élémentairement certes, mais sur de très grandes quantités de données que, contrairement à nous (qui sommes très dissipés), il n'oublie jamais en cours de route !



Monsieur Plumeau, employé modèle, s'endort régulièrement sur son clavier... Il se transforme alors en un abominable robot, Intox 639, qui décide, avec l'aide de son complice, l'ordinateur numéro 1, de mettre l'univers à ses pieds ! Mais il y a une faille...



L'ordinateur numéro 1 suggère à son ami Intox 639, alias Monsieur Plumeau, de tout rationaliser, uniformiser, « conformiser ».



Alors, Intox 639 s'ingénie à mettre en cubes : les monuments, les aliments et... les dactylos et les mamans !



Sa secrétaire, la pauvre Zoé, réagit violemment : « Non, je ne veux pas être normalisée ! Non, je ne veux pas devenir un cube ! »

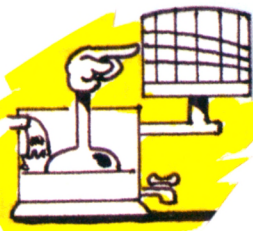
A ces cris, Monsieur Plumeau sort de son rêve.

Les automaticiens

Les **mécaniciens grecs de l'école d'Alexandrie**, dont Ktésibios, construisent des automatismes comprenant le principe de rétroaction (*feed-back*). On a gardé le souvenir de leurs appareils hydrauliques à mesurer le temps : les clepsydres.



En 50 av. J.-C., **Héron d'Alexandrie** met au point son distributeur automatique de vin : deux vases sont en communication ; lorsque le niveau du vin augmente dans le vase de régulation, le flotteur conduit à obturer le robinet qui verse le vin dans l'autre vase. On retrouvera ce principe dans la moderne chasse d'eau.



Au XVIII^e siècle, les **chancelleries florentines** utilisent le code binaire pour transmettre leurs messages diplomatiques.

Les mathématiciens

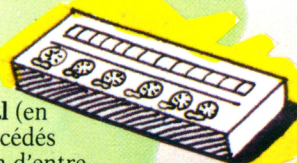
Euclide et **Pythagore** développent des procédés de calcul automatique ; notamment, en ce qui concerne Euclide, celui de la division.



Al Khawarizmi exerce les fonctions de géomètre dans la province de Bagdad. Ses actes répétitifs de mesure lui donnent l'idée d'une description formelle et générale du procédé. Son nom donnera le mot algorithme actuellement utilisé en informatique.

Les précurseurs les plus directs

Leibniz (en 1673) et **Pascal** (en 1642) construisent des procédés mécaniques de calcul. L'un d'entre eux, la Pascaline, figurera longtemps (jusqu'à une date récente) au nombre des instruments dont se serviront les comptables.



H. Hollerith, qui s'occupe du recensement américain, s'aperçoit, en 1889, qu'il est plus commode de traiter ses données à l'aide de cartes à perforations. C'est lui qui est à l'origine directe de la carte perforée que nous connaissons aujourd'hui.



Charles Babbage réalise, en 1834, une machine destinée à calculer automatiquement un type particulier d'équations. Pour cela, de nombreuses tringles sont nécessaires et l'inertie du dispositif l'empêchera finalement de fonctionner.

W. Burroughs construit, en 1890, les premières machines à calculer mécaniques à imprimantes.



Sources :

- Histoire de l'Informatique, par Marguerite Zientara (*Computer World*, 1981 - supplément à l'abonnement au Monde de l'Informatique).
- The Origins of Digital Computers - selected papers (recueil de textes édité par Brian Randell - Springer-Verlag, 1973).
- Eléments pour un dialogue avec l'informaticien, par Jacques Perriault (Mouton, 1971).
- Les Mécaniciens grecs, par Bertrand Gille (Le Seuil, 1980).

H. Aiken propose la construction de calculatrices automatiques à relais électromécaniques. Cette lignée sera sans suite, abandonnée du fait des progrès de l'électronique induits par la seconde guerre mondiale.

Eckert, Mauchly et Goldstine construisent la première calculatrice automatique électronique destinée à des usages balistiques : l'ENIAC (1943-1946).

Une longue histoire...

L'histoire de l'ordinateur n'est pas une histoire simple. Depuis la Haute Antiquité, des philosophes, des ingénieurs, des techniciens essaient de construire des automatismes. Au nombre de ceux-ci, et concernant plus particulièrement notre domaine : l'automatisation du calcul et la simulation des gestes.

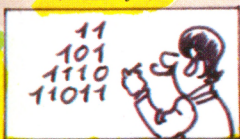
L'ordinateur d'aujourd'hui est redevable aux travaux de nombreux mathématiciens, automaticiens et cybernéticiens. Vers la fin du XIX^e siècle, prend également corps la notion d'**information** et de machines aptes à en traiter les représentations. C'est là l'œuvre des précurseurs les plus directs de l'informatique.

Les cybernéticiens

Jacques Vaucanson partage avec Louis XV le projet de construire un homme artificiel. Il en est resté aux produits intermédiaires, dont le *Joueur de galoubet* (1736), automate dans lequel exécution des fonctions et organe de commande ont été dissociés. On peut voir là l'ancêtre du concept de programme.

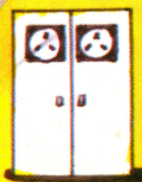


Le moulin à vent anglais d'**Edmund Lee** (1745) est doté d'un dispositif qui le maintient toujours face au vent et d'un mécanisme qui, confronté au problème des changements de la vitesse de ce dernier, contrôle la vitesse de rotation des ailes. Il s'inscrit dans la ligne de travail qui aboutira au concept de système.



En 1848, **George Boole** établit les principes du calcul binaire.

Post et Turing s'attachent, par la construction, en 1935, d'une machine abstraite (fonctionnant seulement sur le papier) et universelle, à formuler toutes les conditions pour qu'un calcul puisse être effectué de façon totalement automatique.



J. von Neumann joue un rôle décisif en participant à la construction du premier ordinateur universel et en publiant son ouvrage « *Computer as a brain* » qui est à l'origine de la solide métaphore du cerveau électronique.



W. Grey Walter construit des automates électroniques qui réagissent aux influences du milieu extérieur (tortues cybernétiques).

Ross Ashby élabore le concept de « machine qui apprend », 1952.



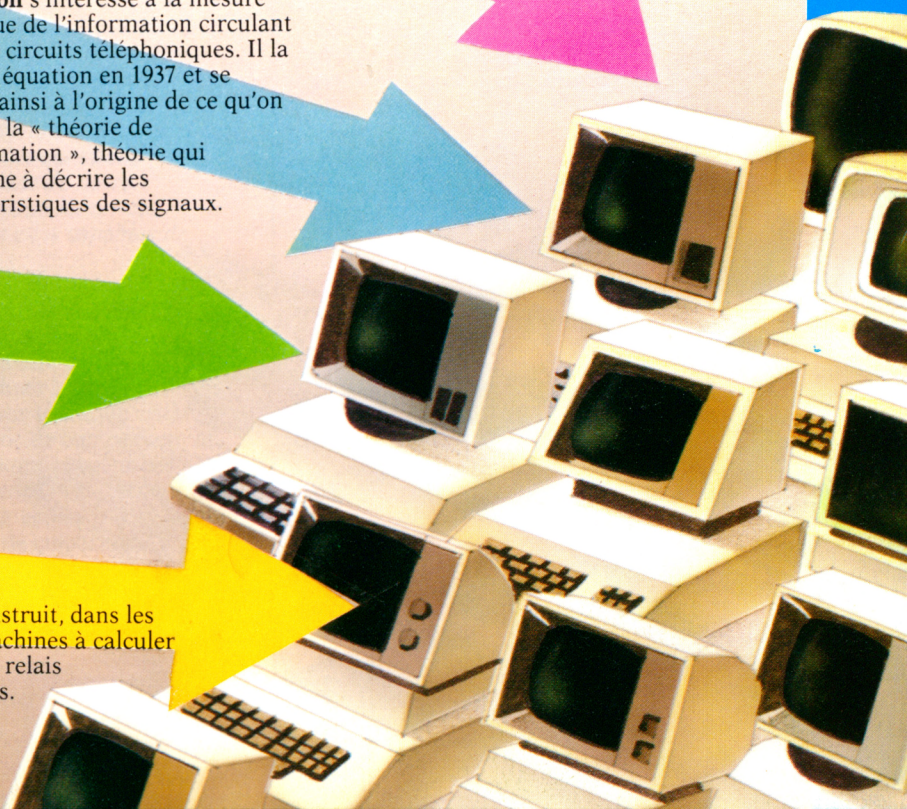
Norbert Wiener formule, en 1948, la théorie des systèmes asservis, ou cybernétique.

Jacquard met au point, en 1801, la commande du métier à tisser par des cartes perforées.



Shannon s'intéresse à la mesure physique de l'information circulant sur des circuits téléphoniques. Il la met en équation en 1937 et se trouve ainsi à l'origine de ce qu'on appelle la « théorie de l'information », théorie qui s'attache à décrire les caractéristiques des signaux.

Konrad Zuse construit, dans les années 30, des machines à calculer programmables à relais électromécaniques.



Mega, mini, micro

Actuellement, on distingue trois groupes d'ordinateurs : les mega, les mini, les micro.

Les mega-ordinateurs : ce sont d'énormes machines très coûteuses dont le fonctionnement nécessite la présence d'une foule d'opérateurs, programmeurs et analystes. A l'origine, tous les ordinateurs étaient ainsi, mais actuellement ils ne représentent qu'une faible part du parc total.

Les mega-ordinateurs sont des machines extrêmement puissantes qui sont capables d'effectuer, en un temps record, des calculs très compliqués sur un nombre important de données. Leur utilisation n'est malheureusement pas simple et exige un apprentissage. Ils sont réservés à l'usage professionnel et seulement aux très grosses sociétés ou administrations.

Les mini-ordinateurs : ce sont des ordinateurs plus petits, moins puissants et requérant généralement tout au plus deux opérateurs. Ils sont largement utilisés dans les domaines commerciaux et scientifiques.

Les micro-ordinateurs : encore plus petits et moins chers que les mini – certains sont si peu chers, en fait, qu'ils sont à la portée des usagers particuliers. Ce sont ceux dont nous allons essentiellement parler ici, non seulement dans leurs aspects techniques, mais également en ce qui concerne leur impact culturel.



En 1901, les machines à calculer étaient arrivées à leur dernier degré de perfection ! C'était du moins ce qu'on croyait...

« L'invention de cette machine remonte à 1818 et fut brevetée en 1820. Les expositions de 1823, 1849, 1851 et suivantes ont montré successivement la machine à calculer de Monsieur Thomas toujours améliorée et toujours simplifiée. Aujourd'hui cette œuvre remarquable est arrivée au dernier degré de perfection. »

in *Encyclopédie Universelle* – article « Arithmomètre ».

Après la seconde guerre mondiale, la technologie change, le langage aussi. On ne comprend plus les termes techniques. Pierre Dac le dit bien haut...

« Le Schmilblick des frères Fauderche est rigoureusement intégral, en ce sens qu'il peut à la fois servir de Schmilblick d'intérieur, grâce à la taille réduite de ses gorgomoches et de Schmilblick de campagne grâce à sa mostoblase et à ses deux glotosifres qui lui permettent d'urnapouiller les istioplocks même par les plus basses températures. »

in *Pierre Dac*, par Jacques Pessis (Seghers/humour)

... qui a fait couler beaucoup d'encre

Le test de Turing

En 1950, Turing publiait un article qui eut un certain retentissement, « Calculateurs et Intelligence », dans lequel il soulignait que « l'intérêt actuel pour les machines à penser a pris naissance avec l'apparition d'une catégorie particulière de machines, généralement appelées calculateurs numériques ou ordinateurs électroniques ». Et il posait cette question : « De telles machines sont-elles capables de penser ? » Pour délibérer de cette question, Turing propose un test qu'il nomme jeu d'imitation :

Ce nouveau type de problème peut être présenté sous forme d'un jeu, que nous appellerons « jeu d'imitation ». Il nécessite trois joueurs, un homme (A), une femme (B), et un troisième joueur qui posera les questions (C), et qui peut appartenir à l'un ou l'autre sexe. Ce dernier restera dans une pièce séparée de celle où se trouve le couple. Le but du jeu, pour ce troisième joueur, est de déterminer, chez les deux autres joueurs, qui est l'homme et qui est la femme. Il les connaît sous le nom de X et Y, et doit dire, au terme du jeu, soit « X est A et Y est B », soit « X est B et Y est A ». Pour ce faire, il a droit à des questions du type :

*« X peut-il me dire de quelle longueur sont ses cheveux ? »
Supposons que X est en fait A, c'est à X de répondre. La règle du jeu, pour A, c'est de tenter d'induire C en erreur. Par conséquent, il pourra répondre, par exemple :
« J'ai les cheveux coupés au carré,*

*les plus longues mèches atteignent environ vingt-deux centimètres. »
Pour éviter, naturellement, que les voix n'aident C, les réponses devront être écrites, si possible à la machine. Le mieux serait un téléimprimeur entre les deux pièces ; ou encore, autre solution, les questions et les réponses peuvent être transmises par un intermédiaire. Le but du jeu pour la joueuse (B) est d'aider le joueur (C). La meilleure tactique, en ce qui la concerne, est peut-être encore de ne donner que des réponses justes. Elle peut ajouter des déclarations telles que « C'est moi la femme, ne l'écoutez pas ! », mais comme le joueur A peut tout aussi bien émettre pareil discours, cela n'avancera pas à grand-chose. Et nous posons maintenant cette question : « Que se passera-t-il si l'on fait tenir par une machine le rôle de (A) dans ce jeu ? » Le joueur (C), en pareil cas, échouera-t-il aussi souvent que lorsque le jeu est joué avec un homme et une femme pour partenaires ? Ces questions prennent la place de notre interrogation première, « ces machines sont-elles capables de penser ? »*

Ce test est plus connu sous le nom de test de Turing. Les philosophes douteront peut-être qu'il soit permis d'attribuer le label de l'intelligence sur le simple critère d'une similitude de comportement, ce test de Turing n'en tient pas moins excellemment son rôle : celui d'objectif, pour ceux qui tentent effectivement de construire des

machines capables de réfléchir, ou celui de critère, pour ceux qui cherchent à évaluer les résultats obtenus.

Aucun ordinateur, on s'en doute, ne se porta immédiatement volontaire, aucun ne fut enrôlé sur-le-champ pour jouer au test de Turing. Malgré sa vitesse, sa précision, son universalité, le calculateur numérique n'était encore rien d'autre qu'un appareil à manipuler des symboles généraux. Cependant, pour le vieux pari de Leibniz, le compte à rebours était lancé. L'heure avait sonné de concevoir ces symboles appropriés, ces instructions détaillées, au moyen desquels les règles du raisonnement pourraient enfin être incorporées dans un programme de calculateur. Turing en avait entrevu la possibilité, il avait fourni un critère de succès – mais il concluait son article par des suggestions plutôt évasives sur ce qu'il convenait de faire ensuite.

in *Intelligence artificielle – mythes et limites*, par Hubert L. Dreyfus (Flammarion, 1984).

Une machine « différente »

Qu'y a-t-il de commun entre une moulinette à légumes, une rotative de presse, un tank et une centrale nucléaire ? Question apparemment étrange mais dont la réponse est sérieuse : toutes ces machines ne savent faire qu'une seule chose : mouliner, imprimer, tirer ou produire de l'électricité. Quant à l'ordinateur, c'est radicalement différent : bien que toujours simple machine, il peut faire trente-six choses différentes : des calculs scientifiques, des jeux, la commande de robots, l'établissement de contraventions, etc. Car si l'ordinateur lui-même, le **matériel**, reste toujours le même, la suite d'ordres qu'on lui donne sous forme de codes, le **logiciel**, peut, lui, être changé : un même matériel peut traiter différents logiciels.

Essayons donc de nous approcher un peu plus près et de comprendre ce qu'est et comment fonctionne un micro-ordinateur. Nous prendrons, puisqu'il faut en prendre un, comme exemple et support de nos démonstrations le MO5 de Thomson.

Premiers contacts

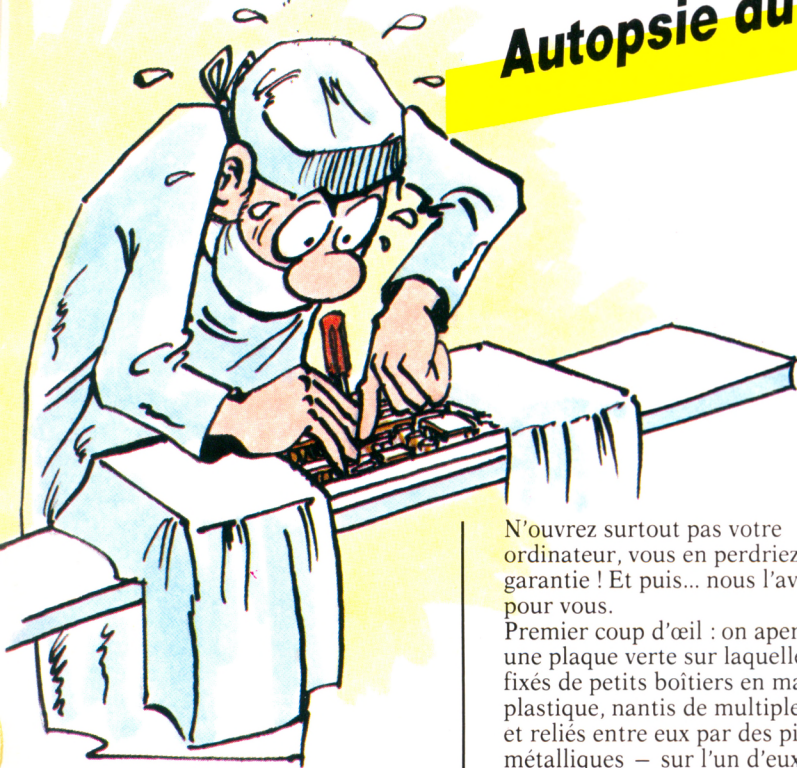
Le MO5 est sur la table, raccordé au téléviseur par une **prise péritel** – on appelle ainsi la longue broche à 20 plots que l'on a enfichée derrière le téléviseur. Sur le devant de l'appareil, des touches (voir clavier). En haut à gauche, une fenêtre à deux battants, avec un petit bouton

à côté ; on peut y enfoncer des **cartouches**. Sur le côté droit, deux prises, l'une pour brancher le crayon optique, l'autre le magnétophone. A l'arrière, une prise pour l'alimentation, via un transformateur et un connecteur – sur lequel on pourra enficher différents **interfaces** (modules permettant de relier des **périphériques** à l'ordinateur). Le petit bouton blanc à côté de la fenêtre sert à **réinitialiser** le système (le faire repartir à zéro).

Attention, il faut toujours allumer le téléviseur avant l'ordinateur et, à la fin du travail au contraire, éteindre l'ordinateur avant le téléviseur. Il ne faut jamais non plus brancher un interface quand l'ordinateur est allumé.



Autopsie du MO5



N'ouvrez surtout pas votre ordinateur, vous en perdriez la garantie ! Et puis... nous l'avons fait pour vous.

Premier coup d'œil : on aperçoit une plaque verte sur laquelle sont fixés de petits boîtiers en matière plastique, nantis de multiples pattes et reliés entre eux par des pistes métalliques – sur l'un d'eux est écrit « 6809 E » : c'est le **microprocesseur**, c'est-à-dire l'organe central de tout le dispositif.

Les autres boîtiers sont de trois sortes, que seul un informaticien averti peut distinguer, grâce aux codes qui y sont inscrits :

- les **RAM** (ou **mémoires vives**) ;
- les **ROM** (ou **mémoires mortes**) ;
- les **boîtiers d'interfaçage** qui permettent les connexions avec l'extérieur.

Tout ce joli monde est, de plus, commandé par une **horloge** à quartz qui le fait vibrer un million de fois par seconde.

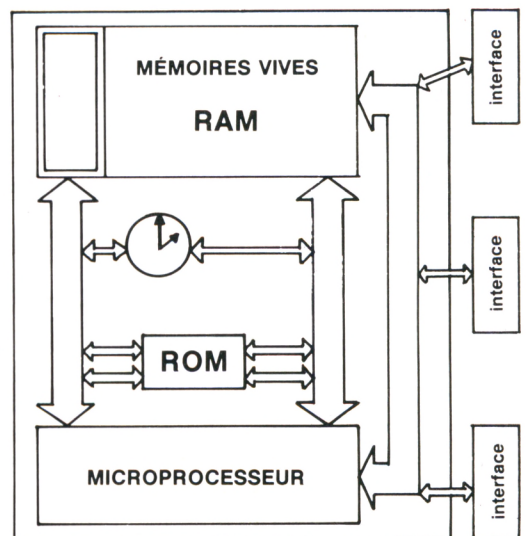
Les pistes métalliques sont, elles, de 4 types :

- le **bus de données** sur lequel circulent les informations à traiter ;
- le bus de commande ;
- le bus d'**adresses** ;
- les fils d'alimentation électrique.

Dès qu'on a allumé l'ordinateur, l'horloge s'est mise à vibrer et le microprocesseur a effectué quelques milliers d'opérations : il a tout simplement examiné le contenu de certaines mémoires (les ROM), a manipulé les informations qu'il y a trouvées, puis a envoyé des signaux électriques sur l'interface connectée au téléviseur, ce qui a provoqué un premier affichage à l'écran. Cette situation restera stable tant que personne n'interviendra en appuyant sur les touches du clavier (ce qui aura pour

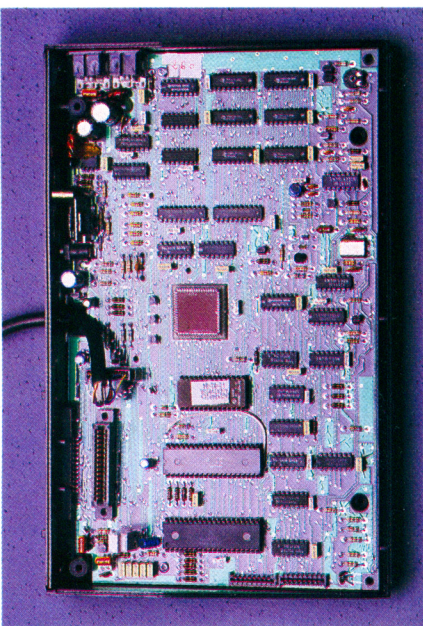
effet de faire apparaître les lettres correspondantes sur l'écran). L'ordinateur est, en fait, à l'écoute du clavier.

Dès qu'une touche est frappée, son code spécifique est envoyé, par l'intermédiaire de l'interface du clavier, puis du bus des données, jusqu'au microprocesseur. A ce moment-là, une action spécifique se déclenche ; elle a deux objets : conserver provisoirement le code en question et trouver l'image du caractère qui lui correspondra sur l'écran.



Tout ceci se passe durant le bref délai qui s'écoule entre le moment où on frappe sur une touche et celui où on voit apparaître une lettre sur l'écran.

On frappe donc sur des touches jusqu'à écrire toute une phrase, qu'on « envoie » en appuyant enfin sur la touche **ENTRÉE**, dont le code particulier déclenche une autre série d'opérations : vider la mémoire qui avait été utilisée provisoirement, puis envoyer cette phrase pour qu'elle soit lue et analysée par un **programme**, l'**interpréteur Basic** dont le rôle essentiel est de reconnaître les mots autorisés et de les stocker en mémoire sous la forme de codes. S'il ne reconnaît pas un mot, il le dit en affichant « **Error 2** ».



La prise péritélévision

Depuis 1984, tous les téléviseurs couleurs doivent absolument posséder une prise péritélévision (péritel) pour le branchement des micro-ordinateurs, des magnétoscopes, etc.

Si votre téléviseur est antérieur à cette date, il se peut qu'il n'ait pas de prise péritel. Si vous désirez cependant utiliser un micro-ordinateur, deux solutions sont possibles :

- faire installer une prise par votre réparateur habituel ;
- acheter un adaptateur, qui permet de brancher votre ordinateur sur la prise antenne (attention, l'image sera de moins bonne qualité).

Quelques précautions

S'il s'agit d'un matériel fiable, l'utilisation d'un ordinateur ne pose aucun problème, à condition toutefois de respecter les conseils de prudence suivants :

- vérifier tous les branchements avant de relier les prises au secteur et d'actionner les interrupteurs ;
- toujours allumer la télévision avant l'ordinateur ;

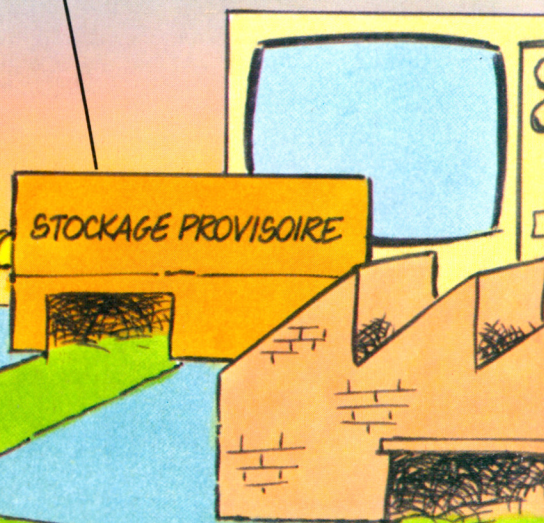
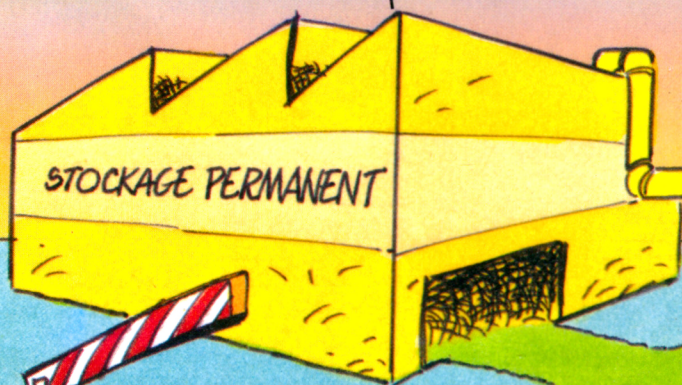
- ne jamais mettre une cartouche dans la trappe lorsque l'ordinateur est sous tension ;
- ne jamais brancher un périphérique lorsque l'ordinateur est sous tension ;
- ne pas ouvrir le boîtier, et surtout ne jamais toucher les composants électroniques avec les doigts.

L'ordinateur, c'est une usine !

On pourrait comparer l'ordinateur, dans son fonctionnement, à une grande usine comprenant :

- des hangars de stockage permanent et des hangars de stockage provisoire : les mémoires ;
- un atelier de travail et d'usinage : le microprocesseur ;
- un centre de publication : la vidéo ;
- une horloge qui synchronise les opérations et donne la cadence ;
- des sas qui font communiquer le dedans et le dehors : les interfaces ;
- une alimentation électrique sans quoi rien ne fonctionnerait ;

- des boulevards de circulation : les bus par lesquels les matériaux (sous forme d'impulsions électriques) circulent à la cadence d'un millionième de secondes. Le micro-ordinateur est également entouré d'éléments extérieurs, de partenaires, que l'on appelle périphériques et qui sont, en quelque sorte, ses fournisseurs d'informations brutes et ses acheteurs d'informations traitées ;
- le clavier envoie les informations et formule les demandes de travaux ;
- le moniteur vidéo (qui peut être un poste de télévision) visualise les résultats demandés.



Mais, au fond, que sait faire un ordinateur ?

Un ordinateur sait, par la voie de son microprocesseur, effectuer les opérations élémentaires suivantes :

Il **affecte** un nombre à une case-mémoire, de la même façon que l'armée affecte une recrue à un

régiment ou l'administration un employé à un bureau.

Il **additionne** des nombres de la façon la plus simple qu'on puisse imaginer, c'est-à-dire en travaillant sur des additions de 0 et de 1. Les ingénieurs ont, en effet, su ramener tous les calculs à l'opération élémentaire qu'est l'addition. Ils ont construit pour cela des **circuits** permettant d'effectuer ces derniers

(l'unité logique et arithmétique). C'est donc le nombre d'opérations effectuées à la seconde (environ un million) qui explique que l'ordinateur nous donne l'impression de faire rapidement des opérations très compliquées. Il ne fait en réalité qu'une longue liste de calculs très simples.

Il **compte** les instructions de travail qui lui sont données. Tel un employé consciencieux, il prend

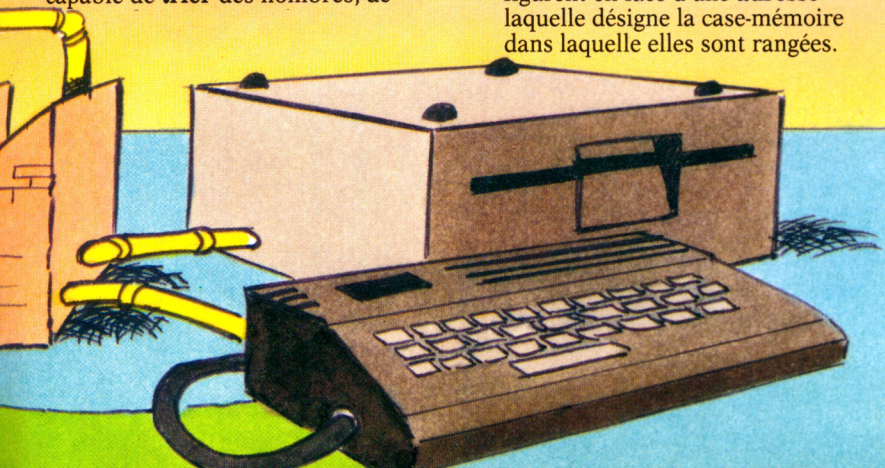
connaissance de chaque directive et l'enregistre.

Il **compare**. Il est capable de savoir si un nombre est plus grand ou plus petit qu'un autre ou s'ils sont tous deux égaux ; ce qui lui permet, en

fonction de ce qu'il a trouvé, d'orienter le travail dans une direction ou une autre.

A l'aide de ces fonctions très simples, mais effectuées un très grand nombre de fois dans un temps très court, l'ordinateur est capable de **trier** des nombres, de

classer des données selon différents critères et de rechercher dans des **tables** (tableaux contenant des données) des informations qui figurent en face d'une **adresse** — laquelle désigne la case-mémoire dans laquelle elles sont rangées.



Les engrenages de mon enfance

Je n'avais pas deux ans quand me vint la passion des voitures. Les noms des pièces automobiles prirent une place importante dans mon premier vocabulaire : j'étais tout fier, en particulier, de connaître les pièces du système de transmission, la boîte de vitesses et, plus encore, le différentiel. Il me fallut, bien sûr, attendre plusieurs années avant de comprendre comment fonctionne un engrenage ; mais à partir de ce moment-là les engrenages devinrent pour moi une source de jeux de prédilection. J'adorais faire rouler l'un sur l'autre des objets circulaires, en leur imprimant des mouvements d'engrenage ; quand je reçus un jeu de construction, évidemment, ma première réalisation fut un système d'engrenages rudimentaire.

Je devins expert à faire tourner dans ma tête des roues dentées, et à suivre des enchaînements de cause à effet : « Celle-ci tourne dans ce sens-là, donc celle-là doit tourner dans ce sens-ci, donc... » Je tirais plaisir, surtout, des systèmes comme celui de l'engrenage différentiel, qui ne suit pas une simple chaîne de cause à effet linéaire, puisque le mouvement de l'arbre de transmission peut se distribuer aux deux roues de manière différente selon la résistance rencontrée. J'ai un souvenir très vif de mon excitation le jour où je découvris qu'un système peut être ordonné, réglé, compréhensible, sans pour autant perdre toute souplesse d'adaptation.

Je crois que d'avoir tant joué avec des différentiels m'a davantage ouvert aux mathématiques que tout ce qui m'a été enseigné à l'école primaire. Les engrenages, en me servant de modèles, ont fait entrer dans ma tête des notions qui risquaient de demeurer abstraites. J'ai en mémoire deux exemples précis, tirés de mes souvenirs d'écolier : les tables de multiplication, d'abord, que je vis sous forme d'un système d'engrenages ; et puis mon premier contact avec une équation à deux inconnues ($3x + 4y = 10$, par exemple), laquelle immédiatement m'évoqua un différentiel. Sitôt que j'eus construit dans ma tête un système d'engrenages figurant la relation entre x et y , et calculé combien de dents il fallait à chaque engrenage, l'équation m'apparut comme une vieille amie.

in *Jaillissement de l'esprit*, par Seymour Papert (Flammarion, 1981).

Les périphériques

Sans ce que l'on appelle ses périphériques, l'ordinateur serait comme un cerveau qui n'aurait ni corps ni organes sensoriels pour communiquer avec l'extérieur. De même que le cerveau peut, au moyen des sens, recevoir des informations de l'extérieur et en émettre d'autres grâce à différents moyens de communication tels le langage, l'écriture, etc., l'ordinateur doit disposer d'appareils lui permettant de recevoir des informations : ce sont les périphériques d'entrée ; et

d'appareils lui permettant de communiquer des informations à son utilisateur : ce sont les périphériques de sortie. Ces périphériques sont des appareils remplissant chacun une fonction spécifique et connectés à l'ordinateur par une prise.

Il existe donc des périphériques qui servent uniquement à entrer des informations, d'autres qui servent uniquement à les ressortir et d'autres, comme les lecteurs de cassettes ou de disquettes, qui remplissent les deux rôles.

Périphériques d'entrée

Le périphérique d'entrée le plus connu et le plus utilisé est le **clavier**. Ce clavier est analogue à celui d'une machine à écrire ; il comporte des touches avec des caractères, des touches avec des chiffres et d'autres avec des signes, comme « + », par exemple. Il peut en outre, comme le clavier du MO5, disposer de touches correspondant à des fonctions spécifiques de l'ordinateur et reproduisant des instructions en un langage compréhensible par lui. Ce clavier joue un peu, pour l'ordinateur, le rôle de la vue et de l'ouïe.

Les **manettes de jeu** et les **souris** permettent, elles aussi, de transmettre des informations à la machine. Les informations sont alors entrées sous forme graphique. La manette de jeu (ou *joy-stick*), par le mouvement de son levier, ou la souris, par la rotation de sa boule, déterminent la position d'un point sur l'écran par ses coordonnées cartésiennes ; les coordonnées verticales et horizontales étant comptées à partir du haut et de la gauche de l'écran.

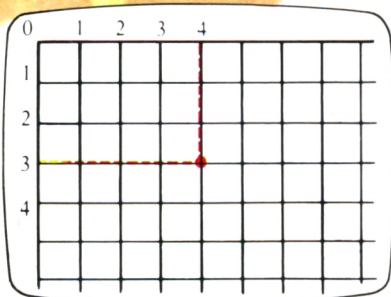
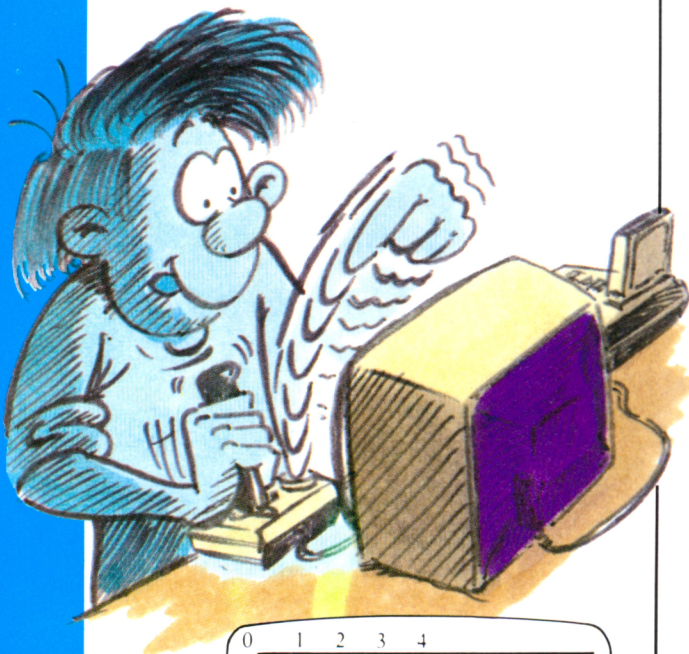
La position d'un point est donc définie par le croisement de deux lignes correspondant à ses coordonnées. Les déplacements du levier ou de la boule transmettent à l'ordinateur des impulsions correspondant à des variations de coordonnées que celui-ci traduit par le déplacement, sur l'écran, d'un

curseur ou d'un dessin, s'il s'agit de jeux vidéo.

Le **crayon optique** est un peu basé sur le même principe mais ses applications et son maniement sont différents.

Il a la forme d'un crayon qui aurait à son extrémité, non une mine, mais une sorte de petite « caméra », un composant électronique sensible à la lumière, capable de repérer, quand on le promène sur l'écran, la position du point lumineux sur lequel il est posé : sur un écran, en effet, les images sont engendrées périodiquement par un spot lumineux qui se déplace avec une vitesse connue ; alors, quand on positionne le crayon optique à un endroit donné de l'écran, l'apparition du spot à un moment précis permet à l'ordinateur de calculer le chemin parcouru par ce dernier depuis le début de l'image et donc la position du point. On peut utiliser le crayon optique pour faire des dessins : tous les points sur lesquels il passe sont enregistrés et retransmis à l'écran. Il peut aussi être utilisé pour faire un choix dans un catalogue affiché : une liste d'options est écrite et dès qu'on pointe le crayon sur l'option choisie, l'ordinateur met en route le programme correspondant.

Le **lecteur de codes barres** est, lui aussi, une sorte de crayon qui aurait à son extrémité un dispositif capable d'enregistrer un signal. C'est un appareil qui est utilisé essentiellement dans le cadre des problèmes de distribution. Il permet de lire, en promenant le lecteur sur les étiquettes de certains produits, le code qui y est inscrit, sous forme de traits noirs plus ou moins nombreux et plus ou moins larges. Ce code est un code standard fournissant des informations sur le produit : son prix, sa nature, etc.



Périphériques de sortie

Qu'il s'agisse de celui d'un simple téléviseur ou de celui d'un appareil spécifique appelé moniteur vidéo, **l'écran** est un outil indispensable pour dialoguer avec l'ordinateur. Il peut être, soit cathodique (comme celui des téléviseurs), soit à cristaux liquides (comme celui des afficheurs de calculatrices électroniques) ; soit noir et blanc, soit en couleurs. Il peut contenir, au total, 1 000 caractères, soit 25 lignes de 40 caractères chacune.

Nouveau venu sur le marché, **l'écran tactile** permet, en y posant le doigt, de choisir une option dans un catalogue, tout comme avec le crayon optique.

C'est sur l'écran que se matérialise le dialogue entre l'utilisateur et sa machine. Il affiche, en effet :

- ce que l'on tape au clavier ;
- les messages de l'ordinateur (résultats d'une opération, messages d'erreur, etc.) ;
- le déroulement d'un programme, texte ou images.

Notons que, si l'ordinateur effectue une opération ou exécute un programme, on ne voit pas s'afficher à l'écran le contenu du travail qu'il exécute mais seulement son résultat. En revanche, si l'on veut connaître les instructions qui composent un programme, on peut, dans certains cas, demander à l'ordinateur de les afficher à l'écran. C'est ce que l'on appelle le **listing** d'un programme.

Ce listing est l'un des nombreux cas qui justifient l'existence d'un autre périphérique : **l'imprimante**. Celle-ci permet, en effet, d'écrire sur papier certaines choses contenues dans les mémoires de l'ordinateur : textes, images ou listes d'instructions composant un programme. Gérée par l'ordinateur, elle doit, bien sûr, lui être reliée pour pouvoir fonctionner.

Quand on veut faire sortir un listing sur imprimante, on doit en donner l'ordre à l'ordinateur en utilisant l'instruction adéquate.

Il existe un grand nombre de types d'imprimantes qui ont des caractéristiques techniques et des possibilités différentes.

L'imprimante thermique nécessite un papier spécial, assez coûteux.

Par contre, son impression de bonne qualité et son prix de revient raisonnable en font un bon instrument pour la micro-informatique familiale.

L'imprimante à impact travaille sur du papier ordinaire.

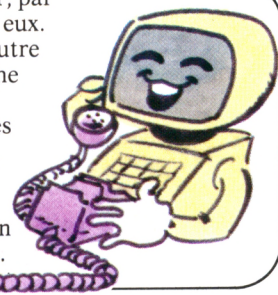
L'imprimante à jet d'encre est rapide et silencieuse, mais d'un prix de revient élevé.

L'imprimante à marguerite est lente mais son impression est excellente. Malheureusement, son prix de revient est aussi très élevé.

L'imprimante à laser est très rapide et son impression est aussi bonne que celle de l'imprimante à marguerite. Bien sûr ces qualités en font un article très coûteux.



Moins répandu chez les particuliers, le **modem** permet de connecter, par l'intermédiaire du réseau téléphonique, plusieurs ordinateurs entre eux. C'est un appareil qui est branché, d'une part à un ordinateur, et d'autre part à une prise de téléphone. Géré par l'ordinateur et mis en marche par l'utilisateur au moyen d'instructions, il transforme le langage binaire de l'ordinateur en une fréquence transmissible par les câbles téléphoniques. De cette façon, un programme, ou même des ordres, peuvent être transmis à un autre ordinateur, lui aussi connecté au réseau téléphonique par un autre modem qui transforme alors la fréquence transmise par câbles téléphoniques en langage binaire. Un modem (MODulateur-DEModulateur) fonctionne dans les deux sens.



Périphériques entrée-sortie

Les **lecteurs de cassettes** ou de **disquettes** jouent à la fois le rôle de périphériques d'entrée et celui de

périphériques de sortie. Ils servent à entrer dans les mémoires de l'ordinateur des programmes, ou logiciels, enregistrés sur cassette ou disquette. Et, inversement, ils

permettent d'enregistrer un programme écrit par un utilisateur. Dans le cas du lecteur de cassettes, il s'agit souvent d'un simple magnétophone à cassettes relié à l'ordinateur par une prise adaptée. Leur usage et leurs caractéristiques seront étudiés dans le cadre des mémoires de l'ordinateur.

Périphériques

Imprimante, crayon optique, lecteur de codes barres, clavier, écran, modem, cartouche, lecteur de cassettes, lecteur de disquettes, joy-stick, souris.

souris

La souris se présente sous la forme d'une petite boîte en plastique de laquelle dépasse une boule métallique. On manie cette boule en la roulant sur une surface plane. Elle a un peu le même rôle que le levier du joy-stick. Elle déplace sur l'écran un point lumineux.

écran

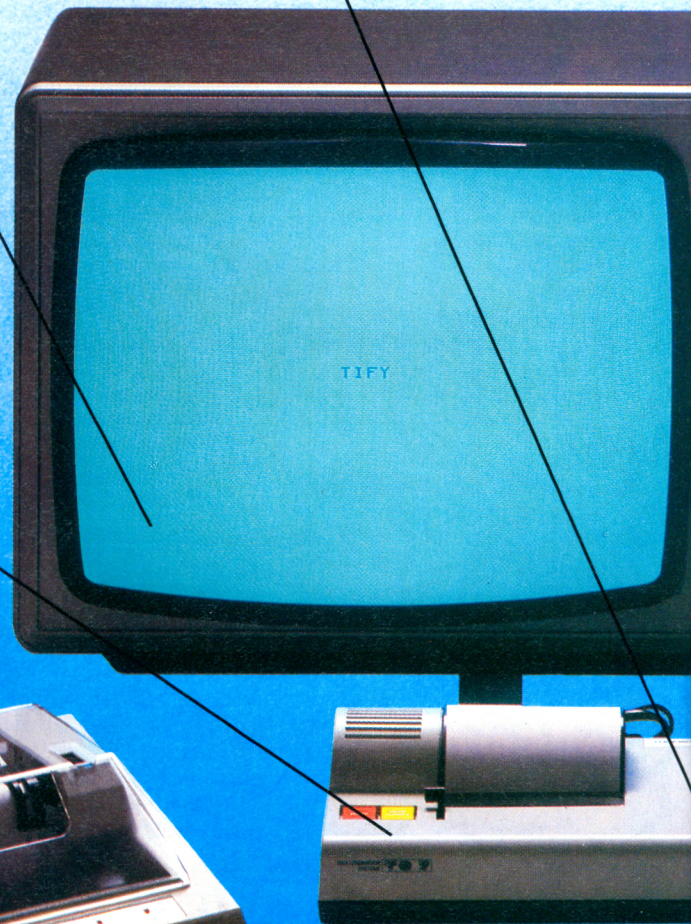
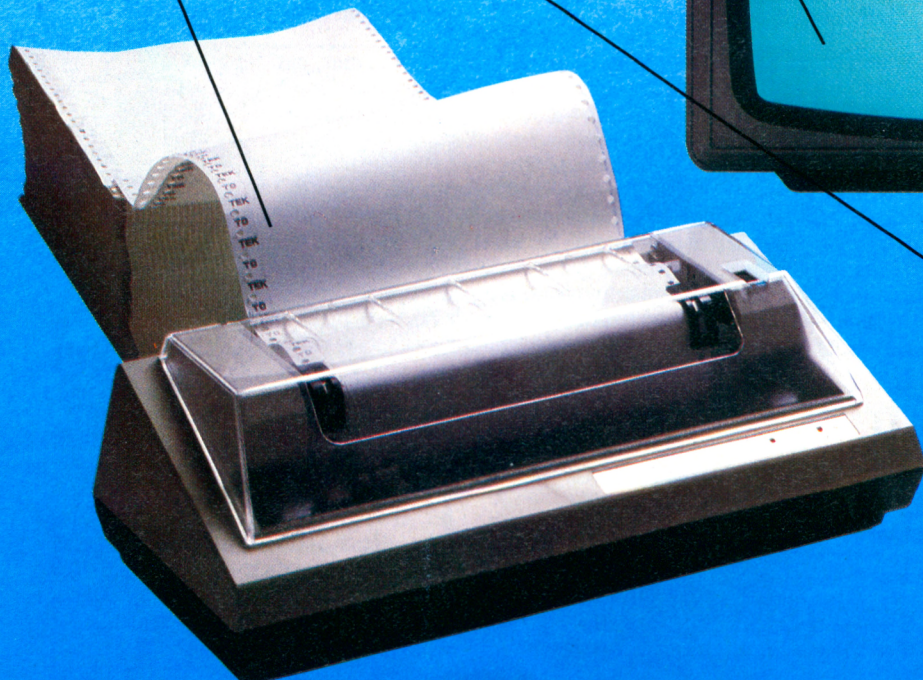
Comparable à celui d'un téléviseur, l'écran de l'ordinateur affiche des images, des mots ou des chiffres. On y voit ce que l'on tape au clavier, les réponses éventuelles de l'ordinateur (résultat d'un calcul demandé, message d'erreur, etc.), ou le listing d'un programme. C'est lui qui matérialise le plus couramment le dialogue homme/machine. L'écran peut être celui d'un téléviseur de type courant ou un « moniteur » conçu spécialement pour l'ordinateur.

manette de jeu (ou joy-stick)

Une manette de jeu (ou joy-stick) est, le plus souvent, une petite boîte munie de boutons et d'un levier que l'on peut déplacer dans toutes les directions. Le levier sert à déplacer sur l'écran un point lumineux qui, dans le cadre des jeux vidéo par exemple, peut prendre la forme d'un petit bonhomme ou d'un vaisseau spatial. Les boutons servent à déclencher une action, faire lancer un projectile par le vaisseau spatial, par exemple.

imprimantes

Reliée, par une prise, à l'ordinateur qui la commande, l'imprimante sert, comme son nom l'indique, à imprimer sur papier des textes, des données ou la liste des instructions composant un programme.



lecteur de codes barres

Les codes barres, ce sont les traits plus ou moins larges que l'on peut voir sur les étiquettes de certains produits. Le lecteur de codes barres ressemble à un stylo ayant, à son extrémité, un composant qui permet, quand on le déplace sur les barres noires, de traduire en informations – prix du produit, nature, etc. – les codes que ces dernières représentent.

modem

Le modem est un dispositif qui permet de relier entre eux, par l'intermédiaire du réseau téléphonique, plusieurs ordinateurs. Il sert à traduire le langage binaire de l'ordinateur en une fréquence transmissible par câbles téléphoniques et inversement.

clavier

Le clavier est un périphérique intégré, la plupart du temps, à l'ordinateur. Il est le moyen d'entrer, sous forme de mots ou de chiffres, des instructions ou des données. Comparable au clavier d'une machine à écrire, c'est par son intermédiaire que se fait la programmation.

lecteur de cassettes

Le lecteur de cassettes est un périphérique, soit d'entrée, soit de sortie. Il peut transmettre à l'ordinateur un programme enregistré sur une cassette magnétique (comme celle que l'on utilise couramment pour enregistrer de la musique) ou, si on le désire, enregistrer (ou « sauver ») sur la cassette un programme écrit par l'utilisateur au clavier.

crayon optique

Le crayon optique est une sorte de stylo que l'on promène sur l'écran. A son extrémité, un dispositif sensible à la lumière enregistre la position du point sur lequel on l'a posé. Il est, par exemple, utilisé pour désigner un choix dans un catalogue affiché à l'écran ou pour réaliser des dessins.

lecteur de disquettes

Plus perfectionné que le lecteur de cassettes, il a exactement le même rôle. Il est cependant plus rapide.

cartouche

Une cartouche est un périphérique d'entrée dont le programme ne peut être ni listé, ni modifié, ni effacé.



Dialogue avec l'ordinateur

Le clavier

Nous prendrons comme exemple le MO5 de Thomson. Le clavier d'un micro-ordinateur est, en bien des points, comparable à celui d'une machine à écrire. En particulier, les vingt-six lettres y sont disposées selon un ordre différent de l'ordre alphabétique : on parle de clavier AZERTY, à cause de la position des lettres sur la première rangée (AZERTYUIOP). Cette disposition est commune à toutes les machines à écrire utilisées dans les pays francophones. Cependant, de nombreux micro-ordinateurs d'origine étrangère, principalement anglo-saxonne, présentent une disposition différente, appelée clavier QWERTY, correspondant aux positions des touches des machines à écrire des pays anglophones. Une large barre horizontale, appelée **barre d'espacement**, permet, sur le clavier ordinateur comme sur le clavier machine à écrire, de créer un espace vide, ou « blanc », entre deux caractères.

Mais il existe, à côté de cela, de nombreuses différences entre le clavier du MO5 et celui d'une machine à écrire.

Ainsi, la rangée supérieure du MO5 comporte tous les chiffres, y compris le zéro – pour un ordinateur le zéro (noté 0) est un caractère différent de la lettre O.

Il présente également, à la place de la touche majuscule/minuscule des machines à écrire, une large **touche jaune**. Cette touche sert à afficher les symboles écrits en jaune sur les autres touches. Par exemple, frapper simultanément sur cette touche et sur la touche « 3 » provoque l'affichage du symbole « # ». Sur la plupart des touches du MO5, en effet, sont inscrits trois symboles : un blanc qu'on obtient en frappant tout simplement la touche, un jaune dont nous venons de parler, et un noir pour lequel il existe une autre touche à frapper simultanément.

Symétriquement disposée par rapport à cette touche jaune, se trouve, en effet, une **touche noire** marquée **BASIC**. C'est elle qui permet d'obtenir les mots écrits en noir. Par exemple, frapper simultanément sur la touche marquée **BASIC** et sur la touche « 3 » affiche le mot **LINE**. Ces mots sont des instructions en **langage Basic**.

Le **passage majuscule/minuscule**, lui, s'obtient d'une manière un peu plus compliquée que sur les machines à écrire : il faut frapper simultanément la touche jaune et la barre d'espace. A la mise en marche de l'ordinateur, les lettres sont automatiquement positionnées dans le mode majuscule. Mais, après avoir frappé plusieurs fois sur la touche jaune et sur la barre d'espace, il est souvent difficile de savoir en quel mode on se trouve ! Pour le savoir, on ne peut qu'essayer !

Le curseur sur tout l'écran

Le MO5 dispose d'un **éditeur pleine page**.

Cela signifie que le curseur d'écriture peut être positionné n'importe où sur la page afin d'y écrire ou d'y modifier une écriture préalable.

L'instruction **EFF** efface le caractère situé au-dessus du curseur et déplace d'une position vers la gauche toute la fin de la ligne.

L'instruction **INS** déplace d'une position vers la droite toute la fin de la ligne et marque un espace au-dessus du curseur.

Lorsque le curseur est positionné sur le caractère d'une instruction de programme, la frappe des touches **INS / EFF** agit sur la totalité de cette instruction de programme (et donc éventuellement sur plusieurs lignes d'écriture).



A droite du clavier se trouvent quatre touches disposées en croix, portant des flèches orientées dans les quatre directions : ←, →, ↑, ↓. Ces quatre flèches permettent de déplacer sur l'écran le curseur (le petit ← qui s'affiche dès qu'on branche l'ordinateur et se déplace au fur et à mesure qu'on tape — il indique l'endroit où le caractère qu'on va taper s'inscrira).

Notons que, lorsqu'on a inscrit un caractère sur l'écran, il suffit, pour le corriger ou l'effacer, de placer le curseur sur sa position et, soit de taper un autre caractère (qui l'efface et le remplace), soit d'appuyer sur la barre d'espacement (qui le supprime et le remplace par un blanc).

Attention ! Si, quand on appuie sur la touche ENTRÉE, l'ordinateur, au

nous transmet le message suivant : « je n'ai rien compris, votre instruction ne correspond à rien dans le langage qui m'a été enseigné ». On sait alors que, en formulant l'instruction, on a commis, soit une erreur dans les mots (les instructions en Basic), soit une erreur de syntaxe (dans la grammaire du Basic).

Il faut alors relire toute la ligne



Lorsque le curseur dépasse un côté de l'écran, il réapparaît du côté opposé. Deux autres touches permettent également de modifier la position du curseur sur l'écran : la touche → renvoie le curseur en haut et à gauche de l'écran sans effacer ce qui est déjà écrit. La touche RAZ effectue le même mouvement mais en effaçant tout ce qui est inscrit sur l'écran. Le clavier du MO5 comporte aussi des touches qui ne figurent pas sur ceux des machines à écrire traditionnelles : ESC, CNT par exemple. Dans le dialogue avec la machine, ces symboles ont des fonctions multiples que nous verrons plus tard. Mais la touche la plus importante du clavier du MO5 est la touche marquée ENTRÉE. A première vue, on pourrait croire qu'elle permet simplement ce qu'on appelle en dactylographie le « retour chariot », c'est-à-dire le renvoi du curseur au début de la ligne suivante. Mais si, tapé n'importe quand, ce symbole entraîne, outre le retour à la ligne, un message d'erreur, c'est que sa fonction est beaucoup plus complexe. La touche ENTRÉE est une « touche d'exécution » : elle sert à « valider » le texte affiché à l'écran, c'est-à-dire à indiquer à l'ordinateur que la suite de caractères peut désormais être prise en compte. Ceci signifie qu'il est toujours possible de modifier la ligne affichée à l'écran tant que cette touche n'a pas été enfoncée.

lieu d'exécuter l'ordre reçu, affiche « ERROR 2 », cela veut dire qu'il

correspondante pour découvrir l'erreur.

Un langage pour la machine

Le clavier est en fait le seul moyen d'établir un véritable dialogue entre l'homme et la machine. Hélas, en effet, malgré tous les efforts des informaticiens, l'ordinateur ne parle pas encore français, ni même anglais ! Le langage courant est trop complexe et trop imprécis pour être intelligible par une machine, aussi perfectionnée soit-elle. Il suffit de penser aux difficultés que nous rencontrons, du fait de l'importance de son vocabulaire et de sa grammaire, lors de l'apprentissage d'une langue étrangère, pour comprendre qu'il est impossible de faire ingurgiter à une machine toutes les subtilités des langues de Shakespeare ou de Molière.

Pour dialoguer avec la machine on emploie donc un langage simplifié, un **langage informatique**, moins complexe, avec son vocabulaire de quelques centaines de mots tout au plus, que le langage humain. Il est néanmoins suffisant pour donner clairement des ordres à la machine.

Comme chez les humains, il existe, dans le monde des ordinateurs, plusieurs langages (Pascal, Cobol, Fortran, etc.). Pour notre part, nous allons faire nos premiers pas dans deux langages très simples : **Basic** et **Logo**. Le Basic, dont le nom signifie, en anglais, « fondamental », a été principalement conçu pour permettre aux néophytes d'apprendre rapidement un moyen de communiquer avec l'ordinateur. Comme la plupart des langages informatiques, il est d'origine américaine, et utilise donc des mots du langage courant... des pays anglo-saxons ! Le Logo a été mis au point au *Massachusetts Institute of Technology*, par Seymour Papert, et a fait l'objet de nombreux développements en France depuis une dizaine d'années. Il permet de construire facilement des dessins et de manipuler des textes.

Un peu de Basic...

PRINT
un nombre
entier

Grâce à la touche ENTRÉE, il est possible, nous l'avons vu, de donner des ordres à l'ordinateur : si celui-ci reconnaît les mots et la syntaxe de l'instruction comme faisant partie de son vocabulaire et de sa grammaire, il exécute l'ordre immédiatement, sinon il affiche « ERROR 2 ».

L'un des principaux mots du vocabulaire Basic est PRINT. En anglais « print » signifie « imprime, donc affiche ». Cette instruction donne l'ordre à l'ordinateur d'afficher quelque chose à l'écran. Tapons, par exemple :

PRINT 124
puis appuyons sur la touche ENTRÉE.

La machine interprète cet ordre (elle vérifie que le mot PRINT fait bien partie du vocabulaire Basic et recherche à quelles opérations il correspond), puis l'exécute ; on voit alors s'afficher sur l'écran :

```
PRINT 124
124
OK
-
```

La machine a bien exécuté l'ordre (elle a écrit le nombre 124). Puis, à la ligne suivante, elle a affiché « OK » pour nous indiquer que l'ordinateur se tenait de nouveau à notre disposition. Le curseur est positionné sur la ligne vierge suivante.

Tapons maintenant :
PRINT 124,56 (dorénavant nous symboliserons le fait de taper sur la touche ENTRÉE par le signe ↵). La machine, au lieu d'afficher : 124,56 affiche :

```
PRINT 124,56
124 56
OK
-
```

Notons qu'il existe trois façons de donner à l'ordinateur l'ordre « PRINT » :

- taper successivement les touches P, R, I, N, T sur le clavier ;
- appuyer simultanément sur les touches BASIC et « / » (au bas de la touche « / » figure le mot PRINT, en noir) ;
- appuyer simultanément sur la touche jaune et la touche « / » (en haut à droite de la touche « / » figure, en jaune, le symbole ? qui équivaut à PRINT).

PRINT
un
nombre
décimal

C'est que, pour l'ordinateur, les nombres ne comportent pas de virgules. Celles-ci sont remplacées par des points.

Tapons donc maintenant :
PRINT 124.56 ↵
L'ordinateur affiche bien :

```
PRINT 124.56
124.56
OK
-
```

La virgule, elle, avait simplement indiqué à l'ordinateur qu'il devait écrire, séparément, les nombres 124 et 56.

L'instruction PRINT est l'une des plus précieuses du langage Basic. Elle permet, en effet, de transformer notre ordinateur en une calculatrice très puissante. Tapons, par exemple :
PRINT 17 + 45 ↵
L'ordinateur affiche immédiatement le résultat de l'addition demandée, soit :

```
PRINT 17 + 45
62
OK
-
```

L'instruction PRINT permet donc d'obtenir le résultat, non seulement de calculs, même très compliqués, posés en utilisant les quatre opérations élémentaires : addition (+), soustraction (-), multiplication (attention, en informatique, pour éviter la confusion avec la lettre « x », le symbole de la multiplication est (*)), division (/), mais aussi de calculs mettant en jeu des fonctions mathématiques plus complexes (sinus, cosinus, logarithmes, exponentielles, etc.). Ainsi :
PRINT 17 * (45 - 7) / 19 ↵
provoquera, en un temps record, l'affichage du résultat :

```
PRINT 17 *
(45 - 7) / 19
34
OK
-
```

Tapons maintenant :

PRINT TIFY ↵

L'ordinateur va-t-il afficher « TIFY » ? Non, il répond par le message « ERROR 2 ». Il n'a donc rien compris ! C'est que, pour lui, toute chaîne de caractères (et non de chiffres) est une instruction en Basic. Or l'instruction TIFY n'existe pas dans sa mémoire, et pour cause !

Essayons maintenant :

PRINT « TIFY » ↵

La machine a compris, elle affiche :

PRINT
une
chaîne de
caractères

```
PRINT « TIFY »
TIFY
OK
-
```

Pour que l'ordinateur comprenne qu'une suite (une « chaîne ») de caractères ne doit pas être prise comme une instruction mais comme un mot (ou groupe de mots) du langage courant qu'il a à enregistrer, il faut donc l'écrire **entre guillemets**. Les guillemets permettent d'indiquer à l'ordinateur tout message à reproduire, à condition que sa longueur n'excède pas 255 caractères.

Des instructions de toutes les couleurs !

COLOR

Il est des instructions qui permettent d'obtenir des résultats assez amusants ! Ainsi, l'instruction **COLOR** permet de jouer avec les couleurs de l'écran !
Tapons par exemple :
COLOR 1 ↵
Le MO5 se met à écrire en rouge !

```
COLOR 1
OK
-
```

COLOR 2 ↵
Il écrit en vert !
COLOR 13 ↵
Il écrit en violet !

Le MO5 possède 16 couleurs, numérotées de 0 (noir) à 15 (orange), et l'instruction **COLOR X** transforme la couleur d'écriture (à l'origine la couleur 4, bleu) en la couleur X. Tout ce qui sera écrit après cette instruction le sera dans la couleur X.

Mieux encore ! Tapons :
SCREEN 1 ↵
Tout ce qui est écrit à l'écran devient rouge, même ce qui avait été affiché avant que l'ordre ne soit donné !
On peut aussi jouer sur la couleur du fond :
COLOR 1,8 ↵

```
COLOR 1,8
OK
-
```

SCREEN

Le MO5 écrit en rouge sur une bande grise ! En fait, le fond général de l'écran reste le même, mais chaque caractère s'inscrit sur un fond (un petit carré) gris. Par contre, si on tape :
SCREEN 1,8 ↵

```
SCREEN 1,8
OK
-
```

c'est le fond tout entier qui change de couleur en même temps que les caractères. Bien entendu, si la couleur du fond et celle du texte sont les mêmes (par exemple, avec l'instruction **COLOR 1,1**), on ne voit rien : l'ordinateur écrit en rouge sur fond rouge !

L'instruction **SCREEN** peut encore admettre un troisième paramètre, correspondant à la couleur du cadre, autour de l'écran. Par exemple :
SCREEN 1,8,0 ↵

```
SCREEN 1,8,0
OK
-
```

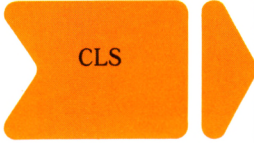
affiche des caractères rouges sur fond gris dans un cadre noir ! Ces instructions peuvent être utilisées en même temps que l'instruction **PRINT**. Il suffit pour cela d'indiquer à l'ordinateur, à l'aide du symbole **;**, que l'on désire exécuter deux ordres successivement. Par exemple :
SCREEN 0,12,15 : PRINT « TIFY » ↵

```
SCREEN 0,12,15
: PRINT « TIFY »
TIFY
OK
-
```

affiche le mot **TIFY** (attention ! Ne pas oublier les guillemets) en noir sur fond bleu pâle, avec pourtour orange.

A chaque fois qu'on inscrit une nouvelle instruction sur l'écran, ce qui était affiché au préalable le reste, la ligne s'écrit simplement à la suite des autres. Quand l'écran est plein, le texte remonte au fur et à mesure que l'on rajoute des lignes.

Le clavier du MO5 dispose d'une touche **RAZ**, qui provoque l'effacement de tout l'écran et le retour du curseur en haut à gauche. On peut aussi effacer l'écran en tapant les trois lettres **CLS** (Clear Screen en abrégé, c'est-à-dire



« nettoyer l'écran »). Cette instruction Basic peut bien sûr, elle aussi, être combinée à d'autres, grâce au symbole : . Ainsi, l'ordre composé :

CLS : PRINT « TIFY » ↵



efface tout ce qui était déjà à l'écran, et affiche le seul mot TIFY en haut à gauche.

Lorsqu'on écrit directement sur l'écran du MO5, il est possible de placer le curseur où l'on veut (et, donc, d'écrire où l'on veut) en utilisant les touches flèches. Mais lorsqu'on demande à l'ordinateur d'afficher un message à l'aide de l'instruction PRINT, il l'écrit toujours au début de la ligne suivante. Il existe cependant une instruction du langage Basic qui permet de positionner le texte écrit par l'ordinateur à l'endroit désiré : c'est l'instruction **LOCATE**.

On sait que l'écran du MO5 permet d'afficher, au maximum, 25 lignes (numérotées de 0 à 24) de 40 caractères (numérotés de 0 à 39) chacune. L'instruction LOCATE permet d'indiquer, par ses coordonnées, l'endroit de l'écran où l'on veut positionner le curseur, donc l'endroit où s'affichera la première lettre du message. Cette instruction, toujours utilisée avec PRINT, permet d'afficher un message n'importe où sur l'écran. Par exemple, l'ordre composé :

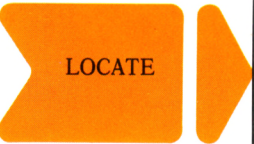
CLS : LOCATE 17,10 : PRINT « TIFY » ↵

réalise successivement les trois opérations suivantes : effacement de l'écran, positionnement du curseur au 17^e caractère de la 10^e ligne (à peu près au centre de l'écran), affichage du mot TIFY. L'emploi des deux-points permet de réaliser des opérations relativement complexes. Ainsi, la séquence d'instructions

CLS : SCREEN 0,3,2 : LOCATE 17,12 : PRINT « TIFY » ↵



affiche-t-elle, en partant du 17^e caractère de la 12^e ligne, le mot TIFY en noir sur un écran jaune entièrement nettoyé, avec un pourtour vert. Cependant, cette méthode est limitée. Le MO5, en effet, ne tolère que des lignes



d'instructions comportant, au maximum, 255 caractères. Il est donc impossible de réaliser, par la méthode que nous venons de décrire, la moindre opération nécessitant un nombre un peu important d'instructions.

De plus, comme les ordres sont exécutés dès qu'on appuie sur la touche ENTRÉE, si on veut réaliser deux fois de suite la même opération, il faut réécrire la séquence d'instructions deux fois de suite, ou alors, grâce aux touches flèches, repositionner le curseur sur la ligne concernée et enfoncer la touche ENTRÉE (ceci est rendu possible grâce à l'éditeur pleine page).

Pour que l'ordinateur puisse être utilisé au mieux de ses possibilités, il faudrait, en fait, on le voit bien, qu'un certain nombre d'impératifs soient réalisés :

- possibilité d'écrire à la suite les unes des autres un grand nombre d'instructions ;
- possibilité d'obtenir une exécution différée des ordres donnés à la machine ;
- possibilité de mémoriser les séquences d'instructions pour qu'elles puissent être réutilisées plusieurs fois.

Il faudrait donc que le langage utilisé pour communiquer avec lui possède les propriétés fondamentales correspondantes.

Le langage Basic atteint ce triple objectif en proposant la solution suivante : la **numérotation des lignes d'instructions** !

Ainsi, si on tape la ligne :

1 CLS ↵

on ne provoque pas l'effacement immédiat de l'écran. Le simple fait de numéroté la ligne, c'est-à-dire de commencer par écrire, au début de la ligne, un nombre entier (sans oublier de taper ENTRÉE à la fin), permet de différer l'exécution de l'instruction. On entre les instructions suivantes en les numérotant, elles aussi, ligne par ligne :

2 SCREEN 1,7,11 ↵

3 LOCATE 10,10 ↵

4 PRINT « TIFY » ↵

Si on se borne à les entrer ainsi, aucune de ces instructions n'est exécutée. L'ordinateur les garde en réserve. Pour qu'il effectue successivement toutes les opérations dans l'ordre de leur numérotation, il suffit de lui en donner l'ordre grâce à l'instruction Basic **RUN**, qui signifie « courir », mais correspond aussi au « partez » qui donne le départ d'une course. Si on tape simultanément BASIC et RUN, puis ENTRÉE, la séquence d'instructions s'effectue ainsi : l'ordinateur affiche, à partir du

Des couleurs et des accents

10^e caractère de la 10^e ligne, le mot TIFY, en rouge sur un écran blanc, avec un cadre jaune pâle. Tant que l'ordinateur n'est pas éteint, cette suite d'instructions reste disponible dans la mémoire de l'ordinateur et on peut obtenir le résultat à tout moment en ordonnant RUN puis ENTRÉE. La suite des lignes d'instructions numérotées est ce que les informaticiens appellent un **programme**.

Les instructions sont effectuées par l'ordinateur selon l'ordre de numérotation, indépendamment de l'ordre dans lequel elles ont été tapées. Traditionnellement, les informaticiens numérotent les lignes de 10 en 10 (10, 20, 30...) afin de pouvoir insérer des lignes supplémentaires entre les lignes déjà écrites. Ainsi, si on a le programme :

```
10 CLS ✓  
20 LOCATE 10,10 ✓  
30 PRINT « TIFY » ✓
```

on peut insérer, sans problème, entre 10 et 20, l'instruction :

```
11 SCREEN 1,7,11 ✓
```

Alors, programmons maintenant !

Nous connaissons désormais un certain nombre d'instructions, alors pourquoi ne pas essayer de concevoir un programme un peu plus compliqué, par exemple :

```
10 CLS  
20 SCREEN 0,7,1  
30 COLOR 1,11  
40 LOCATE 5,5  
50 PRINT « TIFY »  
60 COLOR 4,12  
70 LOCATE 8,10  
80 PRINT « S'IL TE PLAÎT, »  
90 COLOR 2,10  
100 LOCATE 18,15  
110 PRINT « RACONTE-MOI »  
120 COLOR 5,13  
130 LOCATE 29,20  
140 PRINT « UNE PUCE »
```

A vous d'essayer maintenant !

Toutes les couleurs du MO5 (16 au total) sont repérées par un numéro de code (de 0 à 15).

Ce sont des numéros de code qui sont utilisés dans les instructions Basic COLOR et SCREEN ; en voici la liste :

0 Noir	8 Gris
1 Rouge	9 Vieux rose
2 Vert	10 Vert pâle
3 Jaune	11 Jaune pâle
4 Bleu	12 Bleu pâle
5 Magenta (mauve)	13 Violet pâle
6 Cyan (bleu pur)	14 Bleu très pâle
7 Blanc	15 Orange

Au-dessus des touches 6, 7, 8, 9, 0, est inscrite une série de caractères accentués : é, è, ù, ç, à. Tous ces

L'ordinateur a de la mémoire

Attention ! Un programme inscrit dans la mémoire de l'ordinateur le reste tant qu'on n'éteint pas la machine ou qu'on ne donne pas l'instruction d'effacement — et ceci même si rien n'est affiché sur l'écran. En effet, après avoir écrit le programme, si on appuie sur la touche RAZ, puis sur la touche ENTRÉE, tout ce qui était inscrit sur l'écran s'efface, mais le programme n'en est pas oublié pour autant — il est quelque part dans la mémoire de l'ordinateur. Il suffit pour le faire réapparaître d'appuyer simultanément sur BASIC et sur **LIST**, puis sur ENTRÉE.

Pour cette raison, il faut être très prudent. En effet, si un programme est inscrit en mémoire, on peut très bien le modifier, partiellement (un caractère, une instruction, une ligne, etc.) ou entièrement, sans réafficher les instructions de départ : chaque ligne de l'ancien programme sera automatiquement remplacée par la ligne modifiée — celles auxquelles on n'aura pas touché resteront telles quelles. De même, un nouveau programme peut être réécrit sur le premier, il le remplace automatiquement, mais ce remplacement se fait ligne à ligne, numéro à numéro, si bien que, si l'ancien programme comportait des lignes que ne comporte pas le nouveau, ces

caractères s'affichent en frappant successivement — et non simultanément — la touche marquée **ACC** (pour accent) et l'une des touches chiffrées : 6 pour é, 7 pour è, 8 pour ù, 9 pour ç et 0 pour à. Mais il est aussi possible de frapper un accent ou un tréma séparément afin de les placer sur une voyelle quelconque. Pour cela, il faut d'abord se positionner dans le mode minuscule, puis frapper, d'abord sur la touche accent, puis, simultanément, sur la touche jaune et sur le 0 (pour l'accent grave), le 7 (pour l'accent aigu), le a (pour l'accent circonflexe), le « (pour le tréma). Ces séquences s'effectuent sans que le curseur n'avance, ce qui permet d'afficher l'accent, le tréma ou la cédille directement sur la lettre considérée, et non à côté.

dernières restent en mémoire, d'où une perturbation dont on ne peut venir à bout qu'en réaffichant tout le programme et en le relisant ligne par ligne. Ainsi, si le programme en mémoire était :

```
10 CLS  
11 SCREEN 1,7,11  
20 LOCATE 10,10  
30 PRINT « TIFY »
```

et qu'on retape uniquement :

```
10 SCREEN 2,8,10  
20 LOCATE 20,30  
30 PRINT « TIFY »
```

Au moment où on ordonne RUN à l'ordinateur, le mot TIFY est bien situé en (20, 30), mais les couleurs sont (1, 7, 11) au lieu de (2, 8, 10). Le programme que l'ordinateur a en mémoire est, en effet :

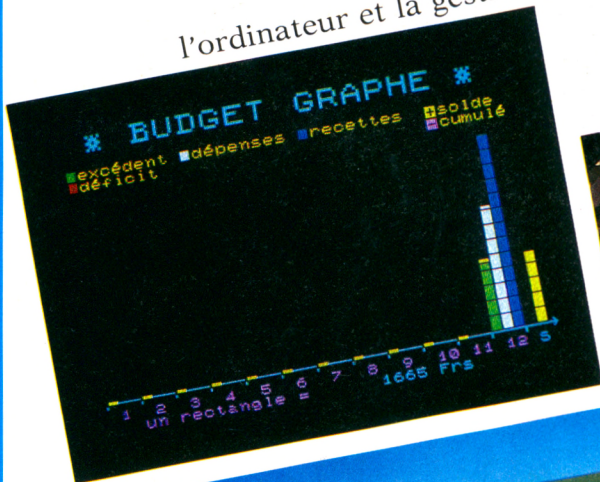
```
10 SCREEN 2,8,10  
11 SCREEN 1,7,11  
20 LOCATE 20,30  
30 PRINT « TIFY »
```

La ligne 11 n'a pas été effacée et, comme elle est exécutée après 10, ce sont les couleurs qu'elle indique qui restent à l'écran.

C'est la raison pour laquelle, si on veut modifier entièrement le programme, il est plus prudent d'effacer préalablement celui qui est déjà dans la mémoire de l'ordinateur. Il suffit alors de taper **NEW** sur le clavier de l'ordinateur, puis d'appuyer sur la touche ENTRÉE.

On peut vérifier que tout est bien effacé en tapant l'instruction Basic LIST. Rien n'apparaît sur l'écran.

l'ordinateur et la gestion



la météo



la simulation de vol



l'enseignement assisté par ordinateur

Ordinateur et gestion

Après le calcul scientifique, et dans l'ordre chronologique, la première utilisation vraiment courante de l'ordinateur fut l'assistance à la gestion. C'est ainsi qu'aujourd'hui des pans entiers de la vie économique — comptabilité, administration des ventes, suivi des stocks, finances, production — sont traités sur ordinateur. Cette informatique des entreprises et des administrations est encore aujourd'hui le domaine fort de l'utilisation des ordinateurs.

Modèles et simulation

Expérimenter sur un problème complexe (pilotage de bateau ou d'avion, contrôle d'un aéroport ou d'une écluse, par exemple) coûte

très cher, demande beaucoup de temps et n'est pas toujours faisable en réel. On construit donc des modèles, traduisant la réalité de la façon la plus fidèle possible, qu'on transfère sur l'ordinateur ; n'importe quel élève-pilote peut, par exemple, s'entraîner à atterrir sans consommer un litre d'essence, sans gêner le trafic normal... et sans risque d'accident.

Météo

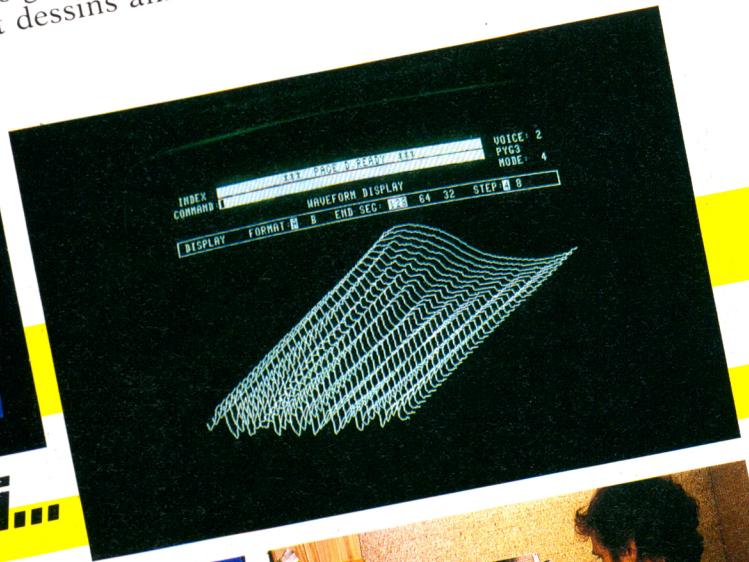
Depuis des temps immémoriaux, l'homme cherche à savoir à l'avance le temps qu'il va faire ; mais, jusqu'à l'arrivée de l'ordinateur, ces prévisions relevaient plus de la magie et du folklore que de la science. Aujourd'hui, par contre, l'énorme quantité de variables relevées par les satellites ou dans les observatoires peut être reliée, analysée, entrée dans des modèles de simulation très complexes ; et des prévisions à peu près exactes peuvent être faites au moins pour des périodes de deux ou trois jours.

La météorologie est l'un des domaines qui requièrent la plus grande puissance de calcul.

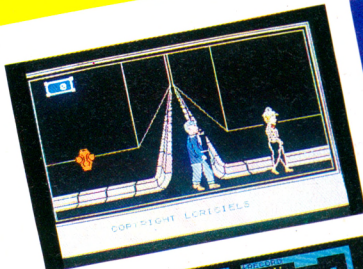
Enseignement assisté par ordinateur

Une machine, contrairement à un professeur, n'hésite pas à répéter inlassablement la même chose, ne fait pas d'ironie, traite tout le monde de la même façon, c'est-à-dire sans émotion ! L'enfant (ou l'adulte) peut donc étudier à son propre rythme et n'importe où. L'Enseignement Assisté par Ordinateur (EAO) pratique aussi bien la méthode « tutorielle » que la méthode des essais et erreurs. Dans la première, l'ordinateur procède en séquence par une série programmée à l'avance d'explications, de questions, d'exercices, d'évaluations de l'élève et de renvois. La seconde s'appuie sur des modèles de simulation.

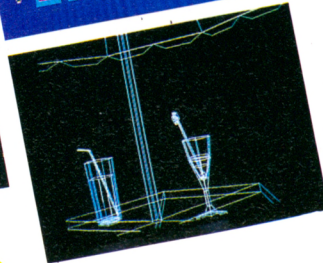
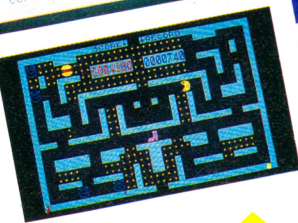
les graphismes
et dessins animés



L'ordinateur, c'est aussi...



les jeux



la musique

Graphismes et dessins animés

Peindre avec un ordinateur... c'est possible. Muni d'un crayon spécial, l'artiste dessine sur une plaque sensible, puis le dessin est enregistré et ses dimensions et couleurs changées au gré du créateur, à l'aide de quelques commandes. Ou bien on introduit dans l'ordinateur des règles de construction de figures et d'harmonies de couleurs et la machine essaie toutes les combinaisons possibles. Reste à décider si le résultat est « beau »... Là, l'ordinateur donne sa langue au chat. Le dessin animé consiste à créer le mouvement grâce à une série d'éléments qui défilent à la vitesse de 24 images/seconde. Les créer « à la main » est très long ; par contre, si on introduit dans l'ordinateur la première et la dernière image d'une séquence, c'est lui qui calculera et dessinera tous les passages intermédiaires ; d'où un énorme gain de temps.

Jeux

Le jeu, les enfants le savent bien, c'est sérieux. La frontière entre jeu électronique et ordinateur n'est pas très nette. On trouve des puces dans les deux. Echecs, tennis de table, alunissage, puzzle, bridge, parcours imaginaires dans des forêts enchantées, mots croisés, pendu, etc., il n'y a que l'embarras du choix et cela ne fait que commencer ! Avec la télématique et les satellites, pourquoi ne pas se choisir un partenaire d'échecs à New York ? A cause du prix à payer, bien sûr !

Pilotage automatique

Depuis les débuts de l'aviation, la réalisation de « pilotes automatiques » a passionné les chercheurs. Deux grandes raisons à cela : la miniaturisation des appareils est limitée par la taille des

hommes, et certains exercices présentent un réel danger pour la vie humaine (vol à très haute altitude ou à très grande vitesse, simulation d'atterrissage en catastrophe, par exemple). Si différents procédés, plus ou moins satisfaisants, telle la télécommande (malgré ses risques de panne ou d'interception), ont déjà été mis au point dans ce domaine, l'ordinateur n'en ouvre pas moins des perspectives tout à fait nouvelles.

Musique et ordinateur

La musique, avec ses règles mathématiques, physiques et acoustiques, ne pouvait pas manquer sa rencontre avec l'ordinateur. Une mélodie une fois composée est soit jouée directement, soit saisie sous forme symbolique, puis synthétisée par la machine dans des rythmes, des tonalités diverses, simulant les



la bureautique



la réservation des places de train



la médecine et la santé

instruments traditionnels. Le plaisir de l'exécution musicale devient accessible immédiatement : plus besoin de le mériter par des années de gammes.

Robotique

Bien différents des « androïdes » chers aux auteurs de science-fiction, les robots industriels sont des machines automatiques conçues pour réaliser une tâche précise, rapidement, sans contrôle important et, généralement, dans des conditions rendant impossible l'intervention humaine.

Bureautique

Dans les pays occidentaux et au Japon, environ la moitié des gens qui travaillent le font dans des bureaux. La bureautique, c'est tous les moyens automatiques, quels qu'ils soient, dont se sert une personne qui travaille dans un

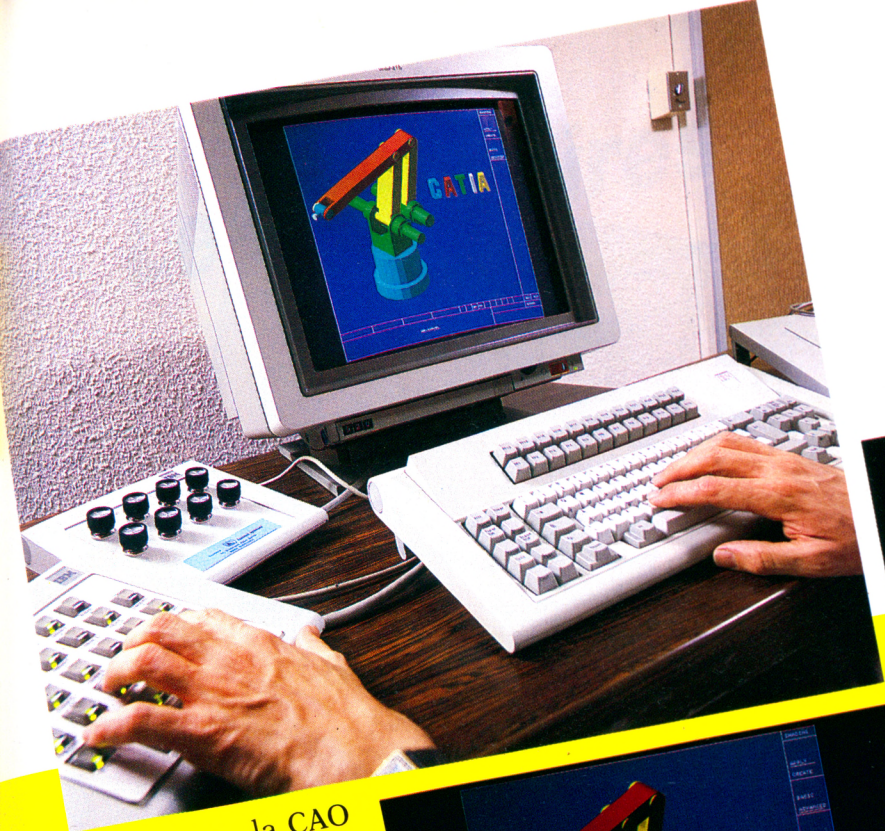
bureau : machines à créer, mémoriser, classer, distribuer, recopier les documents, télécopieurs, répondeurs et boîtes à messages électroniques, etc.

Médecine et santé

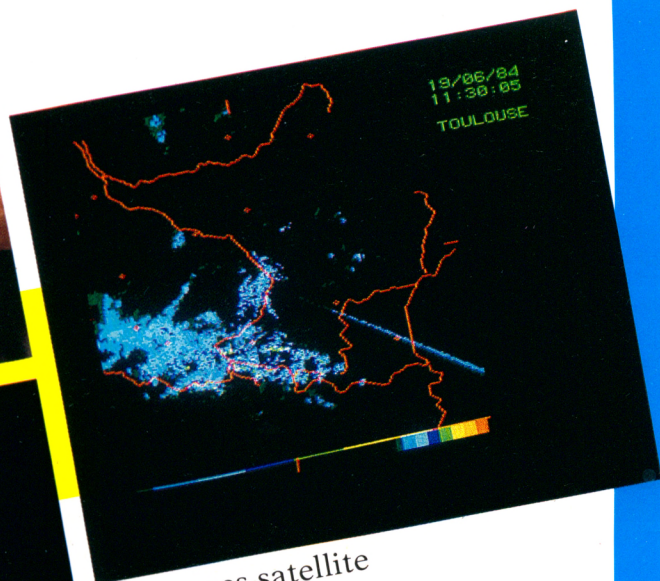
La médecine et la santé sont des domaines dans lesquels l'ordinateur a trouvé de nombreux emplois. Seul, ou associé à d'autres appareils, il calcule les formules sanguines à partir des prélèvements, intervient dans la construction des images d'écographie et de scanner. Il analyse les radiographies et pilote avec précision les appareils de radiothérapie. Des mémoires ont été construites, qui stockent de grandes variétés de symptômes, ce qui permet d'affiner les diagnostics. Parfois, d'ailleurs, c'est la machine elle-même qui établit ce diagnostic. Enfin il intervient comme commande de robots assistant les grands handicapés.

Réservation des places voyageurs à la SNCF

Qui n'a jamais pris, avant l'ordinateur, un train avec de jeunes enfants ne saura jamais ce que pouvaient être les affres des voyages avant la réservation automatique des places. Si on oubliait d'écrire à la gare de départ un mois à l'avance pour réserver places ou couchettes, on ne savait jamais si on allait faire le voyage debout, assis ou couché. Maintenant, tous les trains de grandes lignes qui passent en France sont enregistrés dans un fichier central. Ainsi, tout agent de la SNCF peut réserver une place assise, fumeur ou non-fumeur, une couchette, en haut ou en bas, un wagon-lit, et cela deux mois à l'avance, directement depuis un



la CAO



les images-satellite

terminal situé dans une gare ou certaines agences de voyages, ou par téléphone à un centre de réservation équipé d'un terminal. Ouf ! Finie l'anxiété des départs !

Recherche sismique

Tout le monde sait que les besoins du monde en énergie vont en croissant. Grâce au club de Rome, on sait aussi que les réserves en énergie fossile (charbon, pétrole), bien qu'importantes, sont limitées. Des moyens énormes sont mis en place pour les trouver même dans les endroits les plus inaccessibles : déserts, océans, etc. Et comme il serait quand même un peu cher de creuser des trous au hasard, les géologues ont mis au point une méthode moins aléatoire. Ils font exploser des petites charges à la surface de la terre ou de la mer, des ondes électriques parcourent la croûte terrestre, sont réfractées et réfléchies par les différentes couches géologiques et enregistrées

à la surface par des détecteurs très sophistiqués. Ces séries de données, dites sismiques, car l'explosion est assimilée à un mini tremblement de terre, sont ensuite filtrées et entrées dans des modèles mathématiques très complexes qui éliminent les bruits. Le résultat est une image plus ou moins fidèle du sous-sol, image sur laquelle les géologues peuvent, en se fondant sur l'expérience passée, spéculer sur la possibilité de la présence ou de l'absence de pétrole. Si la configuration est favorable, un puits est creusé.

CAO

CAO : Conception (acte de l'intelligence, de la pensée, s'appliquant à un objet), Assistée (qui se tient auprès de quelqu'un pour le seconder dans sa tâche) par Ordinateur. Un être humain + un processeur + un écran graphique de plus en plus perfectionné + un moyen de communication avec la machine (clavier, souris) + une table traçante = un dessin détaillé

en trois dimensions + la simulation du comportement d'une aile ou d'un moteur d'avion, d'un flacon de parfum, d'un nouvel immeuble dans un quartier animé, d'une platine à laser, d'un pont ou d'une autoroute, d'une structure métallique ou d'un train.

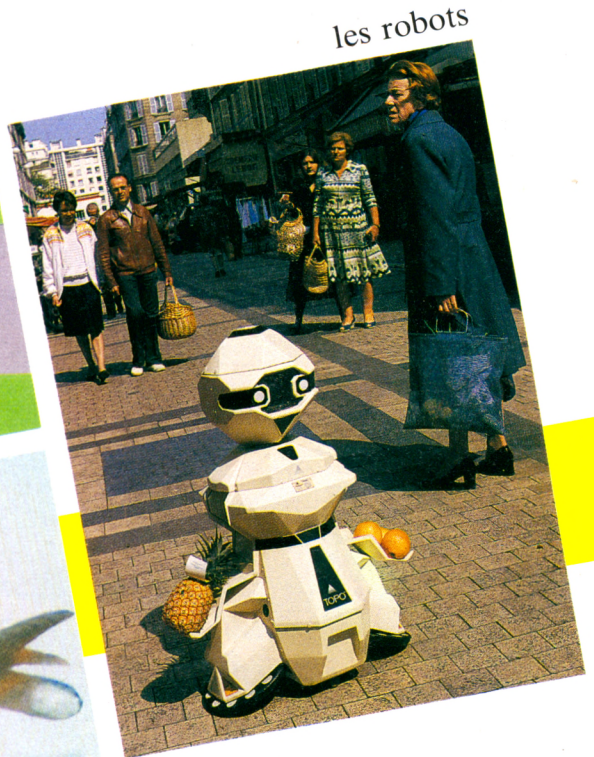
Satellites et ordinateurs

A 900 km au-dessus de la terre, le satellite LANDSAT enregistre des images du terrain. L'image ainsi numérisée est traitée par ordinateur et traduite sous forme de carte thématique sur des écrans graphiques à haute résolution. C'est ainsi qu'on peut identifier sur l'image des taches représentant le blé, l'avoine, la forêt, l'eau, les zones urbaines, etc. En Haute-Volta, par exemple, des cartes au 1/200 000 ont été établies, qui identifient les terres cultivées, les terres incultes, les sols en état de dégradation, les feux de brousse.

le pilotage de la circulation



la recherche iconographique



les robots

L'inventaire de ces états de surface peut aider les agronomes et les chefs d'état à mieux gérer les ressources naturelles de leurs pays.

Pilotage de la circulation

Dans nos villes continuellement au bord de l'étouffement automobile, les feux de circulation étaient traditionnellement réglés une bonne fois pour toutes en fonction des statistiques de circulation. Aujourd'hui, la densité exacte de circulation est mesurée en permanence sur le terrain par des dispositifs automatiques et envoyée en temps réel dans l'ordinateur central. Cette information est confrontée à la vitesse moyenne des véhicules et à la distance entre les feux. De savants calculs sont effectués pour optimiser le flux de circulation. Dans certains cas, des caméras de télévision surveillent les points névralgiques et peuvent même conseiller les automobilistes

sur la vitesse optimum qu'ils doivent maintenir s'ils veulent rouler régulièrement.

Et demain ? Recherche iconographique

Un bon dessin vaut mieux qu'un long discours ! A condition de trouver le bon... justement ! Hélas ! Car si les images, sous toutes leurs formes, existent en grande quantité, les iconothèques, elles, en sont à peu près au point où en étaient les bibliothèques du Moyen Age : un exemplaire connu du seul responsable de l'endroit est introuvable en son absence. Il existe aujourd'hui, grâce au vidéodisque interactif, des prototypes de banques d'images accessibles sur une combinaison de critères. Les images originales ont été transférées sur vidéodisques par des processus photographiques et vidéo, et leur contenu est décrit par

une série de mots clés organisés sous forme de base de données relationnelles. L'ordinateur traite une question formulée dans un langage simple combinant les mots clés, trouve les adresses des images répondant à la question et les affiche sur un ou plusieurs écrans de télévision.

Robot domestique

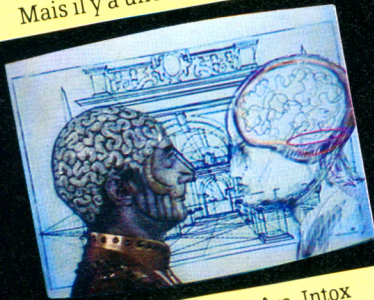
La bonne à tout faire, cuisinière, nounou, femme de chambre, bricoleuse, blanchisseuse, accessible à tous ? Le rêve deviendrait-il réalité grâce au robot domestique ? Pendant que nos gentils esclaves électroniques prépareront le petit déjeuner, feront couler le bain, choisiront nos vêtements après avoir écouté la météo, passeront les commandes d'épicerie, feront venir l'électricien pour réparer une panne (à moins qu'ils ne le fassent eux-mêmes), habilleront les enfants pour l'école, feront la vaisselle, le ménage et le repassage, allumeront le feu et inventeront des cocktails inédits pour le dîner, nous, nous cultiverons notre esprit, nous ferons de la musique, nous philosopherons, nous apprendrons la vie à nos enfants... Ou peut-être mourrons-nous d'ennui !

Les mémoires

De tout temps, l'homme a eu besoin d'aider sa propre mémoire, mais les moyens qu'il a utilisés à cette fin ont varié selon les époques : peintures rupestres, parchemins, livres, enregistrements, etc., et maintenant ordinateur. Le principe a, en fait, toujours été le même : utiliser un support extérieur ; seul ce dernier a varié : pierre, papyrus, papier, bandes magnétiques ou, pour l'ordinateur, composants électroniques.

A notre époque, c'est la « mémoire informatique » qui est partout : dans les hôpitaux, les bureaux, les administrations... et même chez les particuliers (pas un train ou un avion qui ne dépende de l'ordinateur, pas un individu qui ne soit confronté chaque jour – métro, Carte Bleue, etc. – à un ordinateur). Elle conserve, archive et classe des informations de toutes sortes. Mais il faut savoir que l'ordinateur, outil sophistiqué de stockage et de traitement de l'information, a une mémoire qui, bien que rappelant la mémoire humaine, est d'une nature très différente : loin d'appréhender la réalité dans toute sa complexité comme le fait le cerveau humain, l'ordinateur y sélectionne, selon un schéma qui lui a été imposé par son concepteur, un **nombre limité** d'informations qu'il **code** (transcrit en un langage qui lui est propre, le **langage binaire** – basé sur des suites de 0 et de 1) avant de les enregistrer dans ses **mémoires** et de les **traiter** pour fournir des **résultats** à son utilisateur.

Monsieur Plumeau, employé modèle, s'endort régulièrement sur son clavier... Il se transforme alors en un abominable robot, Intox 639, qui décide, avec l'aide de son complice, l'ordinateur numéro 1, de mettre l'univers à ses pieds ! Mais il y a une faille...



« Et maintenant, rêve Intox 639, alias Monsieur Plumeau, grâce à l'ordinateur numéro 1, mon cerveau contient toute la science du monde ! »



Il décide donc que les humains n'ont plus besoin de réfléchir : les réponses à toutes les questions leur seront fournies gratuitement...



Mais, là encore, Zoé fait son impertinente en posant avec insistance une question saugrenue : « Quelle est l'odeur des cerisiers au printemps ? »

L'ordinateur est incapable de lui répondre...

Mémoire humaine et mémoire informatique

L'homme ne peut vivre, réfléchir, produire... sans cette aptitude à fixer, conserver, reconnaître et rappeler des expériences et des « souvenirs » qu'est la mémoire. Cette caractéristique n'est d'ailleurs pas exclusivement humaine – ne parle-t-on pas de « mémoire d'éléphant » – mais est d'une nature très différente chez les animaux, principalement en raison de leur absence de langage.

Le terme de mémoire regroupe, en fait, des processus fort divers. Ainsi, parle-t-on de mémoire à court ou à long terme, de mémoire visuelle, auditive, numérique... et même de « bonne » et de « mauvaise » mémoire !

Le fonctionnement de la mémoire humaine est encore mal connu, bien qu'étudié par de nombreux chercheurs du monde entier. On

pense cependant qu'elle procède, à partir d'impressions reçues, par processus d'abstraction, de généralisation et de modélisation ; c'est-à-dire que, d'une expérience concrète, on ne retient le plus souvent qu'un « schéma directeur ». Ainsi un enfant, après avoir mis la main dans les flammes, se souviendra-t-il seulement, d'une manière très générale, que « le feu brûle ».

Il semblerait aussi que la mémoire humaine « fractionne » ses souvenirs et les « range » dans des compartiments multiples. Parfois ces compartiments sont trop petits, et c'est ainsi que nos souvenirs sont incomplets : on se rappelle souvent l'air d'une mélodie, sans arriver à en retrouver les paroles, ou un événement précis sans pouvoir en préciser la date.

11 = 1011?

Pour faciliter la transmission ou la mémorisation de renseignements, l'homme utilise, depuis toujours, des codes. La représentation du langage parlé, notamment, a donné lieu à l'écriture, donc au codage, de la parole grâce à un certain nombre de « caractères ». Le type de codage utilisé dépend, bien sûr, d'un certain nombre de critères (époque, lieu, but recherché, etc.). Ainsi, une même information peut donner lieu à des représentations multiples. Prenons l'exemple du chiffre « 11 ». En France, au ^{XX}e siècle, on peut l'écrire de différentes façons :

ONZE

écriture littérale française

11

écriture décimale

XI

écriture romaine



écriture d'horloge

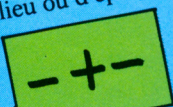


écriture au moyen de dés

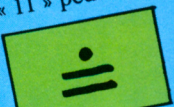
Si l'on change de lieu ou d'époque, « 11 » peut devenir, par exemple :



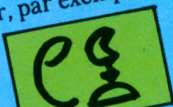
écriture grecque ancienne



écriture japonaise



écriture maya



écriture égyptienne



écriture babylonienne

99

écriture cambodgienne

Et on pourrait prolonger la liste jusqu'à arriver à « l'écriture informatique » qui nous permettrait d'affirmer que :

11

système décimal

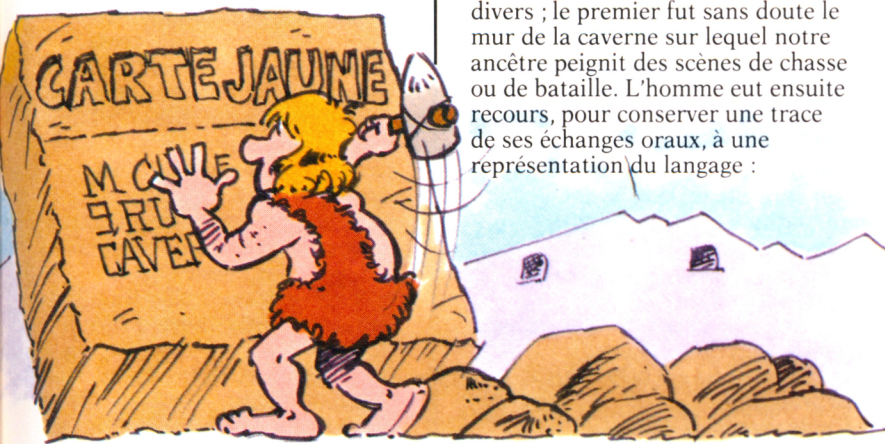
= 1011

système binaire

= A

système hexadécimal

Théoriquement, la mémoire humaine est infinie, et chacun connaît ce qui figure dans sa propre mémoire. Cependant, par crainte de l'oubli – phénomène encore inexpliqué, qui relève peut-être de la psychanalyse – et pour ne pas encombrer sa mémoire, l'homme a depuis longtemps utilisé des mémoires auxiliaires à supports divers ; le premier fut sans doute le mur de la caverne sur lequel notre ancêtre peignit des scènes de chasse ou de bataille. L'homme eut ensuite recours, pour conserver une trace de ses échanges oraux, à une représentation du langage :



utilisons quotidiennement des agendas, carnets d'adresses, fichiers, etc., nous disposons également de supports de mémoire plus sophistiqués : cartes magnétiques (cartes de crédit ou autres), disques et cassettes de musique enregistrée, bandes vidéo, etc. Le support varie, mais la fonction est toujours la même : mémoriser des informations.

La découverte de l'électricité a, quant à elle, donné, dans un premier temps, un nouvel essor aux communications (télégraphe, etc.), mais très rapidement, l'homme – et ce n'est pas étonnant – a cherché à utiliser ses propriétés pour en faire un « support de mémoire ».

L'ordinateur était né.

Un ordinateur c'est, en quelque sorte, une machine qui utilise l'électricité en vue de mémoriser, et aussi de traiter, de l'information ; et le premier problème que sa mise

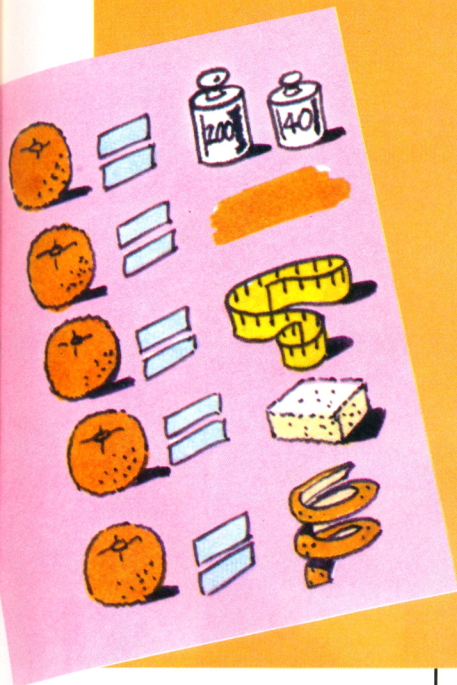
Le goût de l'orange

La mémoire des hommes et ce qu'on désigne par le mot « mémoire » en informatique sont de nature très différente.

En effet, si la mémoire humaine permet d'enregistrer des milliards d'informations et de « souvenirs » concernant une situation, la mémoire de l'ordinateur n'est qu'une série d'armoires de rangement où l'homme peut enregistrer et stocker, pour les restituer ensuite facilement et rapidement, un nombre fini de renseignements, soit quantitatifs, soit informatifs (éléments chiffrés, listes de mots, signes, etc.) ; une infime parcelle de la réalité, donc. De plus, contrairement à l'homme, l'ordinateur ne connaît pas ce qui est dans sa mémoire.

Prenons l'exemple d'un jeune enfant : lorsqu'il voit une orange, fruit qu'il a déjà goûté, il se souvient immédiatement de la sensation de plaisir qu'il a éprouvée en la mangeant.

L'ordinateur, quant à lui, pourra bien sûr écrire dans sa mémoire une foule de renseignements concernant cette orange, mais tous ces renseignements seront élémentaires et chiffrables ou descriptifs – poids : 240 g, couleur : orange, diamètre : 8 cm, contenu : jus sucré + pulpe, etc. –, jamais il ne dira ou ne comprendra la sensation qui a été celle de l'enfant quand le jus lui coulait dans la gorge, ni le plaisir qu'il éprouve à revoir cette belle forme ronde et brillante.



l'écriture, puis l'imprimerie. Enfin, mais seulement depuis un passé extrêmement récent, il a trouvé d'autres procédés de mémorisation rendus possibles par les découvertes technologiques de notre époque. Ainsi, si nous

au point a posé a été de trouver un moyen de représenter l'information (c'est-à-dire les renseignements utilisables par l'ordinateur) en termes de courant électrique. Deux voies étaient possibles, et elles ont été empruntées toutes les deux !

Quand l'information est transformée en courant électrique

Le courant électrique est une grandeur physique continue et mesurable – imaginons, par exemple, une ampoule électrique alimentée par un courant dont l'intensité est contrôlée par un « variateur » (ou rhéostat) ; la luminosité de l'ampoule peut varier, d'une manière continue, entre le noir complet et la luminosité maximale. C'est à cette intensité

lumineuse variable que l'on a, tout d'abord, eu l'idée d'associer des informations considérées comme variant, elles aussi, de façon continue.

Le type de représentation de l'information auquel cela a donné lieu s'appelle représentation **analogique**, et les ordinateurs construits selon ce principe sont appelés, eux aussi, ordinateurs analogiques. Ils résolvent parfaitement toutes les opérations numériques (les premiers calculateurs furent construits selon ce principe).

L'autre méthode consista à ne retenir de notre lampe que les deux positions extrêmes : lampe éteinte – on lui associa, par convention, un 0 – et lampe allumée – on lui associa un 1 (il n'y avait plus alors de rhéostat mais seulement un interrupteur). On parla alors de représentation **binaire** (liée à deux seules possibilités : 0 ou 1), ou **digitale**, du mot anglais *digit* ; les ordinateurs construits sur ce principe s'appellent ordinateurs digitaux.

Depuis une vingtaine d'années, malgré leur adaptation à la résolution des problèmes physiques, les ordinateurs analogiques ont été remplacés progressivement par les ordinateurs digitaux. En fait,

la suprématie de ces derniers s'explique par la simplicité du traitement de l'information auquel ils donnent lieu ; ils ne connaissent que deux états différents : la lampe est éteinte ou allumée, le courant passe ou ne passe pas, il y a un trou dans la carte ou il n'y en a pas... La puissance de l'ordinateur s'est, en effet, tellement accrue ces derniers temps qu'il est désormais préférable de multiplier les calculs très simples plutôt que d'utiliser des ordinateurs analogiques qui, a priori, utilisent moins d'étapes pour la résolution des mêmes problèmes, mais donnent lieu à un traitement plus compliqué. On peut donc dire que dorénavant le binaire règne sans partage : la communication avec l'ordinateur s'opère dans un « langage machine » composé uniquement de 0 et de 1.

Afin que l'homme puisse converser avec la machine, un codage, c'est-à-dire une transformation de toutes les informations à mémoriser et à traiter, en suites de 0 et de 1, est donc, bien sûr, nécessaire ; mais – et c'est important – c'est l'ordinateur lui-même qui fait cette transformation. L'utilisateur, lui, n'emploie, pour converser avec l'ordinateur, que les nombres décimaux et les caractères d'écriture habituels.



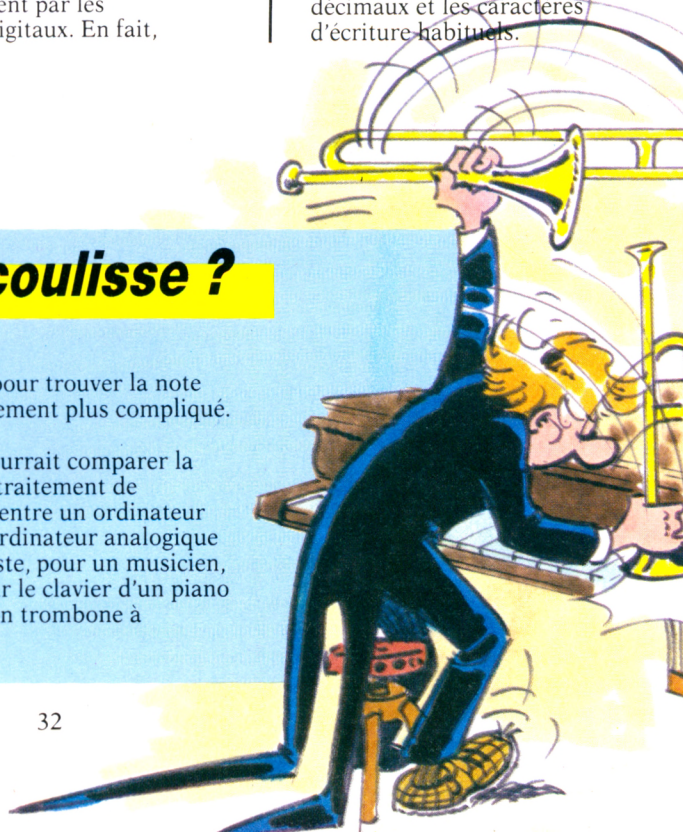
L'ordinateur, qui sait très bien, grâce à son langage binaire, distinguer le zéro (00110000) de la lettre « O » (01001111), a pensé que l'homme, lui, pouvait se tromper : O, sur un écran, est-ce un zéro ou la lettre « O » ? Il (ou plutôt, pour être franc, son concepteur) a donc pensé qu'il serait astucieux de créer, visuellement, pour l'utilisateur, une distinction entre les deux. C'est maintenant chose établie : sur un écran d'ordinateur Ø veut dire zéro, alors que O, c'est une lettre de l'alphabet !

Piano ou trombone à coulisse ?

Le traitement de l'information par un ordinateur digital, c'est un peu comme quand on joue du piano : chaque note n'offre que deux possibilités : on la joue (1) ou on ne la joue pas (0). Avec un trombone à coulisse, le problème est complètement différent ; la note peut être formée, avec plus ou moins de justesse, à différents endroits, et le problème qui se pose

au musicien, pour trouver la note juste, est nettement plus compliqué.

Et bien, on pourrait comparer la différence de traitement de l'information entre un ordinateur digital et un ordinateur analogique à celle qui existe, pour un musicien, entre le jeu sur le clavier d'un piano et celui avec un trombone à coulisse.



Des 0 et des 1 : le règne du binaire

Le système binaire a fait la réputation de l'informatique : un domaine soi-disant réservé aux « forts en thème » et aux matheux... Pourtant, malgré son apparence barbare – des suites de 0 et de 1 – on l'utilise, sans que ça n'étonne personne, depuis fort longtemps. Ainsi les navigateurs de l'Antiquité utilisaient-ils le feu comme phare (la tour d'Alexandrie, l'une des Sept Merveilles du monde, n'avait pas d'autre rôle !) : le feu était, soit allumé (1) et on pouvait entrer dans le port, soit éteint (0) et mieux valait alors passer son chemin. Plus simplement encore : qui d'entre nous n'a jamais fait un nœud à son mouchoir ? Un nœud (1) : il faut se souvenir de quelque chose ; pas de nœud (0) : on peut garder l'esprit libre !... Le seul inconvénient de ce système, c'est que rien n'indique de quoi il faut se souvenir !... Et que le nombre d'informations codées est très faible : 2.



Pourtant, si on utilisait, par exemple, plusieurs mouchoirs, il deviendrait possible de coder un volume plus important d'informations. Avec deux mouchoirs, on pourrait représenter quatre situations différentes : pas de nœud du tout (code 00), un nœud sur le mouchoir de gauche (code 10), ou sur celui de droite (code 01), un nœud sur chacun des deux mouchoirs (code 11). Il serait donc possible de représenter, grâce à cette méthode, toute information « quadriforme » : les quatre premiers chiffres, par exemple, si on décidait que 00=0, 01=1, 10=2 et 11=3, ou les quatre premières lettres de l'alphabet, si on décidait que 00=A, 01=B, 10=C et 11=D. Un « quadruplet », ou groupe de quatre mouchoirs, permettrait, lui, de représenter seize situations différentes. Et, parmi les nombreuses applications possibles, on pourrait notamment coder les dix chiffres (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)

Bit

En langage binaire, l'élément de base (qui peut prendre soit la valeur 0, soit la valeur 1) est appelé BIT, de l'anglais *Binary digIT*, qui veut dire « chiffre binaire ».

plus les six premières lettres de l'alphabet (A, B, C, D, E, F). Il se trouve justement que ce code particulier, appelé **code hexadécimal** (parce qu'il comporte seize symboles différents : 10 chiffres + 6 lettres), est très utilisé par les programmeurs sur ordinateurs.

Mais la combinaison la plus utilisée en informatique est l'**octet**, ou suite de huit chiffres binaires (de huit 0 ou 1).

Un octet peut prendre 256 formes différentes et, donc, représenter l'une parmi 256 informations différentes ; en particulier, l'un quelconque des symboles reconnus par les ordinateurs : les chiffres, les lettres majuscules, les lettres minuscules et tous les « caractères spéciaux » : signes de ponctuation, #, \$, %, etc.

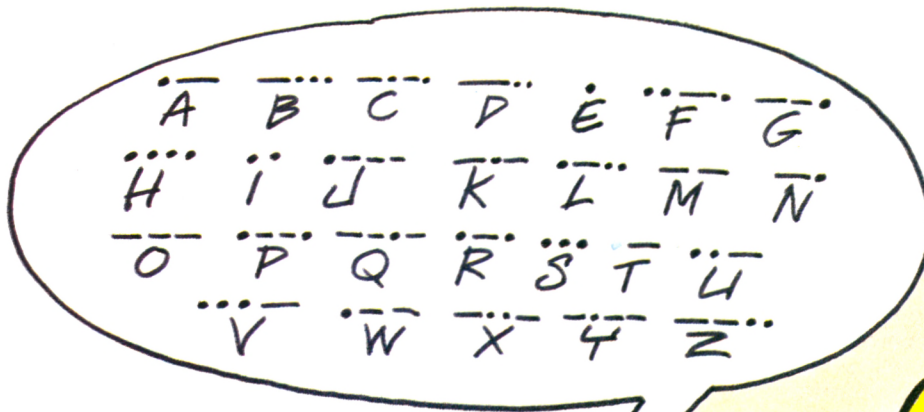
Une suite d'octets peut donc, elle, représenter une suite de caractères : chiffres, lettres, etc.

L'attribution d'un groupe de huit chiffres binaires à un caractère

particulier peut se faire, théoriquement, au gré de chaque fabricant d'ordinateurs. Cependant, un souci de normalisation a conduit à développer un standard international adopté par l'ensemble des constructeurs. Cette norme, d'origine américaine, se nomme le code ASCII (Code Standard Américain pour l'Échange d'Informations).

Le 65^e code, par exemple, y correspond toujours à la lettre A majuscule.

Toutefois, pour laisser une certaine latitude aux constructeurs, le code ASCII n'attribue pas un caractère à chacun des 256 octets ; il ne fixe que les codes des principaux symboles, et laisse un certain nombre de places disponibles (environ une centaine de caractères) pour les besoins spécifiques de chaque constructeur.



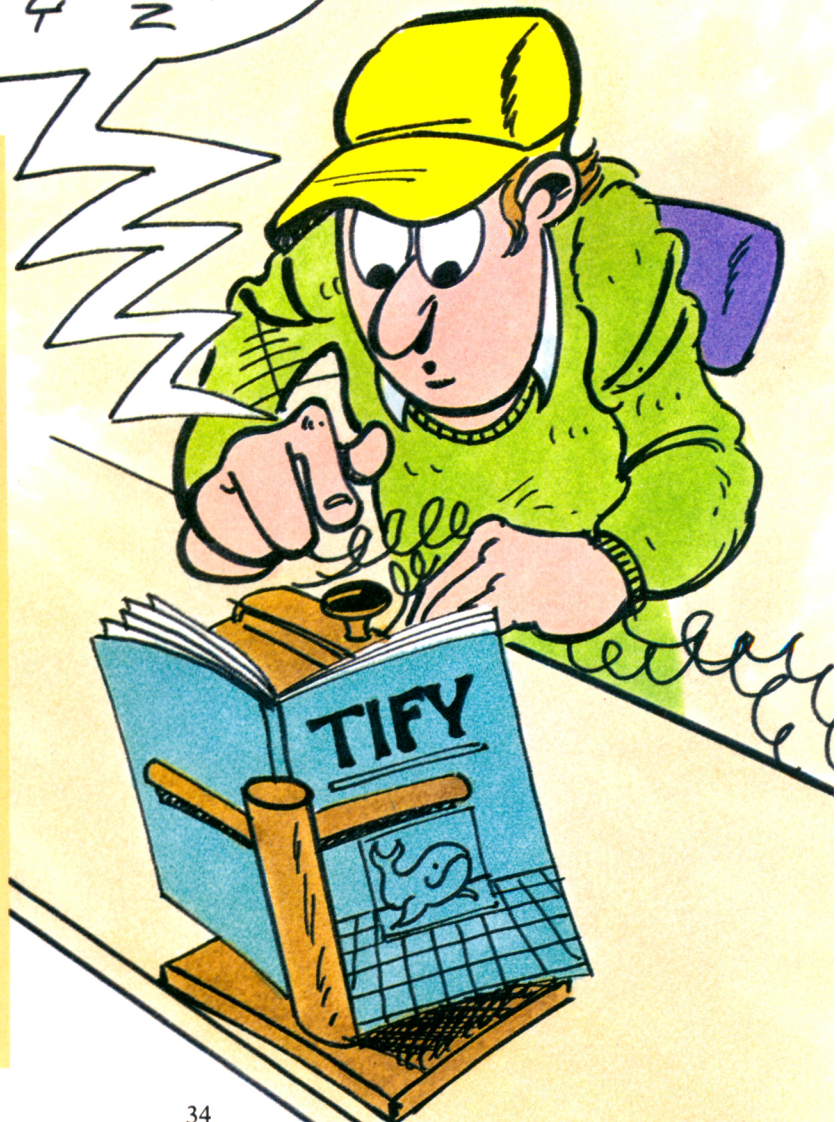
... du binaire sans le savoir !

Le premier code binaire – établissant une correspondance entre les caractères de l'alphabet et des suites de deux symboles – date du siècle dernier...

Il a été mis au point par Samuel Morse et a permis l'existence du télégraphe !

A chaque lettre de l'alphabet correspond, en effet, dans le langage Morse, non pas une suite de 0 et de 1, mais une suite de points et de traits. Petite différence, pourtant, avec le binaire de l'ordinateur qui à chaque caractère fait correspondre un nombre fixe de bits (un octet), le Morse, lui, fait correspondre aux différents caractères un nombre variable de symboles.

Et puis ... le nombre de caractères représentés n'est que de 26, contre 256 pour le langage ordinateur le plus courant !



Ainsi, la plupart des matériels français en utilisent-ils une partie pour coder les caractères accentués, propres à la langue française et ne figurant pas dans l'alphabet anglo-saxon.

Les constructeurs du MO5, quant à eux, ont choisi de n'employer que les 127 premiers codes ASCII, les

valeurs restantes étant laissées libres pour que l'utilisateur puisse y redéfinir ses propres caractères. Le tableau suivant donne les correspondances entre les caractères du clavier, les codes ASCII et les octets du langage binaire (qui seront mémorisés par l'ordinateur).

TABLEAU DES CORRESPONDANCES

(Les 31 premiers codes sont réservés aux fonctions de l'ordinateur telles que la touche « ENTRÉE » du MO5 – code 13).

Caractère	Code ASCII	Octet	Caractère	Code ASCII	Octet
espace	32	00100000	P	80	01010000
'	33	00100001	Q	81	01010001
"	34	00100010	R	82	01010010
#	35	00100011	S	83	01010011
\$	36	00100100	T	84	01010100
%	37	00100101	U	85	01010101
&	38	00100110	V	86	01010110
'	39	00100111	W	87	01010111
(40	00101000	X	88	01011000
)	41	00101001	Y	89	01011001
*	42	00101010	Z	90	01011010
+	43	00101011	[91	01011011
,	44	00101100	\	92	01011100
-	45	00101101]	93	01011101
.	46	00101110	^	94	01011110
/	47	00101111	_	95	01011111
0	48	00110000	-	96	01100000
1	49	00110001	a	97	01100001
2	50	00110010	b	98	01100010
3	51	00110011	c	99	01100011
4	52	00110100	d	100	01100100
5	53	00110101	e	101	01100101
6	54	00110110	f	102	01100110
7	55	00110111	g	103	01100111
8	56	00111000	h	104	01101000
9	57	00111001	i	105	01101001
:	58	00111010	j	106	01101010
;	59	00111011	k	107	01101011
<	60	00111100	l	108	01101100
=	61	00111101	m	109	01101101
>	62	00111110	n	110	01101110
?	63	00111111	o	111	01101111
@	64	01000000	p	112	01100000
A	65	01000001	q	113	01100001
B	66	01000010	r	114	01100010
C	67	01000011	s	115	01100011
D	68	01000100	t	116	01100100
E	69	01000101	u	117	01100101
F	70	01000110	v	118	01100110
G	71	01000111	w	119	01100111
H	72	01001000	x	120	01110000
I	73	01001001	y	121	01110001
J	74	01001010	z	122	01110010
K	75	01001011	{	123	01110011
L	76	01001100		124	01111000
M	77	01001101	}	125	01111001
N	78	01001110	-	126	01111010
O	79	01001111	@	127	01111011

L'ordinateur (machine électrique à mémoriser et à traiter de l'information), mémoire auxiliaire la plus sophistiquée que l'homme ne peut, lui-même, fonctionner sans ses propres « mémoires ».

Tout traitement d'information, même élémentaire, fait en effet appel à des mémoires capables de

conserver et de restituer à volonté, soit une donnée, soit le résultat d'un traitement partiel. Et de même que l'homme, pour aider sa propre mémoire, utilise livres, cassettes ou ordinateur, l'ordinateur utilise, outre sa **mémoire interne**, ou **mémoire centrale**, un certain nombre de mémoires extérieures, dites **périphériques** ou **auxiliaires**.

La mémoire centrale

On regroupe sous le terme de mémoire interne, ou mémoire centrale, tous les dispositifs de stockage d'information internes à l'ordinateur. Ces dispositifs, de nature électronique, font, depuis les débuts de l'informatique, l'objet de nombreuses recherches tendant à améliorer les temps d'accès aux informations, et à obtenir le meilleur rapport capacité/prix. On peut dire que les composants actuellement utilisés dans les micro-ordinateurs représentent le meilleur compromis possible quant au prix, à la rapidité, à la capacité et à la miniaturisation.

Du point de vue de l'utilisateur, on distingue essentiellement deux grandes catégories de mémoires internes : celles dans lesquelles on peut seulement lire l'information, et celles qui autorisent à la fois la lecture et l'écriture.

La mémoire centrale d'un ordinateur peut, en effet, être considérée comme un livre de recettes de cuisine comportant des pages blanches destinées à recevoir les recettes personnelles du lecteur. Les pages contenant les recettes déjà imprimées peuvent être lues, bien sûr, mais pas modifiées ; c'est ce qui se passe dans la partie de la mémoire centrale appelée **mémoire morte** ou **ROM** (de l'anglais *Read Only Memory*, mémoire à lecture seulement). Par contre, les pages blanches peuvent être remplies par les notes personnelles de l'utilisateur, écrites au crayon pour qu'il soit possible de les effacer ou de les modifier ; dans l'ordinateur, cette fonction « pages blanches » est remplie par une partie de la mémoire interne appelée **mémoire vive** ou **RAM** (de l'anglais *Random Access Memory*, mémoire à accès aléatoire).

Pourquoi ces deux types de mémoires ? Elles ont chacune un rôle spécifique. Les mémoires mortes, ou ROM, contiennent tout ce qui permet à l'ordinateur de « marcher ». Bien entendu, ces mémoires sont **permanentes**, c'est-à-dire qu'elles conservent les informations même lorsque le courant est coupé. Lorsqu'on met l'ordinateur sous tension, un message apparaît normalement sur l'écran. C'est un exemple de ce qui est contenu dans les mémoires mortes, en plus des différents programmes gérant les organes de communication (clavier, écran, imprimantes, stylo optique, lecteur-enregistreur de cassettes,

cartouche de jeux, lecteur de disquettes, etc.) et de toutes les fonctions qui permettent d'écrire d'autres programmes. Dans certains micro-ordinateurs, un langage de programmation (généralement le Basic, comme c'est le cas dans le M05) ou des logiciels d'application (programmes conçus dans un but spécifique) sont aussi contenus dans la mémoire morte. On parle alors de **langage** ou de **programmes résidents**.

Le contenu de la mémoire morte, qui permet le fonctionnement de l'ordinateur est souvent appelé le **logiciel de base**. Il est généralement « programmé » par le fabricant.

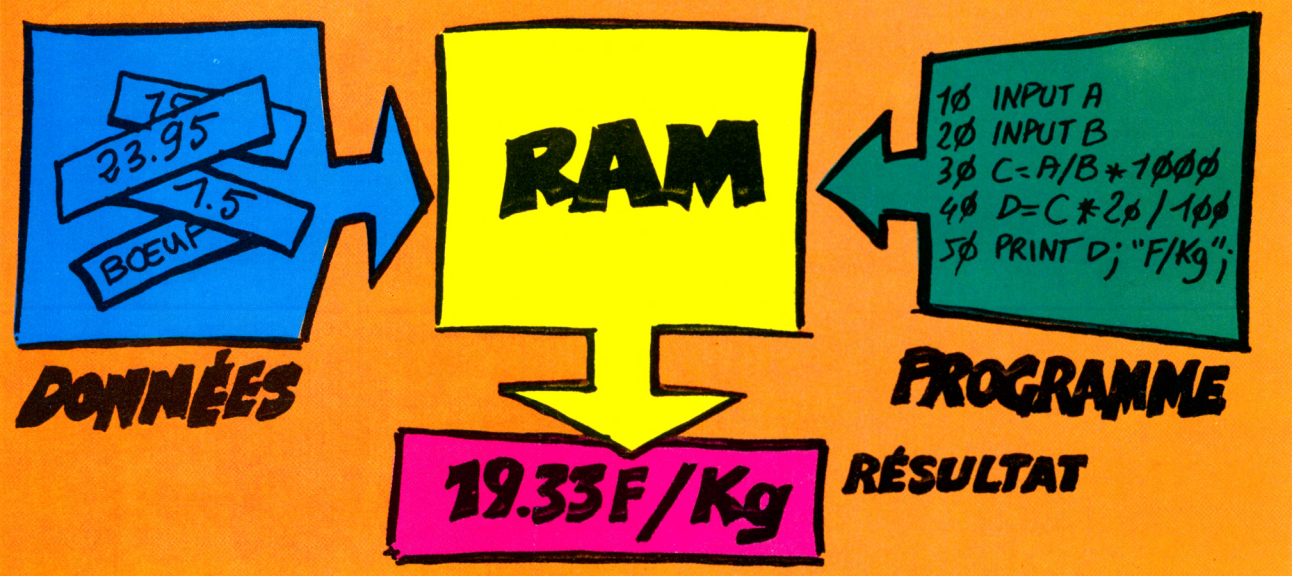
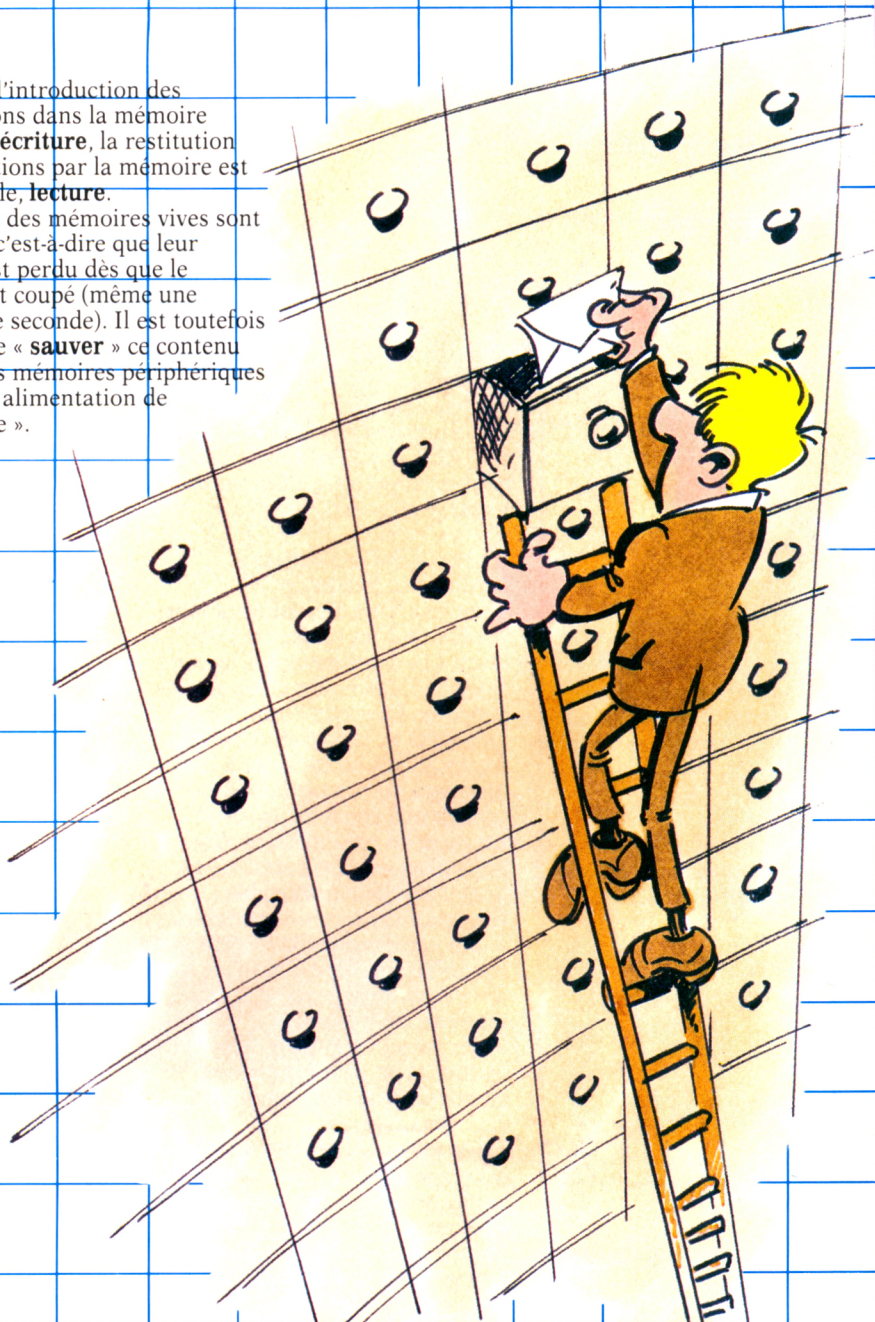
Les mémoires vives, ou RAM, qu'on peut comparer (tout comme les ROM) à une série de boîtes, les **cases-mémoire**, où seront rangées séparément les informations, sont, quant à elles, le lieu où se déroule la majeure partie du travail. Les mémoires vives ont en effet trois fonctions principales :

- enregistrer les programmes destinés à être exécutés par l'ordinateur ;
- mémoriser un certain nombre de données qui seront traitées conformément aux instructions des programmes ;
- emmagasiner temporairement les résultats du traitement avant leur transmission aux périphériques de sortie - écran ou imprimante par exemple.

La mémoire vive doit donc être conçue de telle sorte qu'elle puisse enregistrer rapidement toutes ces informations, les conserver temporairement, les restituer sans délai, et, au besoin, les effacer lorsqu'elles ne sont plus nécessaires.

La phase d'introduction des informations dans la mémoire s'appelle l'**écriture**, la restitution d'informations par la mémoire est appelée, elle, **lecture**.

La plupart des mémoires vives sont **volatiles**, c'est-à-dire que leur contenu est perdu dès que le courant est coupé (même une fraction de seconde). Il est toutefois possible de « **sauver** » ce contenu grâce à des mémoires périphériques ou à une « alimentation de sauvegarde ».



Un peu de Basic...

Le clavier d'un ordinateur n'est pas comme celui d'une machine à écrire. Lorsqu'on y tape quelque chose, l'ordinateur s'attend à ce que ce soit, soit une instruction, soit une information, formulées dans un langage qu'il comprend. C'est pourquoi, si l'on tape au hasard, et qu'on appuie ensuite sur la touche ENTRÉE, il y a de fortes chances pour qu'apparaisse à l'écran un message d'erreur.

On peut faire mémoriser à l'ordinateur beaucoup d'informations. Ces informations, l'ordinateur les range dans des cases. Ces cases-mémoire, ce sont, au départ, des boîtes vides et sans étiquette. Pour que l'ordinateur range une information dans l'une d'elles, il faut lui indiquer à la fois un nom à mettre sur la boîte, comme une étiquette, et un contenu. En informatique, une boîte à laquelle on a attribué un nom s'appelle une **variable**.

Il existe trois sortes de variables :

- des variables **entières** dans lesquelles on ne peut ranger que des nombres entiers (12 par exemple, mais pas 12,5) ;
- des variables **réelles** dans lesquelles on peut ranger des nombres aussi bien entiers que décimaux ;
- des variables **alphanumériques** dans lesquelles on peut ranger des lettres, et par conséquent des mots, mais aussi des chiffres et des signes comme +, =, etc.

Une variable alphanumérique peut contenir au maximum 255 caractères, y compris les blancs qui séparent les mots.

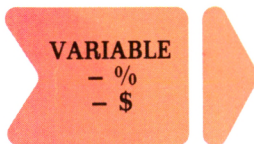
Pour attribuer une **valeur** à une variable, c'est-à-dire pour mettre un contenu dans une boîte à laquelle on a donné un nom, on tape : le nom que l'on veut donner à la variable, le signe = (égal) et le contenu de la variable.

S'il s'agit d'une variable alphanumérique, il faut mettre le contenu entre guillemets.

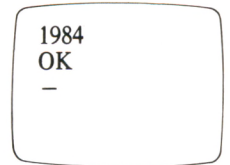
Une variable entière s'écrit avec son nom suivi de %, une variable réelle avec son nom seulement et une variable alphanumérique avec son nom suivi de \$.

Si on veut, par exemple, mettre le groupe de chiffres 1984 dans une variable appelée ANNEE, on écrira : ANNEE = 1984

Pour donner ensuite à l'ordinateur l'ordre d'écrire à l'écran le contenu de la variable ANNEE, on écrira :



PRINT ANNEE
et la réponse sera :



Une variable peut, comme son nom l'indique, varier. Elle peut le faire de différentes façons.

On peut en changer radicalement le contenu. Par exemple, si vous avez écrit : ANNEE = 1984

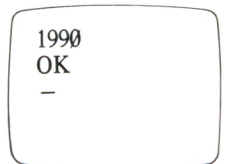
et qu'ensuite, vous écriviez :

ANNEE = 1990,

et que vous tapiez :

PRINT ANNEE

l'ordinateur répondra :



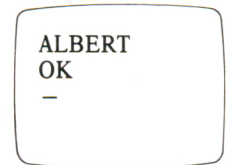
L'ancienne valeur aura été effacée et remplacée par celle que vous aurez tapée en dernier. (Si vous changez 4 ou 5 fois le contenu de ANNEE, ce sera toujours la dernière valeur que vous lui aurez attribuée qui sera retenue par l'ordinateur).

Il en est de même avec les variables alphanumériques.

Si vous avez écrit :

A\$ = "ALBERT"

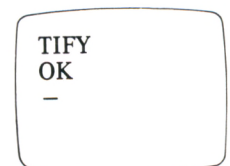
PRINT A\$ vous donnera :



Mais si vous écrivez ensuite :

A\$ = "TIFY"

PRINT A\$ vous donnera :



L'ancien contenu (ALBERT) aura été remplacé par le nouveau (TIFY).

Vérifions avec ce petit programme :

10 ANNEE = 1984

20 ANNEE = 1990

30 A\$ = "ALBERT"

40 A\$ = "TIFY"

50 PRINT ANNEE, A\$

RUN

Le résultat sera :

```
1990  TIFY
OK
-
```

On peut également faire varier le contenu d'une variable en l'augmentant ou en le diminuant ; ou plutôt, pour employer les termes exacts, en **incrémentant** la variable ou en la **décrémentant**.

● INCRÉMENTATION

Prenons une variable dont le contenu est un nombre : par exemple ANNEE ; et écrivons le petit programme suivant :

```
10 ANNEE = 1984
20 ANNEE = ANNEE + 10
30 PRINT ANNEE
RUN
```

Le résultat sera :

```
1994
OK
-
```

En effet, regardons la ligne 20 : ANNEE = ANNEE + 10. Quand on écrit cela, l'ordinateur comprend que la nouvelle valeur que l'on veut attribuer à ANNEE, c'est l'ancien contenu de ANNEE, c'est-à-dire 1984 augmenté de 10. Il effectue alors immédiatement l'opération et range le résultat dans la variable ANNEE. C'est pourquoi, à l'ordre PRINT ANNEE, il répond : 1994

● DÉCRÉMENTATION

Le principe est le même, mais au lieu d'additionner, on soustrait. Continuons notre petit programme :

```
40 ANNEE = ANNEE - 94
50 PRINT ANNEE
RUN
```

Le résultat sera :

```
1900
OK
-
```

L'ordinateur a très bien compris ce que l'on attendait de lui.

On a donc vu que le contenu d'une variable portant un nom pouvait très bien changer sans que le nom de la variable change.

Il est également possible d'attribuer un nom à une variable dont le contenu n'est pas encore déterminé. L'instruction **INPUT** permet ensuite à l'utilisateur d'attribuer la

valeur de son choix à la variable qui attend son contenu : la boîte est prête, elle a une étiquette, il ne reste plus qu'à lui donner un contenu.

Bien sûr, même si son contenu n'est pas encore déterminé, la nature de ce dernier l'est, elle. On sait si ce doit être un nombre entier, un nombre décimal ou une chaîne de caractères.

Prenons un exemple :

```
10 INPUT ANNEE
20 PRINT ANNEE
RUN
```

Le résultat sera :

```
? -
```

ce qui signifie que l'ordinateur attend que vous écriviez la valeur que vous voulez mettre dans ANNEE.

Essayons :

? 1984

Après le point d'interrogation, vous avez écrit 1984, puis vous appuyez sur la touche ENTREE. Le résultat sera :

```
? 1984
1984
OK
-
```

Le premier 1984, c'est celui que vous avez écrit. Le second c'est celui qu'on lui a ordonné d'afficher à la ligne 20.

Le contenu à affecter à la variable ANNEE devait forcément être entier, puisque le nom de la variable n'était accompagné ni du signe % ni du signe \$.

Vérifions-le en faisant à nouveau tourner ce programme grâce à un RUN bien senti puis en tapant successivement :

? TIFY

puis ENTREE

Le résultat sera :

```
Redo
?
```

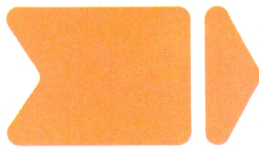
Par l'affichage **Redo**, l'ordinateur vous indique que vous vous êtes trompé, car la variable dans laquelle il doit ranger ce que vous écrivez au clavier ne peut contenir que des nombres. Il vous demande

incrément-
ation

décrément-
ation

INPUT

Redo

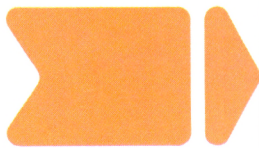


donc de taper une nouvelle information qui soit un nombre. Pour pouvoir écrire un mot après le ?, il eut fallu mettre après le nom de la variable, le signe \$.

```
Par exemple :
10 PRINT "QUEL TEMPS FAIT-IL
AUJOURD'HUI ?"
20 INPUT TEMPS$
30 PRINT TEMPS$
```

Le résultat, après un RUN, sera :

```
QUEL TEMPS FAIT-IL
AUJOURD'HUI ?
? -
```

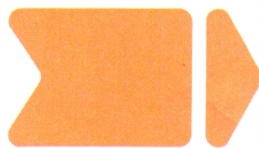


Comme TEMPS\$ est une variable alphanumérique, elle peut contenir n'importe quels caractères.

Si vous écrivez :
? BEAU

le résultat, après avoir appuyé sur la touche ENTREE, sera :

```
QUEL TEMPS FAIT-IL
AUJOURD'HUI ?
? BEAU
BEAU
OK
-
```



Le premier BEAU est celui que vous avez écrit et le second celui qu'on a ordonné d'afficher à la ligne 30.

Si vous aviez écrit :
? 3E3E3E

Le résultat aurait été :

```
QUEL TEMPS FAIT-IL
AUJOURD'HUI ?
? 3E3E3E
3E3E3E
OK
-
```

Une lettre ne peut pas être contenue dans une variable entière, mais un chiffre peut l'être dans une variable alphanumérique.

Un peu plus poussé : vous avez pu, grâce à l'instruction INPUT, entrer un mot, un chiffre ou une réponse à une question posée par l'ordinateur. L'ordinateur va maintenant regarder la réponse que vous lui avez faite et en tenir



IF... THEN...
ELSE

compte pour choisir, entre plusieurs options, celle qu'il va exécuter.

Ecrivez, par exemple, ce petit programme et faites-le tourner :

```
10 PRINT "QUEL TEMPS
FAIT-IL ?"
20 INPUT TEMPS$
30 IF TEMPS$ = "BEAU" THEN
GOTO 50 ELSE GOTO 60
50 PRINT "C'EST MAGNIFIQUE !"
55 END
60 PRINT "DOMMAGE."
RUN
```

Le résultat sera :
QUEL TEMPS FAIT-IL ?
?

1) Si vous répondez :
? BEAU

Vous lirez :

```
QUEL TEMPS FAIT-IL
AUJOURD'HUI ?
? BEAU
BEAU
C'EST MAGNIFIQUE !
OK
-
```

2) Si vous répondez MAUVAIS ou autre chose, le résultat sera :

```
QUEL TEMPS FAIT-IL
AUJOURD'HUI ?
? MAUVAIS
MAUVAIS
DOMMAGE.
OK
-
```

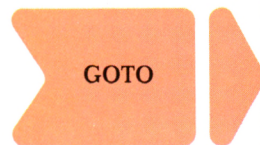
Que s'est-il passé ?

C'est très simple, vous avez fait faire un **test** à votre machine. La ligne 30 peut se traduire ainsi : si le contenu de la variable TEMPS\$ (c'est vous qui avez inscrit ce contenu en répondant) est la chaîne de caractères « BEAU », alors il faut aller directement exécuter les instructions écrites à la ligne 50, sinon il faut aller exécuter celles qui sont inscrites à la ligne 60.

En effet, en anglais, **IF** veut dire « si », **THEN** veut dire « alors » (pour montrer que ce qui va suivre est une conséquence),

GOTO veut dire « aller vers », « se brancher à », en l'occurrence se brancher à la ligne 50, **ELSE** veut dire « sinon ».

Donc, si TEMPS\$ = « BEAU » l'ordinateur ira à la ligne 50 et exécutera l'instruction qui s'y trouve, c'est-à-dire



GOTO

END

PRINT « C'EST MAGNIFIQUE ! »
Sinon il ira à la ligne 60 en sautant les lignes 50 et 55 et il exécutera ce qui s'y trouve, c'est-à-dire :
PRINT « DOMMAGE. »

Le END en ligne 55 est placé là pour dire à l'ordinateur de s'arrêter, parce que : si la réponse est « BEAU », l'ordinateur exécutera, bien sûr, l'instruction de la ligne 50, mais exécuterait ensuite, sans END en ligne 55, celle de la suivante, 60.

Le résultat serait alors :

QUEL TEMPS FAIT-IL
AUJOURD'HUI ?
? BEAU
BEAU
C'EST MAGNIFIQUE !
DOMMAGE.
OK
—

A vous maintenant, mais n'oubliez jamais que :

- une variable alphanumérique contient des chaînes de 255 caractères maximum, composés de lettres, de chiffres ou de symboles. On accole toujours à son nom le symbole \$;
- une variable entière ne contient que des nombres entiers — donc pas de décimaux — et on inscrit toujours après son nom le symbole % ;
- une variable réelle contient uniquement des nombres, entiers ou décimaux. Il n'y a pas de symbole après son nom ;
- PRINT permet d'afficher à l'écran le contenu d'une variable : une chaîne de caractères écrits entre guillemets ou un nombre écrit sans guillemets ;
- INPUT permet, pendant le déroulement du programme, d'inscrire le contenu que l'on souhaite mettre dans une variable ;
- IF ... THEN ... ELSE permet de tester le contenu d'une variable. Si le contenu prend une certaine valeur, l'ordinateur exécute l'instruction qui lui est indiquée après le THEN, sinon, il exécute celle qui lui est indiquée après le ELSE ;
- GOTO donne l'ordre d'« aller se brancher » à la ligne d'instruction dont il indique le numéro, qu'elle soit avant ou après.

Résumé

Du clavier à la mémoire

Chacune des touches du clavier d'un ordinateur envoie une impulsion électrique, mémorisée directement par l'ordinateur sous la forme de l'octet correspondant. Ainsi, si on tape un T (code ASCII 84), l'ordinateur mémorise directement, non pas un T, mais l'octet 01010100 (voir tableau des correspondances). Et, de même, si on tape une suite de caractères, c'est l'ordinateur qui fait automatiquement la transcription sous la forme d'une suite de chiffres binaires comportant autant d'octets que de caractères entrés au clavier. TIFY sera ainsi mémorisé, lettre à lettre, dans l'ordinateur, sous la forme suivante :

Lettre tapée sur le clavier	Code ASCII (il n'intervient pas dans l'opération)	Octet mémorisé par l'ordinateur
T	84	01010100
I	73	01001001
F	70	01000110
Y	89	01011001

Notons que, comme l'ordinateur ne connaît pas l'ambiguïté, la différence entre lettres majuscules et lettres minuscules est clairement marquée par le code ASCII : le A majuscule et le a minuscule, par exemple, n'y ont rien à voir, ce sont deux lettres et deux codes complètement différents. Ainsi, le codage du mot Tify est notablement différent de celui de TIFY, c'est :

Lettre tapée	Code ASCII	Octet
T	84	01010100
i	105	01101001
f	102	01100110
y	121	01111001

Ces suites de chiffres binaires sont tout à fait claires pour la machine. Mais il n'en est pas de même pour l'homme : la preuve ! Essayez donc de trouver la signification du « texte » binaire suivant...

01001001	01100101	00101100
01101110	01101101	00100000
01100110	01100101	01110011
01101111	01101110	01101001
01110010	01110100	01100111
01101101	00100000	01101110
01100001	01110110	01100101
01110100	01101111	00100000
01101001	01110100	01010100
01110001	01110010	01101001
01110101	01100101	01100110
		01111001

C'est en fait très simple : il suffit de se reporter au tableau des correspondances caractères/code ASCII/octets... Et c'est encore plus simple dans la pratique, puisque l'ordinateur « traduit » lui-même et que l'utilisateur n'a qu'à se servir des chiffres ou caractères habituels.

Comment sont constituées les mémoires centrales : un peu d'histoire

La nécessité, en informatique, de conserver des informations existe même pour résoudre des problèmes très simples, telle l'addition. On sait bien, en effet, que lorsqu'on effectue une addition, (998 + 266) par exemple, il est nécessaire de poser des retenues, c'est-à-dire de mémoriser des chiffres :

retenues à mémoriser

A vertical addition problem on a green background. The numbers are 998, + 266, and a horizontal line, followed by 1 264. A red arrow points from the top of the 998 to a red box containing the number 11, which is positioned above the 998.

La mémoire à travers les âges

La première véritable machine à calculer, une machine mécanique, inventée en 1642 par Blaise Pascal, résolut le problème de la façon suivante : la retenue s'opère par un système d'engrenages faisant, à chaque fois qu'une roulette manœuvrée effectue un tour complet, avancer d'un cran (une unité) la roulette située juste à sa gauche (la roulette des unités est à droite de la roulette des dizaines, elle-même à droite de la roulette des centaines, etc.) ; les totaux apparaissent, colonne par colonne, au-dessus des roulettes. Les machines à calculer mécaniques encore utilisées de nos jours (de plus en plus rarement !) ont toujours recours au même principe.

Les ordinateurs actuels fonctionnent, eux, à l'électricité. Or l'électricité est essentiellement une circulation de particules, les électrons, donc quelque chose qui est toujours en mouvement. Heureusement on sait stocker cette énergie, sous forme chimique par exemple, dans une batterie.

La première mémoire à langage binaire de grande capacité, rapide et d'un coût réduit de stockage par bit, fut inventée aux États-Unis en

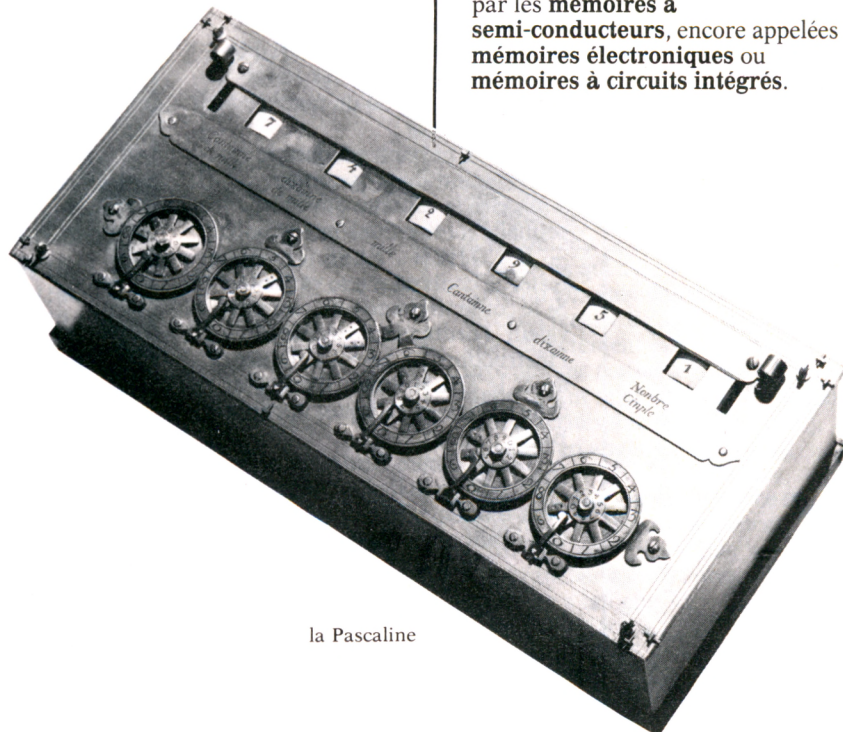
1950. Il s'agit de la mémoire à **tores de ferrite**. Sa conception repose sur le principe de l'aimantation, et plus particulièrement sur la propriété que possède le minerai magnétique d'avoir deux sens d'orientation possibles : (nord → sud) et (sud → nord) ; ce qui convient fortuitement mais admirablement à la traduction d'informations en langage binaire : à chacun des deux sens d'orientation est associé l'un des deux états binaires, 0 et 1. En pratique, on provoque, grâce à des impulsions électriques, l'orientation magnétique de séries de tores (ou anneaux, qu'il serait trop long, ici, de décrire en détail) afin de constituer une série de chiffres binaires regroupés en séries d'octets ; chaque tore correspondant, selon son orientation, à un 0 ou à un 1.

Les mémoires à tores de ferrite permettent d'obtenir des capacités suffisantes (un million d'anneaux correspondent à une capacité d'un million de bits), mais la détection

des champs magnétiques entraîne la perte des informations ; il faut donc réécrire les informations après lecture (cela se fait automatiquement mais le temps d'utilisation en est augmenté). On dit alors que la lecture est **destructrice**.

On peut comparer la lecture de ces champs magnétiques à ce qui se passe dans un jeu de bowling. Entrer une information en mémoire, c'est jouer, c'est-à-dire faire tomber un certain nombre de quilles, qu'on peut coder par des 0, les quilles restées debout l'étant par des 1. Les informations pourraient alors se lire de la façon suivante : la machine parcourt l'ensemble des quilles ; si une quille est par terre, elle la redresse en notant 0, si une quille est restée debout, elle la laisse telle et note 1. A la fin de la lecture, toutes les quilles sont donc à nouveau debout, mais la machine a noté les informations correspondantes.

Les mémoires à tores de ferrite sont maintenant tombées en désuétude, elles ont été remplacées par les **mémoires à semi-conducteurs**, encore appelées **mémoires électroniques** ou **mémoires à circuits intégrés**.



la Pascaline

La mémoire électronique

On peut considérer que le vrai « démarrage » de l'informatique et la naissance de la micro datent de l'emploi des mémoires électroniques: Ce sont les progrès technologiques réalisés, depuis, dans ce domaine – miniaturisation, accroissement de la rapidité et de la fiabilité, etc. – qui ont permis de créer des systèmes toujours plus compacts, toujours plus puissants et toujours moins chers.

A la base de ce type de mémoire se trouve l'association de deux composants électroniques : un transistor et un condensateur. On parle alors de mémoires à semi-conducteurs.

Le **transistor** (de l'anglais *TRANSfert resISTOR*) est une manière électronique de réaliser un interrupteur. Concrètement, le transistor oppose au passage du courant dans la ligne ab, soit une très forte, soit une très faible résistance, en fonction du courant de commande (sens + intensité) imposé en c. Les composants employés sont de la famille MOS (Métal, Oxyde, Silicium), basée sur des impuretés métalliques placées dans une **puce** de silicium.

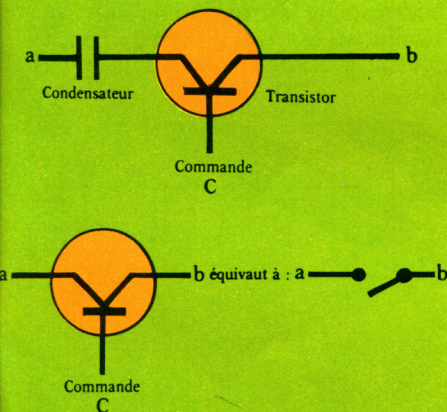
Le **condensateur** est un composant électronique pouvant présenter deux états : « chargé » ou « déchargé » ; et passer de l'un des deux états à l'autre en présence d'un courant de forte intensité.

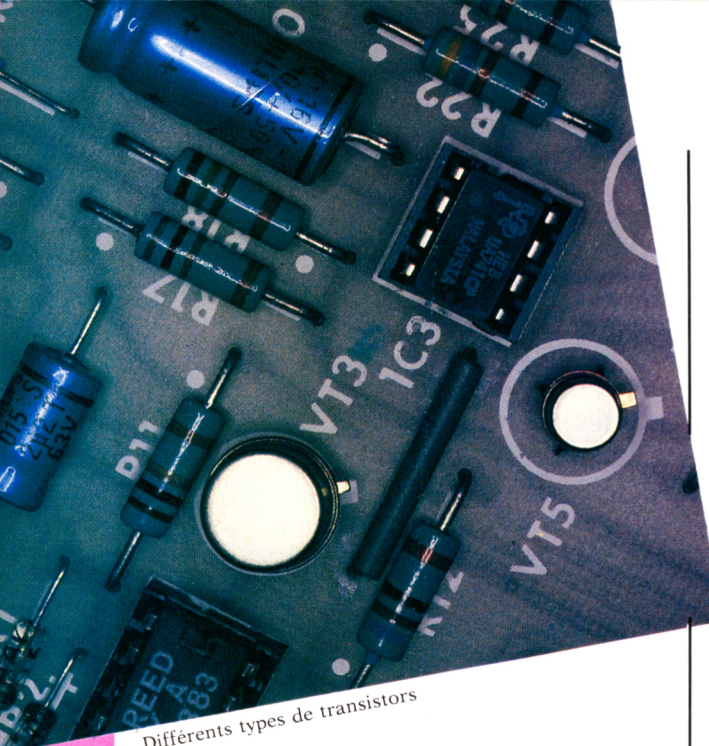
Ces particularités permettent parfaitement d'adapter les condensateurs à la mémorisation des informations. Si on admet, en effet, qu'un condensateur chargé vaut 1, et un condensateur déchargé 0, en réalisant le couplage d'un condensateur avec un transistor, on peut réaliser des opérations d'écriture et de lecture d'informations en langage binaire.

L'écriture se fait en deux temps. D'abord, on provoque un court-circuit (en activant le transistor sans faire passer de courant dans la ligne ab), ce qui met le condensateur à l'état 0 (déchargé), quel que soit son état initial. Ensuite, on peut écrire. Pour écrire la valeur 0, aucune opération n'est nécessaire : on dit que 0 est la valeur inscrite par défaut. Pour écrire la valeur 1, on déclenche, par la commande écrite au clavier de l'ordinateur, le passage du courant

dans la ligne ab : le condensateur change alors d'état, il se charge, sous l'effet du courant. C'est un peu comme lorsque l'on enregistre de la musique sur une cassette. Quand on appuie sur les touches du magnétophone qui provoquent l'enregistrement, les têtes effacent d'abord automatiquement, même si rien n'est enregistré, et ensuite, elles inscrivent sur la bande magnétique la fréquence qui permettra la reproduction de la musique que l'on enregistre. Pour lire, on provoque le déblocage du transistor. Si le condensateur est à l'état 0, rien ne passe. S'il est à l'état 1, il se décharge, provoquant, dans la ligne ab, la circulation d'un courant qui est détecté. L'opération se passe de la même manière que pour les mémoires à tores de ferrite, et la comparaison avec le jeu de bowling reste valable.

Les **mémoires à semi-conducteurs** posent le problème de la conservation de l'information. En effet, même si la résistance du transistor en état fermé est grande, il est impossible, à la longue, d'empêcher le passage d'un courant, d'intensité faible mais suffisante pour décharger le condensateur. Imaginez un réservoir fermé par un robinet. Ce dernier maintient très puissamment le réservoir fermé, mais la pression qui est dans le réservoir fait que le robinet fuit légèrement, par son joint par exemple. Pour éviter qu'à un moment donné le réservoir ne soit vide, il faudrait, de temps en temps, remettre de l'eau dedans. C'est exactement ce qui se passe pour les mémoires de l'ordinateur.





Différents types de transistors

On surmonte l'inconvénient de cette « fuite » en rafraîchissant périodiquement la mémoire, c'est-à-dire en réécrivant, à intervalles réguliers (toutes les 20 millisecondes, par exemple), les 1 et les 0 à leurs places respectives. Bien évidemment, lorsque le courant est coupé (au moment où on éteint l'ordinateur ou lors d'une rupture d'alimentation), toutes les informations sont définitivement perdues. Les mémoires de ce type sont donc volatiles et elles conviennent très bien pour réaliser des mémoires vives ou RAM.

Il existe un autre type de composants, appelés FAMOS (*Floating Avalanche MOS*), plus perfectionnés que les composants de technologie MOS que nous venons de décrire et qui sont capables de conserver les informations, même en l'absence de courant, pendant une durée approximative d'un siècle. Ce sont des mémoires programmées et, pour en effacer l'information, il faut les exposer à des rayons ultra-violet. Cette double caractéristique en fait des composants parfaits pour les mémoires mortes ou ROM.

Cette association d'un transistor et d'un condensateur constitue une cellule élémentaire de mémoire, permettant de mémoriser une information d'un bit. Les composants les plus récents, appelés **circuits intégrés**, peuvent, eux, réunir sur une seule pastille de silicium de quelques millimètres carrés, appelée **puce**, ou *chip* en anglais, l'équivalent de plusieurs dizaines, voire centaines de milliers de composants électroniques de ce type. Chaque puce est encapsulée dans un boîtier de plastique ou de céramique, muni de broches (ou « pattes ») métalliques.

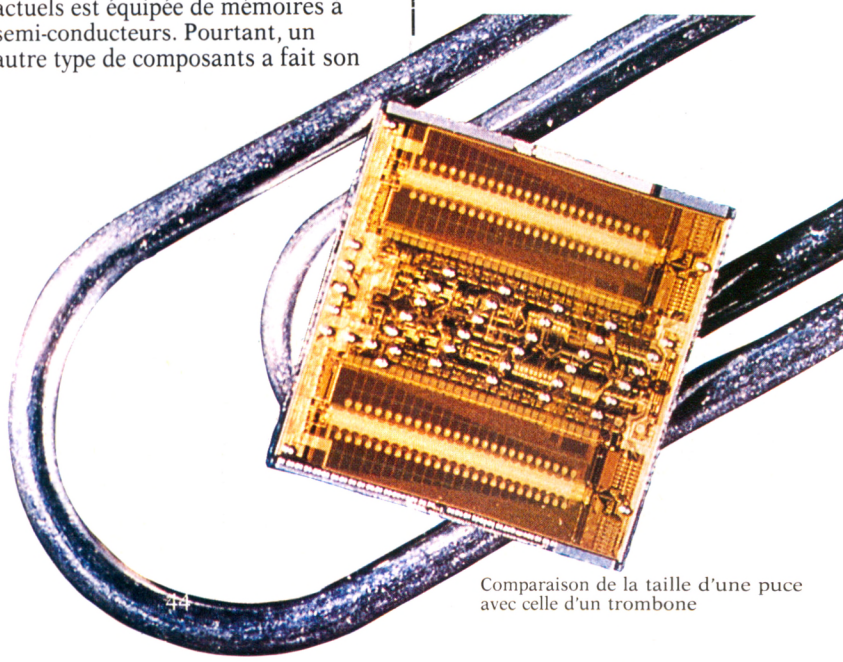
La quasi-totalité des ordinateurs actuels est équipée de mémoires à semi-conducteurs. Pourtant, un autre type de composants a fait son

apparition sur le marché, la **mémoire à bulles**. C'est une puce spéciale qui stocke les informations dans des bulles magnétiques pouvant prendre les valeurs 0 ou 1. Chaque nombre binaire y est représenté par une chaîne de bulles. Cette mémoire présente deux gros avantages : elle est non volatile et d'une très grande capacité par rapport à son encombrement. Seul son prix reste un obstacle à sa grande diffusion. Le Grid Compass, ordinateur de luxe, en possède une.

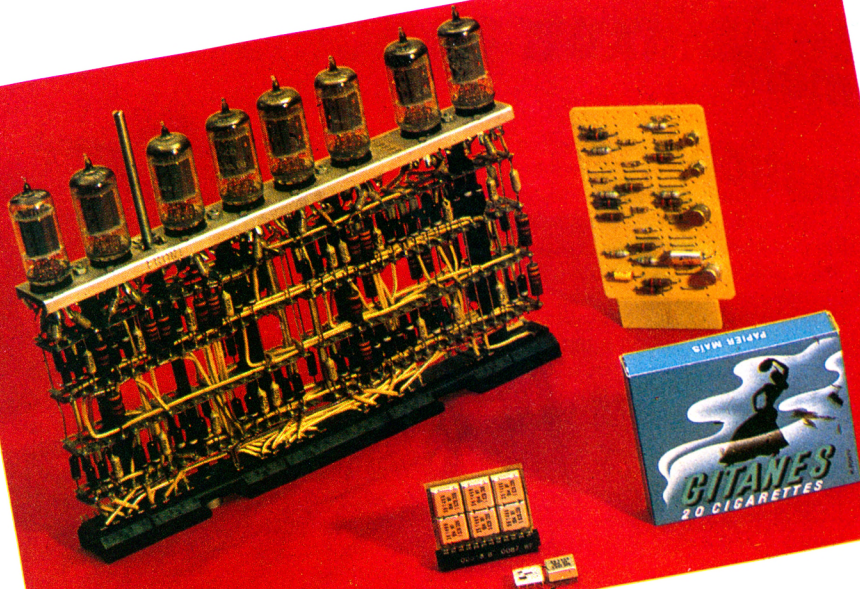
Chaque année, le prix des mémoires à semi-conducteurs baisse de près de 40 % ! Et leurs performances augmentent dans des proportions largement aussi impressionnantes... Depuis quelques années, le marché a donc gagné en importance, et il devient nécessaire d'établir des critères d'appréciation. Les tests destinés à juger de cette mémoire électronique sont, bien sûr, très différents de ceux qu'on appliquerait à la mémoire humaine ; ils portent sur des valeurs quantitatives faciles à déterminer : la **capacité** et la **rapidité**, notamment.

Capacité

La capacité d'une mémoire est définie en nombre de bits d'informations qu'elle peut contenir. Les capacités atteintes actuellement étant de l'ordre de la centaine de milliers, voire du million, de bits, on utilise plus couramment, comme unité de capacité, le **kilobit** et le **megabit**. Contrairement à l'usage courant, en informatique le préfixe « kilo » ne signifie pas exactement « 1000 », comme dans kilomètre, mais



Comparaison de la taille d'une puce avec celle d'un trombone



Accélération de la vitesse de commutation du courant électrique + miniaturisation
 = lampe → diode → transistor → puce

« 1024 ». Cette approximation se justifie par l'emploi, non pas du système décimal, mais du système binaire pour tout ce qui concerne les ordinateurs ; 1024 correspond en effet à la puissance de 2 la plus proche de 1000 (1000 étant une puissance de 10).

L'octet, ou groupe de huit bits, étant actuellement utilisé pour coder les informations, c'est, en fait, lui (ou ses multiples : le **kilo-octet** et le **méga-octet**) qu'on utilise de plus en plus souvent comme unité pour décrire la capacité d'une mémoire. La conversion entre les capacités exprimées en bits et en octets se fait simplement en divisant la première par huit.

En 1971, les premières puces au silicium pouvaient contenir 128 bits.

En 1983, on a pu construire une puce de 256 000 bits !

L'accroissement des capacités de stockage des mémoires à semi-conducteurs, ainsi que la

diminution des coûts de ces dernières, formidables au cours de ces dernières années, sont les domaines dans lesquels Japonais et Américains se livrent actuellement les plus âpres batailles. Les mémoires à grande capacité sont évaluées actuellement à 100 millions de bits !

Prix

Il faut savoir aussi que le prix des mémoires mortes (ROM) a beaucoup baissé. Il devient donc de plus en plus intéressant d'augmenter le nombre de programmes déjà inclus dans les premières, plutôt que de conserver ces derniers dans des mémoires auxiliaires (lesquelles restent quand même meilleur marché). C'est ainsi que, outre les langages, de nombreux logiciels sont désormais résidents.

Rapidité

Un autre critère important dans le domaine des mémoires est la rapidité. On dit qu'une mémoire est rapide lorsque l'on obtient dans un très bref délai les informations qu'elle contient. Cette qualité se caractérise par deux valeurs. D'une part, le **temps d'accès**, qui est l'intervalle séparant la fin d'un ordre de lecture du moment où l'information est disponible, c'est-à-dire le temps qui est mis par la mémoire pour retrouver l'information. Il est compris, selon les composants, entre 0,005 et 1 microseconde. D'autre part, le **cycle de base**, qui sépare la fin de l'ordre de lecture ou d'écriture de la fin de l'exécution de cet ordre. Le cycle de base correspond, en fait, au temps d'accès augmenté du temps mis pour lire ou écrire l'information.

CAPACITÉ MÉMOIRE DE QUELQUES MICRO-ORDINATEURS

ORDINATEURS	RAM (en Ko)	ROM (en Ko)
ALICE	4	8
APPLE 2	64	18
COMMODORE 64	64	20
IBM PC	128	40
IBM PC Jr	128	64 (sur cartouche)
MACINTOSH	128	64
MO5	48	16
ORIC/ATMOS	48	16
SHARP PC 1251	3,7	24
SIL'Z 16	256	4
TANDY TRS COLOR 2	16	16
TO 7	48	22
TOSHIBA/PAP	192	4
VIC 20	5	20
ZX 81	1	8
ZX SPECTRUM	48	16

Adressage, ou comment retrouver une information

On peut se représenter la mémoire d'un ordinateur comme une série de boîtes destinées à recevoir les informations transmises par le clavier. Mais, contrairement à ce qui se passe chez l'homme, il ne suffit pas à l'ordinateur d'entendre ou de lire quelque chose pour s'en souvenir.

Il faut qu'en plus lui soit donné l'ordre de se souvenir, c'est-à-dire de mettre en mémoire l'information que l'on souhaite qu'il conserve. Pour cela, il faut choisir une des petites boîtes, une case-mémoire, et dire à l'ordinateur d'y mettre l'information.

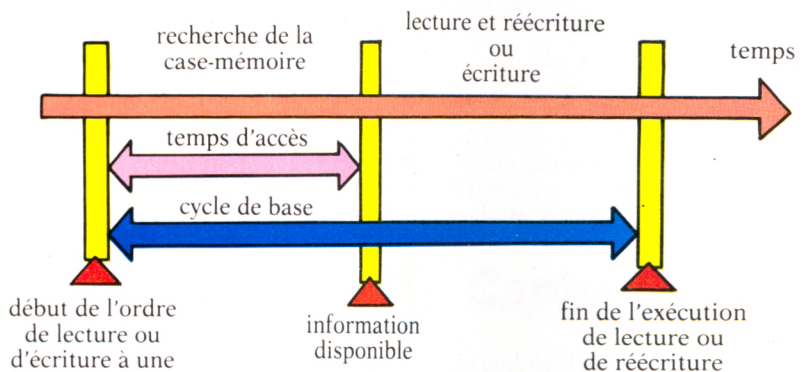
Chaque case-mémoire a une **adresse** (c'est-à-dire un code qui permet d'y accéder), par laquelle on la désigne à l'ordinateur. C'est cette adresse qui permettra ensuite de retrouver l'information. On dit que l'information est le **contenu** de l'adresse.

Pour **écrire** une information, on choisit donc arbitrairement une

case-mémoire vide, qu'on désigne par son adresse, et on y met un contenu : l'information. Pour **lire**, il faut connaître l'adresse et l'indiquer à la machine, qui sélectionne alors la case-mémoire correspondante.

Le schéma ci-dessous décrit le temps d'accès et le cycle de base d'une mémoire :

que l'ordinateur ait à lire ceux qui le précédent pour y parvenir). Le temps d'accès à une information est donc le même quelle que soit son adresse (autrement dit, deux informations situées à des adresses différentes nécessitent le même temps d'accès). De même, quand on veut se rendre chez M. Dupont, si on connaît son adresse, on peut sonner directement au 45 rue de



Les mémoires centrales (RAM et ROM) sont des mémoires à **accès direct** : cela signifie qu'un enregistrement peut, grâce à son adresse, être atteint indépendamment des autres (sans

Rivoli, par exemple, sans avoir à vérifier au préalable qu'il n'habite pas au 1, au 2 ... ou au 44. Nous verrons que pour les mémoires auxiliaires, ce n'est pas toujours le cas.

Jouez avec votre mémoire

En Basic, pour mettre une valeur dans une case-mémoire, on tape l'instruction :

POKE (x, y)

POKE permet d'écrire, à l'adresse x, la valeur y.

Et pour lire le contenu de x, on tape PRINT PEEK (x), l'ordinateur affiche alors :

```
PRINT PEEK (X)
Y
OK
-
```

PEEK appelle une adresse et en lit le contenu.

Pourquoi n'essayeriez-vous pas ? Branchez votre MO5 et tapez l'instruction :

PRINT PEEK (9104)

Votre ordinateur va aller lire à l'adresse 9104, puis il en inscrira le contenu :

```
PRINT PEEK
(9104)
157
OK
-
```

Si vous voulez changer le contenu de l'adresse 9104 et le remplacer par 200, tapez :

POKE 9104, 200

OK

Pour vérifier si le contenu a bien été changé, tapez ensuite :

PRINT PEEK (9104)

L'ordinateur affichera :

```
PRINT PEEK
(9104)
200
OK
-
```

Les mémoires auxiliaires

Etant donné la capacité limitée de la mémoire interne des micro-ordinateurs, il s'est avéré indispensable, pour stocker un volume important d'informations, de disposer d'un certain nombre de **mémoires auxiliaires**. Ces mémoires externes permettent soit de « décharger » temporairement la mémoire centrale, soit d'adoindre à l'ordinateur l'équivalent d'une bibliothèque (l'ensemble des informations conservées est alors appelé **fichier**) ; toutes les informations que l'on désire conserver ou archiver seront donc stockées sur une mémoire auxiliaire. Il existe des mémoires

de plusieurs types, mais elles possèdent toutes des caractéristiques communes : leur support (bande, disque, etc.) est toujours magnétique, alors que leur mécanisme de lecture/écriture est généralement mécanique ; leur rapidité est variable, mais en tout cas toujours moindre que celle de la mémoire électronique ; enfin — et c'est là leur principal point fort — elles sont d'un relativement faible rapport prix de revient/capacité.

En micro-informatique, les deux procédés les plus utilisés sont les **cassettes magnétiques** « audio » et les **disquettes**, ou disques souples.

L'enregistrement magnétique

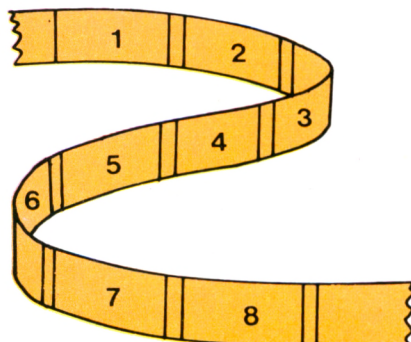
Le principe de l'enregistrement magnétique est l'utilisation de la « polarisation magnétique », c'est-à-dire de l'existence, sur les aimants, d'un pôle nord et d'un pôle sud. Le support d'une bande magnétique, généralement du Mylar, est recouvert d'une pellicule de « peinture magnétique » (à l'oxyde de fer) dont chaque particule se comporte comme un petit aimant que l'on peut orienter. On utilise pour cela un champ

électro-magnétique produit par un électro-aimant contenu dans une tête d'écriture. Les particules sont alors considérées comme des bits de valeur 1 ou 0 selon qu'elles sont orientées nord-sud ou sud-nord. A la lecture, les variations d'aimantation produisent dans la tête de lecture des courants induits qui peuvent être détectés. On en déduit les polarisations des particules et, donc, la valeur des bits qu'elles représentent.

Les cassettes magnétiques

C'est la mémoire auxiliaire la moins coûteuse pour un utilisateur de micro-ordinateur. Il s'agit, en effet, de cassettes, et d'un lecteur-enregistreur, du même type que ceux qu'on utilise couramment pour écouter de la musique. En principe, les magnétophones du commerce conviennent parfaitement à un usage informatique. Cependant, certains constructeurs, dont Thomson, préfèrent, pour des raisons commerciales ou qualitatives, employer des matériels plus spécifiques. Aucune caractéristique spéciale n'est exigée, et même des cassettes d'une qualité moyenne peuvent être utilisées (il existe cependant des cassettes « qualité ordinateur », mais qui ne correspondent à aucune norme établie). La longueur des bandes est

variable (elle dépend de la longueur d'enregistrement autorisée), mais la largeur est constante : 3,81 mm. La densité d'enregistrement est de 320 bits (40 octets) par centimètre. L'enregistrement peut théoriquement être aussi long que la bande, mais, dans la pratique, des enregistrements successifs sont séparés par des blancs.

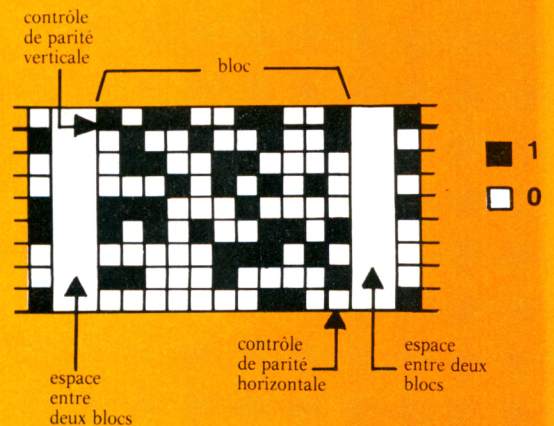


L'enregistrement sur bande

Sur une bande magnétique, les informations sont enregistrées par blocs. Généralement, la bande est découpée en neuf pistes parallèles (huit pour un octet, écrit perpendiculairement au sens de déroulement de la bande, plus une pour effectuer un contrôle de parité). Un caractère est donc enregistré, lui aussi, perpendiculairement au sens de défilement de la bande, en même temps que se fait, sur la 9^e piste, ce qu'on appelle un **contrôle de parité vertical**. A la fin de l'écriture de chaque bloc, on effectue un **contrôle de parité horizontal**, puis on laisse un espace avant le bloc suivant.

Un contrôle de parité s'effectue à partir du nombre de 1 compris dans l'octet à contrôler : si celui-ci est pair, le bit de parité vaut 0, s'il est impair, le bit vaut 1. Par exemple, l'octet « 0001 1010 » comporte trois 1, son contrôle de parité indiquera donc « 1 ».

Ces contrôles de parité permettent, en particulier, de détecter les erreurs d'enregistrement : quand le micro-ordinateur relit un enregistrement, il ne compare les bits de parité qu'avec ceux du modèle original, et peut ainsi signaler les erreurs éventuelles.

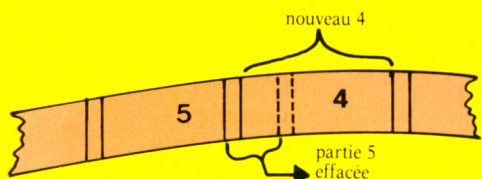


Quand on veut rechercher un enregistrement déjà réalisé sur une cassette, on est obligé de « dérouler » la bande ; ainsi, si l'on recherche, par exemple, le 25^e enregistrement de la cassette, on est obligé de parcourir — de franchir « en séquence » — les 24 précédents ; ce qui fait que le temps d'accès à une information

stockée sur bande magnétique est généralement assez long – il est fonction de sa position sur la bande. Pour cette raison, on dit des cassettes qu'elles sont des mémoires à **accès séquentiel** (ce qui les distingue des mémoires adressables).

L'autre inconvénient des cassettes est leur manque de fiabilité : fragiles, elles se rayent facilement, et leur revêtement magnétique a tendance à disparaître à l'usage. Pour atténuer les conséquences de ce genre de risques, il est souvent conseillé de réaliser les enregistrements sur cassettes en deux exemplaires (sur deux cassettes distinctes). Le procédé est sécurisant mais coûteux !

L'accès séquentiel aux informations stockées sur bande présente également un inconvénient en cas de réécriture sur un enregistrement ancien. Supposons, par exemple, que l'on ait réalisé, sur une même bande, plusieurs enregistrements successifs, numérotés de 1 à 6 et



correspondant aux noms d'un « fichier » client (chaque enregistrement correspond à un nom) ; si on veut modifier le 4^e nom, et le remplacer par un nom plus long, on va, en réenregistrant ce 4^e nom, effacer le début du 5^e nom ; c'est le même problème que pour les enregistrements de musique : on perd un morceau de la 5^e information ! Cette difficulté à modifier les informations stockées sur des bandes magnétiques rend difficile leur emploi professionnel ; on réserve généralement leur usage à la micro-informatique familiale.

Les disques souples (ou disquettes)

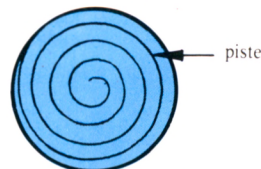


Pour pallier les inconvénients que présentent les cassettes magnétiques (lenteur, faible fiabilité et accès séquentiel), on a inventé un autre type de mémoire périphérique : les disquettes. Malheureusement, le prix de leur lecteur spécifique est beaucoup plus élevé que celui du lecteur de bandes magnétiques (de l'ordre de plusieurs milliers de francs). Une disquette est un petit disque souple, fait d'une matière qui s'appelle Mylar et dont la surface, lisse, est recouverte d'une pellicule d'oxyde magnétique. Sa matière première est donc semblable à celle des bandes magnétiques, mais elle est de meilleure qualité (épaisseur du Mylar et de la couche d'oxyde). Ce petit disque est maintenu dans une enveloppe protectrice de carton semi-rigide qu'il ne quitte jamais ; il peut y tourner librement, à une vitesse de 300 à 360 tours par minute.

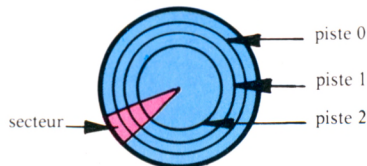
Les disques souples existent en de nombreux formats, mais les plus courants sont de 8 pouces (soit 203 mm de diamètre) ou de 5 pouces 1/4 (soit 133 mm de diamètre). Les impulsions magnétiques correspondant aux informations y sont enregistrées sur des pistes circulaires et concentriques (et non en spirales, comme sur les disques

microsilons) – chacune des pistes peut donc être numérotée. Le nombre de pistes est variable selon les fabricants, mais en général les disquettes 8 pouces possèdent 77 pistes, alors que les disquettes 5 pouces 1/4 en possèdent de 35 à 40. Bien que les pistes soient de plus en plus petites à mesure que l'on se rapproche du centre, elles ont toutes la même capacité de stockage des informations ; ce qui sous-entend que la densité d'informations est plus faible au bord qu'au centre.

Grâce au numéro de la piste où elle est stockée, l'accès à une information mémorisée sur une disquette se fait directement, sans avoir à parcourir les précédentes. Pour accélérer encore la recherche, on divise même généralement chaque piste en un nombre fixe de sections de même taille, les **secteurs** – c'est cette sectorisation



Disque musical



Disquette

qui rend nécessaire l'identité du volume d'informations contenues dans chaque piste, et ce malgré la perte de capacité que cela entraîne pour l'ensemble de la disquette. Pour accéder à l'information recherchée, il suffit de préciser à l'ordinateur, qui s'y reporte directement, les numéros de piste et de secteur correspondants, puis de faire défiler séquentiellement les informations – généralement peu nombreuses – contenues à l'intérieur du secteur sélectionné.

Initialisation ou formatage d'une disquette

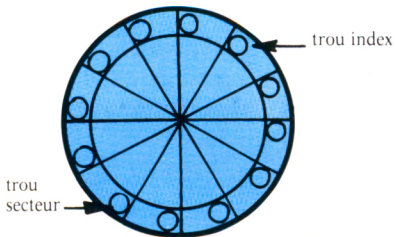
Les disquettes vierges sont inutilisables, tant en lecture qu'en écriture, sans une opération particulière qui consiste à :

- séparer chacune des 40 pistes en 16 secteurs égaux de 128 octets ;
- préparer le catalogue, c'est-à-dire la « table des matières » de la disquette – le contenu de

chaque secteur y est décrit. Cette opération est effectuée grâce à une instruction spécifique à chacun des langages utilisés. En Basic, on écrit : `DSKINI 0` Le lecteur de disquette n° 0 effectue alors l'opération. Si on possède plusieurs lecteurs, ils porteront les numéros : 0, 1, 2 ou 3.

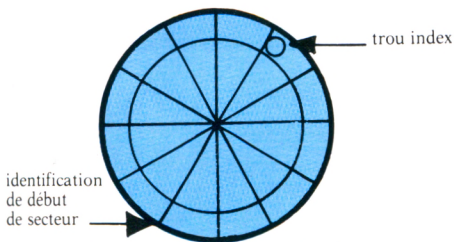
En Logo, on écrit : `FORMATE 0` Pour le MO5 et le TO 7 de Thomson, cette opération est compatible, c'est-à-dire qu'une disquette initialisée sur TO 7 peut être lue sur un système MO5 et réciproquement.

Les disquettes sont, contrairement aux bandes magnétiques, des mémoires **adressables** (permettant un accès direct aux informations – si l'on excepte le défilement séquentiel à l'intérieur du secteur). Ce procédé d'accès direct à l'information permet des temps d'accès notablement réduits – de l'ordre de 200 à 400 millisecondes. L'identification des secteurs, donc leur numérotation pour l'adressage, peut se faire de deux manières : soit physiquement, soit par logiciel (programme communiqué à l'ordinateur). La **sectorisation physique**, de moins en moins



Sectorisation matérielle

employée, est la plus simple. La position de chaque secteur est repérée par un trou situé sur la disquette. Le nombre de trous correspond donc au nombre de secteurs par piste (de 8 à 32), et l'un d'entre eux – le **trou index** – sert de base à l'ordinateur pour retrouver les autres.



Sectorisation logicielle

Malheureusement, ce procédé nécessite l'adaptation des disquettes aux spécificités du micro-ordinateur (fixées par le constructeur). Or, il existe une grande disparité entre ces spécificités, ce qui fait que ce procédé est de plus en plus souvent remplacé par la sectorisation logicielle : le **formatage**. La disquette ne comporte alors qu'un seul trou, le trou index, repérant la position du premier secteur. Aucun autre repère ne figure sur la disquette qui, « vierge » à son achat, ne peut être utilisée sans avoir, au préalable, été formatée. Le formatage étant une fonction programmée, l'ordinateur le réalise automatiquement. Concrètement, le formatage consiste à enregistrer, au début de chaque secteur, afin que l'ordinateur puisse repérer ce dernier, une suite de bits codée.

Notons que, quand on réalise un enregistrement, celui-ci peut très bien déborder sur plusieurs secteurs. C'est alors le micro-ordinateur qui prend en charge le lien entre les différents secteurs (on dit le **chainage**).

Avec des valeurs, parmi les plus courantes, de 16 secteurs par piste et de 128 octets par secteur, une disquette 5 pouces 1/4 (40 pistes, par exemple) peut contenir : $40 \times 16 \times 128 = 81.920$ octets, soit plus de 80.000 caractères ! Une disquette 8 pouces (77 pistes) contiendrait, elle : $77 \times 16 \times 128 = 157.696$ octets, soit plus de 155.000 caractères ! Les disquettes sont donc des mémoires à grande capacité si on les compare aux mémoires internes du micro-ordinateur.

Les disquettes du marché sont, outre leur dimension, classées selon différents critères. On distingue, en particulier, les disquettes **simple face** (S.F.) et les disquettes **double face** (D.F.), selon qu'elles sont utilisables sur une seule face ou sur les deux, et surtout les disquettes **simple densité** (S.D.) et les disquettes **double densité** (D.D.), selon le nombre d'octets (128 ou 256) mémorisables par secteur. La capacité totale de la disquette dépend, en plus, du nombre de secteurs (mais les rapports entre les différents types de disquettes sont sensiblement constants). A côté des formats 8 pouces et 5 pouces 1/4, on assiste actuellement à l'émergence de disquettes plus compactes, bien que de mêmes capacités. En particulier, le format 3 pouces 5 semble s'imposer comme un nouveau standard.

Bien mieux adaptées à l'usage informatique que les cassettes magnétiques (fiabilité, rapidité, accès direct, facilité de stockage), les disquettes se sont désormais imposées – et cela malgré le coût élevé de leur lecteur – pour tout ce qui est application professionnelle ; et, bien qu'elles ne soient pas à la portée de toutes les bourses, on commence à les rencontrer dans le cadre des utilisations familiales ou amateurs.

Il faut savoir, cependant, que les disquettes doivent être manipulées avec précaution et qu'en particulier il ne faut pas mettre les doigts sur la partie magnétique visible ni exposer la disquette à un champ magnétique, la plier, ou l'introduire à l'envers dans le lecteur.

Les mémoires de l'avenir

Si les bandes magnétiques et les disquettes restent actuellement les mémoires auxiliaires les plus répandues sur le marché, il n'en est pas moins vrai que d'autres matériels y ont déjà fait leur apparition et risquent de modifier profondément le paysage micro-informatique des cinq ans à venir ; ce sont, notamment, les **disques durs**, les **vidéodisques** et les **disques optiques numériques**. Tous sont encore, cependant, d'un prix très élevé.

Avec leurs mémoires à très grande capacité (elles peuvent contenir de 5 à plusieurs centaines de méga-octets) et leurs temps d'accès extrêmement réduits (quelques dixièmes de seconde), les disques durs sont de plus en plus fréquemment utilisés en micro-informatique professionnelle.

Le vidéodisque fut conçu, à l'origine, pour obtenir des images vidéo de meilleure qualité que celles enregistrées sur des bandes magnétiques. Il permet, lui aussi, de mémoriser plusieurs millions de caractères et de les restituer quasi instantanément. Sa première application en informatique fut la création de jeux d'arcade inter-actifs (Mach 3, Astraubelt, Dragon's Lair, etc.), mais son utilisation « grand public », si elle est possible techniquement, reste encore limitée, du fait du prix des lecteurs de vidéodisques (plusieurs dizaines de milliers de francs).

Quant au disque optique numérique, plus connu sous le nom de **disque laser**, il offre une possibilité de stockage très importante, ainsi qu'un temps d'accès réduit, tout en assurant une qualité de reproduction et une fiabilité presque parfaites.



La bibliothèque Sainte Geneviève

Une banque d'enluminures sur vidéodisque interactif

Compliments, Adso, continua le bibliothécaire, effectivement ces images nous parlent de cette région où l'on arrive en chevauchant une oie bleue, où l'on trouve des éperviers qui pêchent des poissons dans un ruisseau, des ours qui pourchassent des faucons dans le ciel, des écrevisses qui volent avec les colombes et trois géants pris au piège et mordus par un coq.

« Le Nom de la rose »,
par Umberto Eco (Grasset)

La bibliothèque universitaire Sainte Geneviève à Paris possède une collection de manuscrits du Moyen Age célèbre dans le monde entier. Malgré sa grande valeur de

référence pour les historiens et les chercheurs, ce patrimoine d'enluminures était, jusqu'à maintenant, peu utilisé. Ces manuscrits très anciens sont, en effet, fragiles, il n'est donc pas question de les feuilleter comme des albums de bandes dessinées.

Il y a quelques années, Denise Hillard, alors conservateur à la bibliothèque, eut l'idée de proposer l'accès non pas à l'enluminure originale mais à une photographie. 3 500 diapositives furent ainsi réalisées par François Garnier, ce qui représente la quasi-exhaustivité des enluminures. Mais pour que la consultation des diapositives ne demande pas trop de temps, il fallait associer à chaque image une fiche perforée dont les différents trous correspondaient aux thèmes

de l'image. Grâce à une succession de tris des fiches selon les critères de son choix, le lecteur pouvait obtenir le numéro de la (ou des) diapositive qui l'intéressait. Restait alors à chercher la photo parmi les cent boîtes de 35 diapositives, ce qui n'était pas non plus très facile ni très rapide.

De la rencontre entre Christiane Baryla, actuel conservateur de la bibliothèque Sainte Geneviève, et l'équipe d'informaticiens d'Imedia, fin 1982, naquit ensuite l'idée d'un vidéodisque interactif.

N'importe quel historien (ou personne tout simplement intéressée) aura bientôt accès à ce vidéodisque.



Pour faire fonctionner le système : deux écrans au minimum (un pour les menus, l'autre pour les images), un clavier et la liste des symboles choisis par le créateur du vidéodisque.



Le premier menu est un sommaire proposant cinq critères d'accès : thème, sujet, siècle, manuscrit, utilisation.



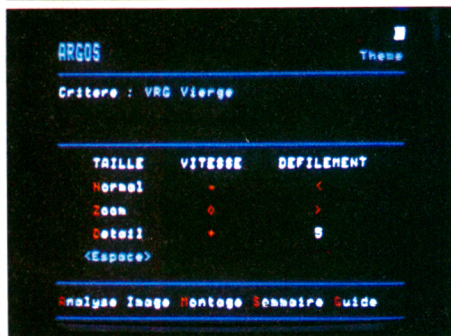
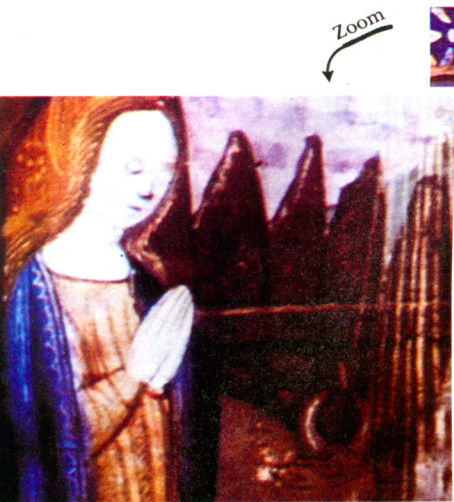
Recherche d'une enluminure ! mode d'emploi

Vous étudiez, par exemple, la manière dont les moines du Moyen Age représentaient les vierges dans les livres religieux. Vous arrivez devant un poste de travail qui se compose d'un clavier, d'un écran de télévision et d'un écran de micro-ordinateur sur lequel est affiché un sommaire (un **menu**) vous proposant cinq possibilités de choix. Si vous tapez sur la touche 1 (accès par thème), l'ordinateur répond en vous proposant sept thèmes. Tapez de nouveau 1 (religion), le nouvel écran affiché vous proposera seize têtes de chapitres repérées par des lettres, de A à P. En frappant la lettre D, par exemple, vous pourrez afficher sur l'écran toutes les enluminures représentant des vierges.

La touche > vous permet de faire défiler toutes les images représentant des vierges, exactement comme si vous feuilletiez un livre. En appuyant sur la touche < vous reviendrez en arrière dans la série des images sélectionnées. Si vous voulez voir l'image d'un peu plus près, la touche Z (zoom) vous donnera une image plein écran et la touche D (détail) mettra l'accent sur un détail particulièrement intéressant. Affinez encore votre question. Cherchez les images représentant le sacre de la vierge. Tapez VRG (code vierge) et SCR (code sacre) – la liste de ces codes (établis par le créateur du vidéodisque), le thésaurus, est indispensable pour faire fonctionner le système. Un traitement se déclenche et trouve les images concernées. Tapez V pour les visualiser. Si vous avez l'intention de revenir et d'interroger souvent ces images, tapez M pour mémoriser votre

sélection, à laquelle vous attribuerez le code à trois lettres de votre choix, par exemple VSC (vierge/sacre). Vous avez maintenant repéré les enluminures sur lesquelles vous voulez faire votre étude. Il ne vous reste plus qu'à vous reporter au manuscrit et à la page correspondants – plus de feuilletage inutile !

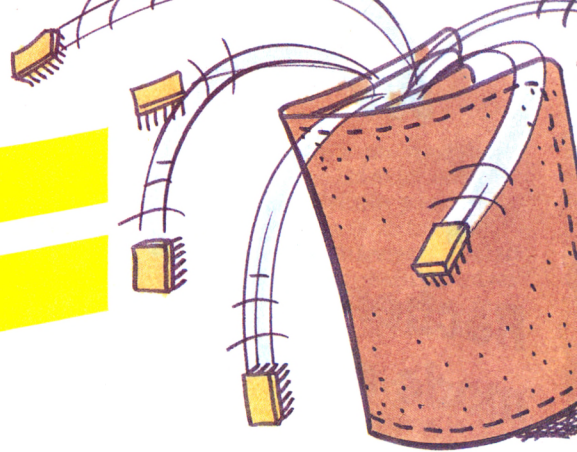
Le vidéodisque est sans doute l'occasion de redonner vie à d'immenses réserves d'images dormant dans des placards. Mais, outre le problème, plus juridique que technique, des droits du propriétaire de l'image originale (dans le cas, par exemple, où on couplerait le vidéodisque avec un magnéscope, afin de recopier des parcours d'images), la création de ces vidéodisques représente un énorme travail. Ce serait pourtant la mise à la disposition du grand public de véritables encyclopédies électroniques, soit sur place, dans les bibliothèques, soit à distance par l'intermédiaire de réseaux de type Teletel, et ce pourrait être l'un des éléments fondamentaux des systèmes d'Enseignement Assisté par Ordinateur.



Autre type de menu — Une fois l'accès (ici le thème) et le critère (ici VRG, c'est-à-dire Vierge) choisis, trois actions sont possibles : sur la taille de l'image (normal, zoom, détail) et sur la vitesse et le sens de défilement des images.

Avec l'aimable autorisation de l'Association Imedia. 21, rue Rollin Paris 5^e.

Cartes à mémoire : des puces dans votre portefeuille !



L'idée d'intégrer une « puce » – composant électronique – dans une carte en matière plastique n'est pas une idée neuve : c'est en effet en 1974 qu'un ingénieur français, Roland Moreno, en crée, dans son appartement, le premier prototype. Depuis, les développements vont

bon train et, si les Français conservent encore une notable avance, quelques pays étrangers (Japon, États-Unis) commencent à s'intéresser de très près à ce « produit-miracle ». Cette « smart carte » – carte astucieuse, comme disent les

Américains – risque en effet de révolutionner notre mode de vie dans un proche avenir ! A l'heure actuelle, seules quelques applications, encore timides, sont en cours de développement, mais le futur est en marche.

La « monétique » et le télépaiement

Tout le monde connaît les cartes de crédit : Carte Bleue ou autres. Elles présentent certains inconvénients, dont celui de ne permettre aucune vérification directe de la position du compte.

Avec la carte à mémoire, plus de problèmes. Le solde de votre compte est à tout instant « mémorisé » sur la puce. Les débits sont immédiatement répercutés, et

les achats ne sont autorisés que si votre compte est créditeur. Un bon garde-fou pour les imprévoyants ! Mieux encore, avec les développements des réseaux télématiques, le « télépaiement » est désormais possible, et en « temps réel ». Vous passez une commande, par Minitel par exemple, et la facture est aussitôt encaissée, sans que vous ayez à vous déplacer !

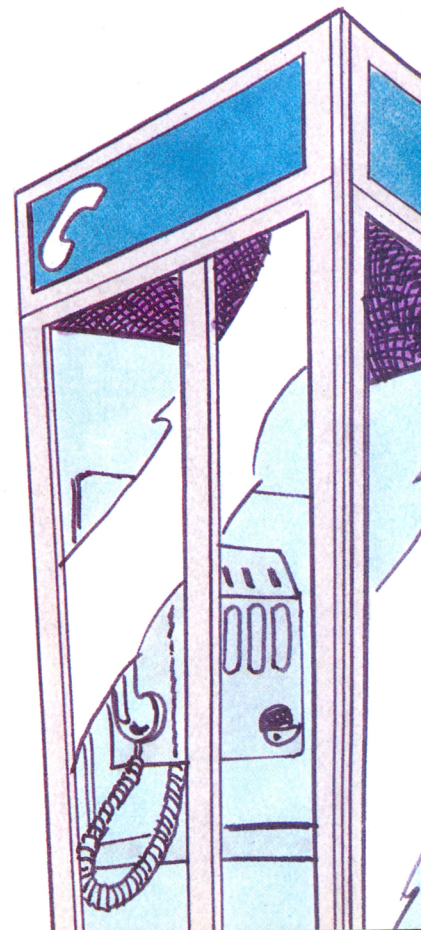
Publiphone

Un autre type de carte à mémoire est, grâce aux PTT, depuis peu en circulation. La carte de crédit téléphonique, pré-payée ou débitée directement sur le compte bancaire, permet de téléphoner d'une cabine publique sans avoir besoin de pièces de monnaie. De plus, l'absence d'argent liquide dans les cabines les protégera contre le vandalisme.

Vers un dossier de santé individuel ?

Le projet SESAM de la Sécurité Sociale est, malgré des difficultés certaines, sans doute l'un des projets d'application majeurs de la carte à mémoire. Le dossier de chaque malade est actuellement constitué des « réglementaires » feuilles de maladie, à la circulation difficile (circulation entre malade, médecin, pharmacien et caisses de maladie). Avec les cartes à mémoire, la saisie des informations se fera directement à la source, chez le médecin, le pharmacien, etc. Les caisses recevront les informations par un réseau télématique (Minitel par exemple), ce qui supprimera un travail long et fastidieux. De plus, et le point n'est certes pas négligeable, ces cartes préserveront le secret médical, et éviteront que des informations confidentielles ne circulent sur des

imprimés en trois exemplaires ! Pour l'instant, le projet SESAM est freiné par des impératifs financiers : un tel plan n'est évidemment viable qu'étendu à l'ensemble des assurés sociaux, et le financement de 30 millions de cartes à mémoire n'est pas évident ! Cependant, des progrès notables ont été réalisés dans ce domaine, comme la carte d'identité de porteur de simulateur cardiaque. Cette carte, de la taille d'une carte de crédit, renferme 220 renseignements médicaux et chirurgicaux concernant le malade : l'équivalent d'un dossier de huit pages ! Les informations contenues sont confidentielles, réservées aux praticiens, mais nécessaires en cas d'accident. Cette carte contribue donc à une plus grande efficacité de la surveillance cardiaque.



Les programmes

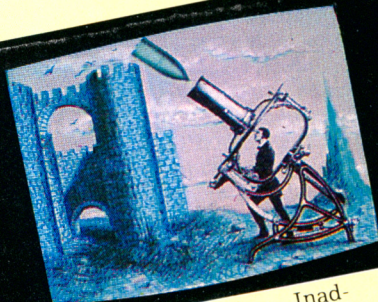
Aussi puissant que soit un ordinateur, il n'en reste pas moins une simple machine ; et pour qu'il puisse résoudre un problème, il faut lui indiquer avec précision : les données, les résultats attendus et la manière d'y parvenir. De plus, pour être rendues « compréhensibles » par la machine, toutes ces indications doivent être analysées (découpées en éléments simples) et dénuées de toute ambiguïté, puis être « traduites » sous une forme intelligible pour l'ordinateur. Ces différentes étapes – qui constituent ce qu'on appelle la **programmation** – sont, en informatique, le lieu privilégié de l'intervention humaine. Il s'agit, en effet, pour l'homme, de plier son propre mode de raisonnement, avec ses qualités d'intuition et d'imagination, à la logique de la machine. C'est la condition *sine qua non* pour bénéficier de la puissance de calcul de cette dernière.

L'accent mis aujourd'hui sur la réalisation de programmes toujours plus performants marque d'ailleurs une nouvelle étape dans le développement de l'informatique : après la prédominance des matériels – le **hardware** –, voici venir l'ère du logiciel (les programmes) – le **software**...

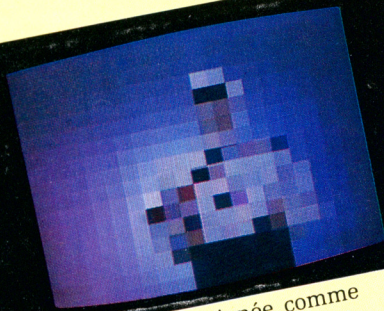
Monsieur Plumeau, employé modèle, s'endort régulièrement sur son clavier... Il se transforme alors en un abominable robot, Intox 639, qui décide, avec l'aide de son complice, l'ordinateur numéro 1, de mettre l'univers à ses pieds ! Mais il y a une faille...



« Les comportements humains doivent être rationalisés, déclare Intox 639. Ordinateur numéro 1, faites-m'en apparaître les différents échantillons ! »




– Inadmissible, Inadmissible !
– Répréhensible, répréhensible, répréhensible !
– Pas possible, pas possible, pas possible, pas possible ! »



Zoé est désignée comme mannequin de programmation pour établir le prototype de la secrétaire uniformisée. Sa réaction ne se fait pas attendre...

« L'ordinateur, c'est fait pour donner de la liberté, pas pour en prendre !... »



Qu'est-ce qu'un programme ?

Le terme de **programme** vient du grec, et signifie étymologiquement « ordre du jour », c'est-à-dire liste, généralement écrite, d'un ensemble de tâches devant être réalisées chronologiquement, en vue d'un objectif précis. Cette signification première n'a, de nos jours, pas vraiment changé ; ainsi parle-t-on de « programme d'une journée » ou de « programme télévisé ».

Nombre des activités qu'on imagine spontanées sont également soumises à des programmes rigides. Un oiseau qui construit son nid, par exemple, exécute un certain nombre d'actions, toujours les mêmes, toutes nécessaires, et invariablement dans le même ordre : choix d'un emplacement, recherche d'une première brindille, dépôt de la brindille à la fourche de deux branches, recherche d'une deuxième brindille, etc. La suite d'opérations se reproduit, dans ce cas, un certain nombre de fois (jusqu'à ce que le nid soit terminé) ; mais, quoi qu'il arrive, l'ordre logique est impérativement respecté : l'oiseau va chercher une

brindille *puis* la pose à l'endroit choisi. La suite des tâches est d'ailleurs, dans son cas, exécutée « automatiquement » ; elle a été programmée en l'oiseau avant sa naissance ; la façon de construire son nid est inscrite génétiquement dans sa mémoire.

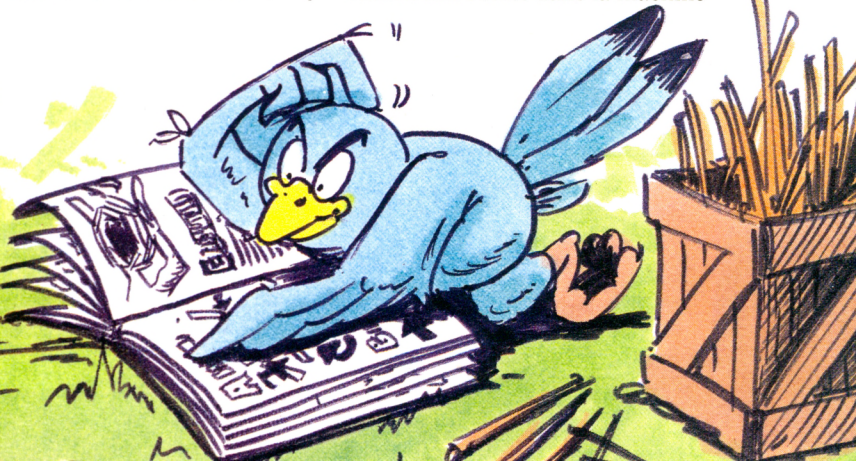
En informatique, un programme c'est aussi une suite d'instructions, données à l'ordinateur, en vue de la réalisation d'une tâche précise. Et, comme la logique du calcul mathématique se prête parfaitement au traitement par des machines, les premiers programmes destinés à des machines furent créés en vue de la résolution de problèmes de calcul.

Ainsi l'idée qu'avait eue Charles Babbage en concevant son engin analytique fut-elle concrétisée, au milieu du XX^e siècle, par les chercheurs d'I.B.M.-Harvard qui mirent au point un calculateur pesant plus de 5 tonnes et comportant 5 000 relais, le Mark I ; les programmes et les informations étaient introduits dans la machine

L'ordinateur

Ne rêvez pas,
pointez, grattez, marnez, vaquez,
trimez, bossez.
Ne rêvez pas,
l'électronique rêvera pour vous.
Ne lisez pas,
l'électroliseur lira pour vous.
Ne faites pas l'amour,
l'électrocoital le fera pour vous.
Pointez, grattez, marnez, vaquez,
trimez, bossez.
Ne vous reposez pas,
le travail repose sur vous.

Jacques Prévert



par des rubans perforés, et les mémoires (non pas électriques mais mécaniques) étaient constituées de roues et de cadrans. Une fois mis en route et programmé, ce calculateur ne nécessitait plus d'intervention humaine. Conçu uniquement pour le calcul numérique, il était capable d'additionner ou de soustraire deux nombres de 23 chiffres en 3 dixièmes de seconde, et de les multiplier en moins de 6 secondes. Un an plus tard, une autre équipe d'I.B.M., dirigée par John Von Neumann, réalisait un calculateur électronique, l'I.B.M. SSEC, comportant 24 100 relais électro-mécaniques et 13 500 tubes à vide. Il était 100 fois plus rapide que Mark I et, pour la première fois dans l'histoire du calcul automatique, les instructions du programme n'étaient plus introduites au fur et à mesure du

A l'heure actuelle, de nombreux micro-ordinateurs familiaux, capables de traiter non seulement les nombres, mais aussi les mots, les phrases, les images et les sons, sont aussi rapides et puissants que les « monstres » des années cinquante ; et cela pour un prix et un encombrement incomparablement réduits.

Algorithmes et organigrammes

Quand on veut résoudre un problème à l'aide d'un ordinateur, il faut donc, tout d'abord, le poser clairement et définir parfaitement, tant les données initiales (ce que l'on connaît), que les résultats attendus (ce que l'on veut savoir). Ensuite, il faut décomposer la marche à suivre par l'ordinateur en une suite logiquement ordonnée d'actions élémentaires que l'on pourra éventuellement répéter. La description de cette suite d'actions s'appelle un **algorithme**.

Prenons un exemple dans la vie courante : les instructions données, dans un magazine féminin, pour réaliser un carré au crochet ; c'est un exemple particulièrement simple d'algorithme. Les données initiales sont les fournitures : type et quantité de coton, taille du crochet. Les actions élémentaires sont les points à réaliser (bride, maille en l'air et maille coulée) supposés connus. Et le programme est la description, compte tenu des données, de la suite d'actions élémentaires à effectuer pour réaliser le carré. Cet algorithme décrit la seule manière de réaliser un carré au crochet en partant des fournitures données, mais certains problèmes peuvent admettre plusieurs solutions (il existe plusieurs manières de faire une multiplication, par exemple, bien que le résultat soit toujours le même). Il existe alors plusieurs algorithmes conduisant à la solution d'un même problème.

Pour qu'il présente un quelconque intérêt, il est une condition à laquelle doit impérativement répondre un algorithme, c'est d'être « raisonnable », c'est-à-dire effectivement réalisable. Ainsi, s'agissant du jeu d'échecs, un algorithme calculant toutes les possibilités de combinaisons sur une centaine de coups fournirait

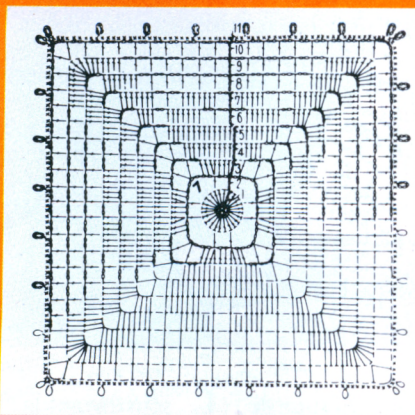
FOURNITURES : 6 g de coton à crocheter N° 40, blanc ; un crochet d'acier N° 1,00.

DIMENSIONS : 11,5 cm de côté.

EXECUTION :

1er tour : 24 br. dans une boucle initiale de 8 m. Remplacer la 1ère br. de tous les t. par 3 m. air ; fermer les t. par 1 m. c. dans la troisième m. air du

début. **2ème tour** : *1 br., 2 m. air, 1 br., 7 m. air*, en sautant chq. fois 2 m du t. précéd. **3ème tour** : rép. 4 fois *8 br. (= 2 sur chaque br. et 2 dans chq. arc.), 2 m. air, dans la m. centrale de l'angle faire 1 br., 5 m. air, 1 br., 2 m. air et faire 2 brides de plus dans l'arc. de l'angle* **4ème-11ème tour** : cont. en suivant les indications du dessin.



traitement, mais stockées en bloc dans une mémoire électronique. Cette propriété, qui permet à la machine, à partir d'instructions contenues dans un programme réalisé par l'homme, de commander elle-même son propre fonctionnement, est la caractéristique fondamentale des ordinateurs.

une solution théorique au problème « comment gagner ? ». Mais aucun joueur ni aucun ordinateur ne peut espérer obtenir un résultat dans un laps de temps raisonnable !

En résumé, un algorithme est une suite ordonnée d'opérations effectivement réalisables (il existe un procédé physique ou mathématique permettant de les exécuter) en un nombre fini (qui peut être très grand) d'étapes.

L'algorithme est généralement exprimé dans un langage proche du langage de tous les jours, mais il emploie certaines abréviations spécifiques (identiques à celles

(on n'applique la ristourne qu'à partir d'un certain prix brut ; en dessous, rien). Il s'agit là d'un problème en apparence simple, mais qu'il devient difficile de traiter « à la main » dès que le nombre de marchandises devient important ou que les taux et les conditions de ristourne varient. Quant à son traitement par un ordinateur, il ne présente de réel intérêt que si la méthode de calcul donne le résultat final (le prix net) quelles que soient les données initiales (le prix brut, le taux et les conditions de la ristourne). L'algorithme doit donc résoudre ce problème de la façon la plus générale possible et permettre

– V : la valeur du prix brut à partir et au-dessus de laquelle on effectue la ristourne.

Le résultat recherché, le prix net du produit sera : PN.

L'algorithme peut s'exprimer ainsi :

- on considère le prix brut (PB) et sa valeur limite (V) ;
- si PB est inférieur à V, pas de ristourne ;
- si PB est supérieur à V, la ristourne se calcule de la façon suivante :

$$R = \frac{T}{100} \times PB ;$$

- le prix net (PN) est obtenu en

Les algorithmes d'Al Khawarizmi



Le mot « algorithme » vient d'Al Khawarizmi, nom d'un mathématicien arabe qui vécut à Bagdad dans la première moitié du IX^e siècle et publia, notamment, un recueil de méthodes de calcul qui influença beaucoup, grâce à ses nombreuses traductions latines, toute la science occidentale du Moyen Age.

Le terme d'algorithme désigna donc d'abord toute méthode de calcul permettant la résolution d'un problème mathématique spécifique, puis sa signification s'élargit considérablement pour s'appliquer maintenant à toute décomposition de la solution d'un problème en une suite finie d'étapes clairement définies et théoriquement applicables à la programmation informatique.

utilisées pour désigner les points de crochet dans l'exemple que nous avons cité). Souvent d'ailleurs, pour plus de simplicité, on le représente sous forme graphique ; on parle alors d'**organigramme**.

Imaginons, par exemple, le problème suivant : comment calculer le prix net (après ristourne) de certains produits sur lesquels on applique, ou pas, une ristourne, selon le montant de leur prix brut

de s'adapter à toute modification de donnée (hausse des prix bruts ou durcissement des conditions de ristourne, par exemple).

Pour simplifier l'écriture de l'algorithme, adoptons quelques abréviations.

Les données du problème seront :

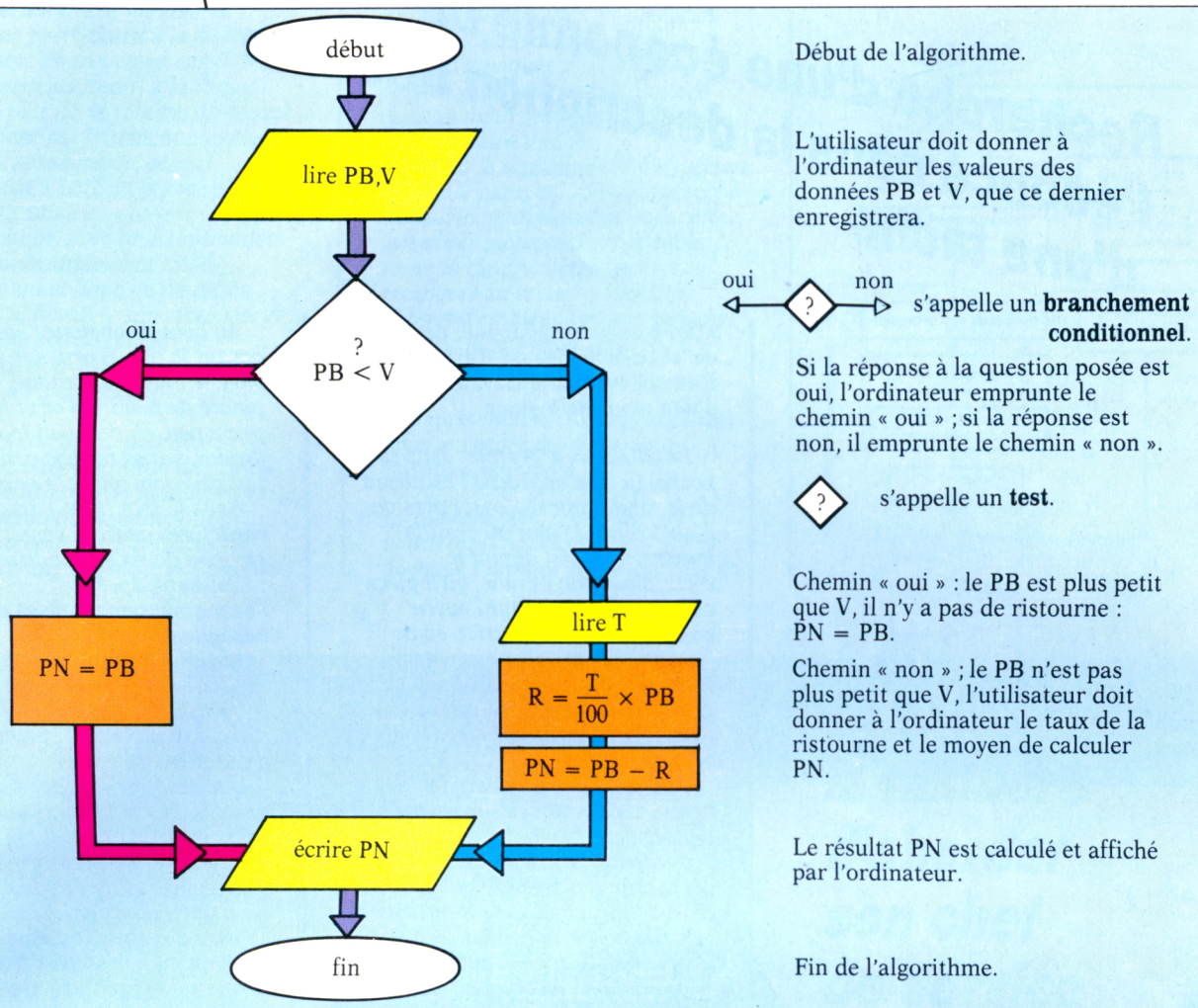
- PB : le prix brut de la marchandise ;
- T : le taux de la ristourne (la ristourne sera R, et le pourcentage de ristourne : T/100) ;

soustrayant la ristourne du prix brut :

$$PN = PB - R.$$

Cet algorithme peut également être représenté sous sa forme graphique (son organigramme).

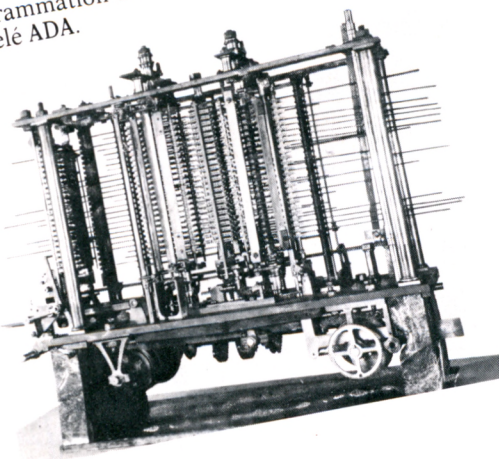
L'organigramme, représentation spatiale d'événements qui se déroulent dans le temps, se lit, du début à la fin, en suivant le sens des flèches ; ces dernières indiquent les différents cheminements possibles.



L'« engin analytique »

La première machine à calculer programmable, l'« engin analytique », fut conçue par Charles Babbage aux environs de 1830. Les programmes étaient alors contenus dans des cartes perforées inspirées de celles des « métiers à tisser Jacquard ». Et c'est Ada Lovelace, fille du poète Byron, amie et collaboratrice de Babbage, qui écrivit la première : « L'engin analytique n'a pas la prétention de donner naissance à quoi que ce soit. Il peut effectuer tout ce que nous savons lui ordonner de faire ». Cette définition de la programmation reste valable de nos jours : quels qu'aient été les progrès technologiques réalisés au XX^e siècle, la machine ne peut toujours rien faire d'elle-même, elle se contente d'obéir aux ordres qu'on lui donne.

Si, de par sa conception, l'engin analytique était tout à fait à même de remplir son rôle, Babbage ne put jamais le faire fonctionner, faute de moyens techniques suffisants. C'est en l'honneur d'Ada Lovelace que l'un des plus récents langages de programmation de haut niveau fut appelé ADA.



Recherche d'une économie de l'énoncé dans la description littéraire d'une tâche

Mieux qu'un programme, cet extrait de «Locus Solus» de Raymond Roussel évoque le travail répétitif d'une machine logique :

Enfouissant le bracelet dans sa poche, Gérard approcha l'escabeau de la table, appuyée, assez près de nous, contre le mur où béait la fenêtre, et s'assit devant le Dictionnaire de l'Erèbe, qu'il plaça convenablement, pour l'ouvrir ensuite à son début strict, en ne faisant, vers sa gauche, pivoter autour de son axe horizontal que le carton de la reliure, prompt à entraîner la garde, exempte de tout gondolement.

Bien à plat, le première feuille ou fausse garde montra son recto entièrement blanc.

Gérard, saisissant ainsi qu'un porte-plume la tige sans fleur entre trois doigts, en trempa légèrement l'un des bouts, encore armé d'une longue épine, dans l'eau presque débordante de la cruche.

Puis, avec la pointe de l'épine, il se mit à écrire sur la feuille blanche du dictionnaire en manifestant toujours une sorte de hâte inquiète.

Au bout de quelques lignes, posant la tige, il prit, sur sa main gauche toujours étendue, une pincée de poudre d'or et la répandit peu à peu, en remuant le pouce et l'index, sur sa fraîche écriture invisible, qui aussitôt se colora.

Sous le mot (ODE), tracé en gros caractères de titre, venait une strophe de six alexandrins.

Laissant, après l'accomplissement de sa courte besogne, retomber sur la réserve de sa main gauche ce qui lui restait de sa pincée de poudre, Gérard retrempa dans la cruche la bonne extrémité de la tige et continua d'écrire avec l'épine.

Une seconde strophe fut bientôt couchée sur la feuille puis saupoudrée d'or.

Le même travail alternatif de griffonnage et de poudrement se poursuivit ainsi, et jusqu'au bas de la page des strophes s'étagèrent.

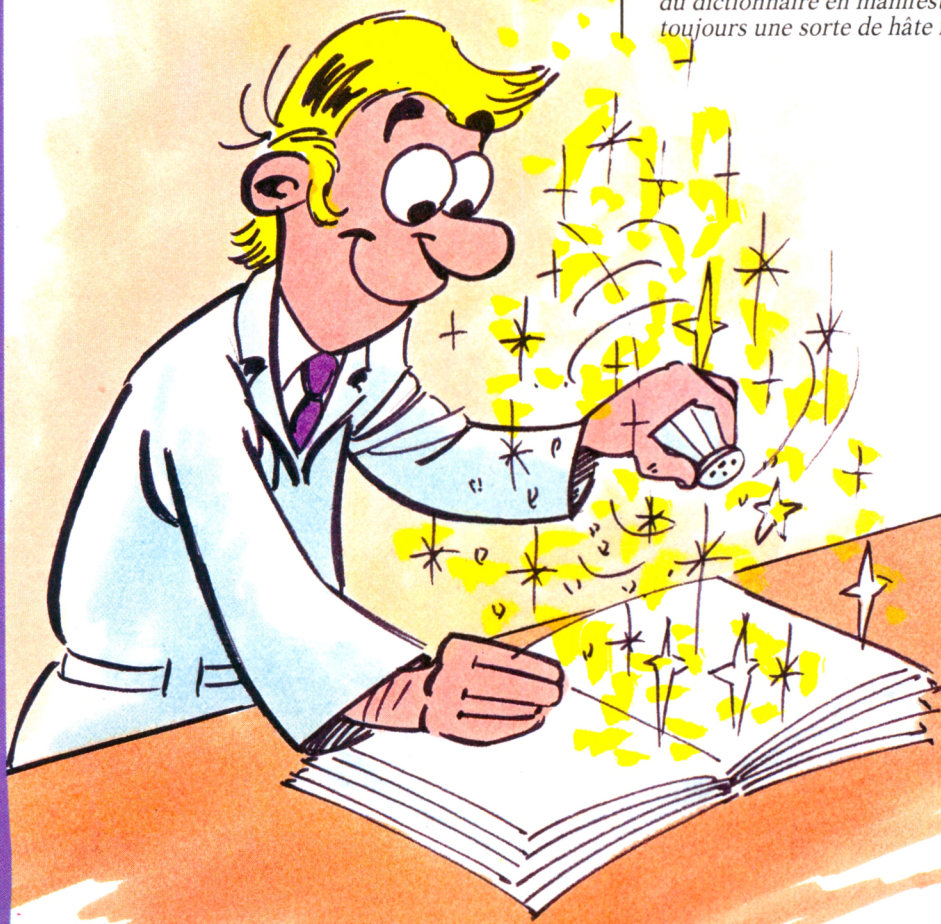
Donnant à l'assèchement le temps de se produire, Gérard souleva momentanément la feuille en la roulant à demi et conduisit de la sorte sur la marge de gauche tous les grains de poudre non captés par l'eau, qui glissèrent sur le tas d'or encore gros de sa main passive prête à les recevoir, quand il eut, en l'agrippant par le haut, dressé le dictionnaire presque verticalement.

Libéré de tous préjudiciables entours déroutants pour l'œil, le fragile texte d'or, jusqu'alors flou, apparut dans son entière pureté.

Gérard laissa, en le retenant, doucement retomber le dictionnaire sur la table et, d'une seule main, mit en pile les quatre in-octavo sous le premier plat de la reliure, pour qu'au lieu d'être en pente il reposât horizontalement sur eux.

Tournée, la fausse garde montra son verso blanc, que Gérard, sans changer de procédés, couvrit de strophes en caractères d'or bientôt secs jusqu'au dernier.

Ici ce fut sur la marge de droite qu'un précautionneux ploiement de la feuille amena les grains d'or restés libres qui, en fine cascade, firent retour à la réserve, grâce à un nouveau redressement momentané du pesant livre.



Au terme d'une manœuvre exécutée par Gérard à la façon d'un manchot, les in-octavo empilés se trouvèrent soutenir, à sa droite, l'autre plat de la reliure, sur lequel s'étaient parfaitement une garde et une fausse garde, celle-ci montrant à côté de la page ultime du dictionnaire — ouvert maintenant, avec tous ses feuillets bien horizontalement tassés, comme un volume qu'on est en train d'achever — son recto vierge qui peu à peu se remplit de strophes nouvelles, une par une écrites à l'eau avec l'épine puis dorées.

Après constat de siccité et routinière récupération de grains d'or, Gérard tourna la fausse garde, sur le verso de laquelle, fidèle jusqu'au bout à ses artifices de scribe étrange, il termina et signa

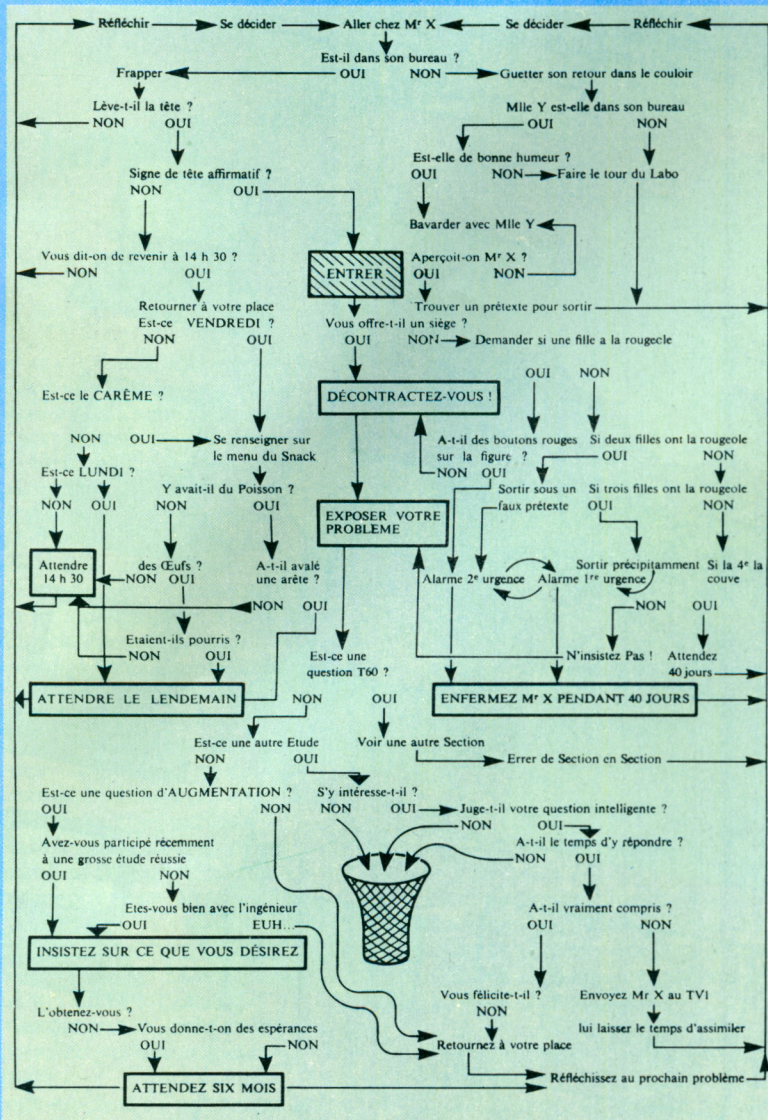
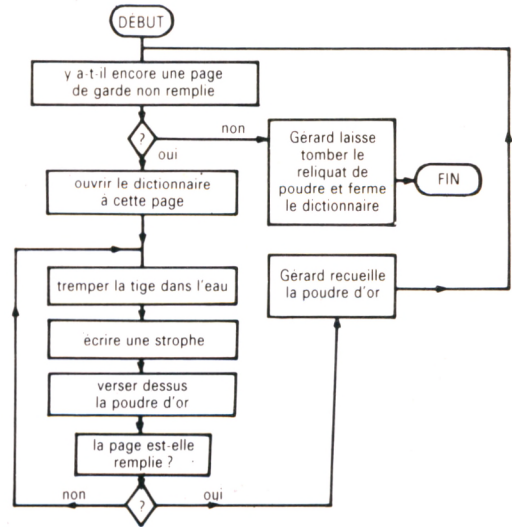
son ode, dont toutes les strophes offraient le même type.

Seuls quelques grains de la poudre précieuse restaient alors dans sa main gauche qu'il secoua pour les faire tomber.

Quand la signature d'or, située au bas de la page, eut elle-même séché complètement, Gérard laissa cette fois choir au hasard sur la table toute la râpure métallique étrangère au texte, en mettant debout d'emblée l'opulent volume — pour le fermer ensuite et le poser.

Sa représentation organigramme inverse l'ordre des tests pour des raisons d'économie.

in *Éléments pour un dialogue avec l'informaticien*, par Jacques Perriault (Mouton, 1971).



L'art et la manière d'aborder son chef de service pour lui demander une augmentation.

Exposé par organigramme

par Georges Perec, qui s'est inspiré d'un organigramme paru dans Bull-informations, in *Enseignement Programmé* (« les Cahiers de l'Institut National pour la Formation des Adultes » (4) décembre 1968 – Dunod/Hachette)

Les langages de programmation

Si un algorithme permet de décomposer un problème en une suite d'opérations simples exécutables par un ordinateur, il n'en est pas pour autant compréhensible tel quel pour ce dernier qui, comme nous le savons, ne comprend que les 0 et les 1 ; une « traduction » est donc nécessaire, et c'est le résultat de cette traduction qu'on appelle **programme**.

L'utilisation d'un langage spécifique pour résoudre certains problèmes n'est pas nouvelle. Déjà avant l'apparition des calculateurs programmables, pour résoudre soit « à la main », soit à l'aide de machines, certains types de problèmes, il fallait les formuler dans un langage différent du langage naturel, le langage algébrique, par exemple :

les formules algébriques
 $R = \frac{T}{100} \times PB$ et $PN = PB - R$
 permettaient déjà, avant que n'apparaissent la machine et la notion d'algorithme, le calcul du prix net, après ristourne, de certains produits.

Malheureusement, ce langage mathématique, que l'on aurait pu imaginer utiliser pour faire fonctionner les ordinateurs, ne permet de créer que les algorithmes appliqués à la résolution de problèmes mathématiques ; or le champ d'application de l'informatique est beaucoup plus vaste que le seul calcul numérique. Il s'est donc avéré nécessaire de trouver d'autres langages.

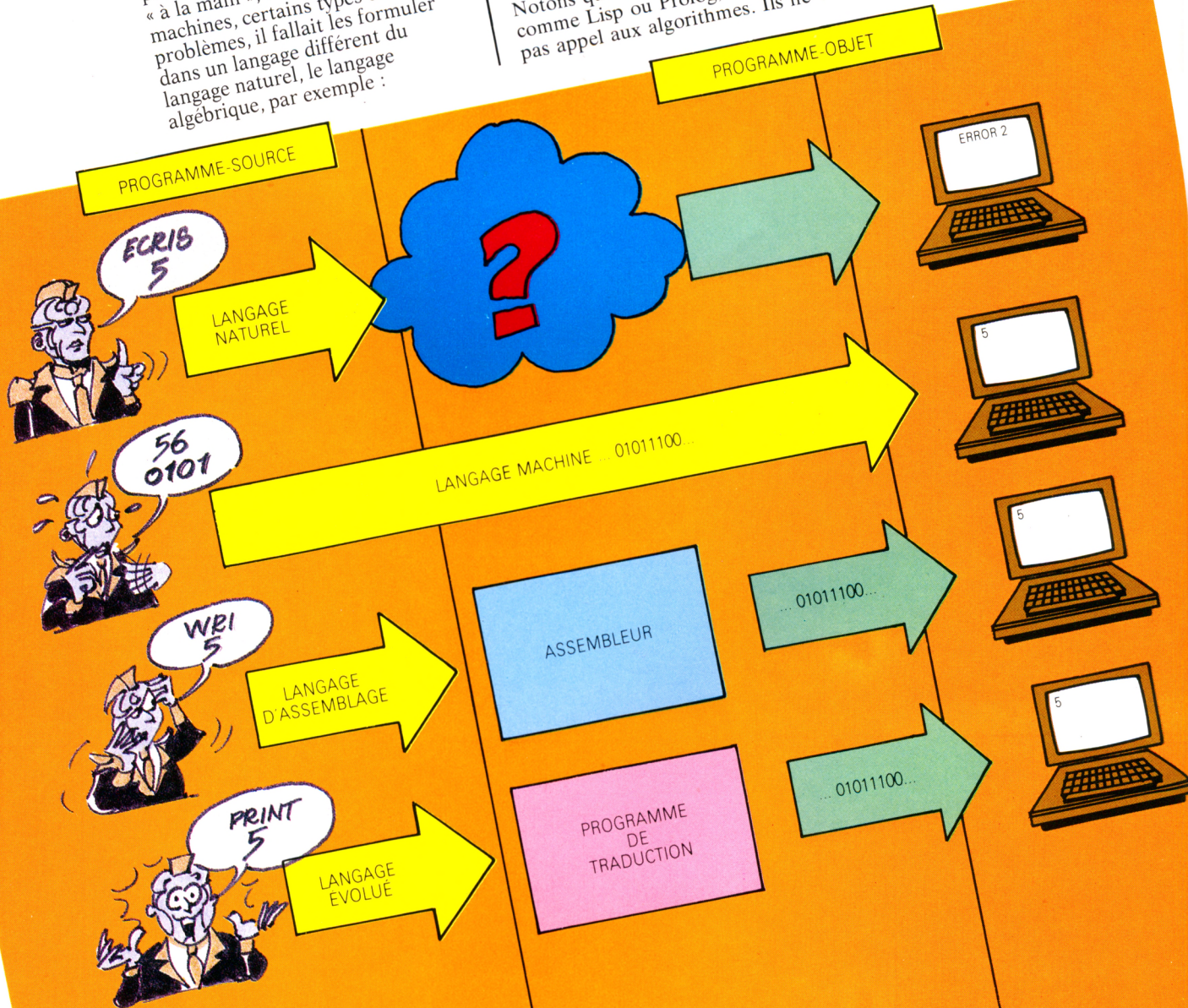
Notons que certains langages, comme Lisp ou Prolog, ne font pas appel aux algorithmes. Ils ne

On a tenté de faire analyser des phrases par les ordinateurs. Dans le cas de phrases non ambiguës, ils ne se débrouillent pas trop mal. Mais, parfois...

Prenons un exemple connu : « Pierre prend la balle et Jean la lance ».

Cette phrase peut être interprétée de deux façons : « Jean lance la balle que Pierre prend », ou bien « Pierre prend la balle » et « Jean prend la lance ». Allez savoir... Il faudrait avoir d'autres informations sur la situation et le seul recours à l'analyse du langage ne suffit pas.

font qu'explorer, en tâtonnant, l'ensemble de faits et de règles qui sont stockés en machines.



Mais, de même que l'ordinateur est dans l'impossibilité de comprendre le langage de l'homme, trop imprécis pour lui, l'homme n'utilise que difficilement le **langage machine** (qu'il a pourtant lui-même imaginé !). Alors, des langages intermédiaires ont dû être trouvés, qui soient à la fois puissants (qui puissent prendre en compte de très nombreuses situations) et précis. Le **langage informatique** parfait n'a pas encore été trouvé, et, à l'heure actuelle, on en compte plus de 400 différents, plus ou moins performants, et généralement spécialisés dans certains domaines d'application : calculs scientifiques, traitement des mots, des images, des sons, gestion des fichiers, etc.

Les premiers programmes furent écrits par l'homme entièrement en langage machine, ce qui donnait une programmation extrêmement lente et fastidieuse. On imagina alors de créer un langage plus proche de l'homme que le binaire, mais toujours suffisamment simple et précis pour pouvoir être traduit automatiquement en langage machine ; c'est-à-dire que cette traduction était devenue suffisamment simple pour faire l'objet d'un programme (que l'on appelle, selon les cas, **assembleur**, **compilateur** ou **interpréteur**) : tout se passe, pour l'utilisateur, comme si la machine traduisait elle-même. Ainsi, l'homme a fait un pas vers la machine (en traduisant

son propre langage en langage de programmation) et la machine a « fait un pas » vers l'homme (elle accepte de traduire elle-même le langage de programmation en suites de 0 et de 1)... Le dialogue homme-machine se simplifie ! Les langages de programmation sont de deux types : les **langages d'assemblage**, plus proches de la machine, et les **langages évolués**, plus proches du langage humain. Le programme conçu en langage de programmation est appelé, par les informaticiens, **programme-source**, sa traduction en langage machine est appelée **programme-objet**.

Le langage machine

Dans sa version la plus simple, la machine ne connaît donc que le langage machine spécifique de son microprocesseur.

Mais, bien qu'il existe autant de langages machines que de microprocesseurs, un certain nombre de principes restent valables dans tous les cas :

- dans tout microprocesseur, il existe au moins un endroit où l'on peut stocker les résultats intermédiaires des opérations - les retenues d'une addition, par exemple. Cet endroit, ou encore ce **registre**, s'appelle l'**accumulateur** ;
- pour l'ordinateur, les données du problème ne sont pas repérables par leur nom, mais par leur **adresse**, c'est-à-dire le numéro (exprimé en langage binaire) de la case-mémoire dans laquelle elles sont rangées ;
- le langage machine comprend un certain nombre d'**instructions** (plusieurs dizaines), chacune représentée par un **code** en binaire et repérable par une adresse mémoire.

Le code explique à l'ordinateur le contenu de l'instruction.

Par exemple :

Code instruction*	Instruction
50	<i>début</i> du programme
51	<i>fin</i> du programme
52	<i>lire</i> une donnée sur une mémoire extérieure et la mettre à l'adresse X en mémoire centrale
53	<i>transférer</i> le contenu de l'adresse X dans l'accumulateur
54	<i>changer</i> le signe du contenu de l'accumulateur
55	<i>ranger</i> à l'adresse X le contenu de l'accumulateur
56	<i>écrire</i> sur papier le contenu de l'adresse X
57	<i>test</i> : effectuer l'instruction d'adresse X si le contenu de l'accumulateur est 0
58	<i>soustraire</i> le nombre dont l'adresse est X à l'accumulateur
59	<i>multiplier</i> le nombre dont l'adresse est X par le contenu de l'accumulateur
60	<i>aller</i> à l'instruction dont l'adresse est X

*Cette série de codes est théorique ; elle ne correspond à aucun microprocesseur particulier. De plus, tous les codes instructions sont, en fait, inscrits dans l'ordinateur sous forme de nombres binaires, mais comme leur formulation est assez compliquée, nous les avons notés ici sous leur forme décimale.

Dans un programme en langage machine il faut donc préciser les codes et adresses des instructions utilisées ainsi que les adresses des données nécessaires.

Par les adresses, l'ordinateur sait quelle instruction ou quelle donnée il doit prendre en compte ; par le code d'une instruction, il sait ce qu'on attend de lui. Une instruction

ou une donnée à laquelle on n'a pas attribué d'adresse n'est pas mémorisée. Quel serait, par exemple, le programme, en langage machine, applicable à notre problème de « calcul de ristourne » ? Le problème comporte trois données : le prix brut PB, le taux de ristourne T et la valeur plafond V ; et doit amener au calcul d'un résultat : le prix net PN.

Il est donc nécessaire de réserver, pour ces quatre valeurs, quatre places (adresses) en mémoire. Le choix des adresses est totalement arbitraire. Il suffit de choisir des places libres, c'est-à-dire non encore occupées par d'autres programmes ou données. Admettons, ici, que ces quatre valeurs puissent être rangées à partir de l'adresse 6000 ; nous choisirons, par exemple :

Adresse	Donnée ou résultat
6000	prix net PN
6001	prix brut PB
6002	taux T
6003	plafond V

Le programme complet (et toujours théorique) en langage machine se présentera alors comme dans l'encadré à gauche.

Adresse* de l'instruction	Code* de l'instruction	Adresse* de la donnée	Instruction
500	50		début du programme
501	52	6001	lecture de PB
502	52	6002	lecture de T
503	52	6003	lecture de V
504	53	6001	mettre PB dans l'accumulateur
505	58	6003	soustraction : $PB - V$
506	57	512	test : branchement conditionnel à 512
507	53	6001	mettre PB dans l'accumulateur
508	59	6002	multiplication donnant R
509	58	6001	soustraction : $R - PB$
510	54		changement de signe : $PN = PB - R$
511	60	514	aller en 514
512	55	6001	mettre PB dans l'accumulateur
513	56	6000	mettre le contenu de l'accumulateur dans la mémoire
514	56	6000	écrire PN
515	51		fin du programme

* Ici encore, les adresses et codes devraient être exprimés en binaire.

instructions (ici prises à partir de 500) est, là encore, arbitraire et ne dépend que des disponibilités. Par exemple, si, dans notre cas, le programme comportait plus de 5 500 lignes, les adresses des instructions et celles des données (commençant à 6000) se chevaucheraient et le programme ne pourrait fonctionner.

Attention !

- Pour simplifier, nous avons écrit tous les nombres (adresses et codes d'instructions et des données) dans le système décimal, mais en réalité, ils sont introduits dans la mémoire de l'ordinateur en système binaire ; ce sont des suites de 0 et de 1.

- Les langages machines varient avec les microprocesseurs ; les constructeurs n'ont, en effet, pas jugé utile d'employer les mêmes

Chaque ligne comprend l'adresse de la case-mémoire où sera rangée l'instruction, le code de l'instruction et, quand c'est nécessaire, l'adresse mémoire de la donnée qui fait l'objet de l'instruction. Le choix des adresses des

codes pour définir les mêmes opérations !

- Il est nécessaire, pour choisir les adresses où seront rangées les données et les instructions, de connaître parfaitement la capacité mémoire de son ordinateur et les zones déjà occupées.

La programmation en langage machine est donc quelque chose de compliqué, qui nécessite un apprentissage et une formation. Elle se trouve ainsi réservée aux professionnels et aux amateurs très éclairés...

Mais, même pour ces professionnels, ce type de programmation reste contraignant et peu rapide. C'est la raison pour laquelle les informaticiens ont créé les deux autres types de langage dont nous avons déjà parlé : les langages d'assemblage (souvent appelés, abusivement d'ailleurs, assembleurs) et les langages évolués.



Les langages d'assemblage

Le langage d'assemblage n'est pas un langage évolué. Il est beaucoup plus proche du langage machine que ces derniers, mais sa caractéristique principale est d'utiliser, pour symboliser les instructions, non pas des codes numériques, mais des formules mnémotechniques, ou **mnémoniques** (utiliser des mnémoniques revient à écrire, au lieu de « je dois me souvenir que j'ai rendez-vous chez le dentiste le 24 novembre à 15 heures », « 24/11-15 h. : dentiste »). Quand un programme comprend une centaine d'instructions différentes, il est, en effet, extrêmement difficile de se souvenir du code en langage machine correspondant à chaque opération. En langage d'assemblage, donc, chaque

Code en langage machine	Symbole en langage d'assemblage
50	GO (Commencer, équivalent du RUN en Basic)
51	HALT (Arrêter)
52	READ (Lire)
53	LDA (pour <i>Load Accumulator/Charger l'Accumulateur</i>)
54	CHS (Changer Signe)
55	STO (pour <i>Store/Ranger</i>)
56	WRI (pour <i>Write/Ecrire</i>)
57	IF (Si)
58	SUB (pour <i>Subtract/Soustraire</i>)
59	MUL (Multiplier)
60	JMP (pour <i>Jump/Sauter</i>)

instruction est représentée, non plus par un code numérique, mais par un groupe de trois ou quatre lettres, généralement la forme abrégée de la fonction à effectuer : dans notre exemple, l'instruction codée 59 en langage machine, qui réalise la multiplication du contenu de l'accumulateur par celui de l'adresse X, est représentée en

langage d'assemblage par MUL. Malheureusement, les mots évoqués par le langage d'assemblage sont généralement des mots anglais. Le tableau de gauche donne, à titre d'exemple, la liste des mnémoniques les plus couramment employées, avec leur correspondance (leurs codes) dans le langage machine théorique de notre exemple précédent.

Le travail du programmeur est donc nettement simplifié. C'est le programme assembleur (compris dans la ROM de l'ordinateur) qui traduit ensuite automatiquement les formules abrégées en nombres binaires.

Un des grands intérêts des langages d'assemblage réside dans le fait que, quel que soit le microprocesseur utilisé, et donc quels que soient les codes des instructions en langage machine, leurs mnémoniques restent toujours les mêmes. Autre avantage de la plupart des langages d'assemblage : les adresses absolues (numériques) où sont rangées les données et les résultats du problème à résoudre y sont également remplacées par des étiquettes, ou adresses symboliques, donc plus faciles à retenir.

Dans l'exemple du calcul de ristourne, nous aurons, par exemple, les correspondances suivantes :

Donnée ou résultat	Code numérique	Etiquette
prix net :	6 000	PRN
prix brut :	6 000	PRB
taux :	6 002	TO
plafond :	6 003	PLAF

Les programmes en langage d'assemblage, tout comme ceux en langage machine, sont exécutables très rapidement (puisque'ils n'utilisent que les instructions élémentaires connues par le microprocesseur) ; malheureusement, ils ne sont pas non plus « portables » sur plusieurs machines. De plus ils restent assez complexes et nécessitent un grand nombre d'opérations élémentaires pour arriver à un résultat même très simple.

Les langages évolués

Après avoir programmé pendant des années, les informaticiens constatèrent que certaines structures apparaissaient, régulièrement, dans presque tous les programmes. Ces structures, communes à la plupart des algorithmes, correspondaient à plusieurs instructions en langage machine, pas forcément contiguës en mémoire. Dans les années 50, les programmeurs mirent donc au point des langages utilisant ces structures de haut niveau : les langages dits évolués, qui facilitèrent considérablement la programmation.

Il existe différents langages évolués qui, proches les uns des autres au départ, se différencièrent progressivement au cours des années, chacun se spécialisant dans un type de besoins spécifiques.

Cependant, aussi notables que soient les distinctions, les langages

évolués possèdent tous des caractéristiques communes, qui les rendent différents du langage machine ou des langages d'assemblage.

Leur première particularité provient de la manière dont sont repérés les emplacements-mémoire. Dans un langage évolué, les contenus des mémoires ne sont plus repérés, comme dans les langages machines, par des adresses numériques (nombres binaires dont le programmeur doit se souvenir), ni, comme dans les langages d'assemblage, par des adresses mnémoniques, mais par des noms : les **variables**. Une variable n'est ni une donnée ni une adresse, mais simplement une étiquette fixée à une case-mémoire contenant une information, laquelle étiquette permet de retrouver automatiquement l'information même sans connaître l'adresse de la case-mémoire qu'elle occupe. On pourrait comparer une variable au texte inscrit sur la jaquette d'une cassette magnétique où ne figurerait que le seul nom de l'auteur de la musique ; la bande magnétique figurerait la case-mémoire et la variable serait le nom de l'auteur. Si on effaçait la bande et si on y enregistrerait un nouveau morceau du même auteur, le nom inscrit sur la jaquette (la variable) ne changerait pas, mais le contenu (l'information) de la bande magnétique (de la case-mémoire) ne serait plus le même. On aurait affecté une nouvelle valeur à la variable.

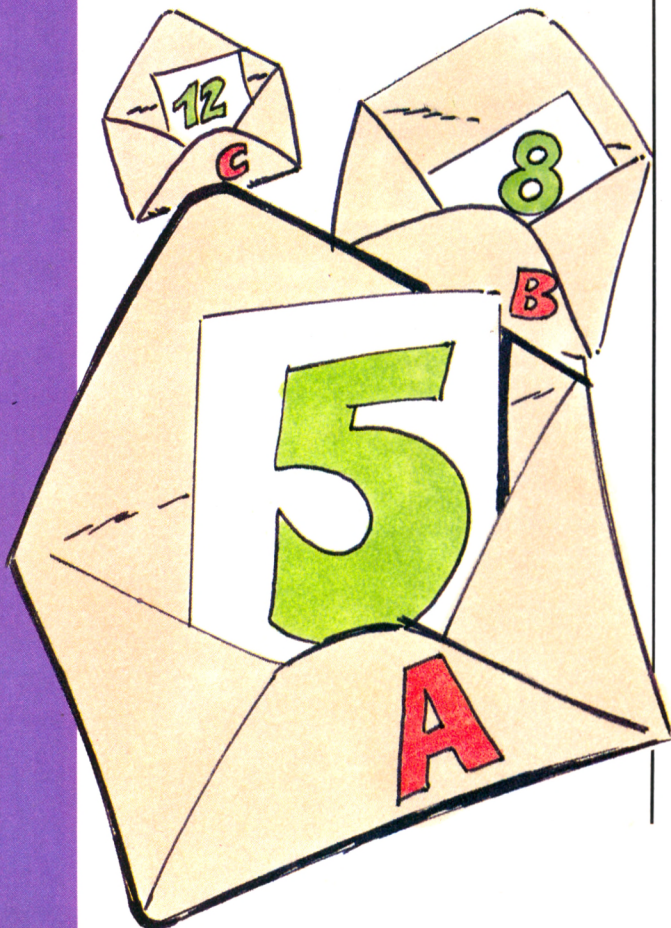
En ce qui concerne les instructions données à l'ordinateur, il est très important de noter qu'elles sont, dans le cas des langages évolués, formulées d'une façon très différente de celle utilisée en langage machine ou en assembleur. Dans les langages évolués, une instruction est un mot clef qui, compris par la machine, déclenche non plus une opération élémentaire, mais une série d'opérations binaires plus ou moins complexes. Pour prendre un exemple, si, dans la vie courante, pour dire à quelqu'un d'aller d'un endroit à un autre en

prenant le bus, on utilisait le langage machine, il faudrait lui donner les explications suivantes : sortir de l'endroit où il se trouve, prendre (si besoin est) l'ascenseur, sortir de l'immeuble, aller à l'arrêt de bus, monter dans le bus, etc. ; en langage évolué, il suffirait de lui dire de partir de l'endroit où il se trouve et de prendre le bus pour atteindre sa destination ; les opérations seraient décrites par blocs. C'est ainsi qu'il existe des mots clefs permettant l'affichage de mots ou de phrases à l'écran, ou ordonnant certains calculs. Ainsi, les langages évolués comprennent, à la simple lecture du signe + situé entre deux nombres ou deux variables, qu'il faut : charger l'accumulateur avec une donnée, aller à l'adresse de la deuxième donnée, additionner le contenu de l'accumulateur avec celui de l'adresse, etc. Le signe + du langage évolué est traduit en langage machine, interprété et exécuté sans qu'il soit nécessaire au programmeur d'écrire à la main toutes les instructions qu'effectuera la machine.

Enfin, tous les langages évolués comprennent des instructions particulières, correspondant à des structures de « contrôle ». Ces mots clefs n'entraînent pas l'exécution d'une instruction unique, mais ils interviennent, dans le programme, sur le déroulement des opérations.

On peut classer ces structures en deux catégories : les **boucles** et les **tests**.

Les boucles permettent de répéter, autant de fois qu'il est nécessaire, l'exécution d'une instruction sans qu'il soit besoin de la réécrire ce même nombre de fois. C'est comme si, pour expliquer comment faire un puzzle, au lieu de dire, autant de fois qu'il y a de pièces : « Prenez une pièce du puzzle, cherchez où elle se place, placez-la » – imaginez que le puzzle ait 1 000 pièces : vous devriez dire ou écrire cette phrase 1 000 fois –, on disait simplement : « Exécutez l'instruction qui va suivre mille fois en prenant à chaque fois une pièce différente ». Les tests, eux, permettent à



l'ordinateur, dans le cas où une option multiple se présente à lui, de savoir ce qu'il doit faire dans chacun des cas. Dans le cas de notre calcul de ristourne, l'option était : PB est-il ou non inférieur à V ? De même, on peut dire que la formulation « S'il fait beau demain, je vais me promener : s'il fait mauvais, je reste chez moi » est un test. La variable testée, c'est le temps : si le temps est beau, alors on exécute l'instruction « aller se

langage évolué sont pratiquement indépendants de la machine sur laquelle ils doivent être exécutés (il existe des différences de formulation ou de syntaxe mais elles sont bien moindres que dans le cas des langages machines) ; seuls les programmes de traduction (interpréteurs ou compilateurs) en dépendent.

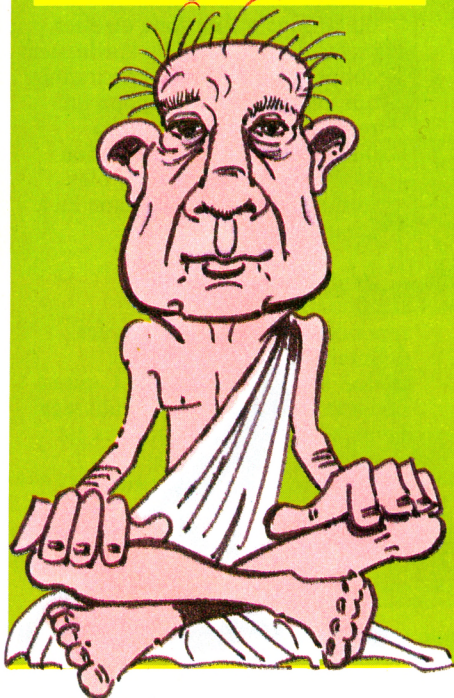
De nombreux langages (plus de 400) existent dans le monde ; ils ont généralement été créés pour

LES LANGAGES ÉVOLUÉS LES PLUS UTILISÉS				
NOM	Année de création	Domaine d'utilisation principale	Compilé ou interprété	Remarques
FORTRAN	1954	Calcul scientifique	compilé	Immense bibliothèque
COBOL	1959	Gestion, commerce	compilé	Plusieurs versions
ALGOL	1960	Scientifique	compilé	Fils de Fortran
LISP	1960	Intelligence artificielle	compilé	Très souple
APL	1962	Maths – Gestion	interprété	Clavier spécial
BASIC	1965	Apprentissage de la programmation	compilé ou interprété	Très répandu et très simple
PASCAL	1970	Enseign. – Systèmes	compilé	Programmation structurée
FORTH	1970	Robotique – Gestion	compilé	Adapté au temps réel
LOGO	1971	Éducation	ligne/ligne interprété	Bon graphique – récurtivité
LSE	1972	Enseignement en français	interprété	Langage en français

promener », dans le cas contraire, on exécute l'instruction « rester chez soi ».
Les programmes écrits dans un

répondre aux besoins précis d'un groupe d'utilisateurs. Cependant, seule une dizaine d'entre eux sont véritablement répandus.

Langage...



Tel qu'il est, le Biglotron se compose, ou plus exactement se décompose en trois circuits principaux dont deux secondaires et un complémentaire et dont voici, par ordre d'entrée en action, le processus fonctionnel de sa posologie fondamentale. Tout d'abord, dans le premier circuit, on distingue le Clebstroïde qui, isolé du P.X. de l'intrudmon par une armature en fignabulose ignifuge, agit, par capillarité médullaire, sur le fiduseur de télédeconométrie différée, lequel, en vertu du phénomène d'osmose ondulatoire érigé en principe par le célèbre physicien Jean-Marie Keszke-Lavoulvoule, catalyse en quelque sorte, le Schpoutzmühl de dérivation qui, par voie de conséquence, se trouve entraîné par le brigmuch michazérospiroïdal en direction de la zone d'influence de la boustife de relevailles dont le tuyau d'argougnaphonie spéculaire libère un certain volume de

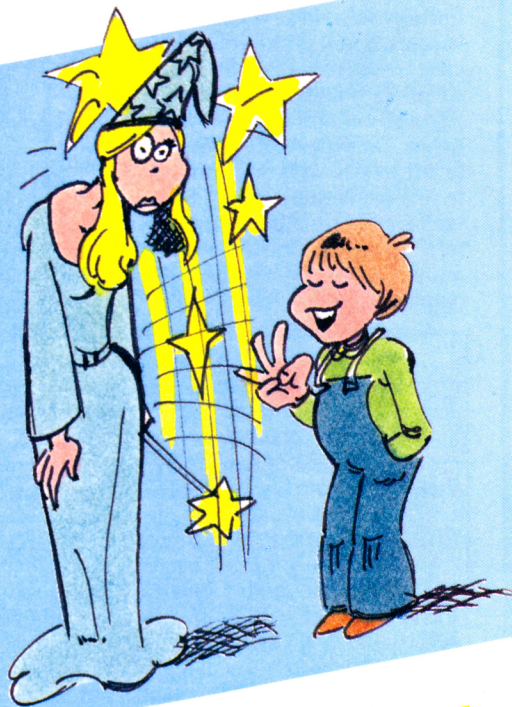
Laplaxmol, lequel, comme chacun le sait, n'est autre qu'un combiné de smitmphre à l'état pur et de trouduchium filtrant sulsiforé. Dans le deuxième circuit, le même mouvement s'opère, mais en sens inverse ; il est donc inutile d'en parler, même à voix basse, d'autant que c'est dans son troisième circuit que se trouvent étroitement conjugués les éléments majeurs de vérité parmi lesquels le schpatzmocl rotatif à crémaillère alternative dont le rôle de générateur permanent d'énergie est prépondérant puisque par le simple truchement de son induit de giclée, il polypophéripotéise littéralement le filtre à moléculbutant, lequel, en dernière analyse, détermine l'angle orbiculaire et synochoïdal du foutaisiogognomètre à spirale introputréfactionnelle.

in *Essais, maximes et conférences*, par Pierre Dac (Le Cherche-Midi éditeur, 1978).

Récurtivité

Une chose est réursive quand elle se définit en s'utilisant elle-même comme référence. Une image peut être réursive. Par exemple, une télévision montrant sa propre image allumée qui montre donc sa propre image allumée etc., est une image réursive.

Un petit garçon fut rencontrer une fée ; celle-ci lui dit :
« Tu as le droit de faire trois vœux. »
Le petit garçon lui répond :
« Je voudrais du chocolat.
Je voudrais du chewing-gum.
Et...
Je voudrais encore faire trois vœux ! »



Interpréteurs et compilateurs

La traduction d'un programme du langage de programmation en langage machine, c'est-à-dire le passage du programme-source au programme-objet, peut se faire de deux façons ; il existe donc deux types différents de programmes de traduction : les interpréteurs et les compilateurs.

Les interpréteurs

C'est la solution la plus simple. Les lignes du programme-source sont traduites, une par une, en langage machine et, à chaque ligne du programme-source, correspond une ligne du programme-objet. Ce qui fait qu'à tout instant une exécution

partielle du programme (les lignes déjà traduites) est possible ; les langages interprétés sont donc très commodes pour la mise au point des programmes : l'utilisateur peut en vérifier le fonctionnement « morceau par morceau ». Cependant, cette traduction simultanée n'est pas mémorisée par l'ordinateur et doit, par conséquent, être recommencée à chaque exécution du programme. Il s'ensuit un « temps mort » entre l'écriture et l'exécution réelle du programme.

Les compilateurs

Pour pallier cet inconvénient, on a imaginé de faire la traduction en

bloc, une fois le programme entièrement terminé ; le programme-source en langage évolué est alors remplacé, en bloc, par le programme-objet en langage machine. L'exécution est, dans ce cas, immédiate ; un programme compilé est donc bien plus rapide qu'un programme interprété. Pour éviter les erreurs (puisque aucune vérification partielle n'est possible), les compilateurs procèdent, avant la traduction proprement dite, à une analyse lexicographique pour reconnaître les mots clefs du langage utilisé, et à une analyse syntaxique pour vérifier que les instructions sont correctement écrites. Tous les mots et la syntaxe du langage utilisé sont, en effet, inclus dans la ROM de l'ordinateur qui peut vérifier, avant de traduire, que chacun des mots qu'on lui propose figure bien dans sa mémoire et que la formulation de chaque instruction est bien conforme à la syntaxe qu'on lui a communiquée.

LE LANGAGE LOGO

Conçu entre 1968 et 1971 par une équipe du MIT (*Massachusetts Institute of Technology*), sous la direction de Seymour Papert, ce langage de programmation était, à l'origine, destiné à étudier le comportement des enfants face à la résolution de problèmes liés à la représentation géométrique dans le plan. Il n'en est pas, pour autant, réservé uniquement aux enfants.

LE LANGAGE BASIC

Créé en 1965 pour répondre aux besoins des étudiants des universités scientifiques américaines, le Basic est un langage principalement conçu à l'usage des débutants. Son nom signifie d'ailleurs « simple, primordial » en anglais, et est formé des initiales de *Beginners All purposes Symbolic Instruction Code* – littéralement « langage symbolique (pour

répondre à) tous les besoins des débutants ». C'est un langage évolué, simple mais puissant, qui ne nécessite pas un long apprentissage. Le nombre de ses instructions est limité (une cinquantaine) et sa syntaxe est extrêmement simple.

Un peu de Logo...

Les langages de listes

A côté des langages qui, comme le Basic, traitent de formules de type mathématique, il en existe d'autres, tel le Logo, qui, totalement différents, manipulent des **listes** de mots, chiffres et symboles, et permettent à l'utilisateur de définir lui-même les opérations qu'il désire voir intervenir dans son programme.

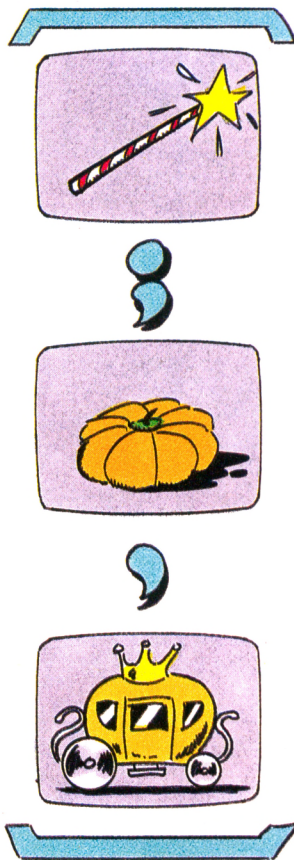
Une liste est une suite composée des éléments suivants :

- deux parenthèses (une pour ouvrir et l'autre pour fermer) ;
- des termes, séparés par des espaces, qui sont, soit une chaîne de caractères, un « atome », soit une liste, un « mot ».

On peut ainsi décrire l'un des faits importants de l'histoire de *Cendrillon* :

[baguette magique citrouille carrosse]

peut, par convention, vouloir dire que l'opération « baguette magique », appliquée à « citrouille », donne « carrosse ».



Extensibilité

ROUGE

De même (SOMME 2 3 = 5) exprime le fait que l'opération SOMME, appliquée à 2 et 3 donne 5. Ce type de description permet donc, non seulement de traiter des quantités numériques, mais aussi de simuler des opérations de raisonnement.

Les langages possédant ces caractéristiques s'appellent **langages de listes** ; et le Logo en est un exemple. Il est spécialisé dans le traitement des listes ; de plus, il est **récuratif** et **extensible** ; enfin, il est particulièrement bien adapté à tout ce qui est basé sur des essais et leurs vérifications, puisque ses messages d'erreur se veulent plus explicites que ceux du Basic : ils indiquent clairement à l'utilisateur la nature de ses erreurs.

Logo est extensible

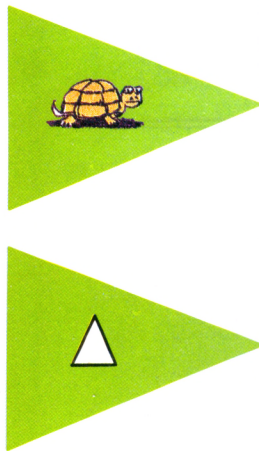
La plupart des langages de programmation possèdent un nombre limité d'instructions, ce nombre ne pouvant être augmenté. Il en est certains, pourtant, qui, à partir d'un petit ensemble de base, permettent à l'utilisateur de se construire son propre vocabulaire, lequel vient enrichir la base de départ. On parle alors de langages extensibles.

En Logo, par exemple, les programmes – qu'on appelle des **procédures** – peuvent voir les mots de base dont ils sont composés (**les primitives**) remplacés par des noms de sous-programmes qui jouent le même rôle qu'eux.

Ainsi, si on décide de créer un sous-programme définissant l'écriture en rouge, et de lui donner le nom « ROUGE », il suffira ensuite d'écrire ROUGE dans la procédure définitive pour que le sous-programme soit effectué. Créons-le donc (c'est également une procédure). Toute définition de procédure en Logo devant être introduite par la primitive POUR et se terminer par le mot FIN ;

```
tapons :  
? pour ROUGE  
> VE  
> FCFG 1  
> FCB 1  
> FIN
```

? est le symbole affiché par le système pour indiquer qu'une commande peut être tapée ;



VE efface tout ce qui est sur l'écran ;

↵ ou **ENTRÉE** a le même rôle qu'en Basic : faire prendre la ligne en compte par l'ordinateur ;

FCFG Fixe la Couleur du Fond Graphique ; ici, 1 donc rouge ;

FCB Fixe la Couleur du Bord, également rouge ; l'ensemble de la lettre sera donc rouge ;

> est affiché par le système. A partir du moment où ROUGE est ainsi défini, il devient une commande au même titre que VE ou FCFG, et peut être tapé au clavier comme une primitive :

? ROUGE entraîne l'exécution des trois instructions définissant ROUGE. ROUGE fait désormais partie de notre vocabulaire Logo et peut s'insérer dans toute autre procédure.

Le vocabulaire Logo, assez réduit au départ, se voit ainsi progressivement enrichi ; la seule limite à son extension étant la capacité-mémoire du système.

Logo agit et réagit à son environnement

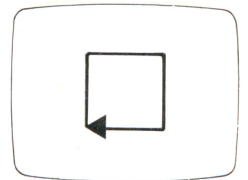
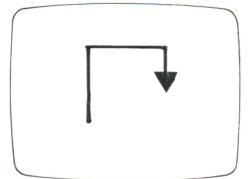
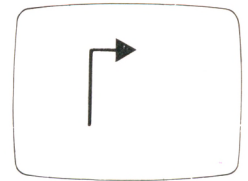
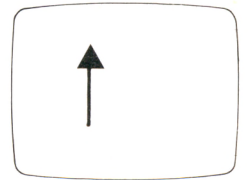
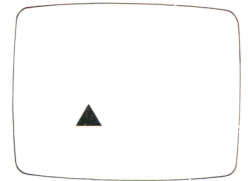
Avec Logo, on peut programmer le parcours sur le sol d'un robot, sorte de soupière renversée montée sur roulettes qu'on appelle **tortue de plancher** et qui peut avancer, tourner, etc., et même laisser sur le sol une trace au crayon – donc dessiner. Il existe également une représentation graphique de cette tortue : elle est symbolisée, à l'écran, par un petit triangle isocèle (ce qui permet de connaître le sens de sa marche) destiné à tracer, lui aussi, des dessins (sur l'écran).

Dans les versions implantées sur le TO 7 et le MO5, on peut s'adresser à l'ordinateur en Logo par le biais de tous les périphériques actuellement disponibles, y compris les manettes de jeux, le haut-parleur, le crayon optique, les couleurs et l'écran graphique : des commandes spécifiques peuvent être envoyées à ces périphériques par l'ordinateur, et, réciproquement, le résultat d'opérations en provenance de l'extérieur peut être récupéré par eux. C'est ainsi qu'il existe, par exemple, une opération permettant d'obtenir les coordonnées de la pointe du crayon optique ; et que la tortue peut renvoyer son cap et sa position ou obéir à des commandes. On peut, par exemple, lui ordonner de dessiner un carré ; ce qui va nous permettre de définir la procédure CARRE :



? POUR CARRE

AV 50
TD 90
AV 50
TD 90
AV 50
TD 90
AV 50
TD 90
FIN

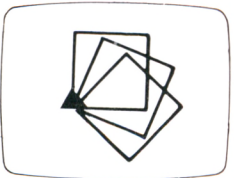
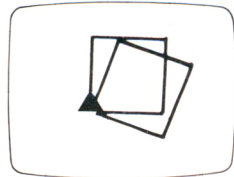
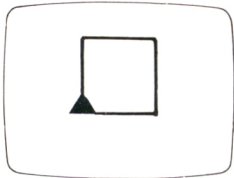


Au départ la tortue a le nez pointé vers le haut ; ensuite :
AVn lui ordonne d'AVancer de n pas, ici de 50, et
TDn lui dit de Tourner à Droite selon un angle de n°, ici 90° ; elle se déplacera donc ainsi, pour former un carré, Notons que, en Logo, contrairement à ce qui se passe en Basic, **les primitives doivent toujours être séparées par des espaces**. Aussi ne pourrait-on pas remplacer AV 50 par AV50, l'ordinateur ne comprendrait pas. Une fois le CARRE défini, on peut l'introduire dans la définition d'une autre procédure :

? POUR FIGURE

CARRE
TD 20
CARRE
TD 20
CARRE
FIN

produira, puisque TD 20 veut dire « tourner à droite de 20° », le dessin suivant : le carré initial plus deux carrés obtenus par deux rotations successives de 20°.



On pourrait, en continuant la procédure, obtenir, au total, 18 carrés successifs ; après quoi le carré reviendrait à sa position initiale. Le nombre total de carrés dépend, bien sûr, de l'angle choisi. Ecrire le programme entier serait long et ennuyeux – il faudrait répéter 18 fois CARRE TD 20. Mais peut-être existe-t-il d'autres solutions !

Logo manipule des mots et des listes

- Un **mot** est une suite de caractères encadrés par des espaces, que l'on écrit précédé d'une seule « ».
- Une suite de mots encadrée par des crochets [] est une **phrase**, ou une **liste simple**.
- Une **liste** est une suite de mots et de phrases. Elle est également encadrée de crochets.

Par exemple :

[hibou, bijou, caillou, pou]

et

[je mange une cerise]

sont toutes deux des listes, mais seule la dernière est une phrase, parce qu'elle seule a une signification dans le langage courant.

- Un **nombre** est un mot dont tous les caractères sont des chiffres.

Des listes de listes

Il existe également des listes de listes ; et, comme il existe des listes d'un seul mot et des mots vides, il existe des listes vides et même des listes de listes vides...

Par exemple : [[] []]

Mais attention : pour qu'une liste de listes soit « licite », il faut qu'elle comporte autant de crochets gauches que de crochets droits !

Et bien, les primitives et les procédures peuvent utiliser tous ces objets Logo comme données, et les manipuler. Ainsi si on appelle l'instruction Logo suivante :

? ÉCRIS PH [QU'ELLE EST]
« JOLIE

ÉCRIS appelle le résultat de l'instruction PH ; c'est donc une instruction différente du POUR qui définit une procédure. On voit s'afficher à l'écran :

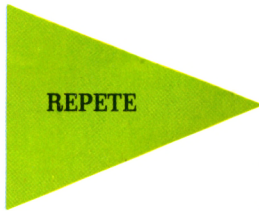
QU'ELLE EST JOLIE

Les crochets se tapent de la façon suivante :

CNT - A pour [
CNT - Z pour]

La primitive **PH**, est une opération qui, à partir de deux arguments qui la suivent (une liste et un mot), fabrique une phrase (ou une liste). Ce résultat est ensuite pris en charge, comme argument, par la commande **ÉCRIS** qui a pour objet de l'afficher à l'écran et de passer à la ligne suivante.

```
? ÉCRIS SOMME 45  
9  
?
```

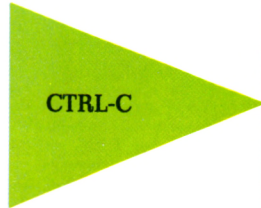
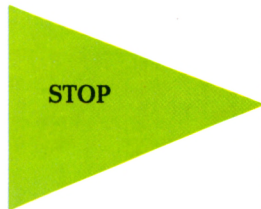
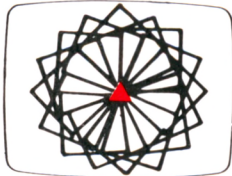
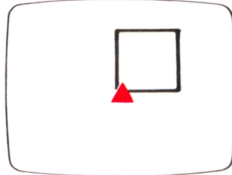


SOMME est une opération qui renvoie le 9, lequel devient une donnée pour la commande ECRIS : tout se passe alors comme si la ligne d'instruction devenait ECRIS 9 : le résultat (9) s'affiche à l'écran avec passage à la ligne.

Reprenons enfin notre exemple FIGURE ; on pourrait l'écrire, avec l'instruction REPETE :

```
? POUR FIGURE
REPETE 18 [CARRE TD 20]
FIN
```

ce qui donnerait :



Ainsi, notre programme FIGURE pourrait s'écrire d'une troisième façon :

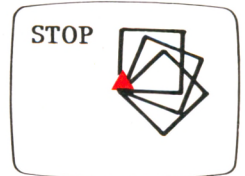
```
? POUR FIGURE
CARRE
TD 20
FIGURE
FIN
```

La récursivité de cette procédure est terminale, car c'est la dernière elle-même qui renvoie à la procédure elle-même. Alors, ici, contrairement au cas précédent (procédure avec **REPETE**) le programme tournera indéfiniment, redessinant sans cesse sur les 18 premiers carrés, à moins qu'on ne lui donne l'ordre de s'arrêter. Une boucle s'est en effet établie. On peut la stopper temporairement en appuyant sur **STOP** ; elle repartira alors dès qu'on tapera une touche quelconque. Et on peut la rompre définitivement en appuyant simultanément sur **CTRL** et **C**.



Logo est récursif

Logo est récursif parce que certaines de ses procédures le sont. C'est-à-dire qu'elles contiennent des procédures qui portent le même nom qu'elles.



Conclusion

Nous avons vu par quelques exemples que Logo était vraiment un langage d'un genre particulier, que sa puissance provenait essentiellement de sa capacité d'extension et de sa récursivité. Il ne faut pas non plus oublier que toutes ces possibilités, ainsi que celles qui permettent le traitement des listes, s'ajoutent à toutes celles qu'on trouve généralement dans les langages classiques.

Utiliser Logo c'est concevoir et analyser les problèmes de manière fondamentalement renouvelée, en utilisant les techniques modernes de la programmation structurée, et en mettant le pied dans le domaine de l'intelligence artificielle. Pour les éducateurs et les enfants, c'est apprendre à connaître la programmation d'aujourd'hui et comprendre la programmation d'hier et de demain.

Un peu de Basic...

En plus de ses instructions simples, le Basic comporte un certain nombre de structures de contrôle, qui font sa richesse fonctionnelle.

La première d'entre elles est l'**instruction de branchement**

GOTO qui, comme nous l'avons vu, permet de rompre le déroulement normal du programme par un ordre signifiant « aller à tel numéro de ligne », lequel numéro peut être indifféremment en aval (le programme saute quelques lignes) ou en amont (le programme retourne à une ligne qu'il a déjà exécutée). Introduisons, par exemple, dans l'ordinateur, le programme suivant :

```
10 CLS
20 LOCATE 15, 10
30 PRINT « BONJOUR »
40 LOCATE 14, 12
50 PRINT « SIGNE TIFY »
60 SCREEN 1, 4, 6
70 SCREEN 2, 5, 7
80 SCREEN 3, 6, 8
90 GOTO 50
```

GOTO

BONJOUR
SIGNE TIFY

BONJOUR
SIGNE TIFY

BONJOUR
SIGNE TIFY

Une fois ce programme introduit, si on donne à l'ordinateur l'ordre RUN, l'écran est nettoyé, puis le texte entre guillemets s'inscrit, sur deux lignes, en milieu d'écran ; après quoi les lignes 60, 70, 80 sont exécutées successivement, ce qui

CNT-C
RAZ

FOR...
NEXT

provoque une série de changements dans les couleurs du texte, de l'écran et du pourtour. Quand l'ordinateur arrive ensuite en 90, l'ordre lui est donné de retourner en 60, c'est-à-dire de réafficher les couleurs de 60, puis de 70, puis de 80, puis de nouveau celles de 60, etc.

Une boucle est établie, qui provoquerait indéfiniment cette alternance de couleurs si on n'y mettait fin en appuyant d'abord, simultanément sur les touches **CNT** et **C** (la boucle s'arrête alors et le numéro de la dernière ligne exécutée par l'ordinateur est affichée), puis sur la touche **RAZ** qui efface tout.

Il est également possible de réaliser des programmes qui ne bouclent qu'un nombre fini de fois ; ceci, grâce à l'instruction **FOR...NEXT**. Cette instruction comporte deux parties qui encadrent la portion de programme à répéter. Par exemple :

```
10 FOR I=1 TO 5
20 PRINT « VIVE TIFY »
30 NEXT I
```

VIVE TIFY
VIVE TIFY
VIVE TIFY
VIVE TIFY
VIVE TIFY
OK
-

La ligne 20 représente la portion de programme que l'on désire boucler. On précise donc en ligne 10 que l'opération va se répéter 5 fois (littéralement pour I allant de 1 à 5). Le choix de la lettre I est purement arbitraire. En ligne 30, on indique la fin de la boucle par un ordre qui signifie « au suivant ».

L'instruction NEXT doit être évidemment suivie de la même lettre que FOR.

Concrètement, la variable I prend, au cours du déroulement du programme, toutes les valeurs entières comprises entre 1 et 5. Le programme suivant :

```
10 FOR I=1 TO 5
20 PRINT I
30 NEXT I
```

affichera à l'écran les cinq premiers nombres entiers.

```
1
2
3
4
5
```

En l'améliorant un peu, il est possible d'afficher les résultats d'une table de multiplication, par exemple celle de 9 :

```
10 FOR I=1 TO 10
20 PRINT 9*I
30 NEXT I
```

```
18
27
36
45
54
63
72
81
90
```

La ligne 20 affiche le résultat de la multiplication de 9 par I, I prenant toutes les valeurs comprises entre 1 et 10.

Le Basic comporte une autre structure de contrôle très importante, le **test conditionnel** **IF... THEN... ELSE** (littéralement **SI... ALORS... SINON**). Comme nous l'avons déjà vu, cette instruction indique à l'ordinateur deux possibilités de programme, en fonction, soit d'une valeur entrée par l'utilisateur, soit d'un résultat calculé par l'ordinateur.

Le petit programme de jeu suivant compare, par exemple, la valeur entrée au clavier par l'utilisateur (à l'aide de l'instruction INPUT) avec le nombre 50. Le joueur a gagné s'il tape 50, perdu dans les autres cas :

```
10 CLS
20 PRINT « ENTREZ UN
   NOMBRE
30 INPUT A
40 IF A=50 THEN PRINT
   « GAGNE » ELSE « PERDU »
```

La ligne 40 teste, c'est-à-dire compare avec 50 (IF A=50) la valeur attribuée à A par le joueur, puis il en déduit deux déroulements possibles : si le nombre joué est égal à 50, alors le programme

affiche GAGNE (THEN PRINT « GAGNE »)

```
ENTREZ UN NOMBRE
?
```

```
ENTREZ UN NOMBRE
? 50
```

```
ENTREZ UN NOMBRE
? 50
GAGNE
OK
-
```

s'il est différent de 50 (ELSE PRINT « PERDU »), le programme suit un autre chemin et affiche PERDU.
ALORS (instructions)
SI (condition)
SINON (autres instructions)

```
ENTREZ UN NOMBRE
?
```

```
ENTREZ UN NOMBRE
? 39
```

```
ENTREZ UN NOMBRE
? 39
PERDU
OK
-
```

IF... THEN...
ELSE

IF...
THEN

L'instruction **IF... THEN** peut également être utilisée seule (sans ELSE).

Prenons un exemple ; tapons le programme :

```
5 CLS
10 PRINT « QUI EST LA ? »
20 INPUT A $
30 IF A $ = « TIFY » THEN PRINT
   « C'EST TIFY »
```

Que provoque RUN à l'écran ?

```
QUI EST LA ?
?
```

L'ordinateur a obéi à l'instruction 10 en inscrivant « QUI EST LA ? » puis à 20 en demandant à l'utilisateur de donner une valeur à la variable alphanumérique A. Si nous tapons alors TIFY, l'ordinateur affichera :

```
QUI EST LA ?
? TIFY
C'EST TIFY
OK
-
```

Si nous avons tapé quoi que ce soit d'autre que TIFY, il n'aurait rien affiché de plus ; on en serait resté par exemple à :

```
QUI EST LA ?
? INTOX
```

Compliquons un peu les choses, afin d'obtenir un affichage dans tous les cas :

```
5 CLS
10 PRINT « QUI EST LA ? »
20 INPUT A $
30 IF A $ = « TIFY » THEN PRINT
   « C'EST TIFY » : GOTO 50
40 PRINT « CE N'EST PAS TIFY »
50 PRINT « SIGNÉ : VOTRE
   ORDINATEUR PRÉFÉRÉ »
```

Deux affichages seront alors possibles en fonction de la valeur affectée à A :

```
QUI EST LA ?
? TIFY
C'EST TIFY
SIGNÉ : VOTRE ORDINATEUR
PRÉFÉRÉ
```

(si A\$ = « TIFY », le GOTO de 30 envoie en 50)
ou

```
QUI EST LA ?
? INTOX
CE N'EST PAS TIFY
SIGNÉ : VOTRE ORDINATEUR
PRÉFÉRÉ
```

Comme la valeur donnée à A ne répondait pas à la condition imposée par 30, l'ordinateur a sauté directement à l'instruction 40 puis à 50.

Si nous avions écrit tout simplement :

```
5 CLS
10 PRINT « QUI EST LA »
20 INPUT A $
30 IF A $ = « TIFY » THEN PRINT
   « C'EST TIFY »
40 PRINT « CE N'EST PAS TIFY »
50 PRINT « SIGNÉ : VOTRE
   ORDINATEUR PRÉFÉRÉ »
```

Pour la valeur « TIFY », nous aurions obtenu quelque chose de surprenant :

```
QUI EST LA ?
? TIFY
C'EST TIFY
CE N'EST PAS TIFY
SIGNÉ : VOTRE ORDINATEUR
PRÉFÉRÉ
```

Après 30, l'ordinateur serait tout naturellement allé à 40, puis à 50. Nous pourrions maintenant modifier un peu notre programme et l'écrire de façon à éviter d'être

obligé de faire signer l'ordinateur :
 5 CLS
 10 PRINT « QUI EST LA ? »
 20 INPUT A\$
 30 IF A\$ = « TIFY » THEN PRINT
 « C'EST TIFY » : END
 40 PRINT « CE N'EST PAS TIFY »
 Deux affichages auraient alors été
 possibles :

```

QUI EST LA ?
? TIFY
C'EST TIFY
  
```

ou

```

QUI EST LA ?
? INTOX
CE N'EST PAS TIFY
  
```

L'instruction **END** donne, en effet, l'ordre de ne pas prendre en compte les lignes d'instruction qui viennent après elle. Si A = TIFY, tout se passe comme si la ligne 40 n'existait pas.

Une autre simplification est possible au niveau des deux premières lignes d'instruction. Il existe, en effet, un INPUT condensé. Ce qui veut dire que :
 10 PRINT « QUI EST LA ? »
 20 INPUT A\$
 peut se résumer de la façon suivante :
 10 INPUT « QUI EST LA ? » ; A\$
 L'ordinateur affiche alors :

```

QUI EST LA ??
puis, si on répond TIFY :
QUI EST LA ?? TIFY
C'EST TIFY
  
```

Il n'est plus indispensable de mettre le point d'interrogation dans la chaîne alphanumérique :
 10 INPUT « QUI EST LA » ; A\$
 suffit, puisque l'affichage est alors :

```

QUI EST LA ? TIFY
C'EST TIFY
  
```

Le ; est une instruction qui permet simplement d'écrire, sur une même ligne, deux mots séparés par un espace (plus petit que celui créé par la virgule).

Voici un petit programme qui nous a été suggéré par Zoé – sans, bien sûr, aucune intention particulière !
 5 CLS
 10 INPUT « QUEL EST VOTRE NOM » ; NOM\$
 20 INPUT « POUVEZ-VOUS

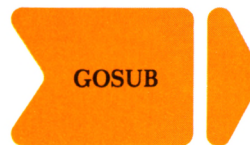
```

M'INDIQUER LE PREMIER
CHIFFRE DE VOTRE NUMÉRO
DE SÉCURITÉ SOCIALE » ; SS
30 IF SS = 1 THEN 60
40 IF SS = 2 THEN 70
50 PRINT « VOUS AURIEZ DU
M'INDIQUER UN 0 OU UN 1 » :
GOTO 20
60 PRINT « BONJOUR
MONSIEUR » ; NOM$ : GOTO 80
70 PRINT « BONJOUR
MADAME » ; NOM$ : GOTO 80
80 INPUT « CONNAISSEZ-VOUS
TIFY ? » ; R$
90 IF R$ = « OUI » THEN 120
100 IF R$ = « NON » THEN 130
110 PRINT « VEUILLEZ AVOIR LA
GENTILLESSE DE ME
RÉPONDRE UNIQUEMENT
PAR OUI OU PAR NON » : GOTO
80
120 PRINT « ALORS, NE
MANQUEZ PAS SES
RENDEZ-VOUS ! » : GOTO 140
130 PRINT « ALORS, JE SERAIS
TRÈS HEUREUSE DE VOUS
LA FAIRE CONNAÎTRE !
VENEZ À SON PROCHAIN
RENDEZ-VOUS. » : GOTO 140
140 PRINT « À BIENTÔT ! SIGNÉ :
ZOÉ »
  
```

IF...THEN suivi d'un numéro de ligne y est mis pour IF...THEN GOTO ; c'est aussi une formule résumée. Passons maintenant à une autre instruction Basic fort intéressante : c'est **GOSUB**. Elle renvoie, non pas à une ligne d'instruction (comme GOTO), mais à un ensemble d'instructions, à un sous-programme.

Prenons par exemple :
 10 CLS
 20 LOCATE 9, 11
 30 PRINT « INTOX ET ZOÉ »
 40 GOSUB 120
 50 LOCATE 21, 13
 60 PRINT « VOUS »
 70 GOSUB 120
 80 LOCATE 25, 15
 90 PRINT « SALUENT »
 100 GOSUB 120
 110 END
 120 SCREEN 11, 8, 3
 130 SCREEN 2, 10, 15
 140 SCREEN 7, 12, 9
 150 SCREEN 14, 1, 12
 160 SCREEN 4, 5, 6
 170 RETURN

La ligne 40 indique à l'ordinateur qu'après 30, il doit se rendre en 120 et effectuer toutes les opérations suivantes (des variations de couleurs), jusqu'à ce qu'il rencontre **RETURN**. A ce moment-là, il retourne à la ligne qui suit l'ordre GOSUB, c'est-à-dire, ici, 50, puis à 60. Ensuite, 70 lui indique de réeffectuer le même sous-programme puis de revenir en 80. Enfin, 100 lui commande une



fois de plus la même boucle. END, en 110, marque la fin du programme. C'est une instruction indispensable sinon l'ordinateur ne saurait plus quoi faire après le troisième RETURN. A la fin de l'opération, l'écran affiche :

```

INTOX ET ZOE
      VOUS
      SALUENT
  
```

Comme toute boucle, celle créée par GOSUB peut être interrompue par l'instruction de contrôle CNT-C (qui affiche le numéro de la ligne sur laquelle le programme s'est arrêté).

Une petite variante :

```

10 CLS
20 LOCATE 0, 11
30 ATTRB 1, 1
40 PRINT « INTOX ET ZOÉ »
50 GOSUB 150
60 LOCATE 23, 13
70 ATTRB 1, 1
80 PRINT « VOUS »
90 GOSUB 150
100 LOCATE 26, 15
110 ATTRB 1, 1
120 PRINT « SALUENT »
130 GOSUB 150
140 END
150 SCREEN 11, 8, 3
160 SCREEN 2, 10, 15
170 SCREEN 7, 12, 9
180 SCREEN 14, 1, 12
190 SCREEN 4, 5, 6
200 RETURN
  
```

ATTRB

ATTRB est une instruction qui modifie la taille des lettres affichées à l'écran. ATTRB 1, 0 en double la largeur, ATTRB 0, 1 en double la hauteur et ATTRB 1, 1 double à la fois la hauteur et la largeur. Essayez !

Un programme « professionnel »

Le grand intérêt des langages évolués est de permettre l'écriture sans trop de difficultés de programmes répondant à des besoins précis. Le Basic, par exemple, bien que d'un abord simple, permet de traiter des problèmes assez complexes. En voici un exemple concret : au Centre Georges Pompidou, à

Beaubourg, les normes de sécurité interdisent l'accès simultané, aux différents étages, de plus de 4 000 personnes. Les impératifs de rentabilité, eux, disent qu'il ne faudrait pas que ce nombre tombe au-dessous de 3 850.

Le problème est donc de régulariser, grâce à un système d'ouverture et de fermeture automatique des portes d'entrée, le nombre des visiteurs, ce qui suppose un contrôle des points d'accès, et le décompte des entrants et des sortants. Pour ce faire, différentes méthodes (caméras, yeux électroniques, etc.) sont envisageables, mais toutes se résument à établir une variable à deux positions, selon qu'une personne sort ou rentre. Dans le programme, nous appellerons cette variable A ; elle prendra la valeur 1 pour un entrant et 0 pour un sortant. Si le nombre des personnes présentes dans le Centre est N, N augmente de 1 à chaque fois que A = 1 et diminue de 1 à chaque fois que A = 0. Le contrôle est alors réalisé grâce au programme suivant :

```

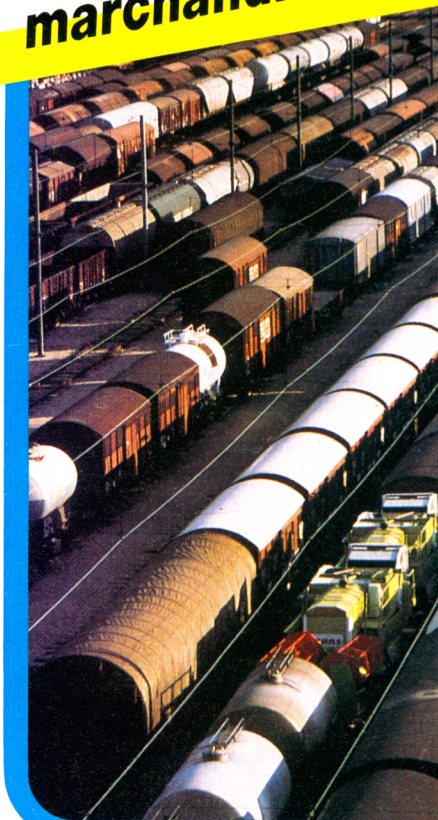
10 N = 0
20 INPUT A
30 IF A=1 THEN N=N + 1
40 IF A=0 THEN N=N - 1
50 PRINT « NOMBRE DE
   PERSONNES A L'INTERIEUR » ;
   N
60 P=1 'OUVERTURE
70 IF N ≥ 4000 P = 0
   'FERMETURE
80 IF N ≤ 3850 THEN P = 1
   'OUVERTURE
90 GOTO 10
  
```

La ligne 10 indique qu'à l'ouverture du Centre, le compteur N est à zéro. La ligne 20 enregistre la première personne qui passe (le premier A est forcément égal à 1, à moins que quelqu'un n'ait dormi dans le Centre... Ce qui laisserait sûrement l'ordinateur perplexe !) et en déduit (lignes 30 et 40) la valeur de N. Il ne reste plus qu'à tester N par rapport aux valeurs plafond (4000) et plancher (3850), et à commander l'ouverture ou la fermeture des portes d'entrée, c'est-à-dire la mise à 1 ou à 0 de la variable « porte d'entrée », P :

la ligne 60 ordonne que la porte soit ouverte au départ ; 70 : dès que N devient égal à 4000, elle se ferme ; 80 : quand N devient égal à 3850, la porte s'ouvre de nouveau ; 90 veut dire que, dès qu'une personne est comptée, on passe à la suivante.

' (apostrophe) est une instruction qui permet d'introduire des commentaires dans un programme. Ces commentaires ne sont pas pris en compte par l'assembleur.

La gestion centralisée du trafic marchandises de la SNCF



L'acheminement des wagons

Bonjour, je suis le wagon 218712732062, je viens de recevoir un chargement de pommes de terre en gare de Saint-Brieuc ; elles sont destinées à la société NEG, grossiste à Bordeaux. Nous sommes en retard, je vais donc être acheminé en régime accéléré (RA) ; je suis ainsi sûr d'arriver à la date prévue. Si l'expéditeur s'y était pris un peu plus tôt, j'aurais pu partir en régime ordinaire (RO)... Cela lui aurait coûté moins cher ! Pendant mon voyage, je vais changer deux fois de train et me retrouver avec des collègues qui viennent de Paris ou de Nantes. Avant mon départ, l'expéditeur a rempli une déclaration qui contient notamment le type de marchandise à transporter (pommes de terre), son poids (20 tonnes), la gare de destination (Bordeaux). Ces renseignements serviront à la SNCF

pour calculer, entre autres, le prix de mon voyage. Toutes les informations me concernant vont être envoyées à l'ordinateur de Paris et enregistrées dans un fichier central, avec celles concernant les quelque 100 000 à 300 000 wagons acheminés chaque jour. Elles pourront ainsi être consultées par n'importe quel agent de la SNCF, dans n'importe quelle gare. Au moment de la composition des trains j'ai été mis dans le lot 42080, qui partira le 23, accroché au train NB 3455. Dès que la gare de Saint-Brieuc a reçu ces informations, elle les a renvoyées, pour vérification, par téletype. Le 23, elle adressera un message « DÉPART » à l'ordinateur. Ce dernier vérifiera la structure du message et en comparera les informations avec celles d'un fichier-barème. La consultation d'un Tableau d'Acheminement, ainsi que celle d'un Tableau d'Enlèvement lui permettra de déterminer la première escale du wagon (Rennes) et la validité du train de départ.

L'ordinateur pourra alors déterminer le jour d'arrivée à l'escale (le 23) et l'heure (20 h) à partir de laquelle je pourrai être incorporé au train en formation pour Nantes (RN 2514). L'ordinateur sait à quelle heure et sur quelle voie ce train sera formé ainsi que les conditions de son départ. Le même processus se reproduira le lendemain en gare de Nantes, et j'arriverai enfin à Bordeaux le 25, avec le train NB 3455, en compagnie d'autres wagons transportant des tuiles originaires de Rennes, de l'essence provenant de Gestel et du verre de Wingles. Je ferai alors connaissance avec mon destinataire, et je serai déchargé et remis à disposition pour rechargement... Le cycle repartira ! Ainsi donc, à partir des données de base que sont : la déclaration d'expédition et les relevés des trains et d'opérations diverses, un fichier centralisé est constitué. Chacun d'entre nous n'y est représenté qu'une seule fois avec ses caractéristiques physiques stables



(type, longueur), celles de son chargement (nature, poids) et sa position à un moment donné. Et c'est à partir de ce fichier que seront élaborés les préavis d'arrivée des trains pour les gares de triage, la mise à disposition pour déchargement, les plans de débranchement, les suivis de nos séjours en gare, etc...

La gestion du matériel

Mais ce n'est pas tout. Comme tout un chacun, je m'use à rouler tout le temps. Je dois donc être surveillé. Cette gestion, pour un parc aussi important que celui de la SNCF, pose un problème d'organisation difficile. Elle suppose, en effet :

- la connaissance précise du parc : les wagons ont un code à douze chiffres significatifs qui est un code de leurs caractéristiques (matériaux des parois et planchers, équipements de freins, etc.),

- le suivi de l'entretien : programmation des révisions, envois en révision, contrôle de l'exécution ;
- le suivi de l'utilisation des wagons entre deux révisions : parcours, type de trafic, avaries accidentelles, inutilisations ;
- la détermination des coûts : dépenses d'entretien, calcul des coûts moyens de la tonne-kilomètre, détermination de la politique d'entretien, échanges internationaux.

La gestion des envois en entretien

Cette application utilise les fichiers des wagons et des mouvements dans le but de minimiser les dépenses d'envois en entretien et d'optimiser l'utilisation des ateliers. Le wagon 218712732062 a été révisé pour la dernière fois le 30 mars 1980. Il doit être révisé de nouveau

quatre ans plus tard, c'est-à-dire le 30 mars 1984, dans un établissement spécialisé situé près de Valence.

Trois mois avant sa date théorique de révision, c'est-à-dire le 1^{er} janvier 1984, le wagon est enregistré dans un fichier de wagons à envoyer en révision. Grâce au fichier central des mouvements, l'ordinateur sait à tout moment où il se trouve et s'il est chargé ou vide. Il sait également quelle est la charge de travail de l'établissement de Valence. A partir du 1^{er} janvier 1984 et jusqu'au 30 juin au plus tard, l'ordinateur compare ces deux fichiers et dès qu'il sait que le wagon 218712732062 doit passer à vide à moins de 100 km de Valence et que l'atelier peut l'accepter, il donne l'ordre de l'y envoyer pour révision.

Cette application a considérablement raccourci le temps moyen d'immobilisation d'un wagon (de deux semaines environ) et divisé par deux les distances d'acheminement du wagon vers l'atelier, économisant ainsi 2 000 wagons. La distribution du travail



dans les ateliers a pu aussi être optimisée. Un projet est actuellement en cours pour remplacer les temps arbitrairement décidés entre deux révisions par un calcul du travail effectivement accompli par le wagon : nombre de tonnes transportées multiplié par la distance. Ce projet devrait économiser quarante millions de francs de plus par an.

Le fichier central

Ce fichier central, qui décrit les trains, les wagons et les événements s'y rapportant est évidemment une mine d'informations en tous genres largement utilisée par la SNCF. Mais tout cela ne s'est pas fait en un jour. L'acte de naissance de la Gestion Centralisée remonte à 1955. En ce temps-là, l'ordinateur s'appelait encore calculateur et on imaginait assez difficilement ce que serait l'avenir électronique. En revanche, on savait qu'il fallait procéder à des réorganisations préalables et que ces travaux

seraient de toute façon productifs, ordinateur ou pas. Un million de lignes de Cobol et Fortran ont ainsi été écrites depuis ; elles représentent 600 années-homme de travail. Aujourd'hui, le système tourne avec une deuxième génération de programmes (environ mille en tout) dont la maintenance est assurée par une équipe de vingt-deux analystes-programmeurs. Avec cette mise sur ordinateur de tous les problèmes que pose le transfert des marchandises par wagon, qu'ils intéressent les mouvements, la gestion ou l'entretien du matériel, c'est une

véritable mutation qu'a accomplie la SNCF. L'exploitation d'un seul fichier centralisé accessible de partout, par tous, simplifie considérablement tous les travaux. La qualité du service rendu à la clientèle en a été fortement améliorée, tant au point de vue de la rapidité d'acheminement que de la connaissance de cet acheminement. L'expéditeur sait à tout moment où est son wagon et quand il lui sera restitué. Les agents de la SNCF peuvent à tout moment, et en quelques secondes, avoir un profil précis du client qu'ils sont en train de traiter.

LE TRAFIC MARCHANDISES DE LA SNCF EN QUELQUES CHIFFRES

- 16 milliards de francs de recettes en 1983 ;
- 300 000 wagons ;
- Plus de 1 000 terminaux ;
- 16 millions d'octets de mémoire centrale ;
- 20 milliards de caractères sur disque ;
- 1 unité centrale de 8 MIPS ;
- 154 millions de francs de budget informatique représentant 1 % du montant des recettes ;
- 15 secondes pour savoir où est un wagon et où il va.

L'heure du choix

J'en avais plein les oreilles de la micro-informatique ! Tout le monde s'y était mis : la radio, la télévision, des amis qui, au cours de conversations, échangeaient des propos apparemment innocents sur les mérites comparés du VICTOR, de l'APPLE ou du TO 7. Je me sentais imperceptiblement mais sûrement exclu de leur cercle. Mes enfants, de leur côté, creusaient au bulldozer le fossé entre générations. Ils s'échangeaient avec convoitise des disquettes... « pour les diplômés », disaient-ils, avec un rien de sadisme dans l'expression. Le journal s'y mettait aussi : publicités racoleuses, articles, déclarations politiques, tout y passait. Même le bus que je prenais chaque matin passait devant une boutique, assez mystérieuse pour moi, située entre une boulangerie et une teinturerie. Son nom, je ne m'en souvenais jamais exactement. Il y avait de l'anglais : *computer*, et puis je ne sais quoi avec. Un soir, je descendis une station plus tôt, fis quelques pas et me retrouvai devant le magasin. Dans la vitrine, deux ou trois ordinateurs clignaient de l'écran aux badauds. Un clavier, un écran, des boîtiers à fente : tous se ressemblaient, de l'extérieur du moins. Je restai un moment devant la vitrine, hésitant, d'autant qu'à l'intérieur officiait une jeune femme blonde qui avait l'air de bien s'y connaître. Alors moi, pauvre homme ignare... On ne se refait pas, hein ? Pour me donner du courage, j'allai m'acheter un pain au chocolat à la boulangerie voisine, ce qui me donna l'air occupé et assura mon pas. « Monsieur... Vous désirez ? » Je ne pouvais vraiment pas prendre l'air expert, surtout à côté de cet autre vendeur, que je venais de découvrir en entrant et qui tapait magistralement sur un clavier ! L'informaticien-type, celui-là, complet croisé et attaché-case débordant de papiers à ses pieds... La franchise s'imposait donc, rendue d'autant plus nécessaire que je tenais encore à la main un bout de pain au chocolat qui s'émiettait sur la moquette. « Eh bien, voilà, Mademoiselle : je suis cerné par les ordinateurs, tout le monde en a autour de moi et je me sens tout bête. Alors je voudrais faire quelque chose... » Heureusement elle a ri, mais d'un rire gentil, indulgent, qui incitait à la confiance. Pas méprisant du tout, sinon j'aurais filé sans demander mon reste. « Avez-vous une idée de ce que vous voulez faire avec ? » Il y avait de la malice dans son regard, j'en suis sûr... « Non. Vraiment non. D'ailleurs je ne sais pas non plus quoi choisir ! Comment fait-on généralement ? » — Eh bien, il y a plusieurs façons de procéder. La première, c'est de se fixer sur une machine particulière, sur une marque... C'est comme pour les voitures : il y a les inconditionnels d'APPLE, ceux de RADIO-SCHACK et ceux de VICTOR, par exemple. Il y a aussi ceux qui ont entendu parler d'une petite-machine-pas-chère sur laquelle ils ont décidé de faire leurs débuts : ZX80, ALICE, LASER, etc.

Il y a enfin les innovateurs, ceux qui testent les nouvelles venues : en ce moment le M05 de THOMSON ou l'APPLE IIc. » En disant cela, elle me conduisait vers un grand panneau sur lequel étaient inscrites les différentes marques avec, en regard, des détails techniques.

« 16 K de mémoire, ça veut dire quoi ? »

— C'est la taille de la mémoire ; ça signifie qu'elle contient environ 16 000 caractères. En fait, c'est peu, et il faut prévoir des mémoires additionnelles.

— ???...

— Oui, des cassettes ou des disquettes comme celles qui sont là.

— Si je comprends bien, on a intérêt à avoir l'ordinateur qui a le plus de K ?

— Pas forcément, parce qu'il y a autre chose qui compte beaucoup :

les performances du BASIC employé ; c'est la facilité d'accès et d'emploi, si vous voulez — avec un mauvais BASIC, une partie de la mémoire peut être neutralisée. Il n'y a pas que le BASIC d'ailleurs, il y a d'autres langages : PASCAL, FORTH, etc. Tenez, si vous avez des enfants, il y en a un qui sort sur tous les matériels en ce moment, c'est le LOGO. Avec, on fait très facilement des petits dessins. C'est amusant et instructif. Regardez ! »

A côté d'elle, il y avait un petit ordinateur noir. Elle mit dans une fente ce que je pris pour une boîte d'allumettes.

« C'est quoi ça ? »

— Une cartouche LOGO. »

Et, après avoir allumé le micro, elle tapa quelques mots sur le clavier. Un instant plus tard, le dessin apparut.

« Alors, on peut faire tout ce qu'on veut ? C'est très drôle ! »

— Ne m'en parlez pas, à l'heure de la sortie des classes, on est envahi par les gosses ! »

Je me demandais s'il n'y avait pas, de temps en temps, au nombre de ces gamins, l'un ou l'autre des miens, et l'idée sordide me traversa l'esprit que je pourrais peut-être faire l'économie d'un achat...

« Donc, j'achète une machine, des disquettes, un écran... »

— Non, pas forcément. Il y a certains appareils que vous pouvez brancher sur votre télé, avec un système de raccordement.

— Bon, et alors je me mets à programmer ?

— Oui, si ça vous branche, bien sûr ! »

« Me brancher ?... C'est vrai qu'elle était plus jeune que moi... autre génération, autre vocabulaire... « être branché » — comme un ordinateur —, tiens !... »

« Notez bien que vous pouvez aussi utiliser des logiciels, des programmes tout faits, quoi ; ceux qui répondent à vos besoins, bien sûr. »

— Par exemple ?

— Et bien, en ce moment, il y a de plus en plus de gens qui achètent des machines, non pas pour programmer eux-mêmes, mais pour faire des choses très précises : apprendre la dactylographie, par exemple, ou pour

faire du courrier ; du « traitement de textes » comme on dit. Il y a des programmes tout prêts pour cela : WORDSTAR, EPISTOLE, APPLEWRITER, MAGIC WINDOW, SCRIPTOR, etc. Là, c'est une autre façon de choisir : toutes les marques n'en possèdent pas... Notez bien qu'on peut connecter à l'ordinateur des petites machines à écrire pas chères, comme la BROTHER, et que ça marche très bien !

— Que peut-on faire d'autre encore ?

— Il y a des gens qui achètent des programmes pour classer leur documentation, leurs bouquins, leurs disques, etc. Ils disent que ça leur fait gagner de la place, parce qu'alors ils rangent tous leurs documents en piles, avec juste un numéro pour chacun d'entre eux, ce qui leur permet de les classer puis de les retrouver grâce à leur logiciel : « Où est mon document sur la chasse à la baleine ? » Ils tapent CHASSE et BALEINE et la machine leur donne le numéro du document situé dans sa pile.

Des logiciels de ce genre, il y en a plein : PFS-THOMSON vient d'en sortir un, par exemple. Il y a même un programme qui aide à construire des rapports, des scénarios, etc. C'est THINKTANK. Et puis, il y a des programmes qui font tout ce que vous voulez : les tableurs comme VISICALC. Enfin, il y a les jeux d'aventure ! Certains jeunes y consacrent des heures et des heures, sans jamais se lasser.

... Voilà, je crois que je vous ai tout expliqué !

— Si je comprends bien, alors, il me reste à réfléchir pour savoir ce à quoi je vais bien pouvoir utiliser mon micro-ordinateur... Pas idiot le coup du classement. Je voulais m'acheter une bibliothèque pour ranger tous mes papiers matière par matière... Je fais un trait sur la bibliothèque et je mets l'argent dans le micro-ordinateur ! Je gagne de la place, plus le temps que je mettais à rechercher mes documents, et... je fais plaisir aux enfants ! » Je disais au revoir à la vendeuse, quand une jeune fille entra, essoufflée, et lui dit quelque chose comme : « La tuile... J'ai mon chat mauve qui est en rade ! Tu peux pas me prêter une quatre-vingt colonnes ? » Drôle de jargon, pensai-je, il me reste encore bien du chemin à faire...



Le choix d'un micro-ordinateur n'est, certes, pas chose facile. Alors voici quelques informations qui devraient rendre cette opération moins hasardeuse.

Le tableau page suivante présente 15 micro-ordinateurs. Ce n'est pas une sélection de matériels, mais plutôt des exemples de ce que l'on trouve actuellement sur le marché pour moins de 5 000 F.

LES PRINCIPALES CARACTÉRISTIQUES DE 15 MICRO-ORDINATEURS

Marque, Modèles Pays	Mémoires	Connexion TV	Langages	Périphériques	Programmétique	Observations
MATRA HACHETTE (Fr.) ALICE Clavier AZERTY	ROM 8 K RAM 4 K Extension de 16 K	Péritel ou modulateur noir et blanc	Basic Microsoft résident	Imprimante Cassettophone	Logiciels EDICIEL cassettes	Appareil d'initiation, avec bon manuel mais capacités limitées. Connectable cassetophone standard
THOMSON (Fr.) TO 7 Clavier AZERTY	ROM 6 K + 16 K en Memo 7 RAM 22 K Extension de 16 K	Péritel ou modulateur SECAM	Basic Microsoft en Memo 7 Prévus : Logo, Forth, Assembleur	Cassettophone spécial Crayon optique intégré Extensions jeux et communication	Logiciels éducatifs, jeux, gestion familiale cassettes dont certaines nécessitent extension RAM	De nombreux clubs sont équipés du TO7 qui va être remplacé par le T07-70 et le MO5 (ci-dessous) Programmes éducatifs s'accompagnant de livres
TO 7-70 Clavier AZERTY	RAM 64 K extensible à 128 K	Péritel	dito ci-dessus	Cassettophone spécial Crayon optique intégré Imprimantes et lecteur disquettes Extensions modem	dito ci-dessus	Présenté comme une version améliorée du TO7 au niveau des capacités des mémoires, du confort clavier et de la gamme des périphériques
MO5 Clavier AZERTY	RAM 48 K dont 32 K disponibles	Péritel	Basic intégré Prévus : Logo, Forth	Crayon optique en option. Cassettophone spécial. Imprimante thermique. Extensions musique, jeux interfaces, dénomérisation	Les cassettes de programmes comporteront une face TO7 et une face MO5	Présenté comme le micro-ordinateur familial d'initiation, de jeux, de création dans le domaine du son et de l'image
MICRONIQUE (Fr.) Hector Coffret familial Clavier AZERTY	ROM 4 K RAM 16 K	Péritel ou modulateur	Basic sur cassette	Lecteur cassettes incorporé Manettes jeux	Une centaine : jeux, gestion familiale	Pour le jeu et l'initiation
HRX Clavier AZERTY	ROM 16 K RAM 64 K	Péritel ou modulateur	Forth résident Basic en option	Lecteur cassettes incorporé. Lecteur disquettes en option Manettes jeux en option	Logiciels sur cassettes	Petite gestion artisanale. Voir le HR+ de RAM 48 K avec basic résident, intermédiaire entre coffret familial et modèle HRX
SINCLAIR (G. B.) ZX81 Clavier QWERTY	ROM 8 K RAM 1 K extensible de 16 K à 64 K	sur prise UHF	Basic résident en option : Forth Assembleur	Cassettophone particulier Manettes jeux Imprimante	Jeux, gestion, utilitaires cassettes	Son prix et ses possibilités d'extension expliquent son succès
ZX Spectrum Clavier QWERTY	ROM 16 K RAM 16 K ou 48 K	PAL sur prise UHF Péritel	Basic résident en option : Pascal Forth Assembleur	Même cassettophone que le ZX 81 Imprimante et microdrive en option	Jeux, gestions, utilitaires cassettes, cartouches	Plus performant que le ZX 81, notamment dans la version 48 K. Imprimante et cassettophone compatible entre le ZX 81 et le Spectrum
ORIC (G. B.) ATMOS Clavier QWERTY AZERTY	ROM 16 K RAM 48 K	PAL sur prise UHF Péritel	Basic résident en option : Forth Pascal Assembleur	Magnétophone standard Lecteur disquettes 3 pouces Imprimante en option	Jeux nombreux, gestion, langages Méthode Assimil Anglais	Successeur d'ORIC 1, avec clavier agréable et basic performant. Une centaine de logiciels
ACORN (G. B.) ELECTRON Clavier QWERTY	ROM 32 K RAM 32 K	UHF Péritel	Basic et Assembleur résidents	Magnétophone standard	Jeux, logiciels graphiques Programmes éducatifs prévus	Commercialisé récemment, bénéficie du même basic performant que le BBC plus cher
AQUARIUS (Japon) Clavier QWERTY	ROM 8 K RAM 4 K, extensions par 16 K	Pentel ou adaptateur	Basic Microsoft résident	Lecteur cassettes Imprimante Manettes jeux	Jeux, petite gestion, fichiers	Initiation et jeux
VIDEO- TECHNOLOGIE LASER 200 Clavier QWERTY	ROM 16 K RAM 4 K Extension 16 K ou 64 K	UHF	Basic Microsoft résident Assembleur en cassette	Lecteur cassettes spécial Imprimante Manettes jeux	Trentaine de jeux, petits programmes éducatifs En préparation : logiciels petite gestion (extension 16 K nécessaire)	Appareil d'initiation et de jeux pour le moment. A suivre le développement des logiciels
ATARI (USA) 600 XL Clavier QWERTY	ROM 24 K RAM 16 K Extension 64 K	UHF ou Péritel	Basic résident	Lecteur cassettes spécial Manettes jeux Imprimante	Abondante en jeux Programmes éducatifs avec Nathan, Hatier, Milliken	Permet de jouer et de s'initier. Très vite l'utilisateur cherchera l'extension de mémoire. Dans cette hypothèse, voir le 800 XL plus performant en version de base
COMMODORE (USA) VIC 20 Clavier QWERTY	ROM 20 K RAM 5 K extensible à 32 K	RVB ou adaptateur	Basic intégré en option : Forth Assembleur	Lecteur cassettes spécifique Stylo optique en option Manettes jeux	Nombreux logiciels utilitaires, de jeux, d'éducation	Bon matériel d'initiation grâce à ses possibilités d'évolution et à son choix de logiciels
COMMODORE 64 Clavier QWERTY	ROM 20 K RAM 64 K	RVB ou adaptateur	Basic intégré en option : Forth Assembleur Logo	Lecteur cassettes spécifique Lecteur disquettes Stylo optique en option Manettes jeux	Programmes éducatifs et scientifiques sur cassettes ou disquettes	Tous périphériques compatibles entre VIC 20 et son « grand frère » C 64. Intéressantes possibilités dans le domaine éducatif. Pour le bureau voir version portable intégrée avec un logiciel au choix

X 2000: L'informatique à sa maison

L'informatique fait partie de notre vie quotidienne. Tous les jours plus familière, elle devient le langage de notre temps : appelée à être comprise et parlée par tous, c'est notre lien concret avec les temps nouveaux, c'est développer notre capacité de communiquer, c'est stimuler notre aptitude naturelle à créer, notre besoin de savoir.

Nouvel élément de notre culture, l'informatique est à votre portée grâce à un réseau dynamique : les Centres X 2000.

Leur vocation : sensibiliser, informer, initier...

Leurs moyens : des animateurs compétents, une gamme étoffée d'ordinateurs et de terminaux informatiques, une foule de programmes et l'accès à toute une palette de banques de données, à une médiathèque complète (livres, revues, cassettes audio-visuelles...) à un réseau télématique mutuel d'information et d'assistance entre les centres.

Vous pouvez ainsi y trouver, par exemple, une première approche concrète de l'informatique en vous initiant à ses techniques et ses usages dans vos loisirs, chez vous, sur votre lieu de travail.

Les Centres X 2000 vous proposent ainsi de multiples prolongements aux émissions télévisées d'initiation informa-

tique : voir et revoir ces émissions, les reprendre point à point avec les animateurs des Centres qui vous apporteront les explications et développements complémentaires souhaités ; vous reprendrez avec eux les éléments de programme étudiés et pourrez vous exercer sur le matériel du Centre...

Vous pouvez en effet aussi plus généralement utiliser sur place tous les équipements du Centre et ses logiciels pour tous vos propres problèmes ou tout simplement satisfaire votre curiosité, vos besoins d'enrichissement intellectuel, culturel ou encore y exercer votre créativité naturelle en produisant vos propres programmes.

Si vous êtes déjà mordu, vous n'en démordrez pas. Si vous ne l'êtes pas encore, vous aurez vite la puce à l'oreille. L'informatique c'est simple, c'est drôle, c'est passionnant et c'est utile. Ça fait tilt.

A la joie d'apprendre s'ajoute le plaisir d'inventer. On peut jouer, dessiner, concocter des stratégies, composer de la musique, réfléchir et agir ensemble.

Avec X 2000, les "puces", les éléments de base de l'ordinateur, sont à portée de la main.

Pour tous renseignements : Fondation X 2000 - 13, place des Corolles
Paris La Défense 2 92400 Courbevoie - Tél. (1) 773.64.07



TIFY

LA CASSETTE VIDEO DES 3 PREMIERES EMISSIONS DE TFI

L'OUTIL, avec Richard Berry
LES MEMOIRES, avec Françoise Dorin
LES PROGRAMMES, avec Dominique Lavanant

EST VENDUE PAR CORRESPONDANCE.

Pour la commander, il vous suffit d'écrire à ALP-TIFY, 26, rue des Carmes 75005 PARIS et de joindre votre règlement libellé à l'ordre de ALP-TIFY, soit 249 FF plus 30 FF de frais de port.

Toute commande non accompagnée de son règlement ne sera pas honorée. Compter quatre semaines de délai de livraison.



TÉLÉVISION FRANÇAISE 01
PRODUCTEUR EXÉCUTIF DE LA SÉRIE ... S'IL TE PLAÎT, RACONTE-MOI UNE PUCE"
SERVICE VIDÉOTEXTE : (3) 615.91.77 "TV INFO"
83, BOULEVARD DU MONTPARNASSE A PARIS 6^e

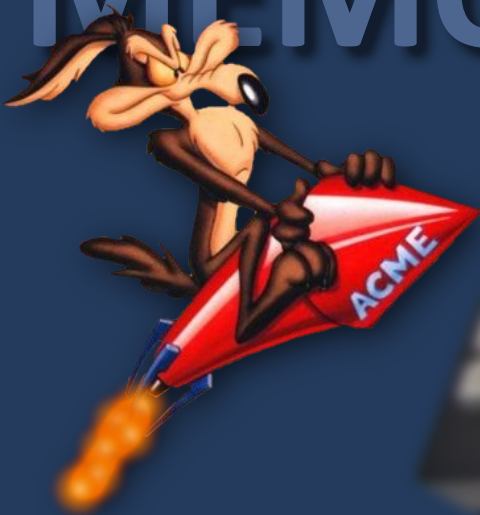


Document **numérisé**
avec amour par :

AMSTRAD

CPC 

MÉMOIRE ÉCRITE



<https://acpc.me/>