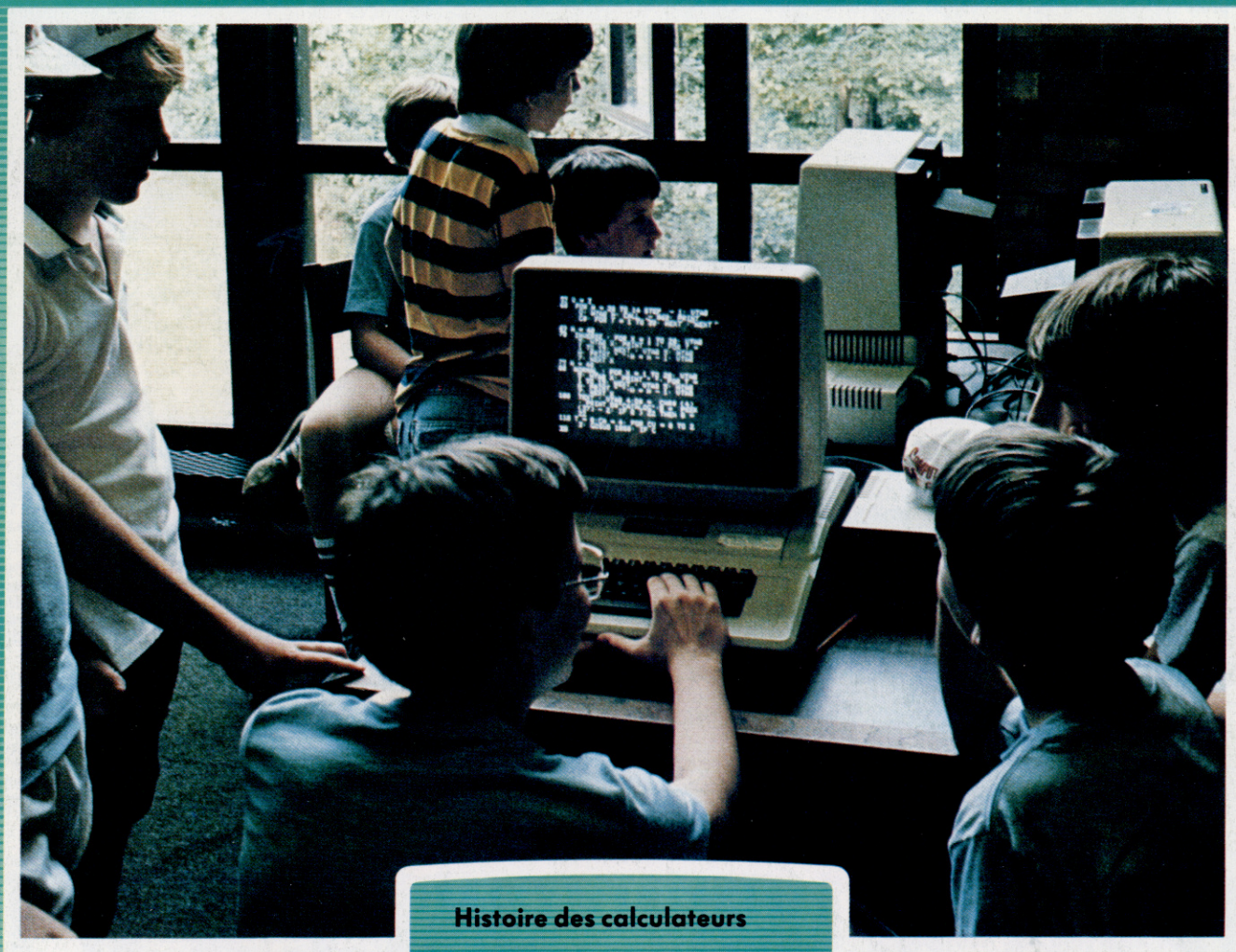


# abc

N° 5

COURS  
D'INFORMATIQUE  
PRATIQUE  
ET FAMILIALE

## INFORMATIQUE



Histoire des calculateurs

Essayez le «BBC Model B»

L'addition en binaire

Un calendrier en basic

EDITIONS  
ATLAS

# Sommaire

## Matériel



D'excellentes caractéristiques techniques ont fait du BBC modèle B un ordinateur très populaire.

89

## Logiciel



L'ordinateur a déjà fait son apparition dans les salles de classe, et un vaste choix de programmes est offert aux enseignants.

81

## Programmation basic



Comment écrire, à l'aide de nouvelles commandes, un programme qui compte le nombre de jours avant Noël.

98

## Le marché



Tout programme rédigé par l'utilisateur doit être traduit en langage machine pour que l'ordinateur puisse le comprendre.

84

Nous devons les puces de l'électronique à trois siècles de travaux théoriques.

86

Une unité à cassette permet de programmer sans la crainte de perdre le contenu de la mémoire.

94

## Mots de passe



Les nombres binaires 1 et 0 peuvent être additionnés à l'aide de simples opérations logiques, ET, OU et NON.

92

L'ordinateur peut stocker des milliers d'octets dans sa mémoire et se souvient exactement de leur localisation précise.

96

## Prochain fascicule



● Nous présenterons l'Atari 400 et l'Atari 800. La grande qualité de leur graphisme a permis la création de jeux stupéfiants.

● Nous explorerons les changements que l'informatique apportera au sein même de votre foyer.

# abc

INFORMATIQUE

cours d'informatique pratique et familiale

EDITIONS ATLAS

Édité par ÉDITIONS ATLAS s.a., tour Maine-Montparnasse, 33, avenue du Maine, 75755 Paris Cedex 15. Tél. : 538-52-70.

Belgique : ÉDITIONS ATLEN s.a., Bruxelles.

Canada : ÉDITIONS ATLAS CANADA Ltée, Montréal Nord.

Suisse : FINABUCH s.a., ÉDITIONS TRANSALPINES, Mezzovico.

Réalisé par EDENA s.a., 29, boulevard Edgar-Quinet, 75014 Paris. Tél. : 320-15-01.

Direction éditoriale : J.-Fr. Gautier. Service technique et artistique : F. Givone et J.-Cl. Bernar. Iconographie : J. Pierre. Correction : B. Noël.

### VENTE

Les numéros parus peuvent être obtenus chez les marchands de journaux ou, à défaut, chez les éditeurs, au prix en vigueur au moment de la commande. Ils resteront en principe disponibles pendant six mois après la parution du dernier fascicule de la série. (Pour toute commande par lettre, joindre à votre courrier le règlement, majoré de 10 % de frais de port.)

Pour la France, s'adresser à ÉDITIONS ATLAS, tour Maine-Montparnasse, 33, avenue du Maine, 75755 Paris Cedex 15. Tél. : 538-52-70.

Pour les autres pays, s'adresser aux éditeurs indiqués ci-dessous.

### SOUSCRIPTION

Les lecteurs désirant souscrire à l'ensemble de cet ouvrage peuvent s'adresser à :

France : DIFFUSION ATLAS, 3, rue de la Tave, 28110 Lucé. Tél. : (37) 35-40-23.

Belgique : ÉDITIONS ATLEN s.a., 55, avenue Huart-Hamoir, 1030 Bruxelles. Tél. : (02) 242-39-00. Banque Bruxelles-Lambert, compte n° 310-0018465-24 Bruxelles.

Canada : ÉDITIONS ATLAS CANADA Ltée, 11450 boulevard Albert-Hudon, Montréal Nord, H 1G 3J9.

Suisse : FINABUCH s.a., ÉDITIONS TRANSALPINES, zona industriale 6849 Mezzovico-Lugano. Tél. : (091) 95-27-44.

### RELIEZ VOS FASCICULES

Des reliures mobiles, permettant de relier 12 fascicules, seront en vente en permanence chez votre marchand de journaux.

**ATTENTION** : ces reliures, présentées sans numérotation, sont valables indifféremment pour tous les volumes de votre collection. Vous les numéroterez vous-même à l'aide du décalque qui est fourni (avec les instructions nécessaires) dans chaque reliure.

En vente tous les vendredis. Volume I, n° 5.

ABC INFORMATIQUE est réalisé avec la collaboration de Trystan Mordrel (secrétariat de rédaction), S.I.-André Laroche (traduction), Ghislaine Goullier (fabrication), Marie-Claire Jacquet (iconographie), Patrick Boman (correction). Crédit photographique, couverture : Photo Tannenbaum-Sygm.

Directeur de la publication : Paul Bernabeu. Imprimé en Belgique par Proost, à Turnhout. Distribution en France : N.M.P.P. Tax. Dépôt légal : février 1984. 3842. Dépôt légal en Belgique : D/84/2783/27.

© Orbis Publishing Ltd., London.  
© Editions Atlas, Paris, 1984.

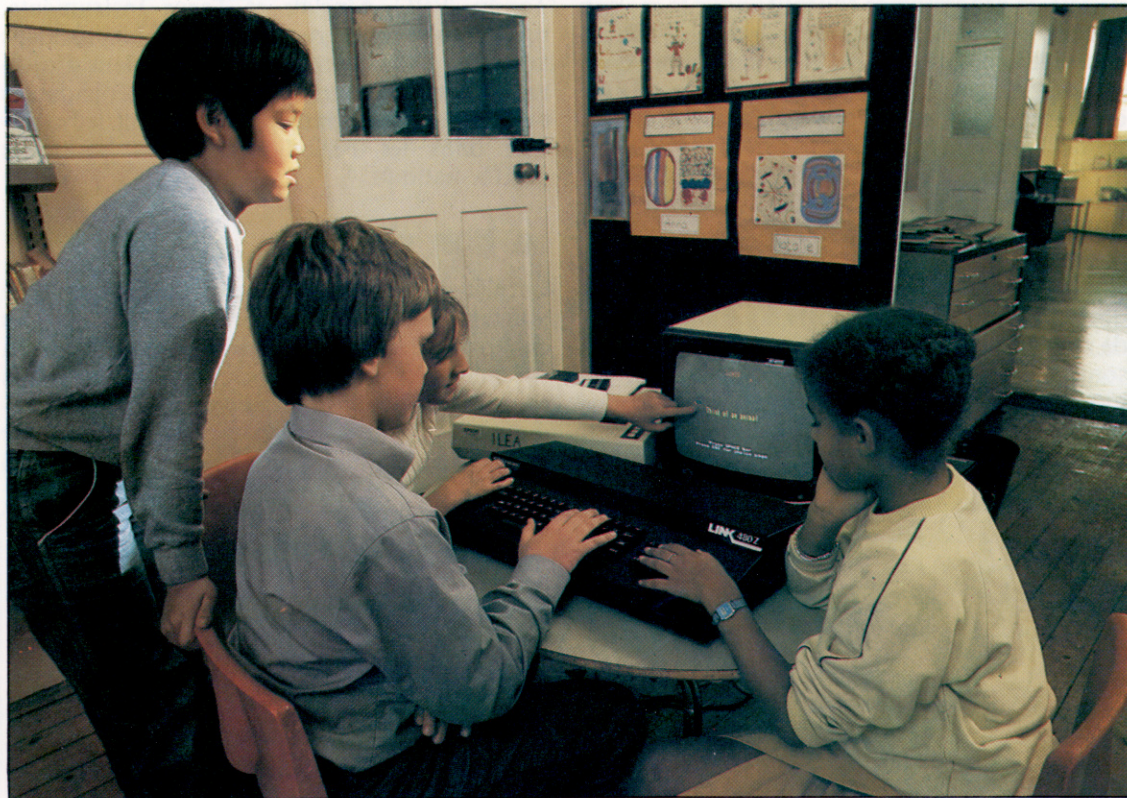
### A NOS LECTEURS

En achetant chaque semaine votre fascicule chez le même marchand de journaux, vous serez certain d'être immédiatement servi, en nous facilitant la précision de la distribution. Nous vous en remercions d'avance.

Les Éditions Atlas



# L'ordinateur à l'école



## L'ordinateur à l'école primaire

La Grande-Bretagne prépare l'avenir avec une politique scolaire qui vise à initier les enfants à l'usage d'un ordinateur dès leurs premières années à l'école. Le but n'est pas de familiariser l'enfant avec la nouvelle technologie, mais plutôt de se servir de l'ordinateur comme outil pédagogique. L'ordinateur est un professeur très patient qui peut enseigner une variété de sujets et qui permet à chaque enfant de travailler à son propre rythme. Avec l'ordinateur l'enfant ne fait pas qu'apprendre, il comprend comment un problème est analysé et résolu. (Cl. Ian Dobbie.)

**L'ordinateur a déjà fait son apparition dans les salles de classe, et un vaste choix de programmes éducatifs est déjà offert.**

Les écoles primaires auront bientôt un ordinateur, et plusieurs lycées en possèdent déjà un. Aujourd'hui, les ordinateurs ne sont plus réservés aux étudiants en informatique, mais sont utilisés également en littérature, en mathématiques et dans l'enseignement des langues étrangères.

Il existe plusieurs programmes d'enseignement pour ordinateur domestique, mais les enseignants se plaignent souvent de leur mauvaise qualité. Ces programmes observent rarement à la fois les règles pédagogiques et informatiques.

Les programmeurs ont rarement l'expérience de l'enseignement, et les enseignants qui essaient de programmer pour la première fois font souvent des erreurs élémentaires.

Même si un programme écrit par un enseignant ne quitte généralement pas sa classe, il est possible qu'il soit utilisé ailleurs. Le programme, pour être utilisé par d'autres personnes, doit être accompagné d'une excellente documentation.

Le programmeur peut répondre que les étudiants n'ont qu'à taper LOAD, mais il oublie souvent que tous les détails doivent être fournis à un débutant.

Quelle est l'utilité d'un logiciel éducatif? L'analyse du travail pédagogique indique qu'une bonne partie du temps des enseignants est dévolue à la répétition des notions mal comprises par les élèves. L'ordinateur accomplit cette tâche à merveille, sans jamais se fatiguer ou s'énerver. L'élève qui prend plus de temps que ses petits camarades pour assimiler le cours sera enchanté de trouver un « professeur » électronique qui puisse s'adapter à son rythme particulier d'apprentissage. La grande complexité des bons programmes d'enseignements entraîne des investissements élevés. Seules les grandes entreprises peuvent prendre part à la réalisation de programmes éducatifs. En France, l'association de Matra et de Hachette commence à commercialiser des logiciels de mathématiques.

Une bonne programmation implique la prévision des erreurs pouvant être faites par un débutant. Cela est très important, l'efficacité du programme en dépend. Un programme doit non seulement obéir lorsque les bonnes touches sont pressées, mais aussi ne rien faire quand une mauvaise touche est pressée. C'est la partie la plus difficile de la programmation. Le programme doit pouvoir récupérer les erreurs d'un enfant en lui laissant toujours l'impression

que les ordinateurs sont faciles et agréables à utiliser.

Malgré ces problèmes, il existe une grande variété de programmes éducatifs pour la maison et pour l'école. Les ordinateurs sont de merveilleux outils pédagogiques, et lors du choix d'un programme pour votre enfant il vous sera utile de comprendre les différentes façons d'utiliser un ordinateur.

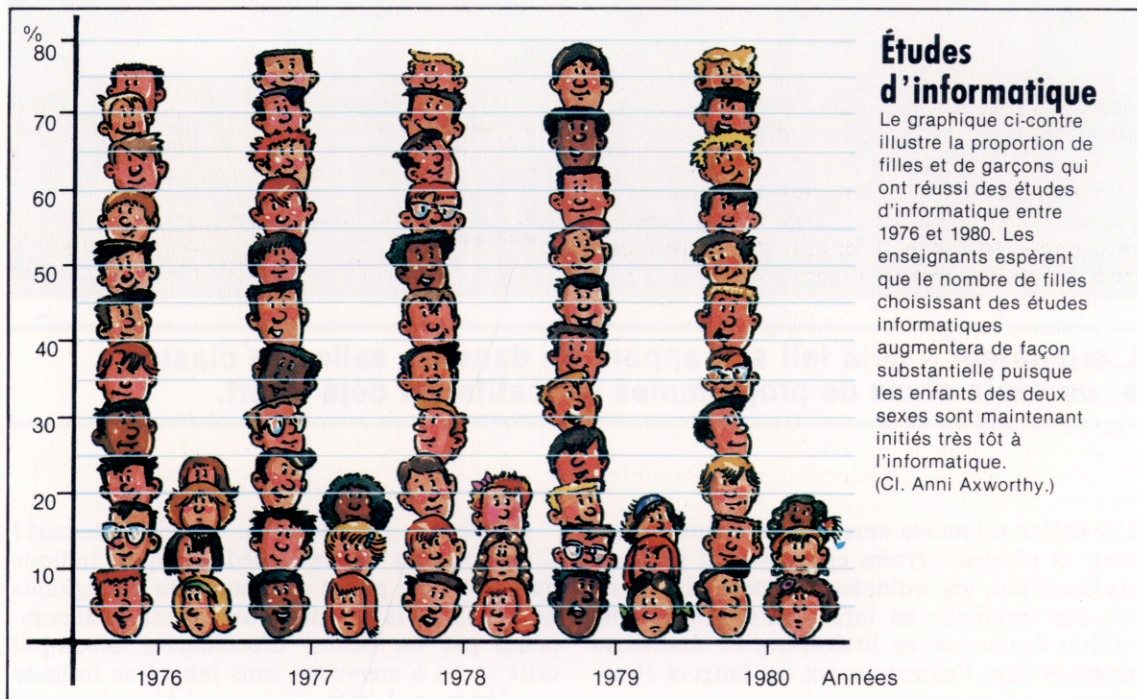
Un ordinateur peut être utilisé pour enseigner presque toutes les matières. Un bon programme fascinera et motivera l'enfant.

La plupart des programmes éducatifs sont fondés sur des exercices. Des exemples sont donnés à l'enfant, et le programme lui demande de résoudre un problème similaire. Les réponses sont normalement notées, l'enfant reçoit un message d'encouragement s'il donne la bonne réponse, ou le programme lui demande gentiment de recommencer s'il a fait une erreur.

beaucoup misé sur l'enseignement. Il existe une grande variété de programmes éducatifs pour les ordinateurs Apple, PET de Commodore, Tandy, BBC, Sinclair et Texas Instruments, mais d'autres constructeurs seront bientôt dans la course. Il y a d'un côté le constructeur d'ordinateurs, et de l'autre les sociétés indépendantes de logiciels. Ces dernières annoncent leur produit dans les périodiques et les boutiques d'informatique.

## Le bon choix

La plupart des programmes pour les moins de huit ans se concentrent sur l'apprentissage des chiffres et des lettres. Texas Instruments est un spécialiste en la matière. De nombreux parents sont séduits par la vaste gamme de logiciels produits par TI et Scott, Foresman & Co. pour l'ordinateur familial TI-99/4A. Il s'agit d'un



Le choix d'un programme dépend de plusieurs facteurs : l'âge de votre enfant, la marque de votre ordinateur et les matières étudiées à l'école.

Si vous n'avez pas encore acheté votre ordinateur et si vous croyez qu'il aura surtout une fonction pédagogique, demandez quel type d'ordinateur est utilisé à l'école. Si vous pouvez acheter un modèle semblable, les programmes de l'école pourront être exécutés à la maison. Plusieurs écoles offrent aux parents des copies des programmes utilisés en classe. Si vous avez déjà acheté votre ordinateur et s'il n'est pas compatible avec celui de l'école, ce n'est pas très grave, l'enfant saura quand même en tirer profit.

Il est normal qu'il y ait plus de programmes éducatifs offerts pour les ordinateurs les plus populaires, mais certains constructeurs ont déjà

ordinateur 16 bits et ses programmes écrits en langage machine sont probablement meilleurs que ceux qui sont écrits pour les ordinateurs 8 bits actuellement disponibles pour le marché grand public.

Cela est prouvé par des programmes de TI comme « Grammaire pour débutants », « Addition et soustraction » (1 et 2) et « Nombre magique ». Ces programmes sont stockés dans des cartouches qui peuvent être enfichées facilement par vos enfants dans le TI-99/4A. Et si vous connaissez le synthétiseur vocal du 99/4A, vous devez savoir que plusieurs programmes parlent avec une voix très agréable. Le problème est que certains programmes ne sont pas encore traduits en français. TI offre également une excellente version du langage LOGO, bien que cela tombe dans la seconde catégorie des outils d'apprentissage.

Il y a un très bon choix de programmes pour ordinateurs domestiques. Et plus l'enfant grandit, plus le nombre et la diversité des programmes deviennent importants. Les programmes pour le groupe de huit à onze ans sont de complexité et de qualité variables, et la plupart s'efforcent d'approfondir les techniques de base et de développer les aptitudes de l'enfant. La plupart des ordinateurs offrent également des programmes d'apprentissage de musique et de langues étrangères pour ce groupe d'âge.

Il existe une surabondance de programmes pour les étudiants du lycée. La meilleure façon d'en choisir un est de consulter l'instituteur de votre enfant. Il pourra vous indiquer les meilleurs programmes disponibles.

Une autre catégorie de programmes éducatifs s'adresse aux enfants de moins de treize ans. A cet âge, les enfants découvrent encore comment apprendre, et les programmes qui les incitent à utiliser l'ordinateur pour découvrir le monde seront des plus utiles. LOGO est le programme le plus connu, et une version est offerte sur les ordinateurs Atari, Tandy, Apple, Texas Instruments, Commodore et IBM. De nouvelles versions sont promises pour les ordinateurs BBC et Sinclair. Grâce à ce programme, un enfant de six à douze ans peut explorer le potentiel graphique d'un ordinateur et apprendre la géométrie avec une « tortue ». L'enfant découvre comment écrire des procédures (programmes) pour la tortue et, avec certaines versions, il progresse jusqu'à ce qu'il puisse dessiner un ensemble de « plans-objets » à l'écran. A l'aide de ce programme, l'enfant apprend de façon autonome les règles élémentaires des mathématiques et découvre la notion d'espace.

Il n'est pas facile de sélectionner un bon programme éducatif en raison du très grand choix offert. Nous vous recommandons de vous rendre aux foires informatiques organisées un peu partout. Là, vous pourrez essayer et comparer divers programmes. La pénurie de programmes excellents sur les plans pédagogique et informatique ne devrait pas être de longue durée. Chaque mois, de nouveaux programmes font leur apparition et peuvent aider au développement des capacités intellectuelles de votre enfant.

Les jeux d'arcades, qui ne semblent pas avoir un potentiel pédagogique évident, jouent néanmoins un rôle non négligeable en dédramatisant le monde de l'informatique aux yeux des enfants. En effet, le jeu est la meilleure introduction possible à la programmation, en particulier si le programme implique des échanges constants entre le joueur et la machine, à l'instar des jeux de rôle ou de simulation militaire. Une fois le stade passif dépassé, les enfants sont souvent tentés par la rédaction de leurs propres programmes. Certains lycées, comme celui de Pont-l'Abbé, en Bretagne, encouragent les élèves à le faire, et les meilleurs résultats sont publiés dans la presse spécialisée. Cette relation directe entre l'enfant et l'informatique est indispensable à l'efficacité d'un « professeur » électronique.

## L'agent secret

**YOU ARE AN AGENT  
FOR MI 6**

**YOUR MISSION IS TO  
CATCH A NOTORIOUS SPY**

**HIS AIM IS TO ELIMINATE  
YOUR AGENTS IN  
EACH CITY**

Il y a plusieurs jeux éducatifs offerts actuellement. Celui-ci se nomme « l'Agent secret » et est publié par Heinemann. Votre mission est de capturer un célèbre espion ennemi avant qu'il n'ait éliminé tous vos agents. Pour le capturer, vous devez déchiffrer les indices donnés par ces divers déplacements. La chasse est ouverte...

L'espion ne peut être arrêté que dans une ville, et il ne restera pas au même endroit plus d'une heure ou deux. Lorsque vous pensez savoir où il se trouve, vous pouvez choisir de vous y rendre en train ou en avion. Vous devez décider si la vitesse de l'avion justifie le coût additionnel.

COST 80	DEPARTURES FROM:					
	AIR			RAIL		
DAY 00	DEP.	ARR.	DEST.	DEP.	ARR.	DEST.
TIME 01.58	07.30	09.10	BER	09.43	03.06	VIE
	09.00	11.00	OSL	10.15	22.00	BER
	10.40	15.20	MOS	10.48	02.30	OSL
	11.20	15.50	IST	10.54	16.58	BON
	14.00	16.10	MUN	11.23	15.49	PAR
	16.20	18.50	VIE	12.00	03.43	TRI
	18.00	20.10	OSL	12.09	05.28	MAD
	19.30	23.00	MOS	15.49	11.11	ROM
	19.30	23.20	IST	16.44	16.43	BEL
	19.30	21.30	BER	17.23	11.54	BUD
	20.30	00.00	ROM	17.54	21.26	PAR
	21.00	00.30	MAD	19.55	07.27	BER
	22.30	23.50	PAR	21.31	03.16	BON

**INTELLIGENCE REPORT**  
AGENT OBSERVED SPY IN CITY FROM WHICH  
GRAIN IS TRANSPORTED 47 HOURS AGO



Un voyant clignote à l'écran quand vous recevez un message d'un de vos agents. Si votre agent est éliminé, le message sera intercepté avant que vous ne l'ayez reçu. Vous pouvez engager un autre agent, mais vous devrez le payer.

Vos informateurs vous envoient des rapports mais ceux-ci sont codés, vous devez donc engager des chercheurs. A la fin du jeu, vous connaissez les noms et les emplacements des villes européennes, vous avez compris comment fonctionnent les fuseaux horaires et vous savez ce que gérer un budget signifie. (Cl. Ian McKinnell.)

**IT TOOK 27 DAYS 13 HOURS  
TO CATCH THE SPY**

**IT COST YOU £7703**

**You are dismissed from MI 6**



# Percer le code

**Lorsque vous exécutez un programme écrit dans un langage informatique évolué, un programme interne le traduit instantanément dans le langage machine de l'ordinateur.**

Même si de nombreux ordinateurs semblent effectuer plusieurs fonctions similaires, chaque modèle est unique. Certains fournissent des programmes intégrés, d'autres doivent lire des programmes sur disquettes ou sur cassettes.

Certaines machines disposent d'un programme global qui permet l'entrée de programmes ainsi que l'utilisation d'instructions comme SAVE ou LOAD. D'autres modèles ont besoin de plusieurs programmes distincts pour exécuter ces fonctions.

Cependant, la plupart des ordinateurs fonctionnent selon des principes similaires. Le mouvement de l'information, vers et en provenance des périphériques de stockage externe, est dans tous les cas commandé à partir du clavier. De plus chaque machine peut communiquer avec d'autres périphériques externes comme des imprimantes ou des tables traçantes.

Dans tout ordinateur, trois programmes sont logés simultanément en mémoire : le système d'exploitation, l'interpréteur BASIC et votre programme.

Lorsque vous exécutez votre programme, les trois programmes semblent travailler simultanément. Chaque instruction BASIC de votre programme est traduite par un interpréteur et, une par une, les instructions résultantes en code machine sont transmises au microprocesseur pour y être traitées. Au même moment, le système d'exploitation surveille le clavier et affiche éventuellement ce qui y est tapé.

Si par exemple l'une des instructions de votre programme demande une impression ou une écriture sur disquette, l'interpréteur demandera au système d'exploitation d'accomplir cette tâche.

Cette illusion de travaux simultanés provient de la vitesse incroyable du microprocesseur. Il peut traiter des instructions venant du système d'exploitation et de l'interpréteur à une telle vitesse que cela semble se produire en même temps.

Certaines machines travaillent encore plus rapidement parce que, lors d'une arrivée de données, elles « interrompent » le traitement normal. De cette façon, le système d'exploitation n'a pas à surveiller les sources de données externes comme le clavier ou les lecteurs de disquettes.

Un « éditeur » est un programme de traitement de texte peu évolué. Les éditeurs sont de qualité variable et vous constaterez probablement que l'éditeur de BASIC est aussi bon.

Au lieu d'un interpréteur, vous pouvez utiliser un compilateur pour exécuter vos programmes. Alors que l'interpréteur doit traduire une par une toutes les instructions qu'il rencontre et cela lors de chaque exécution, le compilateur traduit de façon globale et définitive votre programme en code machine. Les programmes compilés sont exécutés plus rapidement que les programmes interprétés.

BASIC est un langage très populaire. Il s'adresse particulièrement aux débutants. Mais les programmeurs plus expérimentés peuvent créer des programmes plus rapides à l'aide d'un assembleur. Ce langage ne ressemble pas du tout à l'anglais et le programmeur doit connaître assez bien le fonctionnement du microprocesseur.

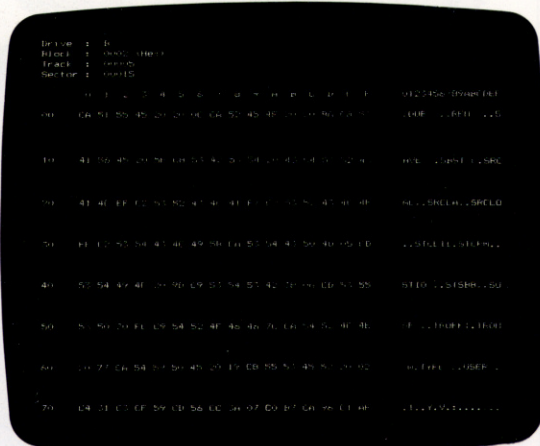
En assembleur, chaque instruction que vous donnez à l'ordinateur a un équivalent direct en code machine. Un langage d'assemblage est composé d'une série d'instructions abrégées comme MVI (*MoVe Immediate*) ou JZ (*Jump on Zero*). Ces abréviations rappellent au programmeur les fonctions associées.

Si vous maîtrisez un assembleur, il ne vous reste plus qu'à apprendre à programmer en code machine, mais cela n'est pas vraiment justifié sauf si vous désirez accélérer l'exécution du programme de quelques fractions de seconde, performance en général peu indispensable.

Le code machine des micro-ordinateurs est généralement écrit en hexadécimal. C'est un système de numérotation en base 16. On compte normalement de 0 à 9, puis on utilise les lettres A à F pour les chiffres 10 à 15. Chaque adresse de mémoire correspond à huit chiffres binaires (bits) et peut être représentée par une paire de nombres hexadécimaux.

## Système d'exploitation de disquettes

Lorsque des programmes sont stockés sur disquettes, les informations sont stockées de façon aléatoire sur la surface de la disquette. Le système d'exploitation de disquettes est un programme qui retrouve l'emplacement des données. Cette illustration montre l'information stockée sur une petite section de la disquette. Cette information est représentée en hexadécimal dans le bloc de gauche et les caractères équivalents sont illustrés à droite. Les codes qui ne correspondent pas à des caractères devant être affichés sont représentés par des points.





Par exemple, le nombre binaire 01011101 est divisé en deux parties : 0101 et 1101. Celles-ci sont traduites en deux nombres décimaux 5 et 13, ce qui correspond à 5 et D en hexadécimal. Par conséquent 01011101 sera désigné par 5D lors d'une programmation en code machine. Utiliser le code machine est certainement la façon la plus lente d'écrire un programme, mais elle donne probablement les temps d'exécution les plus rapides. Voici quelques extraits de programme :

```
BASIC
100 INPUT « Entrez les heures de travail »; HEURES
200 PAY = HEURES * TAUX
```

La première ligne affiche un message à l'écran qui invite l'utilisateur à entrer le nombre d'heures de travail. Si le programme accepte l'entrée, celle-là est multipliée par un taux horaire (défini plus tôt dans le programme) pour donner le montant de la somme à payer.

```
Langage d'assemblage
MVL C, 01
CALL 05
```

La première instruction place la valeur 1 dans une zone de la mémoire du microprocesseur appelée le registre C. La seconde instruction demande au système d'exploitation de prendre la main, le système d'exploitation repasse alors la main à votre programme. Voici une traduction directe de ces lignes d'assembleur :

```
Code machine
0E01
CD0500
```

Une traduction en code machine des deux instructions BASIC comprendrait de nombreuses commandes. Il est évident qu'il est préférable de confier ce travail à un interpréteur ou à un compilateur plutôt que d'écrire un langage d'assemblage ou en code machine.

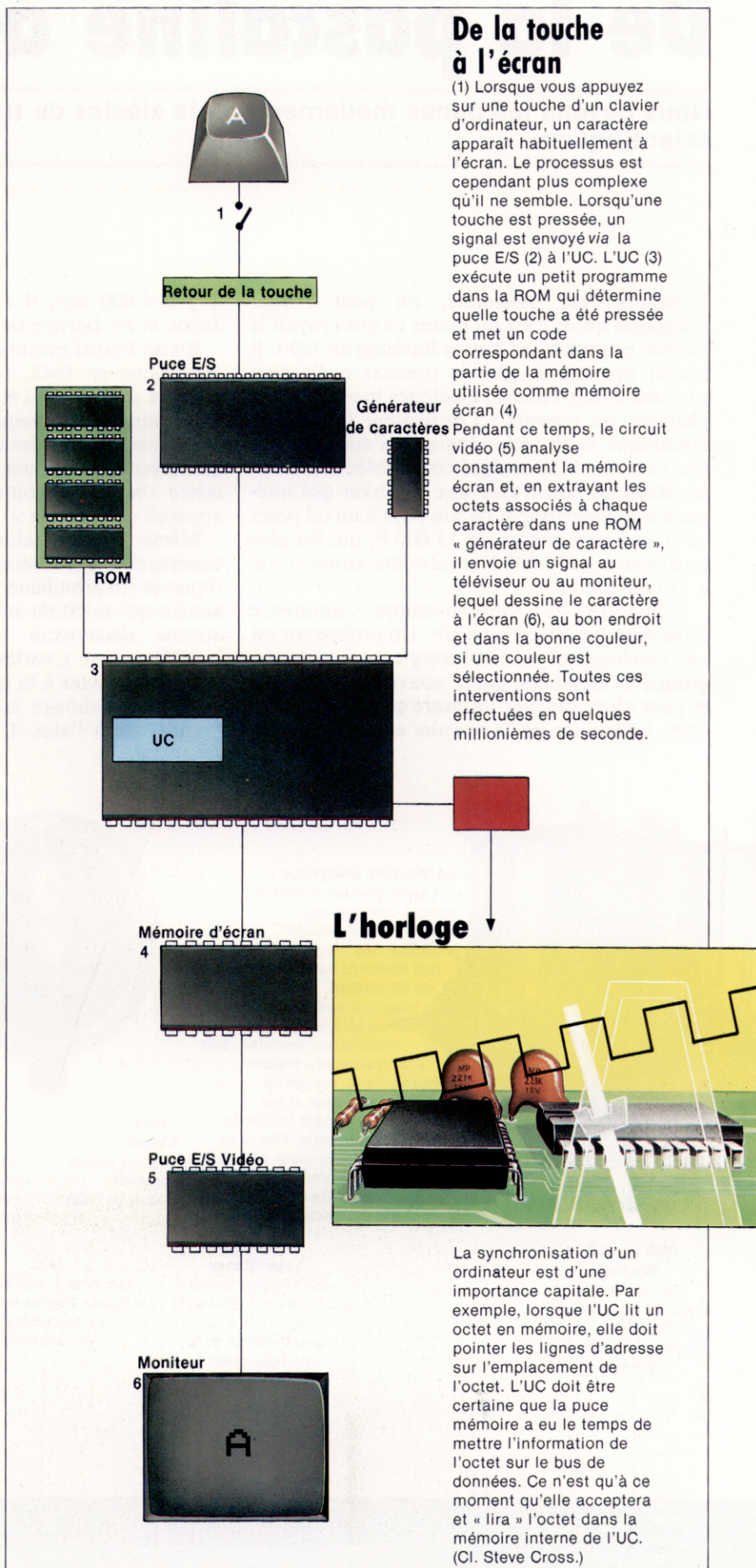
Certaines commandes du système d'exploitation sont presque aussi obscures que des programmes d'assembleur. Voici une commande de CP/M (*Control Program for Microcomputers*).

```
DIR *.BAS
```

Cette commande signifie « lister tous les fichiers dont le suffixe est BAS sur l'unité active de disquettes ». DIR est l'abréviation de *Directory* (répertoire).

Malgré ces commandes curieuses, CP/M est installé sur plus d'un million de machines. Il offre de précieux avantages aux programmeurs professionnels. Les programmes peuvent être transférés facilement entre diverses machines, à condition qu'ils confient à CP/M la gestion des lecteurs de disquettes, du clavier, de l'imprimante et de l'écran.

En choisissant CP/M, les constructeurs qui introduisent une nouvelle machine bénéficient des logiciels existants, et les consommateurs peuvent acheter des machines sans craindre de ne pouvoir trouver les logiciels dont ils ont besoin.



### De la touche à l'écran

(1) Lorsque vous appuyez sur une touche d'un clavier d'ordinateur, un caractère apparaît habituellement à l'écran. Le processus est cependant plus complexe qu'il ne semble. Lorsqu'une touche est pressée, un signal est envoyé via la puce E/S (2) à l'UC. L'UC (3) exécute un petit programme dans la ROM qui détermine quelle touche a été pressée et met un code correspondant dans la partie de la mémoire utilisée comme mémoire écran (4). Pendant ce temps, le circuit vidéo (5) analyse constamment la mémoire écran et, en extrayant les octets associés à chaque caractère dans une ROM « générateur de caractère », il envoie un signal au téléviseur ou au moniteur, lequel dessine le caractère à l'écran (6), au bon endroit et dans la bonne couleur, si une couleur est sélectionnée. Toutes ces interventions sont effectuées en quelques millièmes de seconde.

### L'horloge

La synchronisation d'un ordinateur est d'une importance capitale. Par exemple, lorsque l'UC lit un octet en mémoire, elle doit pointer les lignes d'adresse sur l'emplacement de l'octet. L'UC doit être certaine que la puce mémoire a eu le temps de mettre l'information de l'octet sur le bus de données. Ce n'est qu'à ce moment qu'elle acceptera et « lira » l'octet dans la mémoire interne de l'UC. (Cl. Steve Cross.)



# De la pascaline à l'Apple

Nous devons les puces modernes à trois siècles de travaux scientifiques.

Lorsqu'on est passionné, on peut réussir n'importe quoi ; c'est du moins ce que croyait le citoyen britannique Charles Babbage en 1830. Il réussit presque à créer le premier ordinateur cent ans avant l'heure. Quelques inconvénients gênèrent ses travaux : l'ordinateur devait être mécanique et la technologie de l'époque était peu évoluée. Mais malgré ces problèmes, Charles Babbage construisit une machine qui intéressa le gouvernement de son pays à un tel point qu'il reçut une somme de 15 000 F, qui fut plus tard portée à 170 000 F (équivalant aujourd'hui à 17 000 000 F).

L'histoire de l'informatique commence cependant beaucoup plus tôt. Un ordinateur est une machine qui peut effectuer un calcul sur un groupe de nombres, qui se souvient du résultat et peut alors calculer un autre groupe de nombres. Le boulier fut le premier exemple. Utilisé

il y a 4 000 ans, il est toujours employé au Japon et en Europe de l'Est.

Blaise Pascal inventa la première calculatrice mécanique en 1642, probablement pour faire plaisir à son père qui était inspecteur fiscal. Elle fonctionnait parfaitement, transférant les retenues de la colonne des unités à celle des dizaines au moyen d'un cliquet, un peu comme un odomètre dans une voiture. Pascal nomma cet appareil « pascaline ».

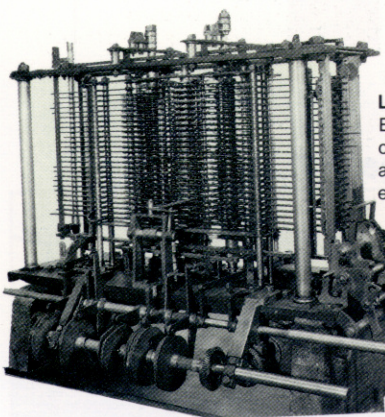
Même si la pascaline ne fut pas un succès commercial, elle intéressa beaucoup les scientifiques et fut grandement améliorée au cours des années qui suivirent son invention. Cependant, aucune découverte significative n'eut lieu jusqu'à ce que Charles Babbage et Ada Lovelace s'intéressent à la question.

Charles Babbage est né en 1791 dans une famille très à l'aise. Il était très doué pour les

### Blaise Pascal

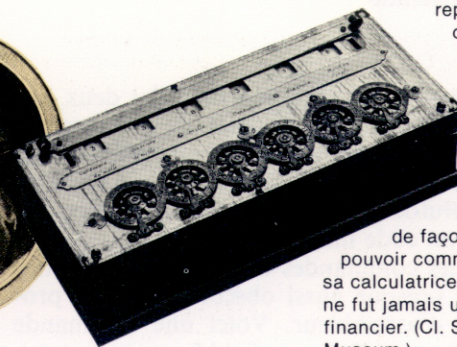
La pascaline fut la première calculatrice mécanique ; elle fut conçue par Blaise Pascal en 1642. Au début, il désirait concevoir une machine qui pourrait, en plus d'additionner et de soustraire, diviser et multiplier. Les roues étaient positionnées à l'aide d'une tige pointue et un mécanisme spécial

reportait les chiffres d'une colonne à l'autre. Pascal obtint du roi de France un brevet d'invention de façon à pouvoir commercialiser sa calculatrice, mais elle ne fut jamais un succès financier. (Cl. Science Museum.)



### La machine analytique

En 1834, Charles Babbage conçut une machine analytique qui pouvait effectuer des calculs avec des nombres comptant plus de 80 chiffres. Cette machine avait plusieurs des caractéristiques de l'ordinateur moderne. Les « programmes » étaient transmis au moyen de cartes perforées et les résultats étaient imprimés automatiquement. Elle avait également une unité de traitement arithmétique et des dispositifs de stockage indépendants.



1623

### Binaire

Francis Bacon utilisait le premier l'arithmétique en base

1671

### Calculatrice

Gottfried Leibnitz réussit à construire une machine qui pouvait multiplier et diviser

1822

### Machine « à différence »

La première calculatrice mathématique de Charles Babbage

1802

### Cartes perforées

Joseph Jacquard construisit un métier à tisser qui mémorisait les motifs du tissu sur des cartes perforées

## Compte à rebours

1000 av. J.-C.

### Boulier

Encore utilisé aujourd'hui, il calcule et mémorise des nombres à l'aide de boules

1614

### Logarithmes

John Napier inventa cet outil qui ramena la division à la soustraction et la multiplication à l'addition

1642

### Machine à additionner

Blaise Pascal inventa sa calculatrice, la pascaline

1000 av. J.-C.

1614

1623

1642

1671

1802

1822





mathématiques et découvrit de nombreuses erreurs dans les tables logarithmiques. Probablement lassé par tous ces calculs, il décida d'entreprendre la construction d'une machine qui pourrait effectuer des calculs complexes.

En 1822, il présenta à la Royal Astronomical Society son premier modèle d'une machine « à différence »; cette machine pouvait construire des tables logarithmiques. Le nom de cette machine vient d'une technique mathématique abstraite nommée « méthode des différences ». Grâce à l'aide de cette société, il put poursuivre ses travaux.

Avec Ada Lovelace, la fille de lord Byron, il prépara le projet plus ambitieux de construction d'une « machine analytique ». Cette machine devait calculer les valeurs de fonctions mathématiques beaucoup plus complexes que les fonctions logarithmiques.

Il dut affronter des problèmes dès le début. La machine ne fonctionnait tout simplement pas. Les plans qui nous sont parvenus montrent que la construction était immense et remplissait le grand atelier de Babbage. Les centaines de pièces devaient être spécialement tournées et la technologie métallurgique de l'époque n'était pas à la hauteur. Sur le modèle miniature qu'il construisit, de légères imprécisions pouvaient être corrigées, mais lorsqu'il essaya de faire fonctionner la véritable machine, ces légères imprécisions devinrent énormes.

Babbage était sur la bonne voie, et s'il avait disposé de meilleures techniques de fabrication,

sa machine analytique aurait probablement fonctionné. La majeure partie de l'architecture logique des ordinateurs modernes s'inspire des travaux de Babbage et il est aujourd'hui considéré comme le père de l'ordinateur.

Pendant ses travaux, Babbage eut l'intuition qu'il était possible de programmer une machine ou de lui montrer comment effectuer une tâche mathématique particulière. S'il avait été capable de le prouver et de construire une telle machine, l'époque victorienne en aurait sûrement profité.

Ce n'est qu'en 1936 que l'intuition de Babbage fut confirmée. La démonstration parut dans un journal presque inconnu, *On Computable Numbers*, publié par un mathématicien de Cambridge, Alan Turing. Turing est inconnu du grand public, mais sa contribution aux travaux qui rendirent possible la conception d'un ordinateur fut essentielle. Les scientifiques savaient depuis longtemps que les mathématiques sont une science qui obéit à des règles logiques, et que si on communiquait ces règles à une machine et si on lui proposait un problème, la machine serait en mesure de le résoudre. Cependant, tous les efforts des plus éminents mathématiciens n'avaient pas réussi à créer une telle machine. Turing décida d'aborder le problème différemment. Il examina le type de problèmes qu'une machine pourrait résoudre en

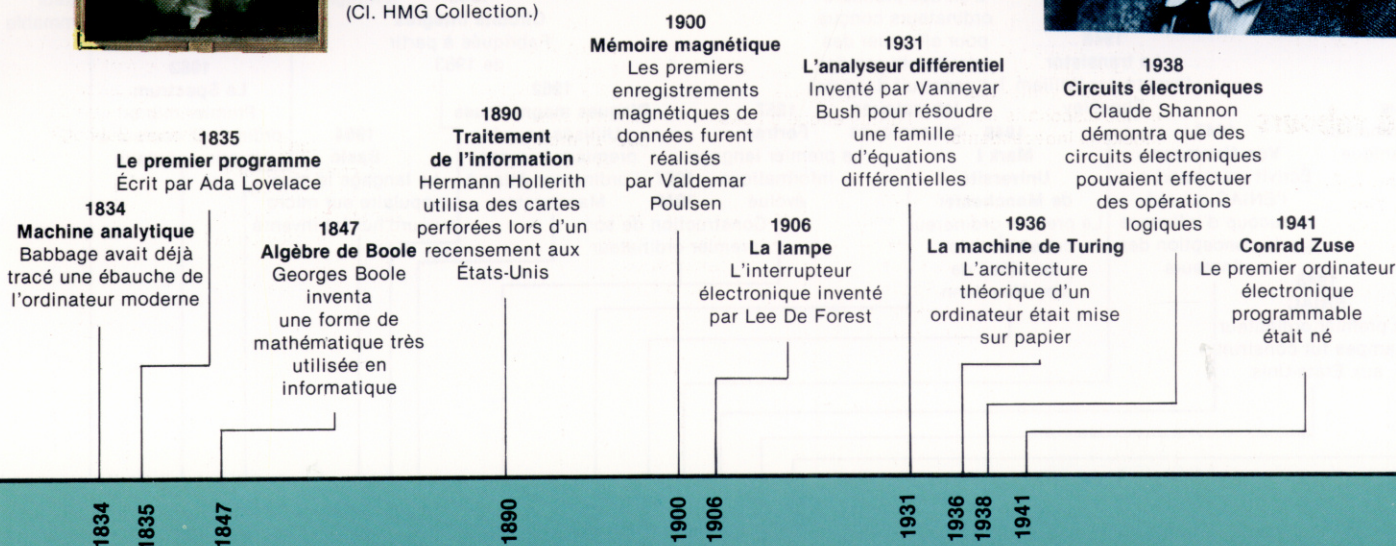
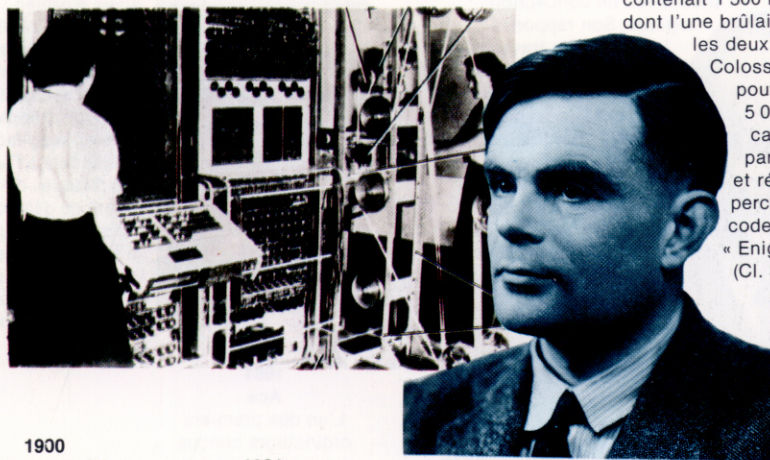
**Alan Turing et Colossus**

Alan Turing prouva qu'un ensemble d'instructions simples pouvait résoudre tout problème complexe. Lui et son équipe créèrent Colossus, l'un des premiers ordinateurs au monde, qui fonctionna pendant la Seconde Guerre mondiale. Cette énorme machine contenait 1 500 lampes, dont l'une brûlait toutes les deux minutes.

Colossus pouvait traiter 5 000 caractères par seconde et réussit à percer le code allemand « Enigma ». (Cl. Science Museum.)



**Le premier programmeur**  
La comtesse Ada Lovelace, femme de Charles Babbage et fille de lord Byron, est l'une des rares femmes à figurer dans l'histoire de l'informatique. Mathématicienne douée, elle écrit de nombreux rapports concernant les travaux de son mari. Elle écrit même des programmes pour ses machines, ce qui fit d'elle le premier programmeur. (Cl. HMG Collection.)





suivant des règles logiques et les énuméra. Cette approche originale lui permit de déduire certaines conclusions qui furent déterminantes dans la poursuite des travaux.

Turing dirigea à Buckinghamshire une équipe de recherche qui travailla au projet le plus secret de la Seconde Guerre mondiale, le premier ordinateur électromécanique. Cette machine déchiffra pendant la guerre les messages codés allemands « Enigma ».

Après la guerre, Turing se rendit aux États-Unis et apporta son concours à la création du premier ordinateur américain. Cet ordinateur fut nommé ENIAC et fut développé à l'université de Pennsylvanie. Il utilisait dix-huit mille lampes; l'une de celles-ci brûlait toutes les deux minutes!

Le nom de Turing est presque inconnu parce qu'il travailla pour MI6; ses travaux furent gardés secrets. Le gouvernement britannique ne révéla leur contenu qu'en 1975.

L'ordinateur progressa, mais on dut attendre l'invention du transistor en 1947 pour rendre possibles des calculs rapides.

Les transistors ont la même fonction que les lampes mais ils sont beaucoup plus rapides, plus

fiables et dégagent beaucoup moins de chaleur. Tout comme les lampes, ce sont des circuits électroniques qui peuvent être ouverts ou fermés et qui peuvent servir à représenter les 0 ou les 1 du code binaire. Au cours des années cinquante et au début des années soixante, des ordinateurs plus gros et plus rapides furent construits et utilisés par de grandes entreprises et par les gouvernements.

Au milieu des années soixante, il vint à l'esprit de certains chercheurs qu'un circuit électronique miniaturisé fonctionnerait tout aussi bien. Les recherches étaient financées par d'énormes capitaux. Les laboratoires commencent à tracer des circuits sur des puces de silicium et à les graver. Avant la fin des années soixante, le « circuit intégré » fit son apparition et l'informatique fit un grand bond en avant.

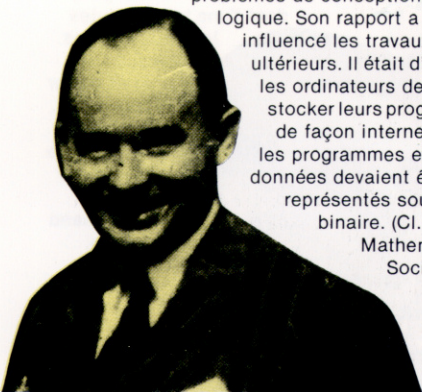
Le circuit intégré mena naturellement à la création de puces comportant de nombreux circuits, et la superposition de plusieurs puces donna le microprocesseur.

Même si le rapport entre les puces et la machine géante de Babbage et d'Ada Lovelace n'est pas évident, l'« architecture » créée par Babbage est toujours utilisée dans le microprocesseur d'aujourd'hui. La théorie qui a permis la conception d'un ordinateur et la preuve qu'apporta Turing de la faisabilité d'un ordinateur ne sont pas dépassées.

**Le Sinclair Spectrum**  
Créé moins de quarante ans après les premiers ordinateurs qui occupaient des salles entières, le Spectrum est petit, compact et peu coûteux. Le Spectrum fut le premier ordinateur offrant un affichage couleur qui coûtait moins de 1 500 F. Les premiers ordinateurs étaient des machines énormes et complexes utilisées par les gouvernements et les grandes sociétés. Aujourd'hui, grâce à des machines comme le Spectrum, la plupart des familles peuvent s'offrir un ordinateur individuel.

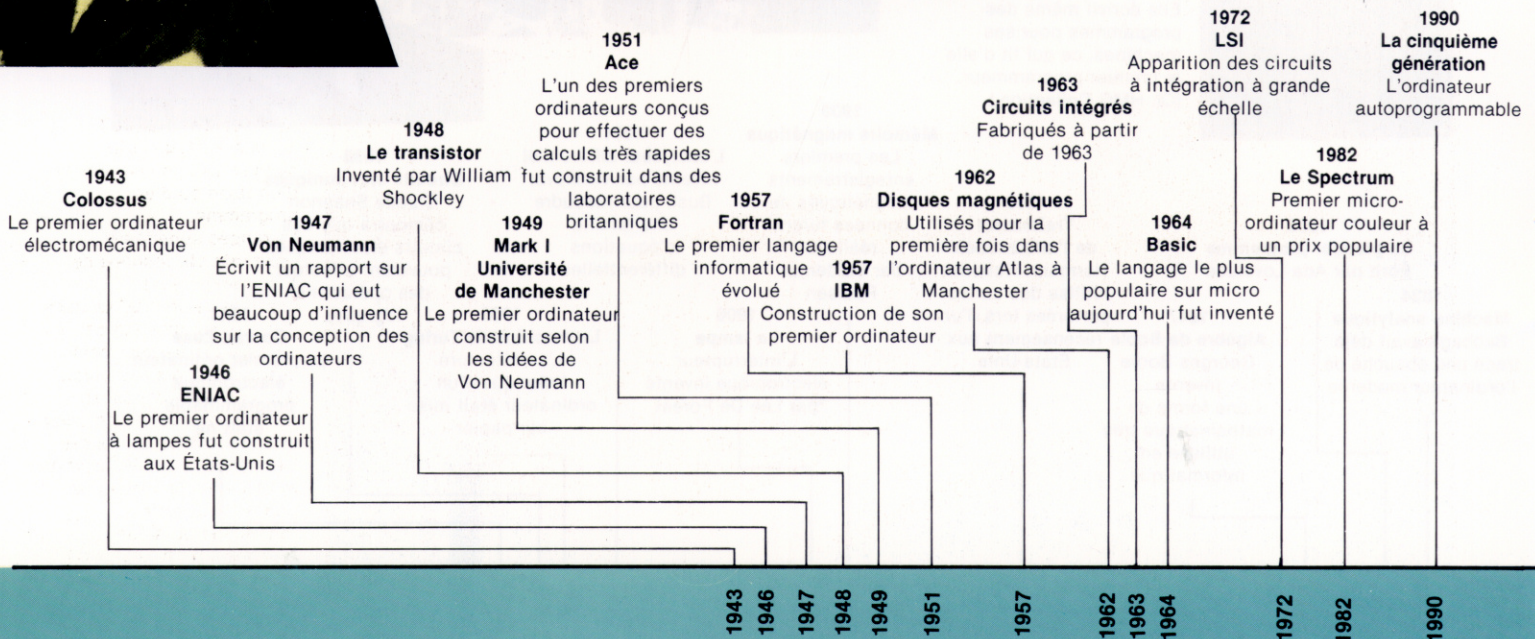
**L'« architecte » de l'ordinateur moderne**

John von Neumann apporta son concours dans le projet ENIAC pour résoudre les problèmes de conception logique. Son rapport a influencé les travaux ultérieurs. Il était d'avis que les ordinateurs devaient stocker leurs programmes de façon interne et que les programmes et les données devaient être représentés sous forme binaire. (Cl. London Mathematical Society.)



**Steve Wozniak**

La réussite de Steve Wozniak est l'exemple même des possibilités offertes par l'industrie informatique. Photographié ici avec sa première invention, l'Apple I (maintenant pièce de musée), Wozniak concevait des ordinateurs à l'école. Même s'il ne suivit pas d'études d'ingénieur, son travail permit de démocratiser l'usage de l'ordinateur. L'Apple II, créé en 1976, fut le premier ordinateur individuel accompagné d'un lecteur de disquettes et fait toujours partie des grands de l'informatique individuelle.





# BBC modèle B

**D'excellentes caractéristiques techniques appuyées par la réputation de la BBC ont rendu cet ordinateur très populaire en Grande-Bretagne. Il est spécialement conçu pour les tâches pédagogiques.**

Le micro-ordinateur BBC modèle B est fabriqué par Acorn Computers à Cambridge et son marketing est assuré par la BBC. Deux modèles furent d'abord proposés, mais le modèle A, moins coûteux et moins évolué, est maintenant retiré du marché. Un grand nombre de programmes ont été développés pour cette machine, à partir des langages de programmation.

Les caractéristiques techniques du BBC sont toujours compétitives malgré le nombre important de nouvelles machines introduites depuis son arrivée sur le marché. Le langage BASIC du BBC est particulièrement bien adapté pour gérer toutes les fonctions spéciales de la machine. Il permet également de développer et d'éditer facilement des programmes.

Il y a huit modes graphiques différents. Cela signifie que l'utilisateur peut choisir une faible, une moyenne ou une haute résolution; cependant, le nombre de couleurs est limité dans le dernier cas. La résolution maximale est de  $640 \times 256$ . La majorité des utilisateurs optent pour un téléviseur, mais un moniteur est proposé pour profiter au maximum du potentiel graphique de

la machine. On retrouve des commandes pour tracer des lignes, des cercles et pour construire des images à l'écran.

L'alimentation se trouve à l'intérieur du boîtier et il y a plus d'interfaces que sur la plupart des machines, ce qui signifie qu'un plus grand nombre de périphériques peuvent être connectés, y compris plusieurs marques de lecteurs de disquettes autres que celui qui a été conçu spécialement pour le BBC.

En plus des interfaces pour lecteur de disquettes, pour imprimante, ou pour dispositif analogique comme un appareil de laboratoire ou un instrument de mesure, il est possible de composer un réseau. Cela est idéal dans une classe, puisque plusieurs utilisateurs peuvent ainsi partager une imprimante ou un lecteur de disquettes.

Mentionnons finalement le « Tube », une interface complexe qui permet de connecter un autre microprocesseur, soit pour accélérer l'exécution, soit pour exécuter des programmes écrits pour d'autres machines. Cependant, peu d'utilisateurs semblent tirer avantage de cette caractéristique.

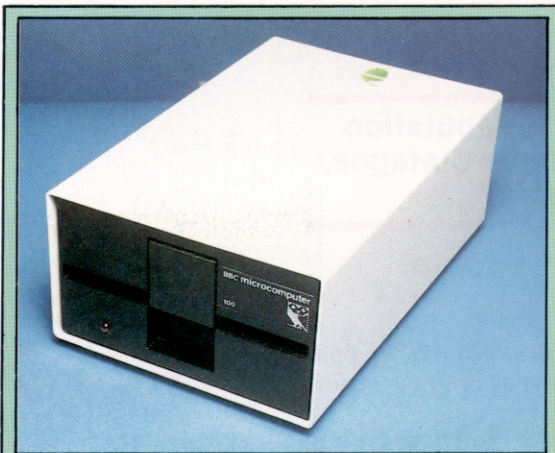


## Le clavier du BBC modèle B

Le clavier est l'un des points forts du BBC modèle B, autant au niveau de la disposition des touches que de la qualité de construction. Les touches sont agréables et permettent une frappe rapide.

Les quatre touches fléchées du côté droit servent à déplacer le curseur lors de l'édition d'un texte ou d'un programme.

La rangée supérieure comporte dix touches de fonction qui sont particulièrement utiles dans des programmes éducatifs puisque l'utilisateur peut choisir une réponse parmi dix réponses proposées. La présence de trois diodes électro-luminescentes est très agréable; celles-ci vous indiquent si le moteur de l'unité à cassette fonctionne ou si les touches « Shift Lock » ou « Capitals Lock » sont activées.



**Le lecteur de disquettes**

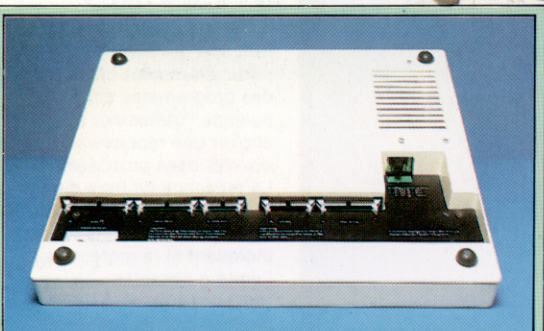
Le lecteur de disquettes pour le BBC est un périphérique très intéressant mais coûte environ 3 500 F et ne stocke que 100 Koctets. Cependant, d'autres constructeurs proposent des lecteurs de disquettes moins coûteux.

**Puces étrangères**

Si vous examinez soigneusement les divers circuits, vous verrez que la fabrication du BBC est une entreprise véritablement internationale. Le modèle B renferme des puces fabriquées en Malaisie, au Japon, au Portugal, en Écosse et aux États-Unis.

**Puce d'interface**

Des adaptateurs d'interface versatiles comme le MOS 6522 se chargent de l'interface avec les périphériques externes. Même si ces puces ne sont pas des processeurs, elles sont aussi complexes que le microprocesseur.



**Connecteurs de périphériques**

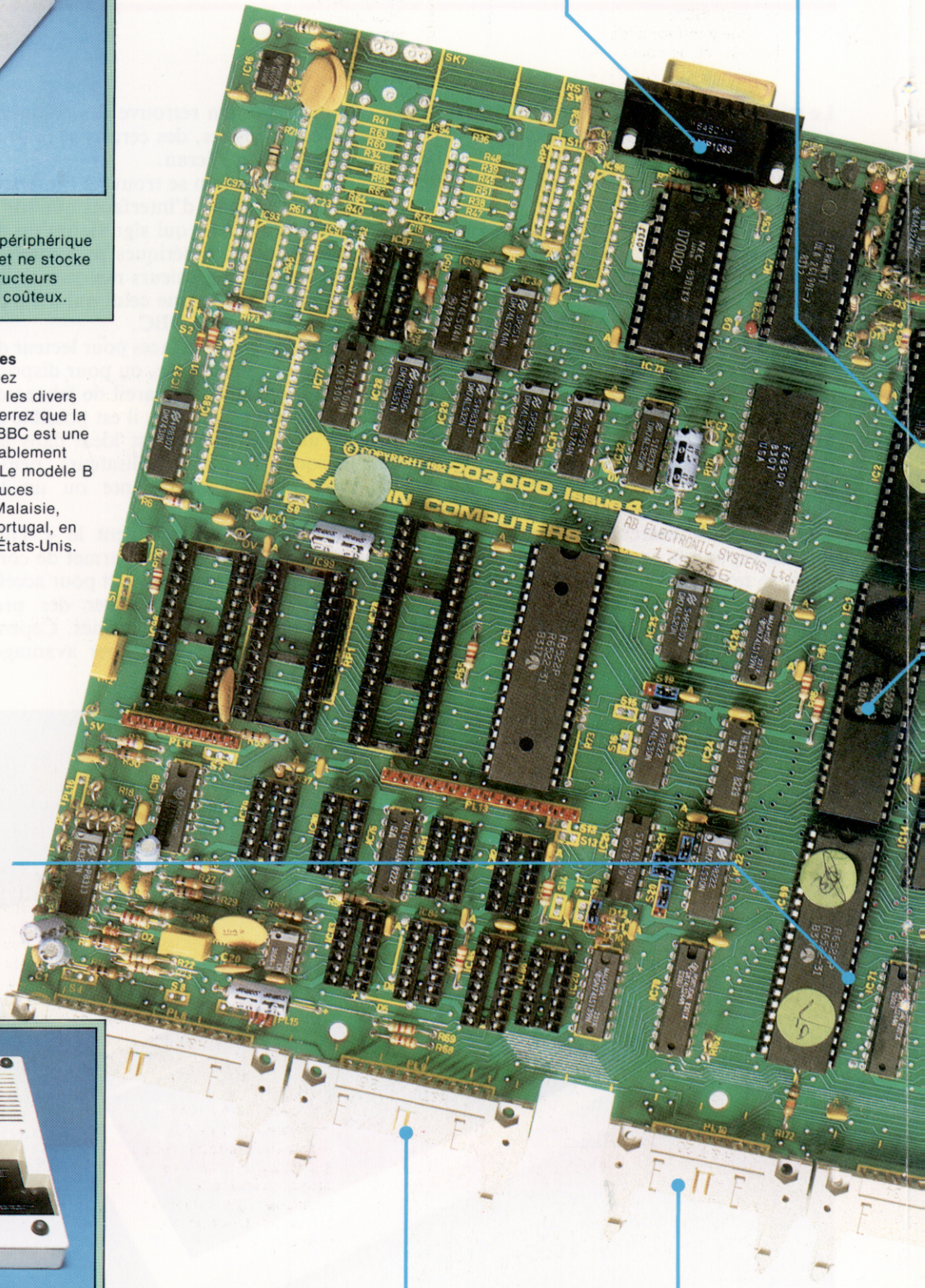
Photo illustrant les connecteurs pour périphériques situés sous la machine.

**Entrée analogique**

Cela permet à l'ordinateur de lire une tension provenant d'un instrument non numérique comme un capteur thermique. Elle est surtout utilisée dans des travaux de laboratoire.

**Contrôleur vidéo**

Cette puce reçoit l'information de la mémoire utilisateur et la convertit en signaux vidéo afin de l'afficher.



**Port imprimante**

C'est ici qu'est branchée l'imprimante parallèle.

**Port utilisateur**

Pour des expériences avec des dispositifs numériques ou avec des circuits logiques conçus par l'utilisateur.

**Interface cassette**

Les programmes peuvent être sauvegardés avec une enregistreuse à cassette de deux façons : à vitesse maximale, ou avec une fiabilité maximale.

**Sortie RVB**

Fournit des signaux distincts pour les composants rouges, verts et bleus du signal vidéo. Cela permet un affichage vidéo d'une excellente qualité.

**Sortie téléviseur**

Elle est reliée à la prise d'antenne d'un téléviseur.

**Port RS232**

Une interface série très rapide pour utiliser un très grand nombre de périphériques.

**Sortie vidéo**

Pour utiliser un écran monochrome.

**Modulateur**

Reçoit le signal couleur venant du contrôleur vidéo et le convertit en une sortie adéquate pour un téléviseur.

**Quartz**

Un quartz qui constitue le cœur de l'horloge et synchronise toutes les opérations.

**Microprocesseur**

Le microprocesseur 6502 se charge de tout le traitement.

**Mémoire utilisateur**

Le BBC modèle B possède 32 K de RAM pour stocker les programmes, les données et les affichages graphiques.

**ULA**

Ce tableau logique non affecté fait le travail de douzaines de circuits utilisés dans d'autres ordinateurs. La pièce métallique montée sur le dessus du composant sert de dissipateur thermique.

**ROM**

Ces deux puces ROM renferment le BASIC et le système d'exploitation, qui est un ensemble de programmes requis par l'ordinateur pour effectuer toutes ses fonctions de gestion interne.

**Le « tube »**

Une interface spéciale conçue par Acorn pour permettre au BBC modèle B de travailler avec d'autres microprocesseurs.

**BBC modèle B****PRIX**

★★★★★

**DIMENSIONS**

409 × 358 × 78 mm

**POIDS**

3 700 g

**UC**

6502A

**HORLOGE**

2 MHz

**MÉMOIRE**

32 Koctets de RAM  
32 Koctets de ROM contenant le BASIC et le système d'exploitation.

**AFFICHAGE VIDÉO**

8 modes graphiques différents.  
32 lignes de 80 caractères.  
La plus haute résolution :  
640 × 256 points. Jusqu'à  
16 couleurs en basse  
résolution.

**INTERFACES**

Téléviseur, moniteurs monochrome et couleur; unité de disquettes, imprimante, entrée analogique, port utilisateur.

**LANGAGE INTÉGRÉ**

BASIC.

**AUTRES LANGAGES DISPONIBLES**

LISP, FORTH, LOGO.

**ACCESSOIRES FOURNIS**

Prises pour lecteur de cassette et pour téléviseur. Guide de l'utilisateur. Casette et brochure de démonstration.

**CLAVIER**

De type machine à écrire avec 74 touches mécaniques, dont 10 touches programmables.

**DOCUMENTATION**

Le guide de l'utilisateur du micro-ordinateur BBC semble avoir été écrit par des gens qui connaissent très bien leur sujet et qui semblent penser que c'est également le cas pour leurs lecteurs. Plusieurs chapitres assez longs sont consacrés aux utilisations spécifiques des programmes système qui commandent l'excellent potentiel graphique, sonore et d'E/S de la machine.

# Portes et additions

Les nombres binaires, 1 et 0, peuvent être additionnés à l'aide des simples opérations logiques ET, OU et NON.

Nous avons vu dans l'article précédent comment des circuits transistorisés assez simples peuvent effectuer des opérations logiques comme ET, OU et NON. Ce qui est surprenant, c'est que ces mêmes « portes logiques » soient utilisées pour effectuer toutes les opérations arithmétiques de l'ordinateur. En logique électronique, les entrées de tension zéro représentent « faux », et les entrées de tension positive représentent « vrai ». L'absence de tension est généralement symbolisée par un zéro (0) et une tension positive par le chiffre un (1). Lorsque les portes logiques servent à effectuer des opérations arithmétiques, les mêmes zéros et uns sont utilisés, mais cette fois ils représentent les membres d'une addition binaire.

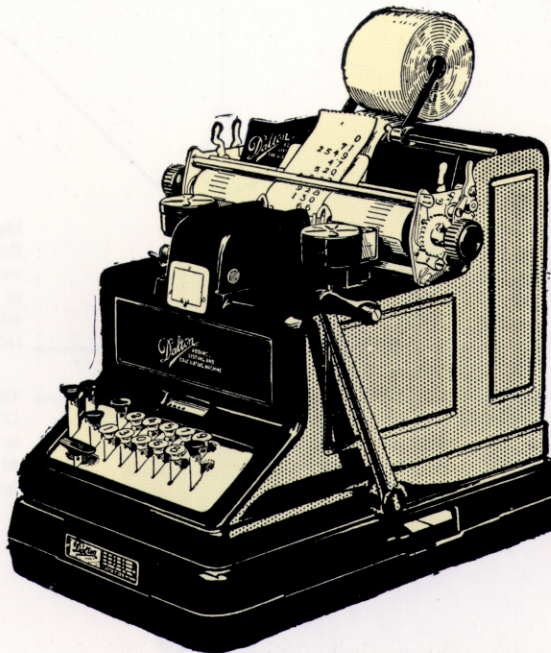
Lors d'une addition de deux chiffres binaires, il n'y a que deux entrées dans le circuit d'addition, et il ne peut y avoir plus de quatre combinaisons d'entrées,  $0 + 0$ ,  $0 + 1$ ,  $1 + 0$  et  $1 + 1$ . Dans notre étude de l'arithmétique binaire, nous avons appris que  $0 + 0 = 0$  (comme en arithmétique décimale). Nous savons aussi que  $0 + 1$  (ou  $1 + 0$ ) = 1 (encore comme en arithmétique décimale). Mais en arithmétique binaire,  $1 + 1 = 0$ . Voici l'illustration de ces quatre opérations.

x	y	z
0	0	= 0
0	1	= 1
1	0	= 1
1	1	= 10

Si nous utilisons une porte OU pour faire l'addition, nous obtiendrions une sortie fautive (0) si les deux entrées étaient fautes (0), et une sortie vraie si l'une des entrées était vraie ( $0 + 1$  ou  $1 + 0$ ).

Jusqu'ici, une simple porte OU semble parfaite pour additionner deux chiffres binaires. Mais attendez. Si les deux entrées sont vraies, la sortie d'une porte OU devrait elle aussi être vraie, ce qui donnerait une mauvaise réponse en arithmétique binaire. La réponse doit être 0 retenue de 1. Cette porte nous donne donc trois bonnes réponses sur quatre, mais trois sur quatre n'est pas suffisant.

Nous avons besoin d'un circuit qui nous donnera 0 si les deux entrées sont 0, 1 si l'une des deux entrées est 1 et l'autre 0, et 0 si les deux entrées sont 1. Cela est plus simple qu'il ne semble. Si nous avons deux portes ET recevant les deux mêmes entrées, mais dont l'une des entrées est préalablement inversée par une porte NON

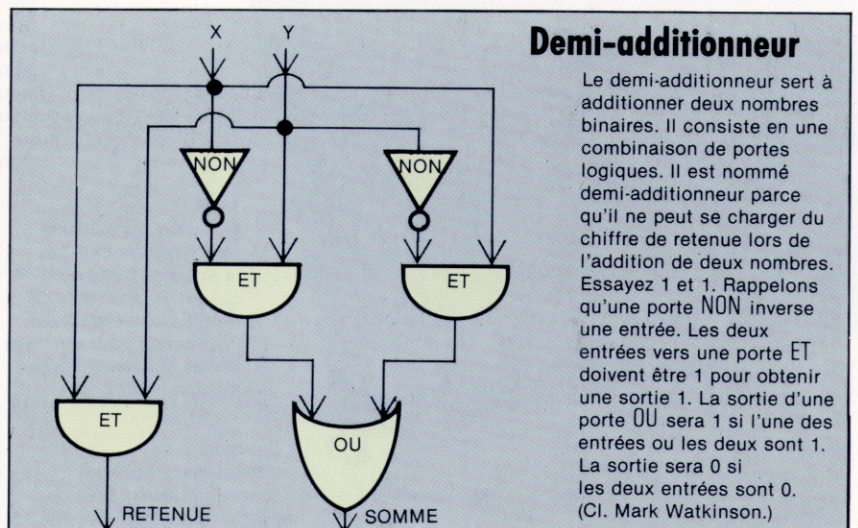


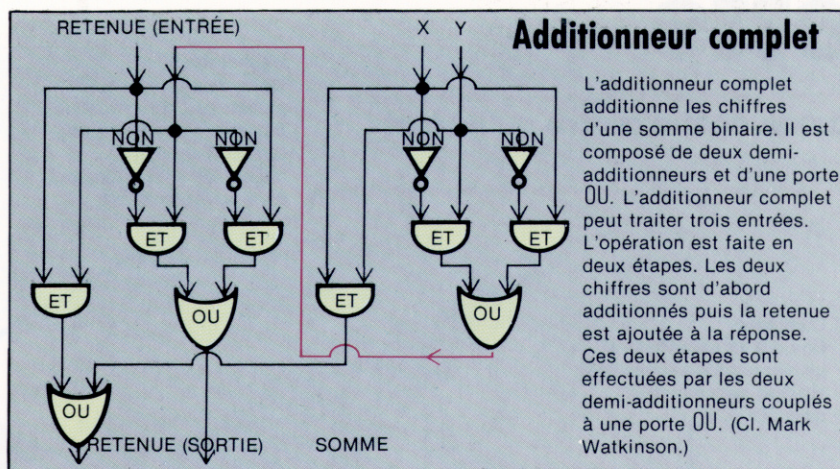
## La machine à calculer

Jusqu'à la récente apparition des calculatrices électroniques, la machine à calculer mécanique était d'usage courant dans les magasins et dans les bureaux. A l'exception de certains raffinements, elle est demeurée inchangée pendant trois cents ans. Le potentiel commercial d'une calculatrice apparut rapidement. Pascal inventa la première machine à calculer et l'offrit à son père qui était inspecteur des impôts. Lorsque Leibniz inventa une calculatrice pouvant diviser et multiplier, la calculatrice était prête à servir le monde des affaires.

vers l'une des portes ET et dont l'autre entrée est inversée vers l'autre porte ET (voir l'illustration), un 0 sur chaque entrée donnerait une sortie 0 sur les deux portes ET et un 1 sur les deux entrées donnerait une sortie 0 sur les deux portes ET. D'autre part, des entrées de 1 et de 0 donneront deux entrées vraies à l'une des portes ET. L'une des portes ET produira donc une sortie vraie. Si les deux sorties des portes ET sont connectées à une porte OU, la sortie de la porte OU ne sera vraie que si l'une des deux entrées est vraie.

Nous avons presque notre circuit, il nous manque encore la retenue. Mais si nous bran-





Examinons cette addition :

$$\begin{array}{r} 011 \\ + 111 \\ \hline 1010 \end{array}$$

Lorsque nous additionnons la colonne 1, nous disons  $1 + 1 = 0$  retenue de 1 et nous écrivons 0 sous la colonne 1. Lorsque nous additionnons la colonne 2, nous disons  $1 + 1 = 0$  + la retenue de la colonne 1 = 1 retenue de 1. Nous écrivons 1 sous la colonne 2 et nous retenons 1 pour la colonne 3. Ici  $0 + 1 = 1$  + la retenue de la colonne 2 = 0 retenue de 1. Nous écrivons 0 dans la colonne 3 et retenons 1 que nous reportons dans la colonne 4. Voici la table de vérité d'un additionneur complet pouvant traiter les retenues.

chons une autre porte ET en parallèle sur les deux entrées, nous obtenons notre signal pour la retenue uniquement lorsque les deux entrées sont vraies. Voici la table de vérité de notre circuit; on nomme ce circuit un demi-additionneur.

x	y	r	s
entrée 1	entrée 2	retenue	somme
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

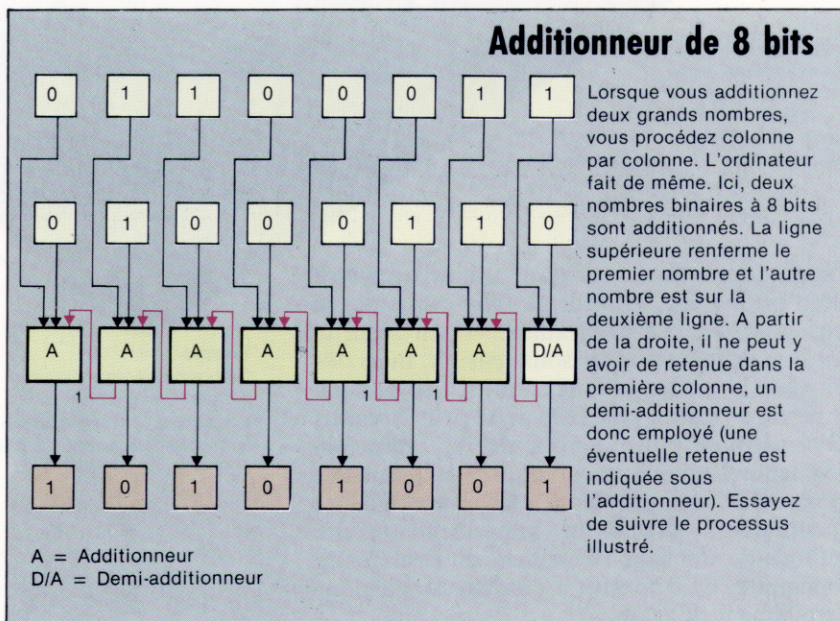
x	y	r(entrée)	r(sortie)	s
0	0	0	0	0
1	0	0	0	1
0	1	0	0	1
0	0	1	0	1
0	1	1	1	0
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

On le nomme un demi-additionneur parce qu'en fait il ne fonctionne qu'à moitié. Il serait parfait si nous ne voulions additionner qu'une seule colonne de chiffres binaires. Cependant, il est généralement nécessaire d'additionner deux octets de données comptant chacun huit bits. L'additionneur à l'extrême droite se doit d'être un demi-additionneur et les autres additionneurs doivent pouvoir accepter trois entrées : les deux chiffres de leur colonne et une retenue venant de la droite.

Un additionneur complet est construit à l'aide de deux demi-additionneurs et d'une porte OU. La « retenue de sortie » de chaque additionneur complet est rapportée dans la retenue d'entrée de l'additionneur de gauche et ainsi de suite pour tous les autres additionneurs complets.

Dans les micro-ordinateurs, la plupart des additions et des autres opérations arithmétiques sont effectuées dans de nombreux circuits additionneurs d'une conception identique à celle que nous venons de décrire. Ces circuits, pour la plupart, font partie de l'unité centrale. Avant l'intégration à grande échelle qui introduisit le microprocesseur, des circuits intégrés ne renfermant que quelques portes étaient d'usage commun. Ces circuits sont généralement nommés puces TTL (ce signifie « logique transistor-transistor », parce que les opérations logiques sont effectuées par des transistors assemblés). Une UC renferme généralement dans une seule puce de petites zones de RAM et de ROM, un très grand nombre de circuits d'interruption et une section nommée ALU ou unité logique arithmétique. L'ALU fait partie de l'UC et renferme toutes les portes logiques et additionneurs requis par l'ordinateur pour effectuer les calculs et pour prendre des décisions logiques.

Il peut paraître étrange que des mécanismes aussi simples et répétitifs puissent réaliser des calculs longs et complexes, dans un laps de temps très court. Pourtant, les ordinateurs les plus performants, des machines de la classe Cray I, sont capables de factoriser des nombres de plus de cinquante chiffres, performance qui semblait encore récemment impossible à la majorité des informaticiens.





# Unité à cassette

La crainte du programmeur est de perdre le chef-d'œuvre qu'il vient de créer à l'écran.

### Haut-parleur

Le haut-parleur de la plupart des enregistreuses à cassette est débranché lorsque l'unité est connectée à un ordinateur ou à un système hi-fi.

### Tête d'effacement

Supprime tout signal enregistré antérieurement sur la bande lorsque le mode enregistrement est sélectionné.

### Compteur

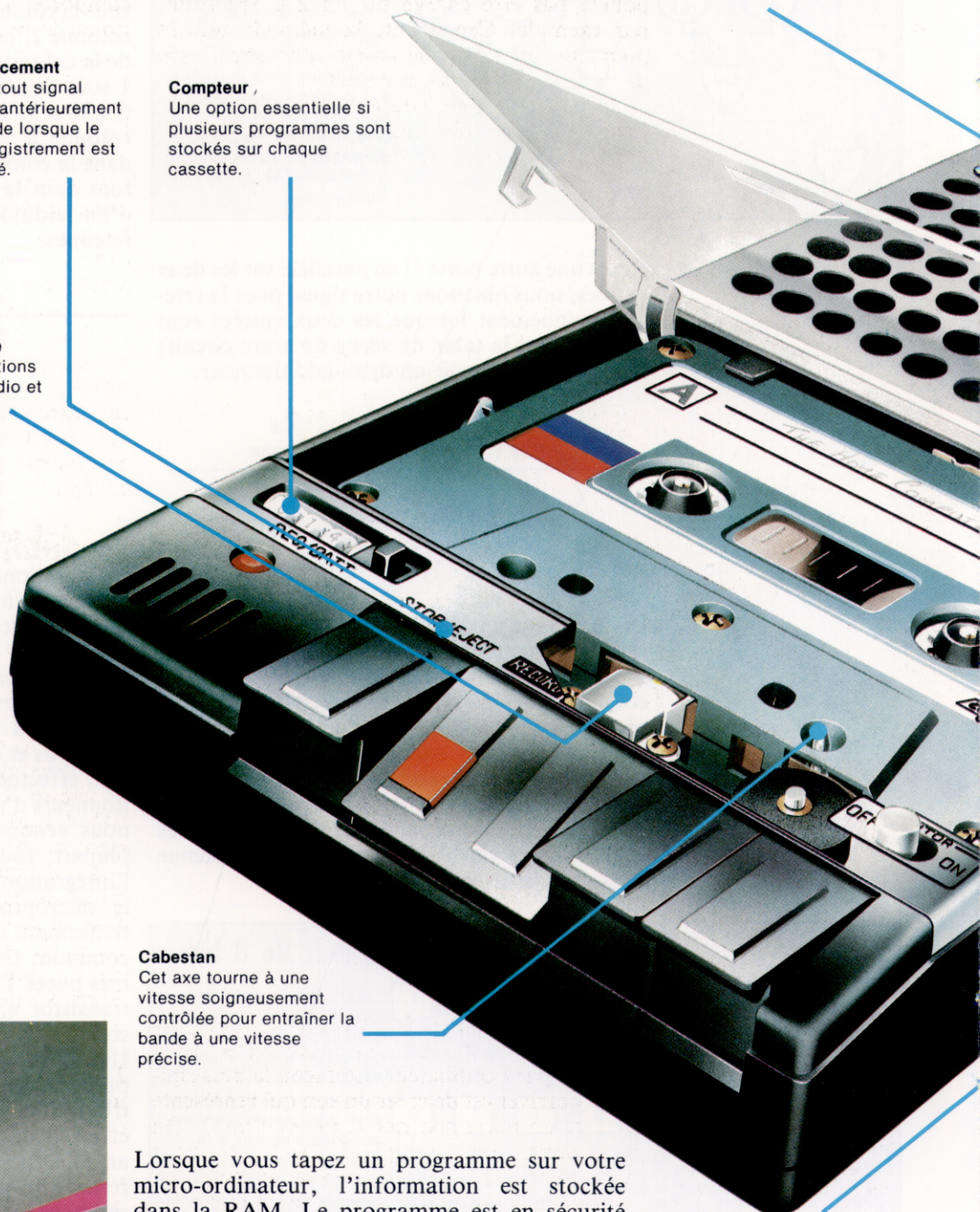
Une option essentielle si plusieurs programmes sont stockés sur chaque cassette.

### Tête de lecture/écriture

Cette tête à deux fonctions enregistre le signal audio et le relit.

### Le Hobbit

Le Hobbit d'Ikon est une unité à cassette dédiée, c'est-à-dire qu'elle a été conçue uniquement pour stocker des données numériques. Elle est mieux qu'une unité à cassette audio puisqu'elle est entièrement commandée par logiciel. Avec cette unité, il n'est plus nécessaire de positionner la bande avant d'effectuer une lecture. Le Hobbit consulte le répertoire et trouve automatiquement le programme sur la bande. (Cl. Ian McKinnell.)



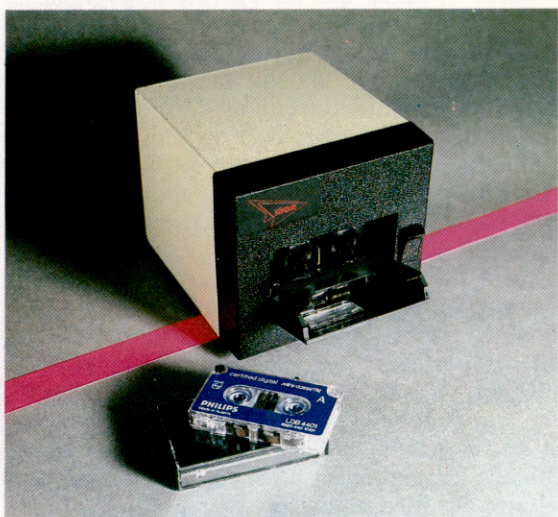
### Cabestan

Cet axe tourne à une vitesse soigneusement contrôlée pour entraîner la bande à une vitesse précise.

### Volume

Ce bouton règle le volume sonore et doit être soigneusement réglé, sinon le programme pourra ne pas être chargé correctement.

Lorsque vous tapez un programme sur votre micro-ordinateur, l'information est stockée dans la RAM. Le programme est en sécurité tant que l'ordinateur est sous tension, mais dès que vous le mettez hors tension, le programme est perdu à jamais. Une telle perte peut être plus frustrante puisque vous devrez retaper le programme lorsque vous voudrez le réutiliser. Pour éviter ce problème, les constructeurs d'ordinateurs proposent généralement une méthode de stockage permanent du contenu de la mémoire. Ce dispositif est indispensable pour le traitement de texte.







### Moteur

Le moteur entraîne le cabestan à une vitesse constante et entraîne également les bobines enrouleuses pendant l'avance et le rebobinage de la bande.

Le support de stockage le plus répandu est le ruban sur cassette. Ce dispositif fut d'abord retenu parce qu'il était déjà disponible et qu'il était peu coûteux. Ce système est maintenant utilisé sur presque tous les ordinateurs familiaux. Le mode de stockage diffère légèrement d'un ordinateur à l'autre; un programme créé et stocké sur un ordinateur Commodore ne pourra pas être chargé sur un ZX Spectrum, par exemple. Cependant, la méthode utilisée pour convertir le programme sous une forme stockable est presque universelle.

Une enregistreuse à cassette audio est conçue pour enregistrer des sons, et votre programme est stocké sous une forme binaire. Les nombres

à la fin de chaque octet, l'ordinateur insère généralement un son additionnel à la fin de chaque octet. On nomme ceux-ci les bits de départ et d'arrêt et leurs valeurs peuvent être 1 ou 0 selon l'ordinateur.

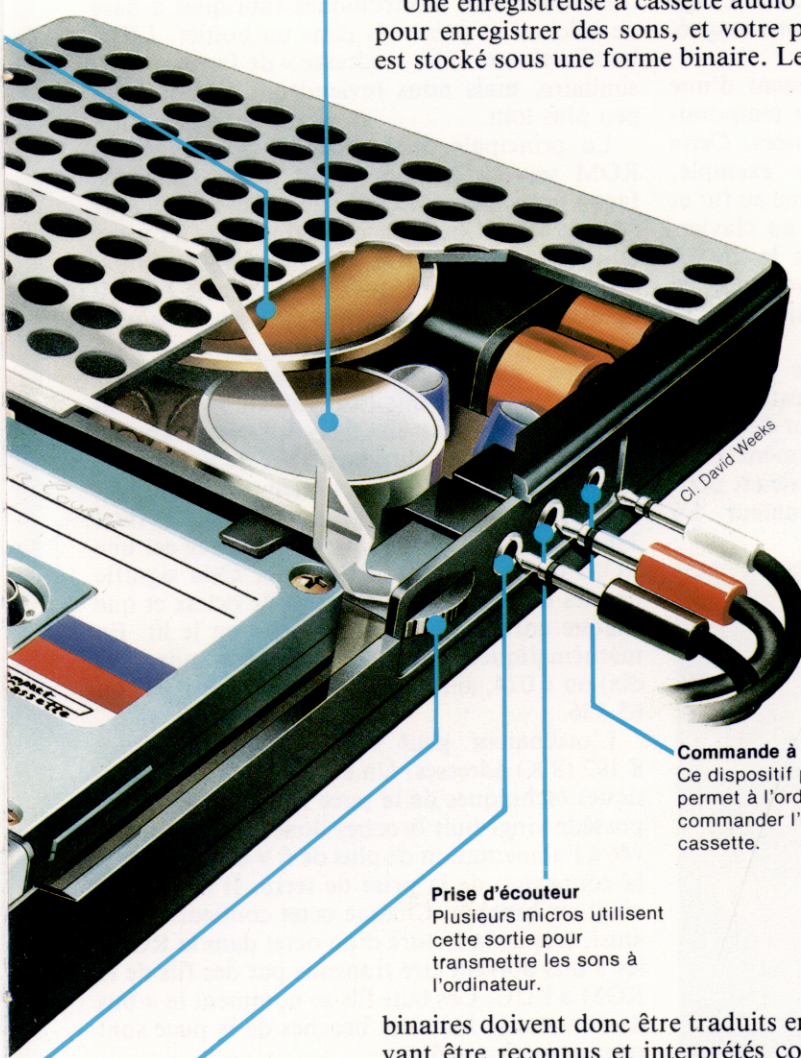
Le programme lui-même est stocké presque de la même manière, sauf qu'il est divisé en segments. Un segment typique compte 256 octets et comporte généralement une information additionnelle permettant à l'ordinateur de vérifier l'intégrité des données associées. Le système utilisé est très simple et se nomme une « somme de contrôle ». Le premier octet du segment renferme le nombre d'octets contenus dans le segment et le dernier octet contient un nombre calculé représentant le total de tous les octets. Lorsque l'ordinateur lit la cassette, il compare les valeurs fournies sur la bande et celles qu'il a calculées; si elles ne sont pas identiques, il annonce l'erreur à l'utilisateur.

Certains systèmes de cassette comme celui du BBC étendent cette vérification jusqu'à nommer et numéroter chaque segment. Si une erreur survient, il est possible de rebobiner la bande sur quelques centimètres et d'essayer de nouveau. A l'opposé, certains systèmes n'annoncent même pas le nom du programme qui est chargé.

## Le débit en bauds

« Débits en bauds » est le terme généralement (et incorrectement) utilisé pour désigner la vitesse à laquelle les sons sont produits et enregistrés sur la bande. Ce nom vient du code Baudot utilisé au début de la télégraphie et désigne le nombre de changement de signaux par seconde. Il est plus exact de parler du nombre de bits enregistrés par seconde. Plus grande est la vitesse d'enregistrement — comprise entre 300 et 1 200 bits par seconde —, plus rapidement vos programmes seront stockés et rappelés dans l'ordinateur. Malheureusement, la fiabilité du système est inversement proportionnelle à la vitesse d'enregistrement; une vitesse de 1 200 bits par seconde est fiable et suffisamment rapide. Certains systèmes offrent un choix de deux vitesses, généralement une vitesse très basse et très fiable de 300 bits par seconde et une autre vitesse moins fiable de 1 200 ou de 2 400 bits par seconde. Les copies de programmes très importants devraient être faites aux deux vitesses pour parer à d'éventuels incidents.

La bande doit être de bonne qualité : vous pouvez vous servir de bandes audio plutôt que de bandes spécialement conçues, mais veillez à choisir des marques réputées et des bandes dont la longueur n'excède pas 60 minutes. La capacité de stockage approximative d'une cassette peut être établie en divisant la vitesse de l'unité à cassette par dix. Cela donne le nombre d'octets stockés chaque seconde sur la cassette; une cassette C-60 de 30 minutes sur chaque côté avec une interface réglée à 1 200 bits par seconde peut contenir environ 432 Koctets de programme.



**Commande à distance**  
Ce dispositif pratique permet à l'ordinateur de commander l'unité à cassette.

**Prise d'écouteur**  
Plusieurs micros utilisent cette sortie pour transmettre les sons à l'ordinateur.

**Prise de microphone**  
Souvent employée comme entrée pour les données venant de l'ordinateur. Cependant, elle ne doit être utilisée que si l'unité ne comporte pas de prise DIN ou auxiliaire. Les boutons de volume et de tonalité doivent être soigneusement réglés lorsque cette entrée est employée.

binaires doivent donc être traduits en sons pouvant être reconnus et interprétés comme des 0 ou des 1 par l'ordinateur. La façon la plus simple d'y arriver est de créer un son qui représente un 1 et un autre son qui représente un 0. On peut choisir par exemple 2 400 cycles pour représenter 1 et 1 200 cycles pour représenter 0.

Lorsque l'ordinateur reçoit la commande SAVE, le tout premier enregistrement sur la bande est un signal sonore constant généré pendant un certain nombre de secondes. Grâce à ce signal, l'ordinateur peut établir une démarcation entre la bande vierge et le début du programme. La première information enregistrée est une série de sons qui représentent les caractères du nom du programme. Chaque caractère est un octet, huit bits, donc chacun d'eux est représenté par huit sons. Cependant, pour indiquer le début et

# Stockage de données

L'ordinateur peut stocker des milliers d'octets d'information dans sa mémoire et se souvient exactement où chaque octet a été placé.

La mémoire des ordinateurs peut stocker des données à long et à court terme. Le stockage à long terme est effectué sur des rubans magnétiques et sur des disquettes.

Les ordinateurs se servent également d'une mémoire à court terme pour stocker temporairement les programmes et les données. Cette mémoire temporaire conserve, par exemple, toutes les instructions d'un programme au fur et à mesure que vous les introduisez au clavier. Mais attention, si vous débranchez le micro-ordinateur ou effacez par inadvertance le contenu de cette mémoire, tout disparaîtra et vous serez obligé de recommencer.

On peut également dire d'une mémoire d'ordinateur qu'elle est interne ou externe. La mémoire interne est située à l'intérieur de l'ordinateur et est normalement entièrement électronique, alors que la mémoire externe est périphérique ou à l'extérieur de l'ordinateur. La

Ces deux types de mémoire sont des dispositifs entièrement électroniques fabriqués à base de silicium et présentés dans un boîtier. L'UC les sélectionne et les « adresse » de façon un peu similaire, mais nous reviendrons là-dessus un peu plus loin.

La principale différence est que les puces ROM servent à stocker des programmes de façon permanente. Le codage binaire de chaque adresse est défini de façon permanente à l'usine et ne peut être modifié. Les ROM sont comparables à des bibliothèques de référence. L'ordinateur peut consulter leur contenu mais ne peut rien y écrire.

ROM est l'abréviation de *Read Only Memory*. On dit que l'ordinateur lit des données lorsqu'il extrait des données dans la mémoire. Il existe plusieurs types de ROM légèrement différents. Le contenu de certaines ROM peut être effacé et reprogrammé. L'Intel 2364 est une ROM typique. Cette puce est une ROM de 65 536 bits, 8 Koctets. Cela signifie que les 64 Kbits sont groupés par octets et que chaque adresse accède à un octet ou le lit. En mathématiques,  $1\text{ K} = 2^{10}$  (deux à la puissance dix) ou 1 024, ainsi que :  $64\text{ K} = 64 \times 1\text{ 024}$  ou 65 536.

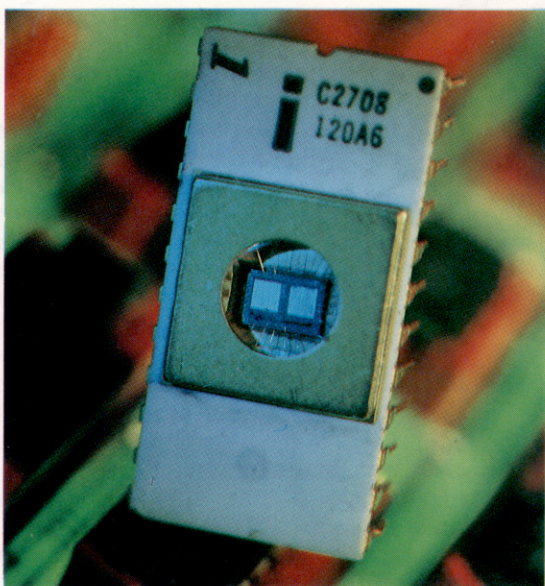
L'ordinateur peut donc choisir l'une des 8 192 (8 K) adresses. Un examen des caractéristiques techniques de la puce 2364 révèle qu'elle possède vingt-huit broches dont l'une est réservée à l'alimentation de plus de 5 V et une autre à la connexion de la prise de terre. Il nous reste vingt-six broches. Chaque octet contient 8 bits ; ainsi, lors de la lecture d'un octet dans la ROM, les 8 bits doivent être transmis par des fils de la ROM à l'UC. Ces huit fils se nomment le « bus de données ». Huit des broches de la puce sont connectées à ces fils.

Cela nous laisse dix-huit broches. L'une des broches n'est pas utilisée. Quatre broches servent à diverses sélections de la puce. Il s'agit d'une broche de « validation de sortie », d'une broche de « validation de circuit », et de deux broches de « sélection de circuit ». Ces broches reçoivent des signaux venant de l'ordinateur.

Les treize autres broches sont des broches d'« adresse ». Chaque broche est connectée à un fil du bus d'adresse et ces fils transmettent sous forme binaire l'adresse de l'octet sollicité. Treize chiffres binaires peuvent donner  $2^{13}$  ou 8 192 combinaisons de 1 et de 0, les treize lignes d'adresses sont donc tout juste suffisantes pour sélectionner chacun des 8 192 octets stockés dans la ROM.

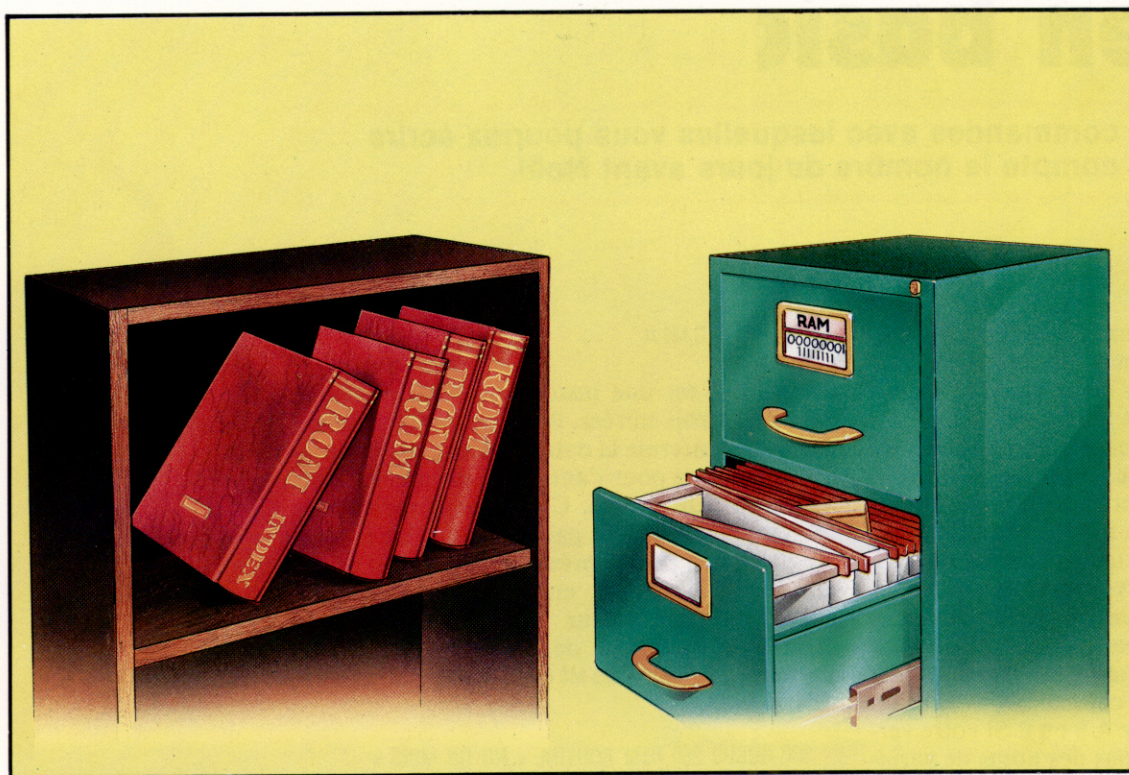
## EPROM

Le problème posé par les ROM ordinaires est que leur contenu est défini lors de la fabrication et ne peut être modifié. Les EPROM (*Erasable Programmable Read Only Memory*) sont beaucoup plus souples. Elles peuvent être reprogrammées en effaçant d'abord leur contenu et en « écrivant » un nouveau programme. Les EPROM possèdent une fenêtre qui laisse passer des rayons ultraviolets, ce qui a pour effet de décharger les condensateurs qui stockent les bits. En l'absence d'ultraviolet, les condensateurs conservent leur charge indéfiniment et le contenu de la mémoire est préservé.



mémoire externe est généralement partiellement mécanique; elle consiste en des dispositifs comme des lecteurs de cassettes, des lecteurs de disquettes ou même des perforateurs et des lecteurs de bandes de papier.

La mémoire interne électronique est généralement nommée la mémoire principale, alors que la mémoire externe est nommée la mémoire de sauvegarde ou la mémoire de masse. Il existe deux types de mémoire interne : la mémoire ROM (mémoire morte) et la mémoire RAM (mémoire vive).

**Mémoire à long terme**

La ROM (mémoire morte) est comparable à un livre puisque c'est un support où les données sont stockées en permanence. Vous ne pouvez modifier ou supprimer les mots imprimés sur une page.

**Mémoire à court terme**

La RAM (mémoire vive) ressemble plus à un système de classement qu'à un livre, puisque l'information peut y être modifiée et qu'elle n'est pas permanente; la RAM est effacée au complet lorsque l'ordinateur est mis hors tension.

Les RAM sont un peu les ardoises de l'ordinateur. Les programmes et les données peuvent y être stockés temporairement pendant le travail de l'ordinateur, des résultats et des données peuvent aussi y être « écrits » temporairement. L'Intel 2114 est une RAM typique. Chaque RAM 2114 contient 4 096 bits de mémoire, et ceux-ci sont disposés en 1 024 « quartets » (demi-octets) de 4 bits. Cela signifie que chaque adresse correspond à 4 bits de données. Deux de ces puces seront donc nécessaires pour produire 1 Koctet de données. Chaque puce 2114 possède dix-huit broches, dont deux sont utilisées pour la prise de terre et pour l'alimentation. Quatre servent de lignes d'entrée/sortie de données. Une est utilisée pour le signal de sélection de circuit (le signal qui sollicite la puce), et une autre annonce après une sollicitation s'il s'agit d'une lecture ou d'une écriture. Les dix autres broches sont utilisées pour le bus d'adresse. Dix lignes d'adresses ne peuvent désigner que  $2^{10}$  adresses, ou 1 024. Si un ordinateur dispose de 64 K de RAM, et s'il utilise des puces Intel 2114, un total de cent vingt-huit puces RAM serait nécessaire, puisqu'il faut deux puces pour stocker un octet. Aujourd'hui on utilise généralement des puces RAM plus denses. En utilisant des puces RAM plus modernes, comme l'Intel 4164, il est possible d'obtenir 64 K de RAM avec seulement huit puces.

Le prix et les dimensions des puces ROM et RAM diminuent à chaque année. Et il est maintenant possible de placer 128 Koctets sur une seule puce. Mais il semble actuellement difficile d'obtenir de plus hautes densités. Les circuits deviennent d'une telle finesse que les techniques optiques actuelles ne sont plus à la hauteur. Les

puces de mémoire de l'avenir à « haute densité » seront probablement fabriquées à l'aide de faisceaux électroniques ou de rayons X.

En gros, on utilise deux types de mémoire RAM, nommés RAM statique et RAM dynamique. Les deux types ont leurs avantages et leurs inconvénients, mais la RAM dynamique est plus utilisée que la RAM statique. Dans les deux cas, le contenu de la mémoire est perdu lors d'une mise hors tension, mais une RAM dynamique doit être régénérée des centaines de fois par seconde. Chaque bit en mémoire doit être régénéré ou réécrit sans ralentir la vitesse d'accès des données. Cela implique donc une synchronisation très précise, ce qui complique le travail du concepteur du circuit.

La mémoire dynamique offre deux avantages par rapport à la mémoire statique. Une mémoire dynamique ne nécessite qu'un seul transistor par bit, alors qu'une mémoire statique en utilise trois. Cela permet une plus grande densité. La plupart des puces RAM dynamiques n'ont que seize broches. L'autre avantage de la RAM dynamique est qu'elle consomme moins d'énergie que la RAM statique. Elle génère donc moins de chaleur et nécessite des alimentations plus petites et moins coûteuses.

La RAM statique a un avantage, c'est la simplicité de ses circuits. Dès que les données sont écrites en mémoire, elles sont conservées sans avoir à être régénérées. Chaque cellule de mémoire d'un bit requiert trois transistors, il est donc difficile d'obtenir des densités élevées permises par la mémoire dynamique. Une mémoire statique consomme aussi plus d'électricité; la chaleur supplémentaire produite peut entraîner l'obligation d'installer un ventilateur de refroidissement.

# Noël en basic

**Voici de nouvelles commandes avec lesquelles vous pourrez écrire un programme qui compte le nombre de jours avant Noël.**

Ce programme révisé la matière des cours précédents et introduit plusieurs nouvelles instructions BASIC. La fonction de ce programme est de compter le nombre de jours avant Noël.

Ce programme commence par énumérer les variables utilisées. Cette pratique n'est pas essentielle, mais elle est recommandée pour rendre vos programmes plus faciles à comprendre. Certaines versions de BASIC permettent de donner des noms de plusieurs caractères aux variables (JOUR par exemple) au lieu de noms à une lettre comme ceux que nous avons utilisés jusqu'ici. Si vous avez la chance de posséder une telle version, des noms comme JOUR, MOIS ou NOJOUR sont plus clairs que A, B ou C. Si votre version de BASIC ne permet pas des noms de variables à plusieurs caractères, lister les noms de variables au début du programme rend le programme plus « lisible ».

Vous commencez depuis peu à faire connaissance avec les rudiments de la programmation. Ces travaux, qui peuvent vous sembler simples, ou même « simples », sont indispensables pour bien connaître les particularités du BASIC. Chaque instruction détaillée dans ce cours vous servira plus tard à résoudre un problème personnel bien plus difficile que celui qui est proposé ci-dessous. Mais tout problème, pour complexe qu'il soit, se décompose en sous-problèmes plus simples dont la résolution peut être confiée à des sous-programmes d'une simplicité comparable à celle de « Noël en basic ». En effet, une évidence doit guider le programmeur débutant, tout programme est composé d'une suite de petits programmes qui facilitent la rédaction et l'exécution de l'ensemble. Alors, un petit peu de patience, et bientôt vous serez capable de rédiger des logiciels scientifiques ou, pourquoi pas, des programmes de jeu passionnants !

Les instructions PRINT commençant à la ligne 230 lanceront l'exécution du programme. Ces instructions annoncent à l'utilisateur la fonction du programme et lui demandent de taper la date dans le format présenté, en séparant les divers éléments par des virgules.

La première nouvelle instruction apparaît à la ligne 300. C'est une instruction de DIMension. Elle sert à déclarer le nombre d'éléments permis dans le tableau portant l'étiquette X. Un tableau est semblable à une variable ordinaire, sauf que la boîte ici compte plusieurs compartiments. A la ligne 300, nous avons créé une variable nommée X comptant douze compartiments. Nous reviendrons sur l'instruction DIM et sur les tableaux un peu plus loin.

```
310 INPUT D,M$,A
```

Cette ligne est une instruction INPUT ordinaire qui attend trois entrées. D est une variable numérique qui renferme la date. A est une autre variable numérique pour l'année. M\$ est une variable légèrement différente. C'est une « variable de chaîne », caractérisée par le signe \$. Une variable de chaîne peut représenter des lettres ou un nombre. Si nous entrons par exemple 23,JANVIER,1983, la valeur 23 est attribuée à la variable D, la chaîne de caractères JANVIER est attribuée à la variable M\$ et la variable A reçoit la valeur 1983.

```
330 GOSUB 560 REM ROUTINE « NO DU MOIS »
```

Cette instruction demande au programme de passer au sous-programme commençant à la ligne 560. Notez également la présence d'une instruction REM sur la même ligne. Il est permis d'insérer une instruction REM sur la même ligne qu'une autre instruction. Ce sous-programme n'est utilisé qu'une seule fois par le programme principal, il aurait donc été tout aussi facile de l'intégrer au sein du programme principal.

Dans la première version du programme, un nombre était utilisé pour désigner le mois et ce sous-programme n'était donc pas nécessaire. Plus tard, nous avons décidé de permettre l'entrée du mois en lettres. Pour convertir ces lettres en leur équivalent numérique, nous avons écrit le sous-programme séparément. Nous n'avons alors qu'à insérer une instruction GOSUB dans le programme principal. Ce sous-programme illustre comment il est facile d'assembler des programmes à l'aide des instructions GOSUB et RETURN.

Ce sous-programme est très simple, mais illustre comment le BASIC manipule facilement les chaînes de caractères. Supposons que nous ayons entré JANVIER comme mois dans l'instruction INPUT. La chaîne de caractères JANVIER est alors affectée à la variable M\$. La première ligne du sous-programme est :

```
560 IF M$ = « JANVIER » THEN LET M = 1
```

Cette instruction compare le contenu de M\$ avec les caractères placés entre guillemets. S'ils sont identiques (ici ils le sont), l'instruction affecte la valeur 1 à la variable M. Ne confondez pas les variables M et M\$. Elles sont différentes. Seule M\$ est une variable de chaîne. Après cette vérification, le programme continue et vérifie tous les autres noms de mois en attribuant à M la valeur

numérique correspondant au mois entré. C'est-à-dire 1 pour JANVIER, 3 pour MARS et ainsi de suite.

Lorsqu'il atteint la ligne 680, le BASIC revient au programme principal à la ligne située après l'instruction GOSUB, la ligne 340. Elle renferme une instruction REM sans commentaire. Cette instruction REM ne sert qu'à introduire une ligne vierge pour faciliter la lecture du programme.

Les lignes 350 à 370 forment une boucle FOR-NEXTI. Cette boucle incrémente la valeur de I, de 1 à 12. La variable I est l'indice du tableau X dans la ligne 360. Examinons cette ligne.

```
360 READ X(I)
```

Voici une nouvelle instruction. READ est toujours utilisé avec une instruction DATA associée. Ici l'instruction DATA correspondante se trouve à la ligne 510 :

```
510 DATA 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 25, 0
```

Ces nombres, sauf le dernier, représentent le nombre de jours dans chaque mois de l'année. Les lignes 350, 360, 370 et 510 équivalent à ces douze instructions LET :

```
LET X(1) = 31
LET X(2) = 28
LET X(3) = 31
LET X(4) = 30
LET X(5) = 31
LET X(6) = 30
LET X(7) = 31
LET X(8) = 31
LET X(9) = 30
LET X(10) = 31
LET X(11) = 30
LET X(12) = 25
```

La boucle définie à la ligne 350 effectue un compte de 1 à 12, et permet de définir douze indices pour le tableau X.

Avant de revenir à ce programme, examinons un court programme beaucoup plus simple :

```
10 READ A, B, C
20 LET D = A + B + C
30 PRINT D
40 DATA 5, 10, 20
```

Ici l'instruction READ de la ligne 10 lit le premier élément de données de la ligne 40 et écrit cette valeur dans la première variable. En d'autres mots, elle affecte la valeur 5 à la variable A. READ lit alors le prochain élément de données et le place dans la variable suivante. Ce programme affecte 5 à A, 10 à B et 20 à C. Il additionne alors ces valeurs et affecte le résultat à la variable D. Ce résultat est alors imprimé à la ligne 30.

Revenons à notre programme. Au début de la première itération la valeur de I est 1. La ligne 360 est donc READ X(1). L'élément de données correspondant dans la ligne 510 est 31 (le premier élément). Par conséquent X(1) = 31.

Lors de la deuxième itération, I devient 2 la ligne 360 devient donc READ X(2). Le prochain élément de données de la ligne DATA est 28. X(2) égale donc 28. Ainsi, les douze compartiments du tableau X reçoivent tous une valeur correspon-

```
100 REM LISTE DES VARIABLES
110 REM
120 REM D = DATE DU JOUR
130 REM M$ = MOIS
140 REM A = ANNEE
150 REM I = INDICE 1
160 REM X = TABLEAU DES JOURS DE CHAQUE MOIS
170 REM R = JOURS QUI RESTENT
180 REM M = NUMERO DU MOIS
190 REM L = INDICE 2
200 REM Z = VALEUR ENTIERE DE A/4
210 REM
220 REM
230 PRINT « CE PROGRAMME CALCULE »
240 PRINT « LE NOMBRE DE JOURS QUI RESTENT »
250 PRINT « JUSQU'A NOEL »
260 PRINT
270 PRINT « ENTREZ LA DATE DU JOUR, LE MOIS, L'ANNEE »
280 PRINT « EX 12, JUILLET, 1984 »
290 PRINT
300 DIM X (3)
310 INPUT D, M$, A
320 REM
330 GOSUB 560 REM ROUTINE « NUMERO DU MOIS »
340 REM
350 FOR I = 1 TO 13
360 READ X (I)
370 NEXT I
380 REM
390 GOSUB 750 REM ROUTINE « ANNEE BISSEXTILE »
400 REM
410 LET R = X (M) - D
420 FOR L = M TO 11
430 LET M = M + 1
440 LET R = R + X (M)
450 NEXT L
460 REM
470 IF R = 1 THEN GOTO 500
480 PRINT « IL RESTE »;R;« JOURS AVANT NOEL »
490 GOTO 520
500 PRINT « IL RESTE 1 JOUR AVANT NOEL »
510 DATA 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 25, 0
520 END
530 REM
540 REM
550 REM
560 IF M$ = « JANVIER » THEN LET M = 1
570 IF M$ = « FEVRIER » THEN LET M = 2
580 IF M$ = « MARS » THEN LET M = 3
590 IF M$ = « AVRIL » THEN LET M = 4
600 IF M$ = « MAI » THEN LET M = 5
610 IF M$ = « JUIN » THEN LET M = 6
620 IF M$ = « JUILLET » THEN LET M = 7
630 IF M$ = « AOUT » THEN LET M = 8
640 IF M$ = « SEPTEMBRE » THEN LET M = 9
650 IF M$ = « OCTOBRE » THEN LET M = 10
660 IF M$ = « NOVEMBRE » THEN LET M = 11
670 IF M$ = « DECEMBRE » THEN LET M = 12
680 RETURN
690 REM
700 REM
710 REM
720 REM NOTE : CETTE ROUTINE NE PREND
730 REM PAS EN COMPTE LES ANNEES BISSEXTILES
740 REM A LA FIN DE CHAQUE SIECLE
750 LET A = A/4
760 LET Z = INT (A)
770 IF A - Z = 0 THEN GOTO 790
780 RETURN
790 LET X(2) = X(2) + 1
800 RETURN
```

dant au nombre de jours de chaque mois, sauf le douzième auquel on affecte la valeur 25 (voyez-vous pourquoi?).

```
390 GOSUB 750 REM ANNÉE BISSEXTILE
```

Cette ligne dirige le programme vers un sous-programme qui vérifie si l'année entrée est bissextile ou non. Voyons comment fonctionne ce sous-programme.

```
750 LET A = A/4
760 LET Z = INT(A)
770 IF A - Z = 0 THEN GOTO 790
780 RETURN
790 LET X(2) = X(2) + 1
800 RETURN
```

Une année bissextile est un multiple de 4. Si l'année en question est le début d'un siècle, elle doit aussi être divisible par 400. Pour simplifier le programme, nous ne vérifierons que la divisibilité par 4 de l'année entrée.

La ligne 750 définit A égal à l'ancienne valeur de A (année) divisée par 4. La nouvelle valeur de A sera un nombre entier si l'année est un multiple de 4. Sinon elle aura une fraction décimale.

La ligne 760 utilise la fonction INT pour trouver l'« entier » de A. Si l'année entrée est 1985,  $A/4 = 496,25$ . La nouvelle valeur de A est donc 496,25. Comme la fonction INT arrondit un nombre décimal à l'entier inférieur, LET Z = INT(A) affecte la valeur 496 à la variable Z.

La ligne 770 soustrait Z de A et compare le résultat à 0. Si le résultat est 0, l'année est bissextile (puisque il n'y a pas de fraction décimale dans A). Dans ce cas, l'instruction GOTO dirige le programme à la ligne 790. La ligne 790 ajoute 1 au deuxième élément du tableau (qui était 28, le nombre de jours en février).

Si le résultat de la soustraction de la ligne 770 est différent de 0, X(2) n'est pas modifié et le programme revient à la ligne 400.

La ligne 400 est une autre ligne REM qui ne sert qu'à rendre le programme plus lisible. A la ligne 410, R représente le nombre de jours qui restent. Il égale ici le nombre de jours du mois moins la date entrée.

Si, par exemple, nous avons entré 12, JANVIER, 1983, D égale 12 et M égale 2. X(M) est donc X(2), et le second élément du tableau X est 28 (1983 n'est pas une année bissextile). Par conséquent, R est égal à  $28 - 12$ , c'est-à-dire à 16, le nombre de jours qui restent en février.

Une autre boucle commence à la ligne 420. Celle-ci incrémente la valeur de M. Savez-vous pourquoi nous employons FOR L = M TO 11 et non FOR L = M TO 12? Si M était 2 parce que nous avons tapé FÉVRIER, la ligne 430 l'incrémente à 3. La ligne 440 définit R, le nombre de jours qui restent, égal à son ancienne valeur plus X(M) qui équivaut à X(3) puisque M a été incrémente de 1. La valeur de X(3) est 31, le nombre de jours en mars. Dans la ligne 440, la nouvelle valeur de R est donc  $16 + 31$ . Lors de la boucle suivante, M est incrémente à 4 et le nombre de jours en avril, X(4), est additionné à l'ancienne valeur de R. R vaut donc :  $16 + 31 + 30$ .

La boucle est parcourue une dernière fois lorsque la limite supérieure 11 a été atteinte (à la ligne 420). La ligne 430 ajoute 1 à la valeur de M une dernière fois, M égale donc 12. La valeur de X(12) est le dernier élément de l'instruction DATA, 25. Il y a vingt-cinq jours avant Noël en décembre.

```
470 IF R = 1 THEN GOTO 500
```

S'il ne reste qu'un seul jour avant Noël, cette ligne dirige le programme vers la ligne 500 afin d'obtenir une réponse grammaticalement correcte.

Ce n'est pas plus compliqué que cela. Ce programme BASIC devrait fonctionner sur la plupart des ordinateurs (voir l'encadré « Variantes de BASIC »), à l'exception peut-être du sous-programme « n° du mois »; il existe de nombreuses variantes pour l'instruction LET en BASIC. Si les lignes comme IF M\$ = « SEPTEMBRE » THEN LET M = 9 ne fonctionnent pas sur votre ordinateur, le sous-programme peut être écrit comme ceci :

```
560 IF M$ = « JANVIER » THEN GOTO 900
570 IF M$ = « FÉVRIER » THEN GOTO 910
580 IF M$ = « MARS » THEN GOTO 920
.
900 LET M = 1
905 RETURN
910 LET M = 2
915 RETURN
920 LET M = 3
925 RETURN
(et ainsi de suite...)
```

Cette solution occupe plus de mémoire et n'est pas aussi facile à suivre. Cependant, nous venons de démontrer qu'il y a généralement plusieurs façons de résoudre un problème.

## Variantes de basic

### DIM

Si le programme doit être exécuté sur le BBC, sur le Dragon ou sur l'Oric, la ligne 300 doit être changée de la façon suivante : 300 DIM X(13).

### INPUT

Il n'est pas permis d'entrer une série de valeurs séparées par des virgules sur le Spectrum. La ligne 310 doit donc être modifiée de la façon suivante : 310 INPUT D, et les lignes 312 INPUT M\$ et 314 INPUT A doivent être ajoutées. La date est entrée en trois étapes, les instructions d'impression doivent être ajoutées : 280 PRINT « JOUR », 311 PRINT « MOIS », 313 PRINT « ANNÉE ».

### REM

Les instructions REM, si elles partagent une ligne avec une autre instruction, doivent être séparées de celle-ci par un deux-points (:) si vous utilisez un BBC, un TI99/4A ou un Spectrum. Par exemple : 330 GOSUB 560 : REM ROUTINE « NO DU MOIS ».

### END

L'instruction END n'est pas offerte sur le Spectrum, la ligne 520 doit donc être modifiée : 520 GOTO 1000. Et la ligne 1000 doit être ajoutée : 1000 REM FIN DU PROGRAMME.

*Avec un prof sympa, l'informatique est un jeu d'enfant.*



*En vente dans  
les boutiques spécialisées  
et grands magasins.*

Le basic, c'est la langue de demain, celle que parlent les micro-ordinateurs. Ce nouveau langage, Alice va vous l'apprendre.

Alice, c'est un micro-ordinateur couleur qui se branche sur votre téléviseur. Alice, c'est aussi le livre qui va vous apprendre à parler basic, vous passionner pour la création informatique.

Une main sur le livre, l'autre sur le clavier, vous dialoguerez très vite avec votre micro-ordinateur. Si vous avez déjà quelques notions de basic, vous arriverez très loin dans la connaissance de ce langage.

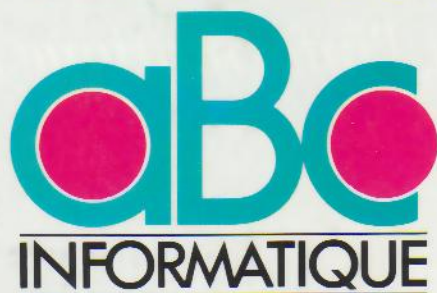
Alors apprenez vite le basic et vous verrez qu'avec un prof sympa, l'informatique est un jeu d'enfant.



# ALICE, LE PROF DE BASIC.

MICRO-INFORMATIQUE MATRA-HACHETTE

# GRAND CONCOURS



## 100 MICRO A GAGNER

# 120000F DE PRIX AU TOTAL!

Cette semaine nous vous soumettons la 3<sup>e</sup> question de notre grand concours réservé aux lecteurs de ABC Informatique.

Il y a 100 micro-ordinateurs Alice à gagner.

Alors tentez votre chance.

Pour participer il vous suffit de répondre à 5 questions. Vous trouverez ci-contre la question n° 3. La 4<sup>e</sup> question

et la question subsidiaire vous seront posées dans le fascicule 6 avec le bulletin-réponse.

### QUESTION N° 3

En quelle année la Comtesse Ana Lovelace écrivit-elle le premier "programme" de l'Histoire?

1802

1835

1847

### Règlement du concours "ABC Informatique - Cours d'informatique pratique et familiale"

#### Article 1

Les Editions ATLAS et la Société ATLEN organisent à l'occasion de la sortie de la nouvelle encyclopédie "ABC Informatique" un grand concours du 20 janvier au 29 février 1984.

#### Article 2

Le concours comporte 4 questions plus une question subsidiaire pour départager les ex-aequo éventuels: 1 question dans le fascicule 3 paraissant le 20 janvier, 1 question dans le fascicule 4 paraissant le 27 janvier, 1 question dans le fascicule 5 paraissant le 3 février, la dernière question et le bulletin-réponse comportant le rappel des 4 questions et la question subsidiaire dans le fascicule 6 paraissant le 10 février 1984.

#### Article 3

Le bulletin-réponse ainsi que les questions et le texte intégral du règlement pourront être également envoyés sur simple demande dûment affranchie adressée à: Editions ATLAS - 33, avenue du Maine - 75755 Paris Cedex 15. Aucune autre correspondance ne sera échangée avec les concurrents.

#### Article 4

Il ne sera accepté qu'un seul bulletin-réponse par personne. Les bulletins incomplets, raturés ou illisibles, ainsi que les envois insuffisamment affranchis, en recommandé ou comportant une adresse incomplète ou illisible, seront considérés comme nuls.

#### Article 5

Pour participer, il suffit de remplir le bulletin-réponse et de le retourner sous enveloppe dûment affranchie aux Editions ATLAS, 33, avenue du Maine 75755 Paris Cedex 15. La date limite d'envoi des réponses est fixée au 29 février 1984 minuit, le cachet de la poste faisant foi.

#### Article 6

Les Editions ATLAS se réservent le droit de suspendre, de modifier ou d'annuler le concours en cas de force majeure ou d'événements indépendants de leur volonté.

#### Article 7

En cas de contestation, le tribunal de Grande Instance de Paris sera seul compétent.

#### Article 8

Le concours est ouvert à toute personne domiciliée en France métropolitaine ou en Belgique à l'exception des personnes visées à l'article 9 ci-après. Les bulletins-réponses devront être adressés, pour la France et la Belgique, à une seule adresse: Editions ATLAS, 33, avenue du Maine - 75755 Paris Cedex 15.

#### Article 9

Les membres du personnel des Editions ATLAS, de la Société EDENA, de la Société ATLEN-Bruxelles, des Agences Robert & Partners, Ecom International et Piment, les auteurs et collaborateurs des publications ainsi que leurs familles ne pourront participer à ce concours.

#### Article 10

Les candidats seront classés selon leur nombre de bonnes réponses aux première, deuxième, troisième et quatrième questions.

La question subsidiaire départagera les éventuels ex-aequo pour les 100 prix. Les réponses à la question subsidiaire seront soumises à un jury souverain composé d'un informaticien, d'un publicitaire et d'un membre du personnel des Editions Atlas; les critères d'appréciation des réponses seront: l'humour, la pertinence, le sens de la formule.

#### Article 11

Prix: le concours est doté de 100 micro-ordinateurs ALICE d'une valeur unitaire de 1200 FF (livrés avec le guide ALICE, une prise Peritel et un câble d'alimentation, sans autres accessoires).

#### Article 12

Les lots ne pourront en aucun cas être échangés contre des espèces.

#### Article 13

Les gagnants seront avisés individuellement avant le 15 avril 1984, et informés de la date à laquelle le prix leur sera remis.

Ces gagnants acceptent par avance la divulgation éventuelle de leur nom et prénom.

Les réponses aux 4 questions seront publiées ultérieurement dans un prochain fascicule de ABC Informatique ainsi que la liste des gagnants avant le 30/6/84.

Cette liste des gagnants pourra être adressée à toute personne qui en fera la demande à Editions ATLAS, 33, avenue du Maine - 75755 Paris Cedex 15.