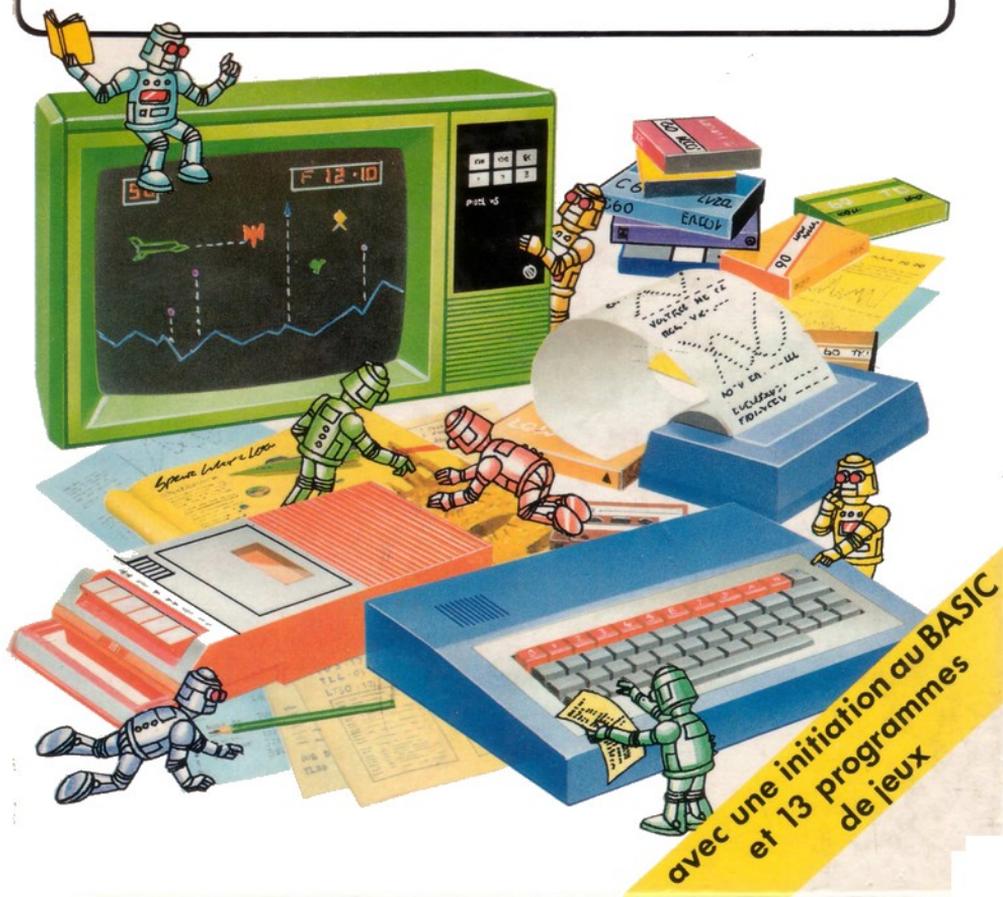


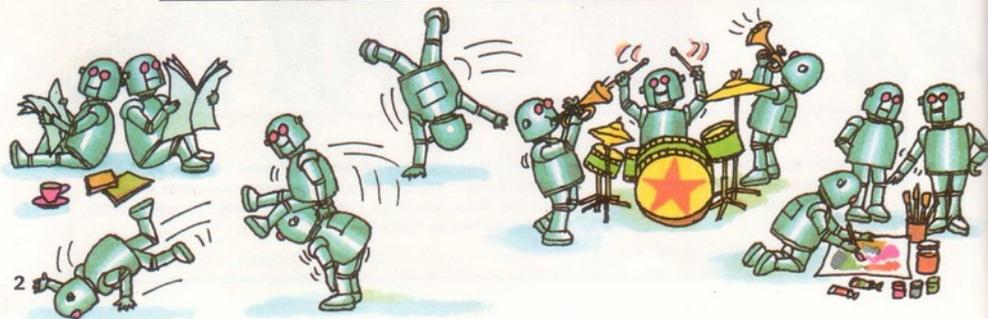
# GUIDE DU MICRO ORDINATEUR



avec une initiation au BASIC  
et 13 programmes  
de jeux

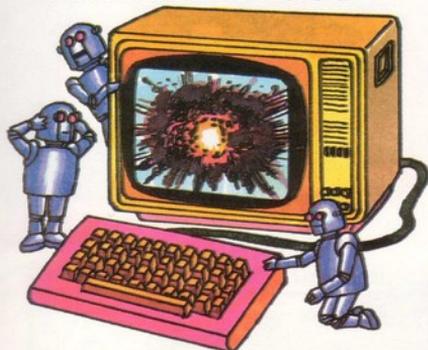
## Sommaire

- 4 À la découverte du micro-ordinateur
- 6 Qu'est-ce que la programmation
- 8 Un coup d'œil sur la console
- 10 Des programmes sur mesure
- 12 Écrivez vos propres programmes...
- 14 Faites-les fonctionner sans peine
- 16 Comment conserver un programme
- 18 Dessins, graphismes et animations
- 20 Bruitages et musique
- 22 Voyage à l'intérieur de la console
- 24 À l'intérieur d'une puce
- 26 Comment une puce travaille
- 28 À propos des puces
- 30 Histoire de la micro-informatique
- 32 Réseaux d'ordinateurs
- 34 Robots et automates
- 36 D'autres utilisations  
de la micro-informatique
- 38 Des périphériques pour  
votre micro-ordinateur
- 40 Le guide des micro-ordinateurs
- 47 Un peu de vocabulaire

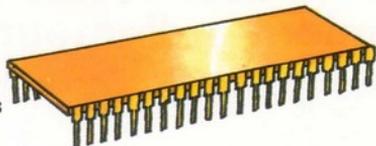


# Avant-propos

L'Introduction à la micro-informatique s'adresse à tous ceux qui veulent pénétrer dans le monde passionnant de la micro-informatique. Abondamment illustrée, elle vous fera connaître les limites et possibilités de ces étranges petites machines que sont les micro-ordinateurs. Et découvrir les secrets de leur fonctionnement.



Les micro-ordinateurs, de petites dimensions, ont de nombreuses applications : jeux, production d'images ou même - avec certaines machines - création de sons, de musique. Vous pouvez aussi les employer à faire très rapidement des calculs fort complexes, tenir votre agenda, répertorier vos disques, vos diapositives...



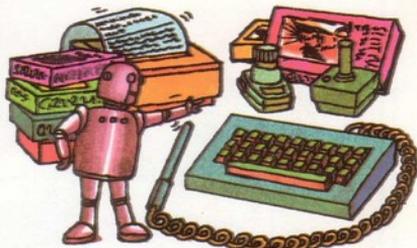
Dans la seconde partie, vous découvrirez le fonctionnement d'un micro-ordinateur et comment il arrive à créer images et sons. Des illustrations très claires vous dévoileront l'intérieur de la machine : ainsi les minuscules « puces » de silicium qui assurent le traitement des informations. Vous apprendrez aussi comment certains micro-ordinateurs peuvent converser avec d'autres à distance et la façon dont ils parviennent à commander robots ou appareils électriques.



Bien que, pour débiter, vous n'ayez besoin que d'un poste de télévision ordinaire relié à votre ordinateur, vous pouvez acheter des équipements complémentaires : un crayon optique pour écrire directement sur l'écran ou des accessoires à utiliser avec des jeux, également décrits dans ce guide pratique.



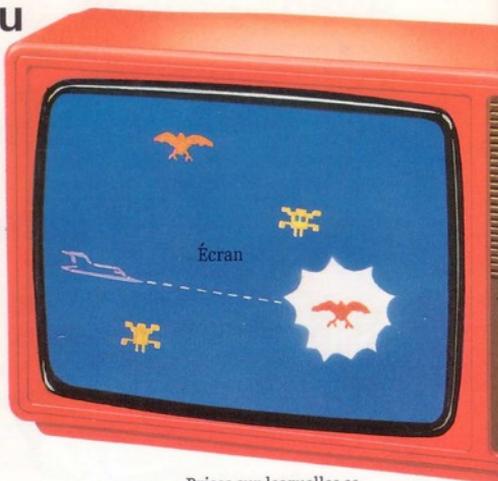
La première partie de ce livre explique comment se servir d'un micro-ordinateur et comment écrire un programme lui indiquant ce qu'il aura à faire. Cela constituera pour vous une introduction concrète et amusante au BASIC, langage de programmation le plus répandu actuellement. Et si vous avez la chance de posséder un micro-ordinateur, vous pourrez créer vos propres programmes de jeux en vous inspirant de ceux que nous vous donnons comme exemples.



Page 40, un guide d'achat présente les micro-ordinateurs les plus répandus, afin d'avoir une vision exhaustive des machines actuellement commercialisées. Enfin, un lexique vous explique les termes habituellement employés en micro-informatique.

# A la découverte du micro-ordinateur

Les micro-ordinateurs grand public ne ressemblent pas tous exactement à celui de l'illustration. Mais, de l'un à l'autre, il existe finalement peu de différences importantes. La plupart d'entre eux sont constitués d'une simple console à relier à un téléviseur normal. Certains sont pourvus d'écrans spéciaux appelés consoles de visualisation. Tous les appareils récents sont fournis avec une notice d'utilisation relativement détaillée qu'il est recommandé de lire avec attention avant de brancher la prise !



## Console

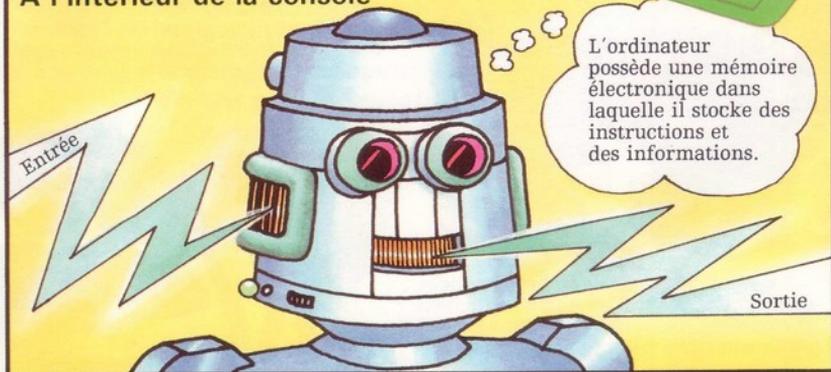
C'est par son intermédiaire que vous communiquerez vos instructions et informations au micro-ordinateur.

Toutes les parties importantes d'un ordinateur sont contenues à l'intérieur de la console.

Prises sur lesquelles se branchent le cordon TV et l'alimentation électrique.



## A l'intérieur de la console



Le « cerveau » d'un ordinateur se trouve généralement sous le clavier et comporte deux parties : la mémoire et l'unité centrale (Central Processing Unit, CPU) qui assure le traitement des informations. Mais, sans programme, sans instructions, le CPU ne peut rien

faire. C'est pourquoi, vous devrez « entrer » en mémoire un programme ainsi que des informations ou des données sur lesquelles travailler. Programmes et données sont appelés « entrées ». Les informations délivrées par l'ordinateur se nomment « sorties ».

## L'écran

Tout ce que vous tapez sur le clavier, ainsi que les informations, dessins ou graphiques délivrés par l'ordinateur, s'affichent sur l'écran. Certains modèles permettent de créer des images colorées. Ce qui présente un intérêt à condition de pouvoir les connecter à un téléviseur couleur.

Cordon secteur du téléviseur

Cordon de connexion de l'ordinateur au téléviseur

Cordon d'alimentation de l'ordinateur

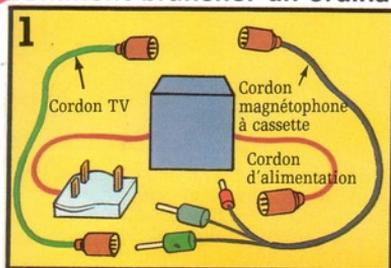
Transformateur

Certains ordinateurs peuvent produire des sons et de la musique.

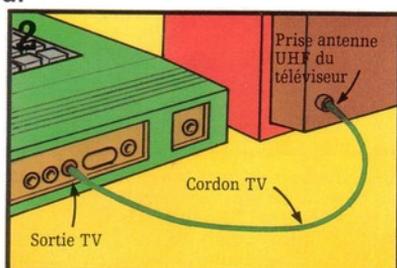
## L'alimentation secteur

Le transformateur sert à abaisser le courant du secteur et à le stabiliser pour qu'il puisse passer dans le micro-ordinateur.

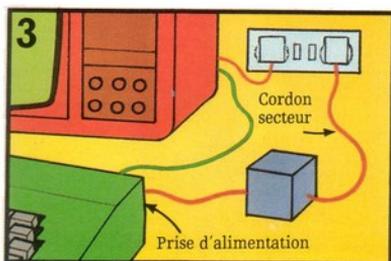
## Comment brancher un ordinateur



De nombreux micro-ordinateurs ont trois cordons. Le premier pour relier la console au téléviseur, le deuxième pour l'alimentation électrique et le troisième pour connecter un lecteur-enregistreur de cassettes\*.



Pour raccorder la console au téléviseur, débranchez le fil de l'antenne TV. Connectez ensuite une extrémité du cordon TV à la prise marquée TV sur l'ordinateur, et l'autre à celle marquée UHF sur le téléviseur.



En branchant le transformateur, faites bien attention à ne pas vous tromper de prise ni à inverser les cordons. Assurez-vous que la TV est branchée; ensuite seulement, mettez les deux appareils sous tension.



Sélectionnez un canal que vous n'utilisez pas pour capter une chaîne. Réglez ce dernier jusqu'à ce qu'apparaisse la mention « READY » ou « OK » sur l'écran. Ce signal varie suivant les modèles.

\* Le magnétophone servira à stocker des programmes ou des données pour l'ordinateur (v. page 16).

# Qu'est-ce que la programmation

Que vous vouliez utiliser votre ordinateur comme jeu vidéo, ou simplement pour additionner plusieurs nombres, vous devez lui fournir un programme, c'est-à-dire une suite d'instructions qui lui indiquent tout ce qu'il doit faire. Et des langages spéciaux existent, pour écrire ces programmes informatiques. Ils sont constitués de mots et de

symboles que l'ordinateur sait reconnaître et convertir en signaux électriques. Les instructions des programmes sont stockées dans la mémoire de l'ordinateur pour être ensuite traitées par l'unité centrale. Le programme s'appelle le logiciel (« software » en anglais). L'ensemble des éléments physiques se nomme le matériel (« hardware »).

## 1 Dire à l'ordinateur ce qu'il doit faire

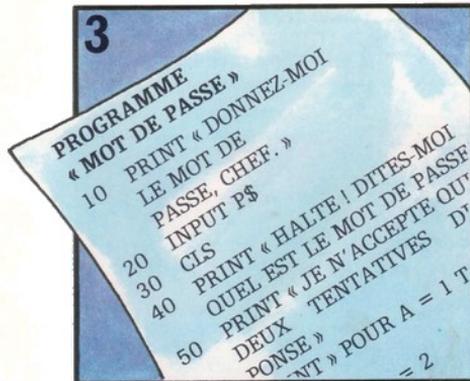


Un ordinateur ne peut exécuter les tâches que vous lui confiez que dans la mesure où vous lui avez dit avec précision ce qu'il doit faire et dans quel ordre il doit opérer. Le programme ci-dessus explique à un robot dont le cerveau est un



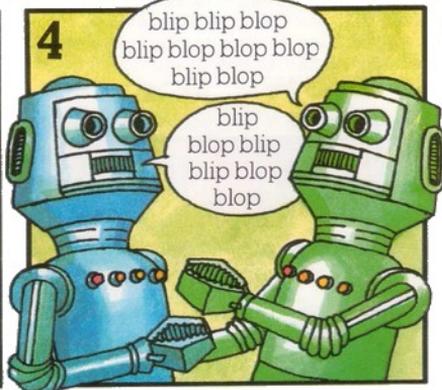
ordinateur, comment il doit s'y prendre pour repeindre une fenêtre. Le robot ne pourra pas exécuter ce travail. Le programme aurait dû préciser qu'il fallait prendre le pot de peinture et le pinceau avant de monter sur l'échelle !

## 3



Cet extrait de programme\* est en BASIC, le langage le plus utilisé par les micro-ordinateurs. Chaque ordinateur contient un jeu d'instructions spéciales, appelé interpréteur, qui traduit le langage de programmation, que vous employez, en code machine.

## 4



Toutes les opérations, à l'intérieur de l'ordinateur, sont faites en code machine. Chaque mot ou élément de ce code consiste en une suite d'impulsions électriques qui circulent dans l'ordinateur.

## La mémoire de l'ordinateur

Un ordinateur a deux sortes de mémoires. L'une conserve en permanence les instructions qui lui permettent de traiter et de restituer informations, données et instructions. L'autre, dite mémoire vive, stocke provisoirement vos données et vos programmes. Son contenu est altéré par une coupure de courant.

La mémoire permanente est appelée ROM (Read Only Memory) ou mémoire morte. L'ordinateur peut seulement y puiser ou y lire des données et des instructions. Vous ne pouvez pas vous en servir pour stocker des informations. L'interpréteur est stocké en ROM.

▲ La ROM est comme un manuel contenant des instructions. L'ordinateur peut le lire, mais ni l'effacer ni y stocker de nouvelles informations.

La mémoire temporaire se nomme RAM (Random Access Memory) ou mémoire vive (MEV). On l'appelle aussi souvent mémoire de lecture-écriture. Tout ce que vous entrez dans votre appareil est stocké ou « écrit » à l'intérieur, pour que vous puissiez le ressortir ou y apporter des modifications.

▲ La RAM, c'est comme un bloc-notes. L'ordinateur peut écrire dessus ou relire ce qu'il a noté. Mais elle s'efface dès que vous débranchez la machine.



### Puissances de mémoire

Les micro-ordinateurs ont des mémoires plus ou moins puissantes, qui se mesurent en nombre de mots stockables écrits en code machine. Chaque « caractère » est appelé un « byte », et 1 024 bytes forment un kilobyte ou 1 K.

Un kilobyte correspond à environ 500 instructions ou données BASIC. Ce qui est suffisant pour stocker un programme simple. En revanche, des programmes plus élaborés peuvent occuper jusqu'à 8 ou 16 K de RAM. Aussi, pour de nombreux ordinateurs, on peut se procurer des packs-mémoire supplémentaires.

# Un coup d'œil sur la console

Le clavier d'un ordinateur ressemble beaucoup à celui d'une machine à écrire. Les lettres et les chiffres sont disposés de la même manière. (Attention, cependant : certains claviers correspondant à ceux des machines à écrire américaines, ne comportent pas de caractères accentués, et les touches de ponctuation ainsi que le A, le Z, le W, le Q... ne sont pas placées de la même façon que sur les claviers français.)

## Touches alphabétiques

Sur la plupart des ordinateurs, vous devez taper un programme en utilisant des touches portant des symboles spéciaux et en écrivant, lettre à lettre, chaque mot avec les touches alphabétiques.

Certains ordinateurs ont des touches différentes.



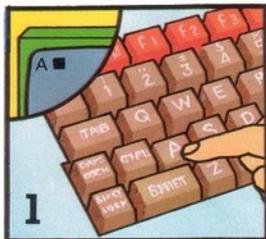
Cette étoile sert de signe de multiplication.

Touche majuscules et blocage majuscules

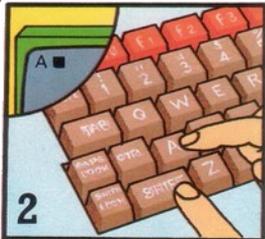
**Barre d'espace**  
Sert à mettre un espace entre des mots ou des symboles.

Touche avec symbole de division

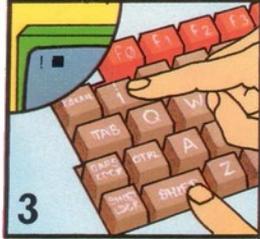
## Utiliser la touche majuscules



De nombreux micro-ordinateurs ne savent écrire qu'en lettres majuscules.



Quelques appareils comportent des minuscules. Pour écrire en majuscules, il faut tenir la touche « shift » enfoncée et appuyer sur la lettre à taper.

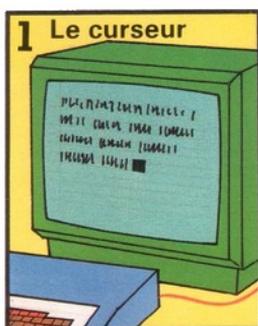


Lorsqu'une touche comporte plusieurs caractères ou symboles, le caractère du bas s'obtient en appuyant simplement sur cette touche, tandis que celui du haut ne peut s'obtenir qu'en pressant simultanément sur « shift ».

### Touches programmables

Ce sont des touches spéciales que vous pouvez programmer vous-même pour exécuter « automatiquement » certaines opérations. Ainsi la coloration des lettres s'affichant sur l'écran. Tous les appareils ne sont pas munis de ces touches.

Le zéro, sur un ordinateur, est représenté par un 0 barré pour le distinguer du o majuscule.



Le curseur est un repère lumineux qui se déplace sur l'écran lorsque vous tapez, pour indiquer la position du prochain caractère.



Si vous voulez changer ou effacer quelque chose, placez le curseur sur les caractères à remplacer ou à effacer à l'aide des touches de contrôle spéciales.

### Un autre type de micro-ordinateur

Ce clavier fait environ le quart de celui représenté à gauche. C'est en fait la taille et la disposition des touches qui déterminent la taille et la forme, car les circuits internes sont peu volumineux. Une place doit toutefois être prévue sur la console pour les prises dans lesquelles viennent se brancher l'imprimante ou le magnétophone à cassette.



Touches de contrôle pour le déplacement du curseur

#### Touche « retour chariot »

A la fin de chaque ligne de programme, vous pressez cette touche pour commencer une nouvelle ligne. Elle a aussi pour fonction d'entrer dans la mémoire vive de l'ordinateur la ligne que vous venez de taper. Cette touche s'appelle soit NEW LINE, soit ENTER ou CR.

#### Touche d'effacement

Vous pouvez effacer vos erreurs simplement à l'aide de cette touche. Sur certaines machines, elle s'appelle aussi RUBOUT ou ERASE.

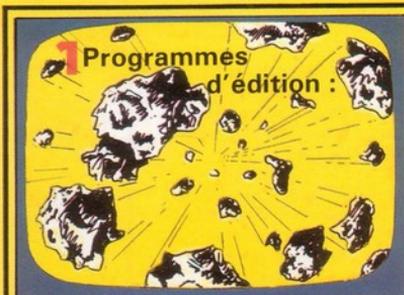
Ce type d'ordinateur est équipé d'un clavier dont les touches ne s'enfoncent pratiquement pas lorsque vous les pressez. Beaucoup d'entre elles portent des mots entiers (instructions, fonctions...) en BASIC, de telle sorte que vous n'avez pas à les taper lettre à lettre.

# Des programmes sur mesure

On trouve des programmes dans des revues ou des livres spécialisés éditant des listes, c'est-à-dire des programmes détaillés ligne à ligne. Vous pouvez aussi acheter des cassettes ou des disquettes contenant des logiciels. Si vous vous sentez l'âme d'un créateur, essayez de concevoir vos propres programmes. Les programmes doivent être écrits dans un langage convenant à votre modèle. Le BASIC est le plus utilisé; mais différentes versions existent, véritables « dialectes » comportant des instructions spéciales. Le programme ne peut tourner si le dialecte est inadapté ou s'il comporte des erreurs.

## Où trouver des programmes

Vous trouverez dans la plupart des magasins de journaux des revues de micro-informatique contenant des



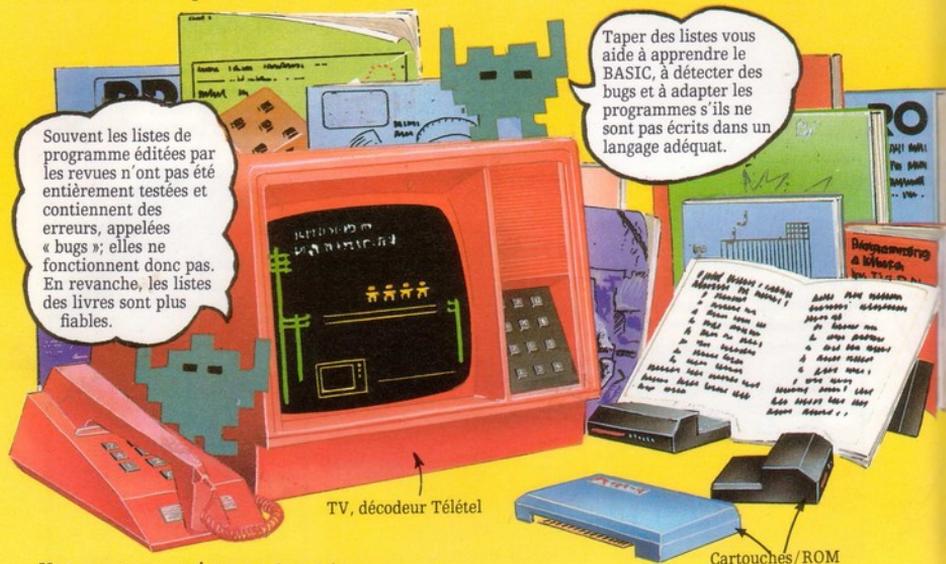
## un large éventail

Vous pouvez acheter toutes sortes de programmes, depuis les jeux d'Arcade avec des images en couleurs et des bruitages élaborés jusqu'à des jeux plus traditionnels comme les échecs.

listes. Certaines sont conçues pour un type précis d'ordinateur. D'autres sont compatibles avec tous. Enfin, des séries entières de livres sont consacrées à l'édition de listes de programmes de jeux.

Souvent les listes de programme éditées par les revues n'ont pas été entièrement testées et contiennent des erreurs, appelées « bugs »; elles ne fonctionnent donc pas. En revanche, les listes des livres sont plus fiables.

Taper des listes vous aide à apprendre le BASIC, à détecter des bugs et à adapter les programmes s'ils ne sont pas écrits dans un langage adéquat.

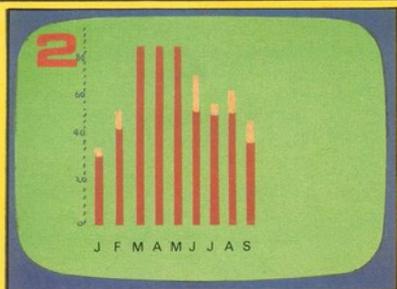


TV, décodeur Télétel

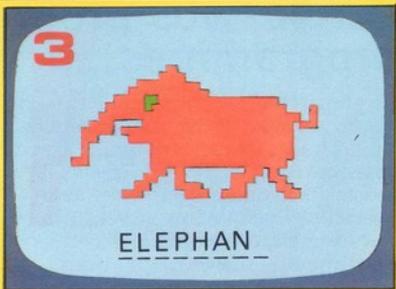
Cartouches / ROM

Vous pouvez aussi connecter votre appareil avec un centre Télétel pour avoir accès aux services videotex par le biais du téléphone ou du réseau Transpac. Télétel vous permet de consulter des banques de données, des pages d'information, de faire du télétraitement, de la recherche de renseignements...

Certains ordinateurs sont prévus pour recevoir des « cartouches de programmes » (logiciels stockés dans des mémoires mortes). Il vous suffit de connecter la cassette sur une prise spéciale montée sur l'appareil pour que le programme soit immédiatement disponible.



Pour des utilisations domestiques, des programmes vous aident à tenir vos comptes bancaires, votre agenda, votre carnet de rendez-vous ou des fichiers (disques, livres, timbres...).



Les programmes de formation constituent des aides pédagogiques appréciables : depuis l'orthographe et les mathématiques jusqu'à l'initiation ou au perfectionnement à la pratique de langues étrangères.

**QUELQUES TRUCS**

1. Assurez-vous que votre programme est écrit dans un langage BASIC convenant à votre micro-ordinateur.
2. A titre d'exemple, 1K de mémoire est suffisant pour contenir un programme de 40 lignes.
3. Si un programme sur disquette ou sur cassette compatible avec votre appareil ne fonctionne pas, écrivez à la société qui le commercialise et mettez-vous immédiatement en relation avec le magasin qui l'a vendue.



Les clubs d'utilisateurs vous donnent l'occasion de rencontrer d'autres personnes intéressées par la micro-informatique et d'échanger programmes et expériences. Pour connaître quels sont les clubs proches de votre domicile, écrivez à une revue spécialisée.



Vous pouvez vous procurer des programmes enregistrés sur cassette. Ce qui implique un magnétophone. Vous pouvez aussi en avoir sur disquettes (*floppy disk*). Ce sont les mêmes, mais ils coûtent un peu plus cher et nécessitent l'emploi d'un lecteur-enregistreur de

disquettes (*drive*). Toutes les boutiques spécialisées ainsi que certaines sociétés de vente par correspondance commercialisent ces disques et cassettes. Vous trouverez leurs coordonnées dans les revues ou les guides d'exposition comme le SICOB ou Micro-Expo.

# Écrivez vos propres programmes...

La plupart des micro-ordinateurs fonctionnent en BASIC, un langage de programmation permettant une grande diversité d'applications. Bien sûr, il existe d'autres langages, et certains spécialistes pensent que le PASCAL, par exemple, est beaucoup plus performant... mais c'est une autre histoire. Voici deux programmes en BASIC. Ce langage, constitué de symboles et de mots, est très facile à apprendre. Le meilleur moyen pour vous initier est d'écrire quelques programmes et de les tester. De nombreuses notices expliquent le BASIC.



Vous trouverez de nombreux termes du BASIC dans le programme « Mot de passe » à droite. Ce programme permet de stopper l'infiltration d'agents ennemis dans une société secrète...

Chaque ligne de programme porte un numéro. Ces numéros vont de 10 en 10 afin de laisser la place à l'introduction ultérieure de nouvelles lignes de programme sans avoir à tout changer. L'ordinateur exécutera le programme en suivant l'ordre ainsi défini.

```
PROGRAMME « MOT DE PASSE »
10 PRINT « DONNEZ-MOI LE MOT DE
PASSE, CHEF. »
20 INPUT P$
30 CLS
40 PRINT « HALTE! QUEL EST LE
MOT DE PASSE? »
50 PRINT « VOUS AVEZ DROIT À
DEUX RÉPONSES. »
60 FOR A = 1 TO 2
70 INPUT A$
80 IF A$ = P$ THEN A = 2 : GOTO 130
90 PRINT « FAUSSE RÉPONSE! »
100 NEXT A
110 PRINT « RESTEZ DEHORS,
VOUS ÊTES UN ESPION »
120 END
130 PRINT « ENTREZ, VOUS ÊTES DES
NÔTRES »
140 END
```

Pour l'utiliser, tapez exactement sur votre clavier le texte de cette liste. A la fin de chaque ligne, pressez la touche RETURN (ENTER, NEWLINE ou CR). Puis, tapez RUN pour que l'ordinateur exécute ce programme.

## 1 Concevoir un programme



Vous faites l'essai d'un kart fabriqué par un ami. La direction vient de lâcher et vous foncez vers une mare aux canards. Pour freiner à temps, il vous faudra absolument taper un code spécial que votre ami a oublié de vous donner. Vous pourrez faire cinq tentatives avant de plonger...

## 2

- 
1. Afficher le titre et les instructions
  2. Tirer une lettre au hasard
  3. Demander au joueur de taper une lettre
  4. Si la réponse est correcte, afficher un message et stopper le programme
  5. S'il est encore temps, donner une autre chance et retourner en 3
  6. Sinon, afficher « SPLATCH » et stopper

La première étape de l'écriture d'un logiciel consiste à mettre noir sur blanc ce que vous attendez du programme... en

clair. Ensuite à isoler chaque étape ou idée et à faire ressortir ce que l'ordinateur devra traiter pas à pas.

PRINT ordonne d'afficher sur l'écran tout ce qui se trouve entre guillemets.

INPUT lui indique d'attendre un message de vous et de le mettre dans une case de la mémoire que vous avez choisi d'appeler P\$.

\$ signifie que l'expression qui suit est une « chaîne » de caractères.

CLS efface l'écran... seulement.

FOR ... TO indique à l'ordinateur combien de fois il va devoir exécuter les lignes 70 à 90.

IF ... THEN indique à l'ordinateur ce qu'il devra faire SI une certaine condition est vraie (vérifiée). Dans ce cas, GOTO indique qu'il faudra passer à la ligne 130. Si la condition n'est pas vérifiée, l'ordinateur passe à la ligne suivante, en 90.

END indique à l'ordinateur qu'il est bien arrivé à la fin du programme.

HALTE, QUEL EST LE MOT DE PASSE?  
VOUS AVEZ DROIT À DEUX RÉPONSES.  
? MICROFOU  
FAUSSE RÉPONSE !  
? MICROGÉNIAL  
ENTREZ, VOUS ÊTES DES NÔTRES

Lorsque vous faites tourner ce programme, l'ordinateur commence par vous demander de choisir un mot de passe et le stocke dans sa mémoire. Il efface ensuite l'écran pour ne laisser aucune trace, puis affiche la question posée au visiteur. En ligne 70, avec INPUT, la machine attend un message du visiteur. Il compare ensuite celui-ci avec le mot placé en mémoire et si c'est le même, il affiche « Entrez... »

3

```
10 PRINT « JEU DU KART FOU »
20 PRINT
30 PRINT « LA DIRECTION DU KART A LÂCHÉ »
40 PRINT « ET »
50 PRINT « VOUS ALLEZ FAIRE UN PLONGEON »
60 PRINT « DANS LA MARE AUX CANARDS À MOINS »
70 PRINT « QUE VOUS NE TROUVIEZ LA LETTRE »
80 PRINT « QUI PERMET DE FREINER, VOUS POUVEZ »
90 PRINT « FAIRE 5 TENTATIVES, PAS PLUS... »
100 LET C$=CHR$(64 + INT(RND(1)* 26 + 1))
110 FOR G=1 TO 5
120 INPUT G$
130 IF G$=C$ THEN G=5:GOTO 190
140 IF G$<C$ THEN PRINT « APRÈS »;
150 IF G$>C$ THEN PRINT « AVANT »;
160 PRINT G$
170 NEXT G
180 PRINT « SPLAAAAAAAAAATSH »
190 PRINT « L'EAU EST-ELLE BONNE? »
200 END
210 PRINT « SCRIIIIIIIII... »
220 PRINT « VOUS AVEZ FREINÉ À TEMPS, BRAVO ! »
230 END
```

Ces programmes ne peuvent pas tourner sur tous les micro-ordinateurs, car certains utilisent des variantes du BASIC. Les instructions qui changent le plus souvent sont CLS qui sert à effacer l'écran et RND qui crée un nombre de manière aléatoire. Si ce programme ne fonctionne pas sur votre ordinateur, consultez votre notice pour savoir quelles sont les instructions équivalentes à utiliser.



4

```
JEU DU KART FOU
LA DIRECTION DU KART A LÂCHÉ ET
VOUS ALLEZ FAIRE UN PLONGEON
DANS LA MARE AUX CANARDS À MOINS
QUE VOUS NE TROUVIEZ LA LETTRE
QUI PERMET DE FREINER. VOUS POUVEZ
FAIRE 5 TENTATIVES, PAS PLUS...
? W
APRÈS W
? Z
AVANT Z
? Y
SCRIIIIIIIII.....
VOUS AVEZ FREINÉ À TEMPS, BRAVO !
```

Maintenant traduisez chaque étape de ce programme en BASIC. Tapez-le ligne par ligne et vérifiez très minutieusement ce que vous venez d'entrer. Voilà ce qui apparaît lorsque ce programme tourne. La lettre affichée

juste après le point d'interrogation constitue la réponse que vous avez faite à l'ordinateur. Vous en saurez plus sur la façon dont fonctionnent les programmes en tournant cette page.

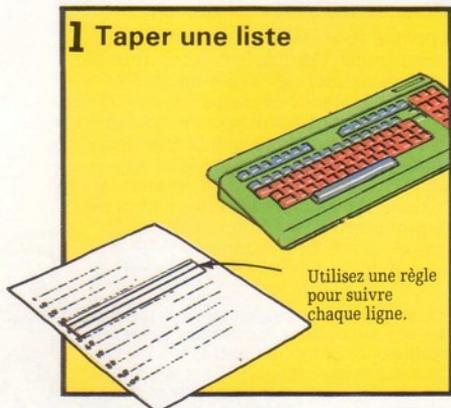
Voir aussi pages 49 à 96.

# Faites-les fonctionner sans peine

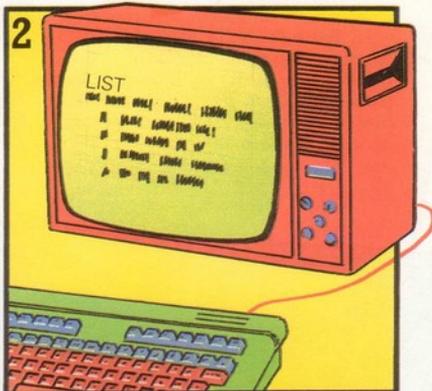
Quand vous copiez une liste sur votre clavier, toutes les lignes du programme vont se mettre en mémoire. Vous pouvez charger un programme dans votre appareil en utilisant un magnétophone à cassette. Si le programme ne tourne pas après que vous avez tapé RUN, il contient

certainement un bug. Certains bugs causent des « plantages » complets et stoppent l'exécution des programmes. D'autres problèmes inattendus peuvent aussi se produire. Vous trouverez ci-contre quelques indications sur les bugs les plus répandus.

## 1 Taper une liste

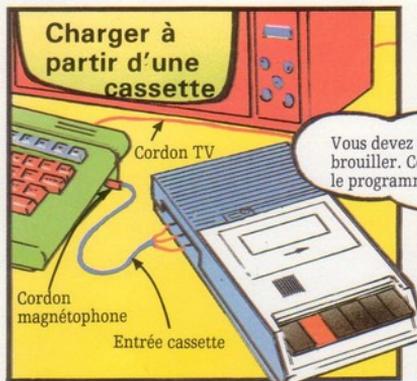


La ponctuation et l'espace sont fort importants, de même que les lettres et les nombres. Une liste doit être tapée avec minutie. Chaque ligne qui apparaît sur l'écran doit être vérifiée. Si le



programme ne tourne pas, tapez LIST pour le visualiser sur l'écran, vérifiez-le une nouvelle fois et portez toutes les corrections avant de le refaire fonctionner..

## Charger à partir d'une cassette



Vous devez voir l'écran TV se brouiller. Ce qui indique que le programme se charge.



Un programme est conservé dans une cassette sous la forme de séries d'impulsions à haute fréquence. Pour le charger sur votre ordinateur, connectez le magnétophone en suivant la notice explicative, placez le potentiomètre entre 7 et 8 et la tonalité dans les aigus. Quand vous tapez LOAD suivi du nom du

programme entre guillemets et que vous pressez la touche PLAY du magnétophone, le chargement doit normalement se faire. Cela peut prendre de quelques secondes à plusieurs minutes. Si vous n'arrivez pas à opérer un chargement, refaites plusieurs tentatives en ajustant le volume et la tonalité.

## Des bugs dans le programme

Cette illustration montre un programme dans lequel se trouvent de nombreux bugs. Le plus commun d'entre eux, c'est la faute de frappe. Si vous ne tapez pas dans un BASIC correct, l'ordinateur ne va pas comprendre vos ordres. Ce type de bug est appelé « erreur de syntaxe »

Oubli de guillemets : les mots après PRINT doivent être mis entre guillemets.

(syntax error). L'appareil délivre sur l'écran des messages d'erreur quand il rencontre quelque chose qu'il n'arrive pas à reconnaître... soit pendant que vous tapez votre programme, soit après RUN ou LIST. Vous en trouverez ici quelques exemples.

20 « VOUS POUVEZ DONNER CINQ RÉPONSES »

10 PRINT "COMBIEN Y A-T-IL DE CROCODILES DANS LA RIVIÈRE?"

Erreur de syntaxe : oubli de PRINT pour dire à l'ordinateur d'afficher cette phrase sur l'écran.

30 LET A=6

40 FOR N=1 TO 5

Instruction inefficace : FOR...TO et NEXT constituent deux parties d'une même instruction demandant à l'ordinateur de répéter plusieurs fois la série d'instructions comprise entre les lignes 40 et celle portant NEXT. Cela constitue une boucle. La partie NEXT de la commande, qui devrait se trouver en ligne 80, n'a pas de numéro de ligne et l'ordinateur ne peut pas la retrouver.

Erreur de syntaxe : ce mot est mal orthographié, la machine ne peut donc pas le reconnaître.

Numéro de ligne manquant : il n'existe pas de ligne 130 dans le programme.

50 ONPUT G

60 IF G=A THEN N=5 :GOTO 130

70 PRINT "FAUX"

NEXT N

90 PRINT "SNAP! VOUS AVEZ ÊTÉ HAPPÉ PAR UN CROCODILE!"

100 END

110 PRINT "OUF! MAINTENANT, FUYEZ A TOUTE PAGAIE!"

120 OK - C'EST TOUT

Erreur de syntaxe : l'expression n'existe pas en BASIC et l'ordinateur ne peut donc pas le comprendre. Il aurait fallu écrire END.

Voici le programme correct :

```
10 PRINT "COMBIEN Y A-T-IL DE CROCODILES DANS CETTE RIVIÈRE?"
20 PRINT "VOUS POUVEZ DONNER JUSQU'À CINQ RÉPONSES"
30 LET A=6
40 FOR N=1 TO 5
50 INPUT G
60 IF G=A THEN N=5:GOTO 110
70 PRINT "FAUX"
80 NEXT N
90 PRINT "SNAP! VOUS VENEZ D'ÊTRE HAPPÉ PAR UN CROCODILE!"
100 END
110 PRINT "OUF! MAINTENANT FUYEZ A TOUTE PAGAIE!"
120 END
```

# Comment conserver un programme

Après avoir entré un programme sur votre ordinateur, vous avez intérêt à l'enregistrer sur un magnétophone à cassette pour éviter qu'il ne soit perdu lorsque vous couperez l'alimentation de votre appareil (la RAM s'efface). Vous pouvez aussi utiliser des disquettes, ce qui est encore mieux si vous devez stocker un grand nombre de programmes. Bien entendu, l'emploi d'une imprimante est

conseillé pour garder des traces sur papier de vos données ou pour sortir des listes. Le magnétophone, le lecteur de disquettes et l'imprimante se branchent sur des sorties de l'ordinateur appelées portes d'accès. Celles-ci contiennent des circuits spéciaux (*interfaces*) qui convertissent les codes machine de l'ordinateur en signaux électriques compatibles avec le matériel utilisé.

## Cassettes

SAVE signifie que vous demandez la sauvegarde d'un programme. LOAD permet de charger un programme déjà enregistré.



Un compteur de bande est pratique pour repérer le début de chaque programme.

Des cassettes de courte durée sont préférables, car elles obligent à ne stocker qu'un petit nombre de programmes et permettent une recherche relativement rapide.

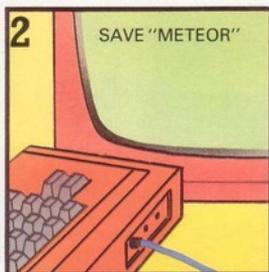
Dans la majorité des cas, un magnétophone à cassette courant fera l'affaire. Attention, cependant : certains ordinateurs, comme le Thomson TO 7, impliquent un enregistreur spécial. Il existe dans le commerce des bandes « données » (*data tapes*), recommandées pour l'enregistrement de programmes ou

d'informations numériques (provenant d'ordinateurs). Mais, *a priori*, n'importe quelle bande de qualité moyenne peut être utilisée. Sauvegarder ou charger des programmes est quelquefois délicat : après plusieurs essais infructueux, nettoyez la tête d'enregistrement/lecture.



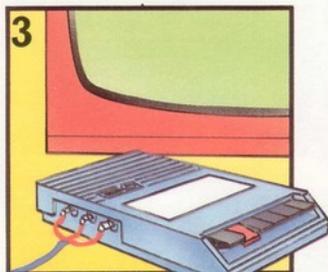
**1 Sauvegarder un programme sur bande**

Connectez le magnétophone à cassette sur l'ordinateur en vous conformant à votre notice. Vérifiez la manière dont vous avez établi les connexions (attention aux inversions de fiches).



**2 SAVE "METEOR"**

Ensuite, tapez SAVE suivi du nom de votre programme entre guillemets, pressez simultanément les touches RECORD et PLAY du magnétophone.



Lorsque la bande passera devant les têtes d'enregistrement du magnétophone, le programme sera sauvegardé sous forme d'impulsions magnétiques.

## L'imprimante



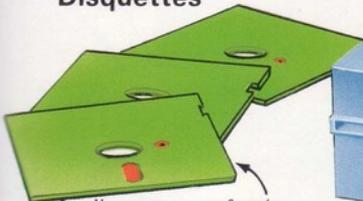
Cette imprimante est très rapide et fournit une bonne qualité de frappe.

Celle-ci coûte beaucoup moins cher, mais travaille plus lentement et sort une moins bonne impression.

Connectée à votre ordinateur, une imprimante permet d'éditer des listes de vos programmes, de copier des fichiers ou des données ou de créer des graphiques. De nombreux ordinateurs utilisent une interface standard appelée RS232 pour connecter l'imprimante (il existe deux types de sorties RS232 : l'une appelée « Série », l'autre « Parallèle » incompatibles entre elles).

Les informations sauvegardées sur papier sont appelées *hard-copy*, copie papier. Vous pouvez tirer de nombreuses copies d'un même programme pour les distribuer à vos amis. Certaines imprimantes travaillent à très grande vitesse : plusieurs lignes par seconde !

## Disquettes

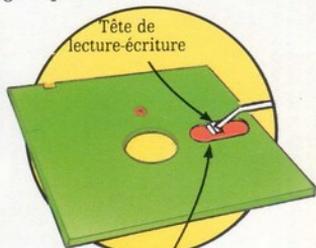


Les disquettes sont enfermées dans une enveloppe de protection, car elles risqueraient de se détériorer au moindre contact avec un corps dur ou gras.

Les disquettes stockent les programmes comme les magnétophones à cassette. La surface des disquettes est lisse, sans sillon. La sauvegarde et le chargement se font à l'intérieur de lecteurs-enregistreurs



Lecteur de disquettes (Disk drive)



La disquette tourne à l'intérieur de son enveloppe de protection.

connectés aux ordinateurs. La tête d'enregistrement-lecture se déplace rapidement à la surface de la disquette à travers une fente pratiquée dans l'enveloppe de protection. Cette tête peut lire toutes les données enregistrées sur le disque ou bien en écrire de nouvelles.

## Pour en savoir plus



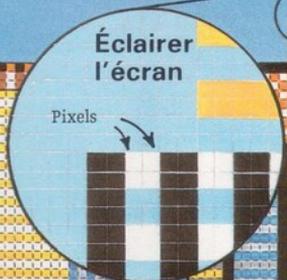
Normalement, lorsque vous conservez un programme, il faut lui donner un nom et le noter dans un répertoire. Pour recharger un programme, il suffira d'en indiquer le nom à l'ordinateur avec éventuellement une instruction particulière (LOAD, RUN...).

Lorsque vous chargez un programme à partir d'une disquette ou d'une cassette, une copie se fait en mémoire. Vous pouvez modifier cette copie ou utiliser des données différentes sans altérer la version stockée dans la cassette ou la disquette.

# Dessins, graphismes et animations

Les ordinateurs produisent des images en allumant sur l'écran de petits points, appelés *pixels*. Les images produites par l'informatique portent le nom de graphismes.

Il est possible de les créer à partir d'instructions simples et de programmes relativement faciles. Certains ordinateurs autorisent même la réalisation d'images en dessinant directement sur l'écran avec un crayon optique ou avec un accessoire spécial appelé table graphique.



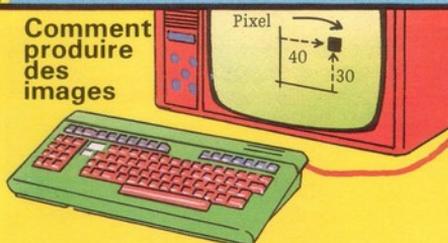
## KING KONG



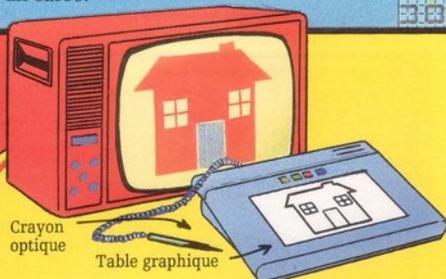
Si vous regardez de près une image informatique, vous percevrez tous les pixels qui la composent. De nombreux ordinateurs peuvent produire des images en couleur s'ils sont connectés à des téléviseurs couleurs : les graphismes sont alors formés de surfaces de pixels de différentes teintes.

Les caractères (comme les lettres ou les symboles) sont, eux aussi, composés de pixels. L'ordinateur divise l'écran en rangées de carrés invisibles, et chaque caractère se forme par l'éclairage de différentes combinaisons de pixels dans un carré.

### Comment produire des images



Dans un programme graphique, vous indiquez à l'ordinateur quel pixel allumer en tapant ses coordonnées. Elles donnent pour chaque point de l'image sa hauteur sur l'écran et sa position en partant de la gauche selon une mesure exprimée en nombre de pixels.



Une table graphique est constituée d'une surface sensible à la pression et recouverte d'une grille. Vous placez votre image à reproduire sur cette grille et retracez ses contours avec un crayon spécial. Automatiquement, l'ordinateur enregistre les coordonnées de chaque pixel.

## Qualité de l'image

1



Pour allumer chaque pixel séparément, il faut une capacité de mémoire très importante. Aussi beaucoup d'ordinateurs opèrent par groupes entiers et tous les pixels d'un groupe ont la même couleur.

2



Un ordinateur doté d'une plus grande capacité de mémoire créera des images à partir de plus petits groupes de pixels et générera ainsi des images plus réalistes appelées graphismes à haute résolution.

3



Un ordinateur sait aussi animer des images en allumant et éteignant des groupes de pixels les uns à la suite des autres si rapidement qu'il en résulte une réelle impression de mouvement.

Le nombre de caractères que l'on peut mettre sur un écran dépend du nombre de carrés, et celui-ci varie suivant les ordinateurs. Avec un modèle qui divise l'écran en 30 colonnes et 20 rangées, il est possible d'afficher 20 lignes de 30 caractères.



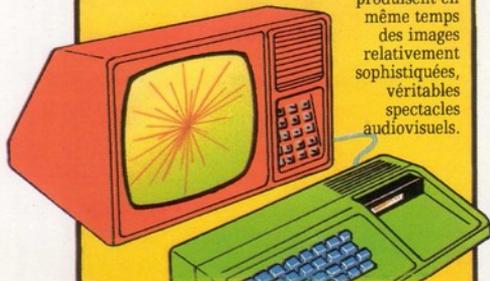
Vous pouvez écrire sur l'écran en le touchant avec un crayon optique. En traçant une ligne, le rayon envoie l'ordre à l'ordinateur d'éclairer des pixels le long de cette ligne. Le crayon peut « voir » ce qui est éclairé sur l'écran et donner à l'ordinateur la position des pixels par rapport au faisceau lumineux.

# Bruitages et musique

De nombreux ordinateurs peuvent jouer des morceaux de musique ou générer des effets sonores. Certains, même, sont doués de la parole. Les appareils capables de créer des sons disposent de circuits spéciaux appelés synthétiseurs. Sur de nombreux appareils, ces synthétiseurs doivent être achetés en option.

## 1 Faire de la musique

Vous pouvez acquérir des programmes musicaux en cassette. Certains produisent en même temps des images relativement sophistiquées, véritables spectacles audiovisuels.



A moins que vous ne préfériez concevoir vos propres programmes musicaux en entrant des instructions correspondant à chaque note. Certains ordinateurs peuvent émettre plusieurs notes à la fois ainsi que leurs harmoniques. Ils disposent ainsi de plusieurs « voix ».

2



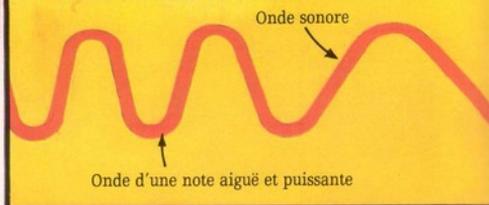
Il y a encore une autre façon de dire à l'ordinateur quelles notes jouer. Des programmes inscrivent en effet sur votre écran une portée qu'il vous suffira de remplir à l'aide d'un crayon optique.

Pour ordonner à l'ordinateur de produire un son particulier, il suffit de taper une commande comme SOUND ou BEEP, suivie de chiffres précisant la note à jouer et sa durée.



Lorsque vous tapez une instruction pour produire un son, l'ordinateur envoie un message au synthétiseur en langage machine indiquant le son à générer. Le synthétiseur produira un signal électrique qui sera amplifié avant d'être transmis à un haut-parleur, dont le rôle

## Plus avant dans le son



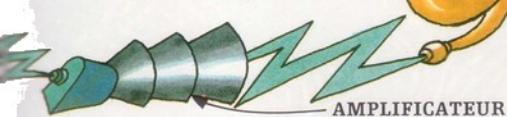
Les vibrations de l'air engendrées par le haut-parleur s'appellent des ondes sonores. Celles-ci ont des formes différentes suivant les sons reproduits. Par exemple, à une note de musique très aiguë et puissante correspondra une onde de fréquence élevée. La valeur des

Vous pouvez programmer certains ordinateurs pour imiter des bruits de pas ou des sonneries de téléphone.

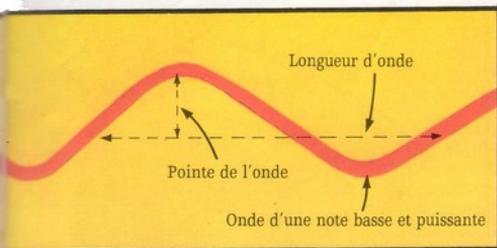


### LE HAUT-PARLEUR

Certains appareils sont équipés d'un haut-parleur intégré dans la console. Mais la plupart utilisent le haut-parleur du téléviseur dont il est possible de régler le volume.



consiste à transformer le signal électrique en ondes sonores. Des signaux différents provenant du synthétiseur feront vibrer le haut-parleur à des vitesses distinctes pour produire des sons différents.



« pointes » de l'onde indiquera la puissance de la note tandis que la longueur d'onde correspondra à la hauteur du son (grave ou aigu). Les variations du volume et des pointes d'un son durant une période de temps, sont appelées « enveloppes ».

## 1 Des machines qui parlent

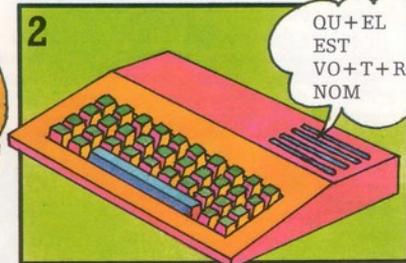


CE  
CH + AT  
EST  
UN  
RO + BOT



Il est plus difficile pour un ordinateur de parler que de produire de la musique, car les sons qui permettent de reproduire la parole sont beaucoup plus complexes. La plupart des mots sont composés de plusieurs sons; ainsi CH + AT ou RO + BOT.

## 2



En utilisant un synthétiseur, l'ordinateur assemble des sons pour créer des mots selon des règles très précises stockées en mémoire. La « synthèse vocale » a été surtout utilisée pour les aveugles ne pouvant pas lire les données affichées sur un écran ou pour l'enseignement de la lecture aux jeunes enfants.

## 3



Il est encore plus difficile pour un ordinateur de comprendre la parole. Et les programmes qui permettent de reconnaître tous les sons et le sens des mots sont extrêmement sophistiqués. Parce que chaque personne a une voix bien à elle.

# Voyage à l'intérieur de la console

Les illustrations de ces deux pages vous montrent l'intérieur d'un petit ordinateur. Tous les modèles possèdent au moins les éléments de base décrits ici; certains – les plus sophistiqués – comprennent davantage de composants.

Les parties les plus importantes d'un ordinateur sont les pastilles, appelées aussi « puces » – les quatre boîtiers noirs bien visibles sur le dessin. C'est sous la forme d'impulsions électriques que les informations circulent dans l'ordinateur. Des circuits en métal, imprimés sur une plaque d'isolant, guident ces signaux à travers les puces. Mais pour en savoir plus, lisez attentivement les pages qui suivent.

## Boîtier ROM

Le programme permanent contenant toutes les données de base essentielles au fonctionnement de l'ordinateur est stocké ici.

## Régulateur de tension

Il règle le courant provenant du bloc d'alimentation secteur et délivre une basse tension (quelques volts).

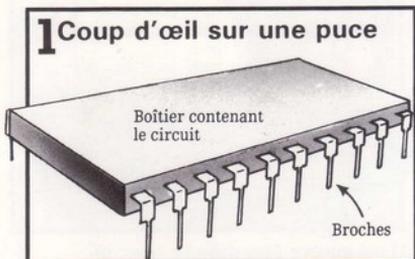
## Circuit imprimé

Support isolant qui porte à sa surface les liaisons conductrices nécessaires à la connexion des diverses composantes de la machine. D'autres composants électroniques, disposés sur ce support, servent à contrôler les échanges de courant électrique : résistances, transistors, condensateurs...

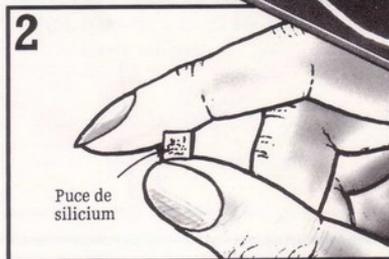
Résistance

## Boîtier RAM

Mémoire vive dans laquelle l'ordinateur stocke provisoirement le programme et les données que vous lui avez entrés ou qu'il a produits.



La puce est une petite surface, en général de silicium, protégée par un boîtier, et capable de supporter un ou plusieurs circuits intégrés, très compliqués et précis. Quant aux broches de métal qui sortent du boîtier, elles servent aux entrées et sorties des signaux électriques.



Cette illustration vous montre la taille véritable d'une puce... pas plus épaisse qu'un ongle et pouvant contenir plusieurs dizaines de circuits différents gravés dans le silicium. Le nom exact d'une puce est circuit intégré. (Integrated Circuit, IC.)

### Unité logique spéciale

Il s'agit d'une puce spéciale, qui contient des instructions ou des commandes spécifiques à chaque modèle d'ordinateur.

**Connecteurs** pour raccorder l'écran TV, l'alimentation secteur, un magnétophone à cassette, une imprimante...

### Modulateur

Cet appareil convertit les informations délivrées par l'ordinateur et destinées à être affichées sur écran, en signaux de télévision.

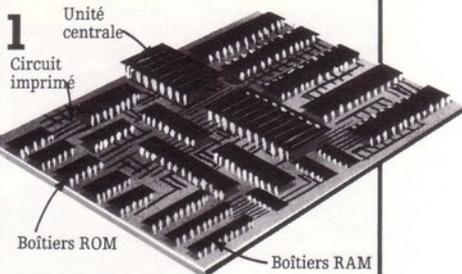
### Microprocesseur

L'unité centrale organise tout le fonctionnement. Elle exécute les instructions de vos programmes et contrôle la circulation des informations provenant de la console, entrant ou sortant de la RAM, de la ROM. Elle contient une « horloge » à quartz battant plusieurs millions de fois par seconde et régulant la circulation des signaux électriques dans l'ordinateur.

**Connecteurs** C'est là que s'enfichent des modules complémentaires (mémoires additionnelles, interfaces...). Les connecteurs de métal conduisent les signaux électriques provenant ou se dirigeant vers ces modules.

Ces trous correspondent aux endroits où les circuits de métal traversent le circuit imprimé pour se poursuivre sur l'autre face.

## Et les gros ordinateurs?



Les ordinateurs les plus puissants ont des capacités mémoires très importantes et comportent un très grand nombre de puces. Ici, un circuit imprimé portant plus d'une dizaine de puces permettant d'accroître les capacités de traitement ou de mémorisation.



Les gros ordinateurs utilisés dans l'industrie comportent des centaines de plaques de circuits imprimés réunies dans des coffrets disposés en rangées dans des locaux réservés au matériel informatique. Ces calculateurs peuvent réaliser une grande quantité d'opérations en même temps.



Un mini-ordinateur est une version plus petite de gros systèmes. Il ne comporte que quelques boîtiers de circuits imprimés et est plutôt utilisé pour réaliser un type défini de tâches (comptabilité ou la constitution et la gestion de banques de données).

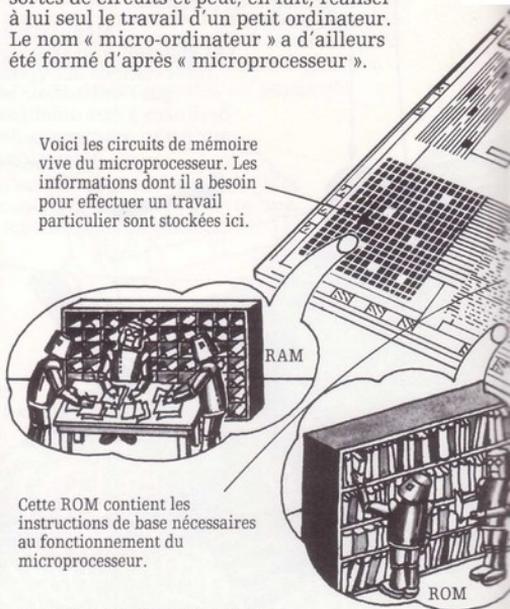
# A l'intérieur d'une puce

Chaque puce qui équipe un ordinateur se compose de circuits conçus pour une utilisation spécifique. L'illustration de droite montre deux puces de silicium très agrandies. L'une est un microprocesseur et l'autre une ROM. Vous pouvez voir les dessins des différentes connexions sur chacune d'elles. Ces connexions sont si petites et si compliquées que des tests très sévères doivent être faits à plusieurs stades de fabrication pour éliminer les puces défectueuses.

## Microprocesseur

Un microprocesseur est souvent assimilé à un véritable ordinateur tenant dans une puce de silicium. Il possède différentes sortes de circuits et peut, en fait, réaliser à lui seul le travail d'un petit ordinateur. Le nom « micro-ordinateur » a d'ailleurs été formé d'après « microprocesseur ».

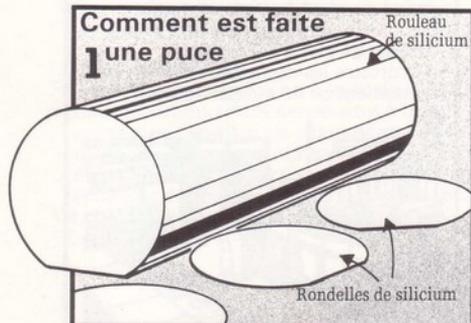
Voici les circuits de mémoire vive du microprocesseur. Les informations dont il a besoin pour effectuer un travail particulier sont stockées ici.



Cette ROM contient les instructions de base nécessaires au fonctionnement du microprocesseur.

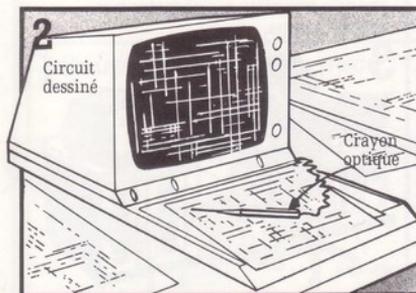
Les circuits, à l'intérieur du microprocesseur, sont reliés entre eux par des « bus ». Les broches métalliques qui sont reliées au circuit imprimé et qui permettent de connecter le microprocesseur à d'autres puces s'appellent aussi des bus.

## Comment est faite une puce



Les puces sont fabriquées à partir de cristal de silicium très pur. Ce dernier est façonné en rouleau, puis coupé en rondelles de 100 millimètres de diamètre sur 0,5 d'épaisseur. Chaque rondelle permet de réaliser 500 puces. Le silicium, très abondant dans la nature, constitue donc une matière première peu chère.

2



Les ordinateurs eux-mêmes participent à la conception des circuits des puces. Ici, un crayon optique est utilisé pour modifier le dessin d'un circuit : celui-ci sera ensuite réduit par l'ordinateur à la dimension exacte d'une puce.

3



Les dessins des circuits sont reproduits sur les puces par procédé photographique avant que les rondelles de silicium soient introduites dans un four. Ensuite un ensemble de phénomènes chimiques permettra aux circuits de se graver dans le silicium.



Tous les calculs et traitements « informatiques » sont réalisés dans les circuits de l'unité logique et arithmétique (Arithmetic and Logic Unit, ALU).

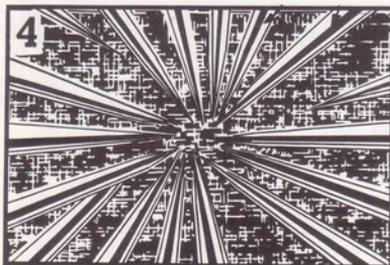
:0058

L'horloge contrôle la vitesse à laquelle travaille le microprocesseur.

### Puce de mémoire



Les circuits d'une puce destinée à mémoriser des données ou des instructions sont semblables à des centaines de petites boîtes empilées les unes à côté des autres. Dans une ROM, chaque boîte contient une information, tandis que les casiers des RAM sont vides.



Plusieurs circuits peuvent être réalisés dans une même puce... ce qui peut prendre plusieurs semaines. Enfin les rondelles sont testées sous microscope pour repérer les puces défectueuses.

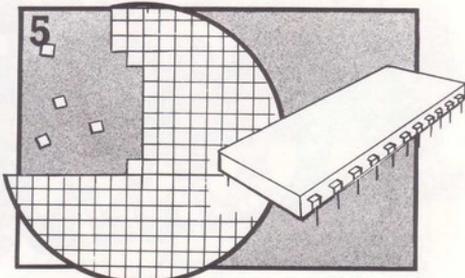
## Le monde des micro-processeurs



Les microprocesseurs sont utilisés pour contrôler ou assurer le fonctionnement de toutes sortes d'appareils. Ils sont si petits et légers qu'ils peuvent être placés dans des objets tels que des appareils photos ou des montres.



Les microprocesseurs ont remplacé de nombreux systèmes électromécaniques aujourd'hui démodés, tels ceux qui équipaient les machines à laver ou les téléphones à pièces.

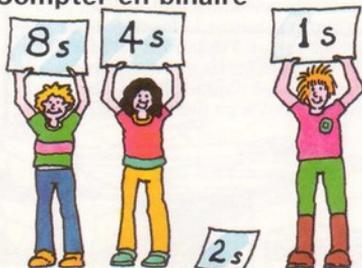


Les rondelles de silicium sont ensuite découpées en puces à l'aide d'un diamant. Les puces défectueuses sont éliminées, les autres placées dans des boîtiers de protection qui peuvent être raccordés à des circuits imprimés.

# Comment une puce travaille

Les circuits qui se trouvent à l'intérieur d'une puce comprennent des centaines de petits composants, appelés transistors, traversés par des impulsions électriques très rapides. Certains de ces transistors sont connectés entre eux de telle sorte qu'ils forment des portes laissant passer ou stoppant – sous certaines conditions – le courant. Grâce à de tels dispositifs, sont créées les suites

## Compter en binaire

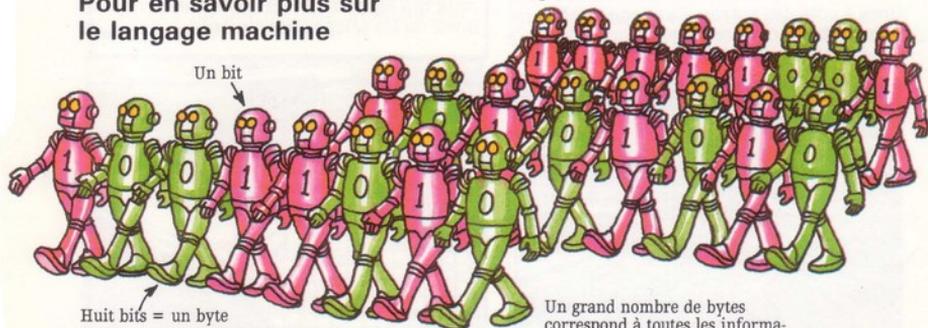


$$(1 \times 8) + (1 \times 4) + (0 \times 2) + (1 \times 1) = 13$$

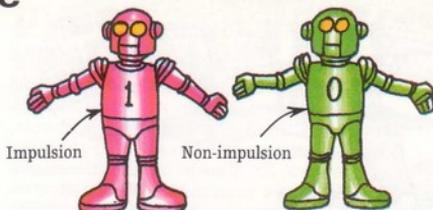
13 s'écrit 1101 en binaire

Les nombres binaires s'écrivent à l'aide de deux chiffres, 1 et 0. Ils sont placés de manière à représenter des valeurs particulières : 1 placé en 1<sup>re</sup> position à droite vaudra 1, placé en 2<sup>e</sup> position il vaudra 2, en 3<sup>e</sup> 4, en 5<sup>e</sup> 8, en 6<sup>e</sup> 16, etc.

## Pour en savoir plus sur le langage machine



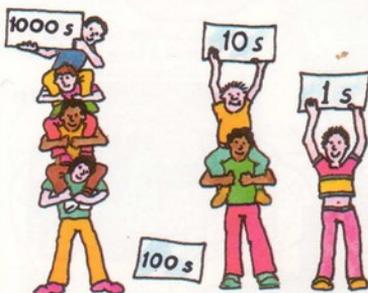
Un bit  
Huit bits = un byte  
Chaque impulsion ou absence d'impulsion est appelée un « bit », abréviation de *binary digit* (chiffre binaire). De nombreux ordinateurs utilisent des groupes de 8 bits pour représenter des parties d'informations. Chaque groupe de 8 bits s'appelle un « byte » ou octet.



Impulsion

Non-impulsion

d'impulsions ou de « non-impulsions ». Le code machine se compose de deux signaux seulement : l'un correspond à une impulsion, l'autre à une absence d'impulsion. Il s'agit d'un code binaire dans lequel ces deux signaux sont représentés par les valeurs 1 et 0.



$$(4 \times 1000) + (0 \times 100) + (2 \times 10) + (1 \times 1) = 4021$$

Le système de numération décimale est fondé sur le même principe que le binaire, mais fait appel à 10 chiffres (de 0 à 9). Ces chiffres ont aussi des valeurs différentes suivant la place qu'ils occupent dans le nombre qu'ils composent.

Un grand nombre de bytes correspond à toutes les informations utilisées par l'ordinateur.

Il y a 256 façons de placer des 1 et des 0 dans un byte. C'est grandement suffisant pour représenter chaque symbole et caractère du clavier par un seul byte, ainsi que pour coder de la couleur et des sons.

## Comment l'ordinateur traite les informations

L'ordinateur traite les informations en faisant passer des impulsions en code machine à travers différents circuits combinatoires, appelés portes, qui réagissent de différentes manières.

Les points auxquels aboutissent les signaux s'appellent des terminaux. Certaines portes reçoivent deux signaux mais n'en émettent qu'un. Voici trois types de portes :



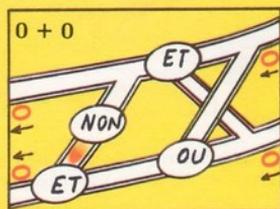
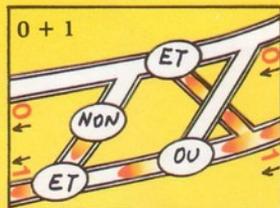
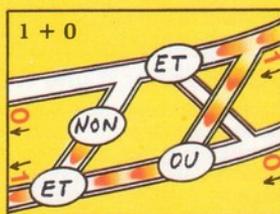
Une porte ET émet une impulsion seulement si elle en a reçu une sur chacun de ses terminaux.

Une porte OU n'envoie une impulsion que si elle en a reçu au moins une sur un de ses terminaux.

Une porte NON ne possède qu'un terminal. Elle n'émet une impulsion que si elle n'en reçoit pas.

## Comment l'ordinateur fait une addition

Cette illustration montre comment le calculateur se sert des portes pour additionner des chiffres binaires ( $1 + 1$ ,  $1 + 0$ ,  $0 + 1$ ,  $0 + 0$ ). Et toutes les instructions ou les commandes que vous donnez à un ordinateur se réduisent à des opérations aussi simples que cela :



La porte ET reçoit deux impulsions mais n'en émet qu'une.

Addition de  $1 + 1$

Chaque impulsion est dirigée vers les deux portes à la fois.

La porte OU reçoit deux impulsions et en émet une.

La porte NON reçoit une impulsion et n'en renvoie donc aucune.

L'impulsion est envoyée à la fois à la porte NON et vers la sortie du circuit.

La porte ET ne reçoit qu'une impulsion et est ainsi bloquée.

Parcours des impulsions le long des connexions à travers les portes.

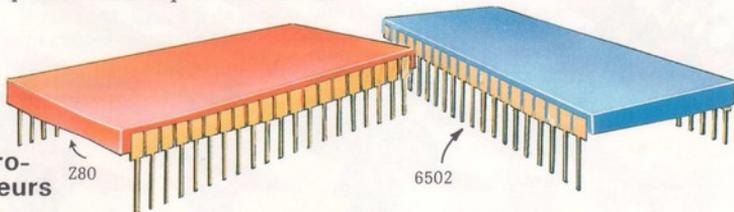
10 en binaire est égal à 2 en décimal.

Voici quelques autres exemples d'addition utilisant le même circuit.

# A propos des puces

La façon dont travaille un ordinateur dépend du type de puce qu'il contient. Les appareils équipés des mêmes microprocesseurs utilisent le même code machine. Les interpréteurs traduisant le BASIC en langage machine sont stockés dans des ROM. C'est pourquoi, lorsque des ordinateurs disposent de ROM identiques, ils peuvent comprendre le même langage BASIC... Le BASIC est un langage qui se rapproche plus de celui des hommes que du code machine. Aussi les ordinateurs ont-ils besoin de « traducteurs », d'« interpréteurs » assez complets. Des langages comme le BASIC sont déjà relativement évolués. Il en existe d'autres, moins évolués, mais plus proches du langage machine, et qui peuvent donc être interprétés plus facilement par l'ordinateur.

## Les microprocesseurs



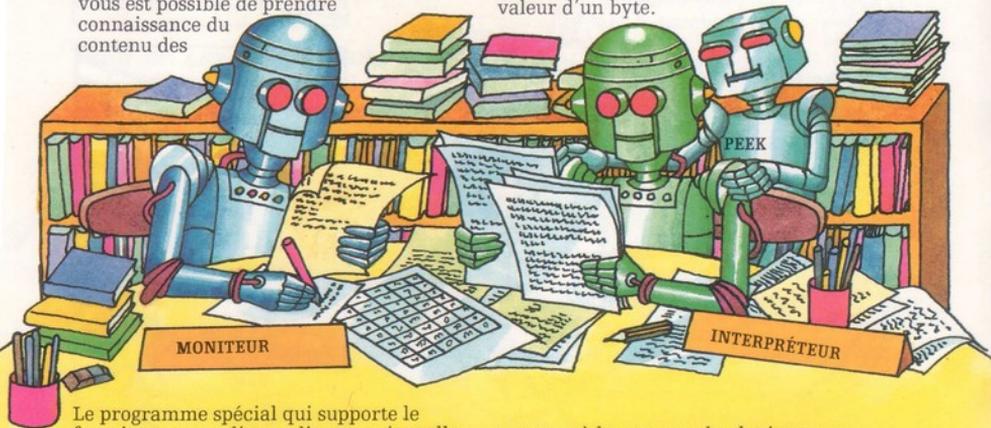
Parmi les divers types de microprocesseurs actuellement sur le marché, deux d'entre eux se retrouvent dans la majorité des micro-ordinateurs grand-public : le Z80 et le 6502. Les différences qui existent entre eux tiennent essentiellement à l'organisation de leurs circuits internes, car leur aspect

est identique. Les instructions concernant les opérations à mener, inscrites dans les ROM, doivent être écrites dans le langage machine correspondant au microprocesseur à commander. Ainsi, pour le Z80, les instructions devront être en code machine Z80.

## A l'intérieur d'une ROM

La ROM est organisée comme un assemblage de petites cases portant toutes des numéros, des adresses et contenant chacune une information (un *byte*). Il vous est possible de prendre connaissance du contenu des

cases-mémoires en tapant l'instruction PEEK, suivie d'un numéro d'adresse que vous aurez choisi parmi ceux indiqués dans la notice de votre appareil. PEEK vous permet en fait d'afficher à l'écran un nombre décimal correspondant à la valeur d'un byte.



Le programme spécial qui supporte le fonctionnement d'un ordinateur s'appelle le moniteur. Il est stocké dans une ROM avec l'interpréteur. Une des fonctions du moniteur consiste à détecter quelle touche a été pressée sur le clavier : recevant un signal électrique de la touche sur laquelle vous avez appuyé, il le

compare à la « carte » du clavier pour trouver quel byte correspond à cette touche. La plupart des micro-ordinateurs utilisent le Code ASCII (American Standard Code For Information Interchange) dans lequel chaque caractère ou signe correspond à un byte.

## A l'intérieur de la RAM

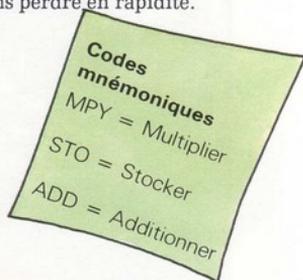
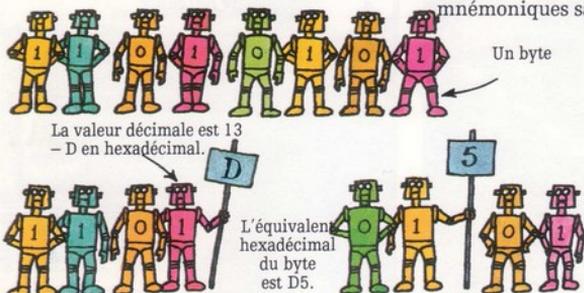
<b>Systèmes variables :</b>	Ces emplacements contiennent des informations telles que les caractères qui se forment sur l'écran.
<b>Mémoire programme :</b>	Là où est stocké le programme.
<b>Liste de visualisation :</b>	L'ordinateur y range une copie en code-machine de ce qui est affiché à l'écran.
<b>Variables :</b>	C'est ici que les données sont stockées.
<b>Ligne en cours et mémoire disponible :</b>	Elles contiennent la ligne qui est en cours de frappe.
<b>Pile calculateur :</b>	Le CPU réalise ici quelques opérations.
<b>Mémoire de secours :</b>	Espace qui sert à mémoriser des informations qui ne peuvent tenir dans d'autres RAM trop pleines.
<b>Pile machine :</b>	Le CPU utilise cet espace pour stocker, par exemple les numéros de ligne de programme.
<b>Pile GOSUB :</b>	Ici est mémorisé le numéro de la ligne de programme sur laquelle doit revenir l'ordinateur après une instruction GOSUB.

La RAM est divisée en plusieurs parties stockant différentes sortes d'informations. Vous pouvez utiliser l'instruction PEEK pour la RAM comme pour la ROM. Pour changer le contenu d'une case mémoire de la RAM, il suffit de taper POKE suivi d'une adresse et d'un nombre (cette opération est impossible avec une ROM qui est une mémoire permanente). La notice de votre ordinateur vous indiquera quelle partie de la RAM « poker » sans risquer de bloquer vos programmes ou de modifier des éléments essentiels au fonctionnement de votre appareil. Vous pouvez utiliser POKE dans l'espace réservé aux systèmes variables. Vous pouvez aussi stocker des données dans la mémoire de secours et les retrouver avec PEEK.

## Programmer en langage machine

Si vous vous « amusez » à programmer en langage machine, votre ordinateur peut exécuter vos instructions très rapidement, sans avoir à les interpréter. Pour les jeux, par exemple, c'est très

utile. Programmer à l'aide de suites de nombres binaires est fastidieux; il est préférable d'utiliser des langages un peu plus évolués faisant appel par exemple aux codes hexadécimaux ou à des codes mnémoniques sans perdre en rapidité.

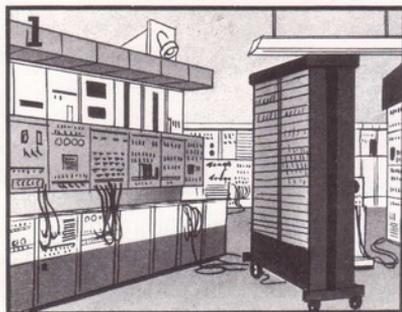


L'hexadécimal est un système de numération à 16 chiffres : les valeurs 10 à 15 sont représentées par les lettres A à F. Un byte de 8 bits peut être écrit sous la forme de deux chiffres hexadécimaux : il suffit pour cela de diviser chaque byte en deux groupes de 4 bits et de transformer ces derniers en chiffres hexadécimaux. Un code mnémonique est constitué par

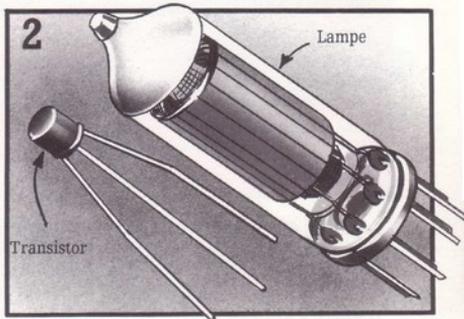
un ensemble de mots abrégés qui correspondent à certaines instructions particulières comprises par l'ordinateur. Chaque abréviation recouvre en fait une « chaîne d'opérations » à exécuter. Les langages de bas niveau sont beaucoup plus faciles à convertir en code machine. C'est pourquoi ils emploient des interpréteurs de taille plus réduite.

# Histoire de la micro-informatique

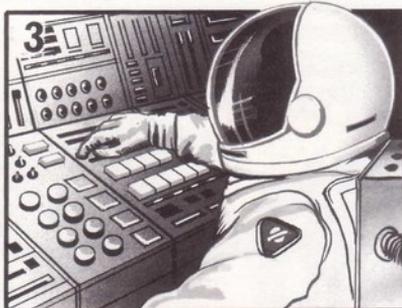
Les premiers calculateurs électroniques furent construits en Grande-Bretagne durant la Seconde Guerre mondiale. A la différence des premières machines à calculer mécaniques, ils étaient programmables et disposaient de véritables mémoires. Ces calculateurs furent utilisés par des chercheurs pour trouver les clés des codes secrets ennemis et pour suivre la trajectoire des avions. Les informations les concernant furent tenues « top secret » durant plusieurs années. Lorsque vint la fin des hostilités, quelques grandes sociétés et des États commencèrent à utiliser des ordinateurs... Mais ils étaient les seuls à pouvoir s'offrir de tels moyens. Depuis, les ordinateurs sont devenus moins chers, plus petits et plus puissants. Cela a conduit au développement de micro-ordinateurs à la portée de tous, même des non-spécialistes.



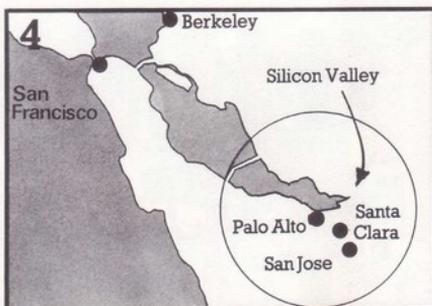
Le premier ordinateur fut construit avant que le transistor ne soit inventé. Ses circuits comportaient environ 18 000 lampes mesurant chacune une dizaine de centimètres de haut, faites de métal et de verre ! Le nombre de pannes était impressionnant et les ingénieurs qui travaillaient avec ces machines passaient un temps fou à localiser les lampes défectueuses...



C'est dans les années cinquante que les transistors sont mis au point aux États-Unis. Capables d'accomplir le même travail que les lampes, ils s'avèrent beaucoup moins chers, plus petits et plus rapides. Peu à peu, les transistors remplacèrent les lampes dans toutes sortes d'appareils électroniques, des postes de radio aux ordinateurs.



Puis, dans les années soixante, le gouvernement américain, engagé dans la conquête de l'espace, a besoin de calculateurs puissants et de petites dimensions pour ses fusées. Il finance la recherche dans le domaine des circuits intégrés, qui consistent en une combinaison de plusieurs transistors à l'intérieur d'une pastille de silicium.



Les puces de silicium connaissent alors un succès énorme et donnent naissance à une nouvelle technique : la micro-électronique. L'endroit où se concentre toute la recherche dans ce domaine fut la Santa Clara Valley, en Californie, aujourd'hui connue sous le nom de Silicon Valley (la vallée du Silicium).

## Les générations de calculateurs

L'ENIAC avait les dimensions d'une maison et pesait 30 tonnes.

Il coûtait un demi-million de dollars et consommait 200 kilowatts.

Il pouvait réaliser deux additions en 3 millièmes de seconde.

Toutes les 7 ou 8 minutes, une lampe tombait en panne !

1945

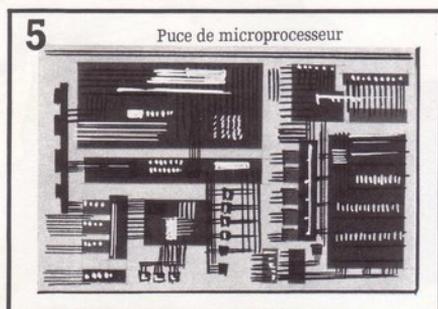
Quatre générations d'ordinateurs se sont succédé, correspondant à des machines de plus en plus petites et puissantes. Les calculateurs à lampes correspondent à la première génération. (Un des premiers d'entre eux, l'ENIAC, nécessita 2 années de construction et fut opérationnel en 1945.) La deuxième génération vit le jour avec l'apparition des transistors. La troisième fit largement appel à l'emploi de puces dans les calculateurs. Enfin, la quatrième correspond à l'invention des microprocesseurs et à l'ultraminutisation des composants.

Une puce est plus petite et plus mince qu'une lentille de contact.

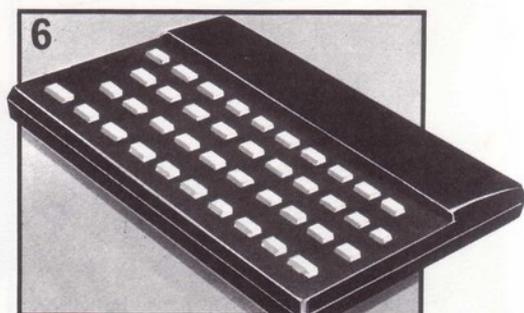
Elle coûte moins de 5 dollars et peut additionner deux nombres en un dix-millionième de seconde.

Elle ne tombe presque jamais en panne et consomme très peu d'énergie.

1980



Un progrès décisif est réalisé en 1971, lorsqu'il devient possible de faire entrer tous les composants d'un ordinateur dans une puce... appelée microprocesseur. Un ordinateur qui, hier, occupait une pièce entière, pouvait désormais tenir sur un morceau de silicium de 5 mm<sup>2</sup>.

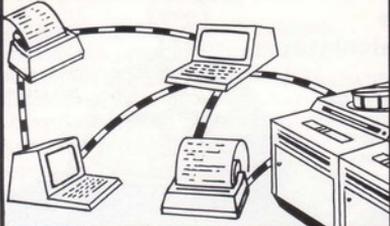


Grâce à cette nouvelle technologie, put être entreprise la fabrication de micro-ordinateurs utilisables pour des travaux ne demandant ni les performances de gros systèmes, ni la compétence de techniciens. Et l'on peut aujourd'hui acheter des ordinateurs de la taille d'un livre et pour le prix de quelques-unes des anciennes lampes.

# Réseaux d'ordinateurs

Vous pouvez mettre en communication deux ordinateurs n'importe où dans le monde, à condition de disposer des connexions nécessaires et de pouvoir se brancher sur un réseau. Vous pouvez utiliser aussi bien votre ligne téléphonique que des réseaux passant par des satellites. Pour se comprendre, les ordinateurs doivent fonctionner avec des programmes spéciaux traduisant les langages et convertissant les vitesses d'émission et de réception des uns et des autres. Cela pour transmettre des informations ou des programmes. Car tout ce qui est dans la mémoire d'un ordinateur peut être transmis à un autre.

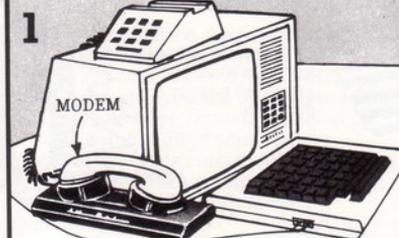
## Utiliser un réseau



Des ordinateurs peuvent communiquer entre eux par des réseaux spécialisés accessibles par l'intermédiaire de lignes téléphoniques normales. Pour les utiliser, il faut se faire reconnaître par un « mot de passe ». On peut ainsi établir des communications entre différents ordinateurs ou périphériques.

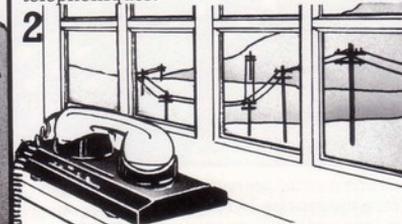
## Un ordinateur en ligne

1

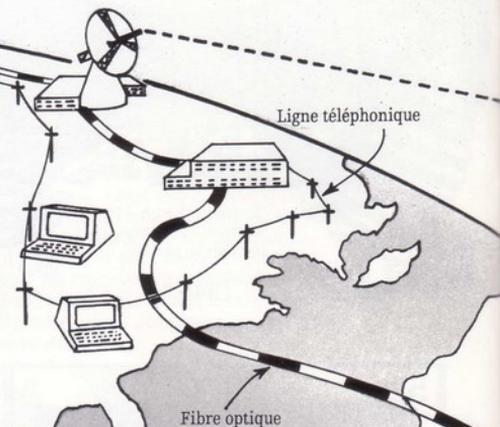


Des ordinateurs peuvent être mis en communication grâce au téléphone, par un MODEM\*. Ce dernier convertit le signal émis en code machine par l'ordinateur en un signal « transportable » par les lignes téléphoniques.

2



Pour recevoir les messages, il faut aussi utiliser un MODEM. Son rôle, cette fois, sera de convertir le signal transmis par les lignes téléphoniques en code machine.



## Envoyer des messages dans le monde entier

Les signaux délivrés par l'ordinateur (signaux « numériques ») peuvent être transmis sous la forme d'ondes radio relayées par des satellites et des stations d'émission-réception. Ces satellites servent aussi à transmettre des communications téléphoniques et des programmes TV dans le monde entier. De nouvelles possibilités de communication sont aujourd'hui offertes par les fibres optiques capables de transporter des informations à la vitesse de la lumière. Les signaux sont convertis en rayons lumineux transmis par les fibres optiques avec très peu de perte d'énergie et grande fidélité.

\*MODEM signifie « MODulateur/DÉModulateur »

## SUPERMARCHÉ

EN PROMOTION AUJOURD'HUI :  
FF.

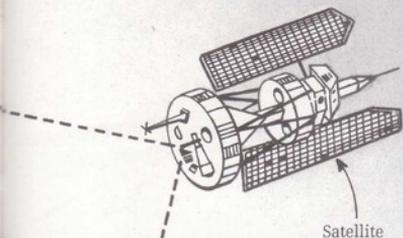
- |                    |                    |
|--------------------|--------------------|
| 1. RAISIN BLANC    | 7,00 LE KG         |
| 2. PAPIER DESSIN   | 0,05<br>LA FEUILLE |
| 3. TROUSSE ÉCOLIER | 11,00              |

Demain, vous pourrez faire vos courses sans vous déplacer, grâce à l'informatique. Il vous suffira de mettre votre ordinateur en rapport avec celui d'un magasin. Vous pourrez alors consulter la liste des articles, passer votre commande et indiquer à l'ordinateur le numéro de votre compte bancaire...

CHER JEAN,

VOUDRIEZ-VOUS BIEN  
M'EXPLIQUER VOTRE  
COMPORTEMENT FACE À  
CET ORDINATEUR

Se connecter sur un réseau peut permettre de transmettre du « courrier électronique ». Au lieu d'écrire une lettre et de la poster, il vous suffit de taper un texte sur le clavier de votre ordinateur, d'établir la connexion avec un autre ordinateur, puis de faire passer votre message sur le réseau.



Satellite

Radar qui reçoit  
les signaux.



Faire son travail à domicile est désormais une réalité : en utilisant des appareils connectés à un réseau aboutissant à un ordinateur central dans lequel sont stockés des informations ou des dossiers.



Depuis peu, certains ordinateurs peuvent se connecter à des centres informatiques vidéotex comme Télétel en France. Grâce à ces réseaux échanger, stocker des informations, entrer en relation avec d'autres ordinateurs dans le monde entier est à votre portée.



Aujourd'hui, quelques établissements d'enseignement sont équipés de véritables réseaux internes de micro-ordinateurs. Les enseignants peuvent, depuis un ordinateur central suivre les progrès de leurs élèves, qui travaillent chacun à son propre rythme.

# Robots et automates

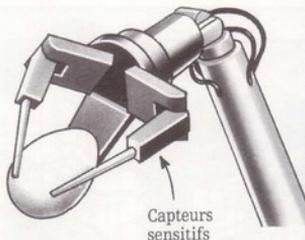
Certains micro-ordinateurs peuvent commander et contrôler divers appareils électriques aussi facilement qu'ils contrôlent leurs écrans d'affichage ou leurs imprimantes. A condition de disposer de la bonne « porte » et de connecteurs sur lesquels se raccorder. Bien sûr, il faut que le signal en code machine du micro-ordinateur soit converti pour convenir à l'appareil. Cette conversion se fait normalement par les circuits d'entrée/sortie de la console. Le microprocesseur assure cette fonction de contrôle.

## Entrée et sortie des signaux de commande



Le cordon de liaison se branche sur la sortie « commande ».

L'ordinateur a besoin non seulement d'envoyer des signaux vers la machine qu'il commande, mais aussi d'en recevoir. La porte de contrôle contient une interface\* par laquelle transitent les ordres (il est possible d'acquérir des circuits d'interface pour la plupart des micro-ordinateurs actuellement commercialisés).



L'ordinateur utilise des capteurs lui indiquant, par exemple, dans quelle position se trouve l'objet qu'il doit déplacer. Un bras de robot contrôlé par ordinateurs dispose de capteurs sensitifs réagissant au moindre contact. Il peut aussi être équipé de cellules sensibles à la lumière.

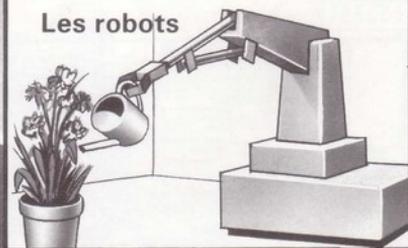
## Contrôler un train miniature

Ce train électrique est contrôlé par un micro-ordinateur relié aux rails par l'intermédiaire d'une porte de contrôle. Il envoie des signaux aux différents éléments du circuit pour changer la position des aiguillages, stopper et faire repartir le convoi.

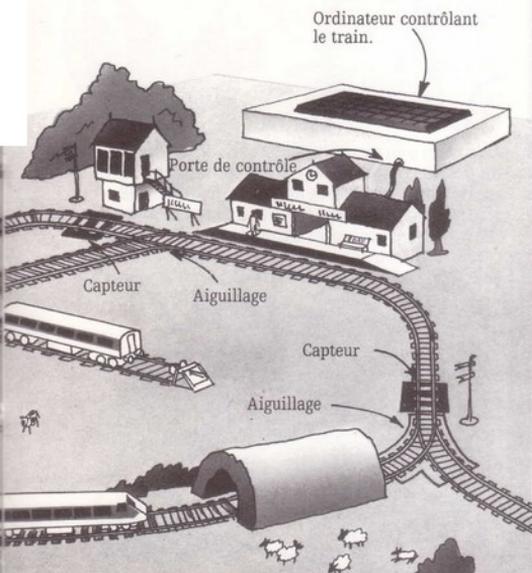


L'ordinateur intervient aussi pour déterminer la vitesse de la locomotive en faisant varier le courant alimentant le circuit. Il peut aussi compter le nombre de tours de circuit fait par le train pour l'arrêter à un moment donné.

## Les robots



Certains ordinateurs peuvent être transformés en robots. Il suffit alors de connecter un bras spécial et d'utiliser des programmes permettant de contrôler la saisie et le déplacement d'objets. Un peu à la manière d'un « cerveau », l'ordinateur ajuste les mouvements du bras en se servant des capteurs.



Lorsque la locomotive passe sur un capteur sensible à la pression, un message est envoyé à l'ordinateur, précisant la position du train sur le circuit et sa direction. En fonction de ces informations et de son programme, l'ordinateur commandera de nouvelles opérations.

## Des ordinateurs dans l'espace

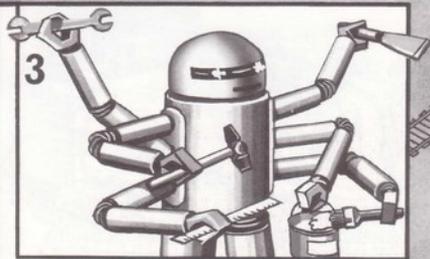
La navette spatiale dispose à son bord d'un micro-ordinateur presque ordinaire. Les sondes non habitées envoyées dans l'espace ont été contrôlées par des ordinateurs embarqués, reliés par radio avec de très puissants calculateurs au sol. Ces



micro-ordinateurs doivent effectuer des opérations compliquées à très grande vitesse. Et ils n'ont pas droit à l'erreur, car ils déterminent la trajectoire, contrôlent les moteurs et leur consommation. Ils sont aussi amenés à surveiller les différentes expériences et à superviser la photographie.



De grands robots sont utilisés dans des chaînes de fabrication. Ils effectuent tout type de tâches. Il s'agit bien de robots, et non pas seulement de machines, car ils peuvent être programmés de différentes façons et sont capables de « prendre des décisions » de manière autonome.



Robot vient du tchèque *robota*, mot utilisé pour la première fois par Karel Capek en 1920. Aujourd'hui, les robots sont utilisés pour des travaux dangereux ou répétitifs. Ils n'ont besoin ni de respirer ni de se nourrir; aussi peuvent-ils travailler dans l'espace ou dans des mines en présence de gaz toxiques.

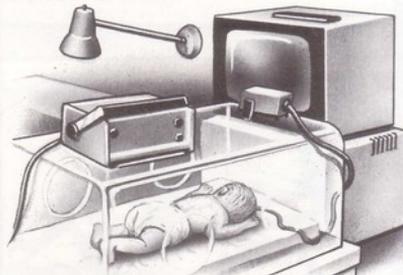
# D'autres utilisations de la micro-informatique

Les micro-ordinateurs ont de nombreuses applications. Petits et puissants, ils parviennent à traiter tous types d'informations converties en code machine, en calculant beaucoup plus vite que le cerveau humain. Stockant un nombre important de données dans un petit espace, ils ont une mémoire totalement sûre... à la différence des hommes.

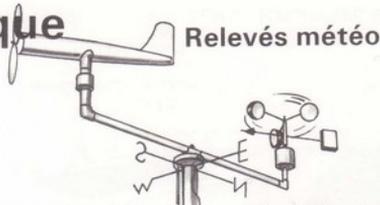
## Des ordinateurs-médecins

AVEZ-VOUS MAL A LA TÊTE?  
OUI  
VOYEZ-VOUS CORRECTEMENT?  
NON  
AVEZ-VOUS LA MIGRAINE?  
NON

De plus en plus, des médecins utilisent des micro-ordinateurs pour établir leur diagnostic. Le patient tape les réponses aux questions posées par l'ordinateur. Celui-ci les compare à une liste qu'on lui a adressée en mémoire afin d'établir un diagnostic et de prescrire d'éventuels soins.



Dans de nombreux pays, des médecins travaillant dans des maternités utilisent des systèmes construits autour de micro-ordinateurs, destinés à surveiller les enfants prématurés ayant des difficultés d'oxygénation. Ces bébés respirent grâce à une machine qui insuffle l'air adéquat dans leurs poumons : un excès d'air endommagerait ceux-ci ; un manque serait dangereux pour le cerveau.



Des micro-ordinateurs sont installés dans de petites stations météorologiques pour traiter les données relevées par différents instruments et pour envoyer les résultats dans des centres plus importants.

## Aider les handicapés



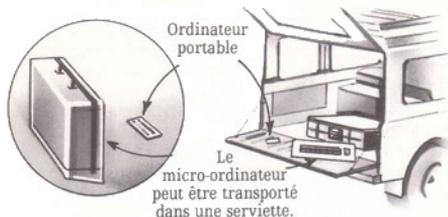
Les personnes muettes ou sourdes peuvent communiquer grâce à l'ordinateur. Il existe aussi des consoles spéciales pour les handicapés physiques. Un mouvement ou une pression légère d'un doigt ou d'une autre partie du corps suffit pour commander l'affichage d'une lettre ou d'un dessin.

## Dessin assisté par ordinateur

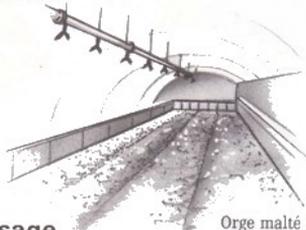


Un ordinateur peut créer et reproduire des dessins en trois dimensions, les faire pivoter pour permettre de les regarder sous des angles différents. En architecture, on peut ainsi dessiner un pont, puis demander à l'ordinateur de calculer les contraintes subies par les pièces et les éléments à déplacer.

## Ordinateurs portables

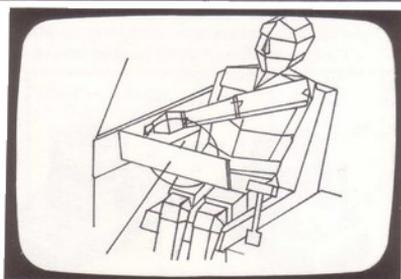


Des personnes qui travaillent en extérieur, comme des géologues faisant de la recherche pétrolière, utilisent des ordinateurs portables afin de pouvoir mettre en mémoire et traiter des données sur le terrain.



## Brassage de la bière

Les ordinateurs sont utilisés dans des brasseries automatisées. Ils contrôlent le mélange de la bière, la fermentation et ajustent les bonnes températures. Les capteurs indiquent à la machine qu'une opération arrive à sa fin et qu'une autre peut commencer.



Dans un espace aussi réduit que la cabine avant d'une voiture, tout doit être conçu pour que le conducteur ait sous la main les contrôles et leviers de commande indispensables. Avec un programme permettant de visualiser conducteur ou passagers, on peut déterminer les meilleurs placements.

## Apprendre avec un micro-ordinateur



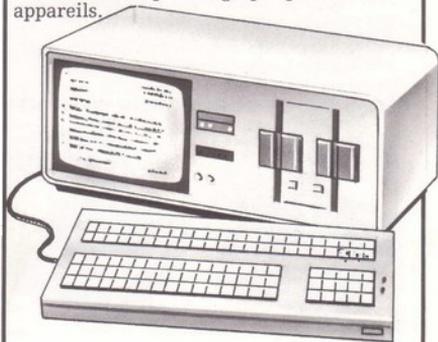
On peut tout enseigner avec un micro-ordinateur, de l'anglais à la cuisine... Si cela vous chante, vous pouvez même « disséquer » un lapin à l'écran, en utilisant un crayon optique à la place d'un scalpel.

## Les ordinateurs au travail



ÉLEVATION DE FACADE

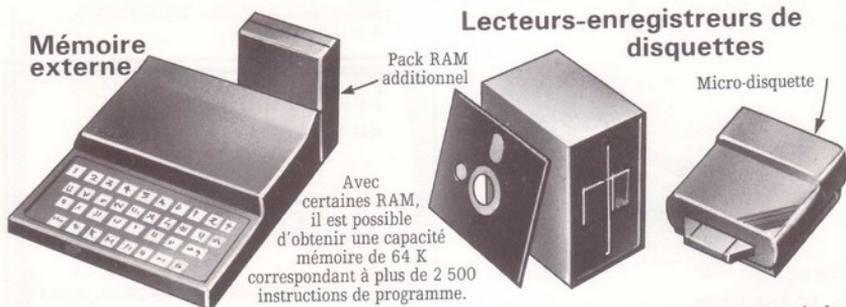
Des artisans, des travailleurs indépendants utilisent les ordinateurs pour traiter leur comptabilité et leur courrier. Plus professionnellement, un architecte ou un dessinateur peuvent recourir aux capacités graphiques de ces appareils.



Des microprocesseurs de « traitement de texte » sont utilisés dans des bureaux pour la réalisation de rapports, la frappe du courrier, etc. Lettres types et documents sont tapés et corrigés sur la machine de traitement de texte, puis stockés sur disque et imprimés.

# Des périphériques pour votre micro-ordinateur

Vous êtes maintenant familiarisé avec votre machine et ses étonnantes possibilités. Mais savez-vous qu'il existe de nombreux périphériques et équipements complémentaires permettant des applications passionnantes : lecteurs-enregistreurs de disquettes, imprimantes, tables graphiques. Ces appareils se connectent à votre ordinateur par l'intermédiaire d'une interface qui assure la conversion des signaux. Et chaque type de périphérique implique une interface différente. La plupart des micro-ordinateurs sont équipés d'origine avec des connecteurs pour magnétophones et télévision; certains comportent aussi des entrées/sorties pour lecteurs/enregistreurs de disquettes, imprimantes, crayons optiques. La plus grande partie des interfaces peuvent être acquises en options si votre console n'est pas équipée.



Avant d'acquérir vos périphériques, vous avez intérêt à accroître la capacité mémoire de votre ordinateur. Vous pourrez ainsi créer des programmes graphiques ou de jeux beaucoup plus élaborés. Attention, tous les appareils n'offrent pas la possibilité d'augmenter leur capacité mémoire. Pour ceux qui le permettent, vous pourrez acheter des RAM s'enchâssant sur un connecteur particulier.

Pour stocker des programmes très longs ou constituer de véritables petites banques de données, il est préférable d'utiliser des lecteurs-enregistreurs de disquettes. Par rapport aux magnétophones à cassette, ils recherchent ou sauvegardent les informations beaucoup plus rapidement. Mais leur prix est plus élevé. Récemment, la marque Sinclair parvint à mettre sur le marché des micro-disquettes stockant jusqu'à 100 K chacune (suffisant pour contenir tout le texte de cet ouvrage).

## Leviers de commande et boutons



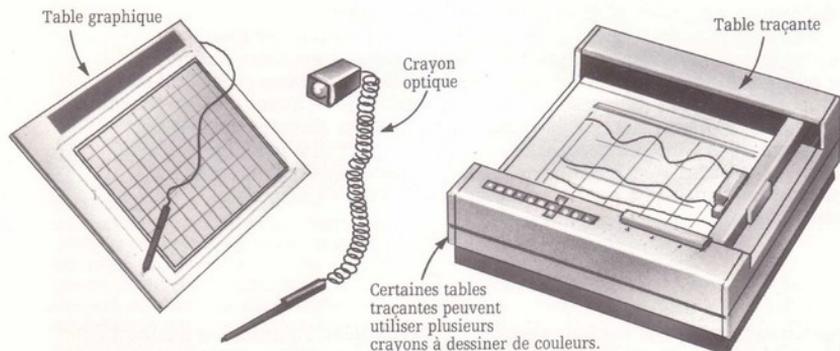
Ils sont surtout utiles pour les programmes de jeux où ils servent à commander les déplacements d'objets (avions, personnages, vaisseaux spatiaux). Plus pratiques et précis que les touches de contrôle du curseur de votre ordinateur, ils rendent vos jeux beaucoup plus vivants. Avec un levier de

commande (*joystick*), vous contrôlez les mouvements d'un objet sur l'écran dans toutes les directions. Le bouton (*paddle*), lui, vous limite à des déplacements horizontaux ou verticaux. La plupart des joysticks disposent d'un bouton séparé servant à déclencher le tir de missiles ou de balles...

## Dessins

Si vous vous intéressez au dessin, sachez que grâce aux tables graphiques vous arriverez à produire sur votre écran des images très complètes et à en tirer des copies sur une « imprimante » spéciale, appelée table traçante. Tous les mouvements de votre crayon sur la table

graphique seront en effet enregistrés et interprétés par un programme chargé spécialement dans votre ordinateur. Si vous ne voulez pas vous lancer dans des dépenses trop importantes, un simple crayon optique pourra vous donner satisfaction dans un premier temps.



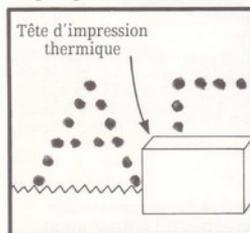
Certains ordinateurs peuvent recevoir une carte ou un pack à haute résolution graphique. Ce qui donne une qualité d'image étonnante grâce à une palette de couleurs plus large et à une augmentation du nombre de pixels qui peuvent être contrôlés individuellement. Par ailleurs,

en mode haute-résolution, votre ordinateur pourra afficher un plus grand nombre de lignes de caractères sur l'écran. Comme la création de graphismes très fins consomme beaucoup de place en mémoire, vous devrez sans doute utiliser des RAM complémentaires.

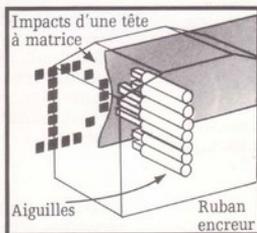
## A propos des imprimantes

Il existe trois grands types d'imprimantes : thermique, matricielle (ou par point), à marguerite. Les imprimantes thermiques, les moins chères, sont surtout destinées à la formation de listes. Les matricielles sont un peu plus chères, mais fournissent une

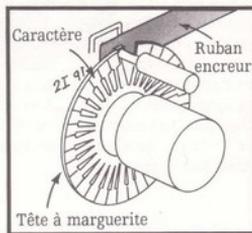
qualité d'impression supérieure. Quant aux imprimantes à marguerite, elles sont souvent hors de portée du grand public, mais leur frappe est remarquable. Pour gagner en vitesse d'impression, certaines imprimantes bidirectionnelles écrivent dans les deux sens (aller/retour).



La tête envoie de petites étincelles sur un papier sensible qui noircit aux endroits « brûlés » par le courant électrique. C'est un ensemble de points noirs qui formeront une lettre ou une figure.



La tête d'impression est constituée d'un ensemble d'aiguilles venant frapper un ruban encreur dans un ordre et selon une cadence variant en fonction de la lettre ou du dessin à imprimer.



La tête d'une « marguerite » ressemble un peu à une roue de vélo. L'extrémité de chaque « rayon » porte un caractère.

# Le guide des micro-ordinateurs

Si vous en êtes encore à vos premiers pas en informatique, n'hésitez pas à consulter le petit lexique ci-dessous. Pour vous y retrouver parmi tous les micro-ordinateurs en vente en France, le meilleur moyen consiste à vous renseigner auprès de vos amis déjà équipés ou de participer aux activités d'un club de micro-ordinateurs. Parallèlement, lisez les bancs d'essais et les tableaux comparatifs publiés dans les revues spécialisées. Enfin, avant d'opter pour tel ou tel modèle, étudiez de très près ce que vous attendez de votre ordinateur, comment vous comptez l'utiliser. Si vous optez pour une machine « bas de gamme », assurez-vous qu'elle peut recevoir des extensions mémoire et qu'éventuellement elle pourra accepter des lecteurs de disquettes et une imprimante.

**Processeur.** Il s'agit du microprocesseur, de l'unité centrale du micro-ordinateur. Les caractéristiques d'un micro-ordinateur comportent habituellement les références du processeur. Les modèles les plus utilisés sont le 6502 et le Z80 (v. page 28). Si vous n'êtes pas encore très expérimenté, ne vous préoccupez pas de ce problème.

**Clavier.** De nombreux ordinateurs ont des claviers semblables à ceux des machines à écrire. Quelques-uns sont équipés de touches micro-sensibles, peu pratiques lorsque vous avez à taper une liste assez longue. En règle générale, la disposition des lettres sur le clavier est la même que pour les machines à écrire (soit de type français : clavier AZERTY; soit de type américain : QWERTY). Il se peut que les touches comportent, en plus des caractères, des symboles spéciaux et des instructions de programmation (Sinclair par exemple) qui s'afficheront sur l'écran sans que vous ayez à les taper lettre par lettre.

**Écran de visualisation.** Les capacités d'affichage d'un ordinateur se comptent en nombre de caractères par ligne et en nombre de lignes par écran. Quelques micro-ordinateurs font défiler le texte automatiquement pour ménager de la place en bas de l'écran lorsque celui-ci est plein.

**Graphiques.** La finesse, la qualité d'une image se mesurent en nombre de points distincts pouvant s'afficher sur l'écran. Le nombre de points par ligne et le nombre de lignes par écran correspondent à la « résolution » permise par l'ordinateur en mode graphique.

**Interfaces.** La plupart des appareils sont équipés d'origine d'interfaces TV et/ou moniteur. Mais ils peuvent aussi disposer d'interfaces pour différents périphériques : imprimantes, unités de disques, joysticks, MODEM Télétel ou coupleur acoustique pour la communication. En achetant votre ordinateur, vérifiez bien qu'il pourra recevoir des interfaces d'extension mémoire, de communication ou de contrôle.

**Logiciels.** Ce sont des programmes sur cassettes, sur disquettes ou simplement imprimés sous forme de listes. Il est bien rare que des logiciels prévus pour tourner sur un type de machine puissent être compatibles avec d'autres machines. Cependant les logiciels de l'ATARI 400 peuvent être utilisés sur l'ATARI 800, ceux du ZX81 sur le Spectrum.

## ZX81 (Sinclair)

Processeur Z80  
RAM 1K, extension 16 ou 32K  
32 caractères sur 22 lignes  
Résolution graphique 63 x 43 points  
Clavier QWERTY



Le ZX81 est un petit micro-ordinateur « bas de gamme ». Ses performances sont néanmoins étonnantes, et il constitue un extraordinaire outil d'initiation à la programmation. En moins de 3 heures, il permet à un néophyte de commencer à réaliser ses propres programmes. Il est équipé d'un clavier à touches micro-sensibles portant, outre les caractères habituels (majuscules seulement), des instructions BASIC complètes (CLS, PRINT, GOTO...) et certaines fonctions (RND, SIN...). Le ZX81 dispose d'une sortie TV en noir et

blanc, d'interfaces magnétophone à cassette et imprimante (spéciale à Sinclair). Cet appareil est certainement celui qui s'est le plus vendu dans le monde et qui dispose de la plus large bibliothèque de programmes dans le domaine des jeux (sur cassettes ou sous forme de listes dans des revues et des livres).

## SPECTRUM (Sinclair)

Processeur Z80A  
RAM 16K, extension à 48K  
22 lignes de 32 ou 40 caractères  
Résolution graphique 256 x 192 points  
Clavier QWERTY



Comme le ZX81, le Spectrum dispose d'un clavier alphanumérique portant aussi des instructions et des fonctions BASIC. Il permet de créer des images en couleur (8 programmables) avec un téléviseur SECAM muni d'une prise Péritel. Outre ses interfaces magnétophone, imprimante et TV, il dispose d'entrées/sorties pour des

lecteurs/enregistreurs de disquettes ou pour communiquer avec d'autres appareils (norme RS232). Il sait aussi créer des sons sur 10 octaves. De nombreux programmes conçus pour le ZX81 fonctionnent aussi sur le Spectrum. En outre, il existe un catalogue important de logiciels spécifiques utilisant toutes les ressources de la couleur.

## PC 1500 (Sharp)

Ce micro-ordinateur de poche peut être alimenté par batterie ou secteur. Réellement portable, il comporte un écran d'affichage, en noir et blanc, à cristaux liquides. Programmes et données peuvent être sauvegardés sur cassettes en passant par une interface spéciale, à se procurer en option. Pour cette « super-calculatrice », il existe aussi une imprimante 4 couleurs (Sharp).



Processeur CMOS  
RAM 3,5K, extension à 7K  
1 ligne de 26 caractères  
Résolution de 7 x 156 points  
Clavier QWERTY  
Dimensions 20,5 cm x 9 cm

## VIC 20 (Commodore)



Processeur 6502  
RAM 5K, extension à 29K  
Résolution de 176 x 158 points  
23 lignes de 22 caractères  
Clavier QWERTY

Petit ordinateur de jeu et d'apprentissage du BASIC, le VIC 20 permet la création de graphismes en couleur et de sons. Il utilise un magnétophone spécial (VIC) pour stocker et charger des données ou des programmes. En version de base, la machine est équipée d'une interface SECAM (TV) et produit des graphiques grâce à des commandes aussi simples que DRAW. Pour les effets sonores le VIC 20

utilise le haut-parleur de la télévision, et peut générer plusieurs sons en même temps. Le VIC 20 comporte en outre les interfaces suivantes : lecteur/enregistreur de disquettes, imprimante, joysticks, crayon optique. Une entrée/sortie RS232 peut aussi être adjointe, ainsi que des cassettes RAM se fixant dans un logement spécial, à l'extérieur de l'ordinateur.

## Newbrain

constructeur : Grundy Business  
importateur : Sanocor International  
processeur : ZX 80A

ROM : 29 KO  
RAM : 32 KO  
écran : 25 lignes par 80 caractères  
résolution graphique : 640 x 250 points

Il offre 4 jeux différents de 255 caractères proposant aussi bien les minuscules accentuées que des symboles permettant des jeux au graphisme

élaboré. Par ses capacités de mémoire, il autorise la conception de programmes très longs ainsi que la gestion d'écrans multiples.

Le Newbrain est livré avec les interfaces permettant l'entrée et la sortie MODEM, l'entrée-sortie imprimante, les entrées-sorties pour deux magnétophones ainsi que le raccordement à un poste de télévision ou à un moniteur vidéo. De nombreux programmes pour cette machine commencent à apparaître sur le marché.

## TI-99/4 (Texas Instruments)

Avec le TI-99/4, vous aurez besoin d'un téléviseur avec prise péritélévision pour visualiser les données, et vous pourrez dessiner en 16 couleurs ! Vous serez étonné également par la qualité des effets sonores passant par le haut-parleur interne de l'ordinateur. Vous pourrez jouer trois notes à la fois, et ce sur cinq octaves. Un synthétiseur de parole, capable de restituer plus de 200 mots, est également disponible.

Pour enregistrer et charger les programmes, utilisez un magnétophone à cassettes normal. Vous trouverez de nombreux logiciels en cassettes, cartouches ou disquettes. D'autres extensions complètent le système :



Microprocesseur 9900  
RAM 16K, extension à 48K  
24 lignes de 29 caractères  
résolution graphique de 256 x 192  
clavier QWERTY

imprimante, lecteur de disques,  
périphériques de jeu et interface RS 232.

## DRAGON (Dragon Data)

Voici un petit micro-ordinateur destiné à l'utilisation familiale. Il se caractérise par un véritable clavier, l'affichage sur écran télévision et l'enregistrement sur magnétophone normal. Neuf couleurs (prise péritélévision) animent avec bonheur le graphisme. Et l'alliance d'une voix et du haut-parleur du téléviseur permet d'obtenir une large gamme de sons. Vous pourrez y connecter une imprimante, un lecteur de disquettes, des périphériques de jeu et des interfaces RS 232 et videotex.



Microprocesseur 6809  
RAM 32K extension à 64K  
16 lignes de 32 caractères  
Résolution graphique de 256 x 192

## ATARI 400 (Atari)



Un clavier QWERTY à touches micro-sensibles, un affichage sur écran télévision pour le micro Atari. Attention, il vous faudra un magnétophone spécial pour charger et enregistrer les programmes sur cassettes. Mais la plupart des logiciels utilisables sur ce

Microprocesseur 6502  
RAM 16K  
24 lignes de 40 caractères  
Résolution graphique de 320 x 192

matériel se présentent sous forme de cartouches qui se fichent directement dans la console. Vous trouverez un grand éventail de programmes de jeux et vous pourrez vous procurer en option des accessoires périphériques. Si vous le souhaitez, adaptez une imprimante et un lecteur de disques. L'Atari 400 vous offre une palette de 16 couleurs (prise péritélévision) et quatre voix.

## TRS-80 version couleur (Tandy, ou Radio-Shack aux États-Unis)

Cassette de programmes

Le TRS-80 couleur fonctionne avec un téléviseur muni d'une prise péritélévision. Il affiche en huit couleurs et peut également générer des sons. Comme extension, vous trouverez une imprimante, des périphériques de jeu, un lecteur de disquettes et une interface RS 232.

Microprocesseur 6809E  
RAM 16K extension à 32K  
16 lignes de 32 caractères  
Résolution graphique de 256 x 192



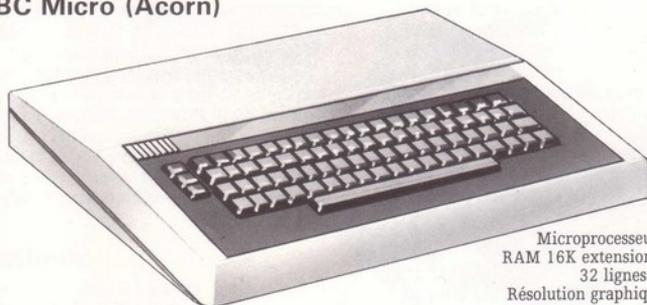
## TO 7 (Thomson)

Premier ordinateur français familial, il présente un excellent rapport qualité x performances/prix, particulièrement grâce à la richesse de son BASIC (en ROM externe) et à ses vastes possibilités graphiques. Parmi les innovations, il faut signaler un crayon optique qui permet de « converser » directement avec l'écran ou de réaliser des dessins sans mettre en œuvre des moyens compliqués. Le clavier (disposition « française »), à touches micro-sensibles, comporte les minuscules accentuées. D'origine, le TO 7 est équipé pour piloter un magnétophone extérieur permettant la sauvegarde des

programmes et des fichiers (cassetophone spécial vendu par Thomson), pour produire de la musique ainsi que des images en couleur (sortie TV par la prise Péritel). Des modules d'extension permettent (en option) l'utilisation de lecteurs/enregistreurs de disquettes, d'imprimantes ou de MODEM.

Microprocesseur 6809  
RAM 8K, extension par bloc de 16K externes  
Caractères aux normes vidéotex (Télétext)  
Résolution graphique de 64 000 points  
Clavier AZERTY

## BBC Micro (Acorn)



Microprocesseur 6502  
RAM 16K extension à 32K  
32 lignes de 40 caractères  
Résolution graphique de 320 x 256

Le BBC est un petit micro-ordinateur performant dont les capacités peuvent être facilement accrues par de nombreuses extensions. L'affichage s'effectue en huit couleurs sur écran télévision et trois voix sont utilisables. La mémoire ROM de 32 K contient le BASIC et l'assembleur qui peuvent être mixés

dans un même programme et un Operating System complet qui, cependant ne peut-être utilisé qu'en adaptant un ZX 80 avec 64 K/RAM. Lecteur de disquettes, imprimante, périphériques de jeu, extension mémoire et interface vidéotex permettent de compléter le système.

## ALICE (Matra et Hachette)

Microprocesseur 6803  
Mémoire RAM 4K extensible à 20 K  
16 Lignes de 32 caractères  
Semi graphique 64 x 32  
Haute résolution 128 x 192 (avec extension mémoire)  
Clavier AZERTY

ALICE est un micro-ordinateur fabriqué en France conçu pour la découverte de la micro-informatique :

Grâce à ALICE les non-initiés peuvent apprendre à parler BASIC (BASIC Microsoft résident). ALICE dispose d'effets sonores musicaux, de neuf couleurs simultanées à l'écran, d'un clavier à touches mécaniques (permettant la frappe rapide d'instructions BASIC) et de connexions vers magnétophone et imprimante standards. C'est le moins cher des micro-ordinateurs disposant de ces fonctions.

Le guide d'ALICE « Découvrez le BASIC », fourni avec la machine, permet d'apprendre à installer



le micro-ordinateur, puis à travailler avec lui en BASIC selon une approche pédagogique simple, claire et sécurisante; il comprend des exercices suivis et corrigés et de nombreux exemples de programmes pour compléter les apprentissages gradués de chaque chapitre.

## APPLE II Plus (Apple)



Microprocesseur 6502  
RAM 48K extension à 64K  
24 lignes de 48 colonnes  
Résolution graphique de 280 par 192  
Clavier QWERTY ou AZERTY

Tout comme le PET, cet appareil s'adresse aux « mordus », aux professions libérales et au public scolaire. Le modèle de base, avec clavier QWERTY ou AZERTY, peut être complété par des modules qui permettent de développer de multiples fonctions; extensions mémoire et cartes grâce auxquelles vous accéderez

à divers langages de programmation. Vous pouvez adapter à l'APPLE II une imprimante, un lecteur de disquettes, une tablette graphique et l'interface vidéotex, etc. Des quantités de programmes les plus divers sont disponibles sur tout support.

Voici quelques autres ordinateurs classés par ordre croissant de prix.

## JUPITER ACE (Jupiter Cantab)

Processeur : Z80  
RAM : 3 KO  
écran : 24 lignes de 32 caractères

Petit ordinateur bon marché qui utilise le langage FORTH au lieu du BASIC. C'est le premier ordinateur dans sa catégorie à utiliser ce langage, donc bibliothèque de programmes très réduite et livres d'explication très complexes.

Le clavier ressemble à celui d'une calculatrice; graphisme et son atteignent une bonne qualité, mais pas de capacités couleur. Une interface imprimante et un lecteur de micro-disquette sont disponibles.

## ORIC 1 (Oric Products International)

Processeur : 6502A  
RAM : 3 KO  
écran : 28 lignes de 40 caractères  
résolution graphique : 240 x 200 points

Micro-ordinateur bon marché offrant seize couleurs et un générateur sonore quatre voies. Clavier de type calculatrice. Interface imprimante Centronics. Il existe une version disposant de 48 K RAM.

## GENIE COLOR (EACA)

Processeur : Z80  
RAM : 16 KO, extension à 32 KO  
écran : 24 lignes de 40 caractères  
résolution graphique : 196 x 96 points

Le GENIE COLOR ressemble beaucoup au TANDY. Il accepte sans grande modification les programmes destinés au TRS 80, mais n'est pas compatible avec les cassettes.

Son clavier est de type machine à écrire et il possède lecteur de disquette, crayon optique et autres périphériques.

## LYNX (Computers)

Processeur : Z80A  
RAM : 48 KO, extension à 192 KO  
écran : 24 lignes de 40 caractères  
résolution graphique : 248 x 256 points

Doté d'un clavier type machine à écrire, ce micro-ordinateur affiche en huit couleurs et possède un convertisseur digital analogique pour produire des sons. C'est un système modulaire, que l'on peut faire grandir à l'aide de nombreux périphériques.

## Commodore 64 (Commodore)

Processeur : 6502  
RAM : 64 KO  
écran : 25 lignes de 40 caractères  
résolution graphique : 320 x 200 points

Le Commodore 64 a le même clavier que le VIC 20 mais une mémoire beaucoup plus importante ainsi qu'un écran de taille standard. Il possède de bonnes capacités graphiques en 16 couleurs et un générateur de sons trois voies.

On peut utiliser sur le 64 la plupart des programmes écrits pour le PET ou le VIC, auxquels il faut ajouter un catalogue spécifique déjà important. Lecteur de disquette, interface modem et cartouches enfichables donnant la compatibilité avec le PET, en font un outil performant.

## HX 20 (Epson)

Processeur : 630  
RAM : 16 KO, extension à 32 KO  
écran : 4 lignes de 20 caractères  
résolution graphique : 120 x 32 points

Ce micro se présente sous la forme d'une petite console intégrant clavier, mini-écran, mini-imprimante et magnétophone. On peut bien sûr, le connecter à un écran T.V., le brancher à une prise de courant ou le faire travailler sur batteries.

Des périphériques sont disponibles : ROM, lecteur de codes/barres, coupleur acoustique pour la transmission par téléphone, et interface RS 232.

# Un peu de vocabulaire

**Animation** : image animée grâce à un programme.

**ASCII (American Standard Code for Information Interchange)** : code standard pour représenter lettres et chiffres par des nombres binaires.

**Base de données** : série organisée d'informations sauvegardées dans la mémoire de l'ordinateur, sur un disque ou une cassette.

**BASIC (Beginners' All Purpose Symbolic Instruction Code)** : langage de programmation le plus répandu, s'adaptant à la plupart des utilisations de la micro-informatique.

**Binaire** : mode de calcul ne recourant qu'à deux chiffres, 0 et 1. Le code machine est un code binaire.

**Bit** : les chiffres 0 et 1 permettent de créer des « nombres » binaires. Dans un ordinateur, un bit correspond à une impulsion (1) ou à son absence (0).

**Bugs** : erreurs dans les programmes qui les empêchent de fonctionner.

**Bus** : voie d'accès électrique à l'ordinateur permettant la transmission d'informations.

**Byte** : la plupart des micro-ordinateurs travaillent avec des groupes de huit bits à la fois qu'on appelle byte.

**Capteur** : élément extérieur à l'ordinateur, qui mesure soit la luminosité, soit la pression ou encore la température et les lui communique.

**Caractère** : un nombre, une lettre ou un symbole. Unité de transmission.

**Chargement (loading)** : transfert d'un programme d'une cassette ou d'une disquette sur un ordinateur.

**Circuits intégrés** : minuscules circuits électriques comportant des centaines de composants électroniques sur une puce de silicium.

**Code machine** : suite d'impulsions électroniques utilisée pour travailler.

**Compatibilité** : deux ordinateurs présentent une compatibilité lorsqu'ils peuvent recevoir les mêmes programmes.

**Copie papier (hard-copy)** : programme ou données sortis par une imprimante.

**Correction (debugging)** : opération consistant à détecter les erreurs dans un programme et à les corriger.

**Données (data)** : toutes informations communiquées à l'ordinateur qui les traitera selon les instructions du programme. Les réponses de l'ordinateur sont aussi des données.

**Entrée (input)** : toute information que vous entrez dans l'ordinateur.

**FORTAN (FORMula TRANslator)** : langage de programmation évolué utilisé surtout par les scientifiques ou les techniciens.

**Graphique** : dessin réalisé par un ordinateur.

**Hexadécimal (hexadecimal numeration)** : système de numération à base 16 chiffres (0 à 9 et lettres A à F), très pratique pour la programmation en langages peu évolués : il suffit d'un nombre de deux chiffres hexadécimaux pour représenter huit bits.

**Interface (interface)** : circuit spécial qui assure la conversion des signaux émis par un ordinateur en signaux compréhensibles par d'autres matériels électroniques et vice versa. Chaque type d'équipement nécessite une interface particulière.

**Interpréteur (interpreter)** : programme résidant en mémoire morte, qui a pour tâche de traduire les instructions délivrées en langage de programmation (le BASIC le plus souvent pour un micro-ordinateur) en langage machine.

**Kilobyte** : 1 kilobyte comprend 1 024 bytes.

**Liste** (ou Listing) : programme écrit, dactylographié ou imprimé sur du papier.

**Matériel (hardware)** : ensemble des éléments physiques utilisés pour le traitement de l'information.

**Message d'erreur** : message que l'ordinateur affiche à l'écran pour vous prévenir qu'il y a un bug dans le programme, vous indiquer sa nature et, éventuellement, vous aider à le localiser.

**Microprocesseur (microprocessor)** : puce qui contient tous les circuits dont un ordinateur a besoin pour contrôler l'ensemble des opérations et effectuer tous les calculs de base. L'unité principale d'un micro-ordinateur n'est autre qu'un microprocesseur.

**Mnémoniques (codes)** : instructions alphanumériques permettant l'utilisation d'un symbolisme mnémotechnique pour représenter un code binaire ou numérique.

**MODEM** : abréviation de MOdulateur/DEModulateur. Convertit les signaux numériques en signaux audio.

**Moniteur (monitor)** : partie de la mémoire morte de l'ordinateur qui indique à l'unité centrale comment opérer.

**PASCAL** : langage de programmation de haut niveau, mais non spécialisé.

**RAM (Random Access Memory)** : mémoire vive dans laquelle sont stockées provisoirement toutes les informations que vous communiquez à votre ordinateur (jusqu'à ce que vous les effaciez ou que vous les modifiez).

**Résolution** : nombre de groupes de pixels que le micro-ordinateur peut afficher sur l'écran. Les images graphiques à haute résolution, très fines et détaillées, sont produites par des ordinateurs capables d'allumer et d'éteindre un grand nombre de pixels de manière indépendante.

**ROM (Read Only Memory)** : mémoire morte dans laquelle sont stockées les instructions indiquant à l'ordinateur comment travailler (moniteur). Les ROM sont des mémoires permanentes qui ne peuvent être effacées.

**Sauvegarde (save)** : « Sortir » un programme de la mémoire d'un ordinateur pour le stocker sur un support externe (disquette ou cassette audio).

**Sortie (output)** : information ou ensemble d'informations délivrées par l'ordinateur.

**Synthétiseur** : circuit électronique permettant de produire de la musique, des bruits ou des mots.

**Transistor** : composant électronique qui stoppe ou envoie des impulsions dans les circuits de l'ordinateur, en relation avec celles qu'il reçoit. Une simple puce contient des centaines de transistors.

**Unité centrale (Central Processing Unit, CPU)** : circuit qui contrôle tout l'ensemble de l'ordinateur et où sont réalisés tous les calculs.

**Unité de visualisation (VDU), Moniteur** : téléviseur ne comportant pas de circuit de réception des chaînes TV, et prévu pour être connecté directement à un ordinateur en vidéo.

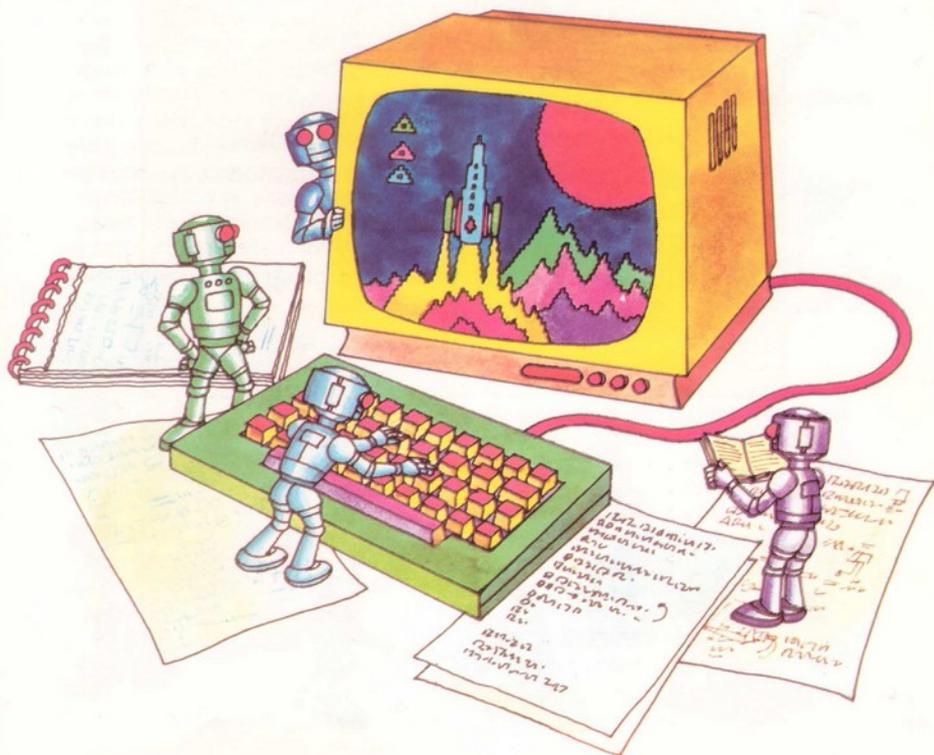
**Variable système** : espace-mémoire en RAM dans lequel sont stockées des informations provenant de l'ensemble de l'ordinateur (par exemple prochain caractère à afficher à l'écran ou adresse de différents espaces).

**Vitesse de transmission (baud rate)** : vitesse avec laquelle des ordinateurs peuvent s'échanger des informations ou en délivrer à destination de périphériques. S'exprime en baud. Un baud correspond à un bit par seconde.

# GUIDE PRATIQUE DU BASIC

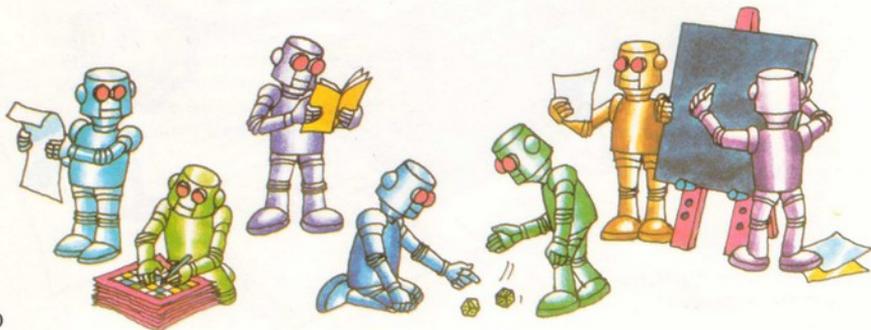
Brian Reffin Smith

Conception graphique : Kim Blundell. Illustrations de G. Round et M. Newton. Édition : Lisa Watts pour l'édition anglaise; Patrick Baradeau pour l'édition française. Révision : Jean-Noël Von der Weid. Assistance technique et Conseil : European Media Business, 9, place des Ternes, 75017 Paris.



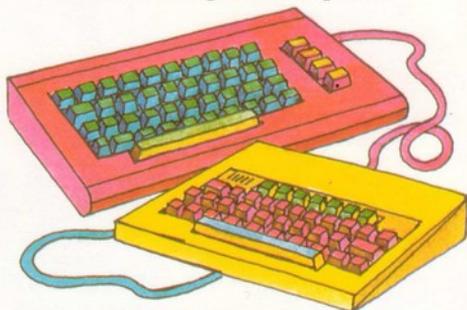
# Sommaire

- 52 Comment fonctionne un ordinateur
- 54 Qu'est-ce qu'un programme ?
  - 56 Écrire un programme
  - 58 Premiers pas en BASIC
  - 60 Entrer des données
  - 62 Comment utiliser INPUT
  - 64 Que faire avec PRINT
- 66 Comment les ordinateurs comparent
  - 68 Programmes en BASIC
  - 70 Dessinons
  - 72 Jouons
  - 74 Faisons des boucles
  - 76 Quelques astuces
  - 78 Sous-programmes
  - 80 Jouons avec les mots
  - 82 Graphiques et symboles
  - 84 Encore des graphiques
- 86 Programmons des poèmes drôles
- 88 Quelques trucs de programmation
  - 90 Réponses aux casse-tête
  - 92 Petit lexique du BASIC
- 94 Pour charger et conserver ses programmes
- 95 Tableaux de conversion

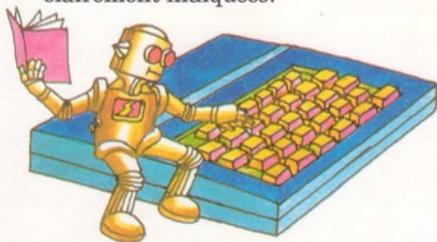


# Avant-propos

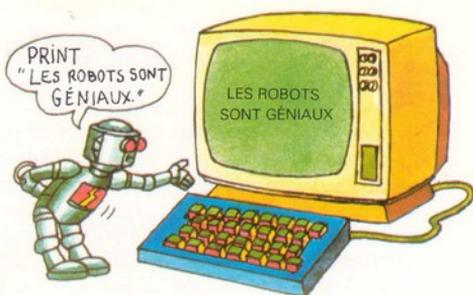
Ce *Guide pratique du BASIC* permettra aux débutants d'apprendre à programmer en BASIC. Le BASIC est le langage de la plupart des ordinateurs domestiques; c'est une façon de donner à la machine des instructions qu'elle comprend.



Point n'est besoin de posséder un ordinateur pour utiliser ce livre, même s'il est plus facile de comprendre les programmes en les essayant. Des différences de fabrication font que de légères variations interviennent dans le BASIC utilisé par tel ou tel appareil. Les instructions proposées ici sont acceptées par la plupart des micro-ordinateurs, et les rares exceptions sont clairement indiquées.



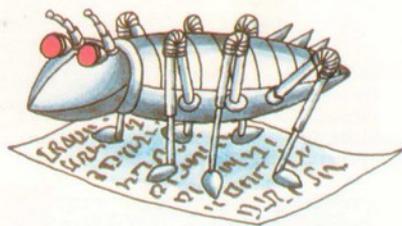
Pour commencer, vous trouverez une introduction générale à la programmation. Puis, au fil des pages, les différents mots-clés du BASIC et quelques programmes types permettant de les mettre en œuvre.



Pour vous entraîner à écrire des programmes, ce livre vous propose des casse-tête à résoudre, des suggestions pour créer ou modifier des programmes (les solutions sont données pages 92 et 93).

À la fin du *Guide pratique du BASIC*, un lexique des mots propres au BASIC et aux ordinateurs, des conseils pour vous aider à programmer, la liste des « bugs » (erreurs qui empêchent le programme de fonctionner) les plus fréquents et des trucs pour les reconnaître finiront de vous armer pour la programmation.

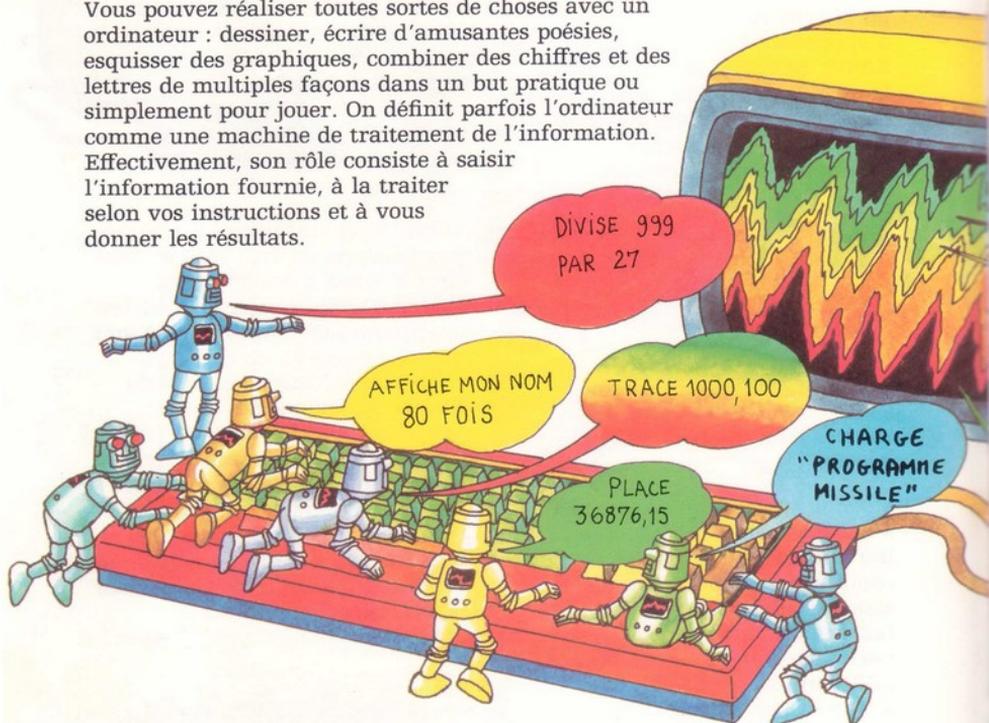
Si vous avez un micro, essayez les



programmes de ce livre, et relevez les instructions BASIC du manuel fourni par son fabricant; il se peut que certaines procédures indiquées ici ne soient pas nécessaires. La meilleure façon d'apprendre le BASIC? essayer de nombreux programmes proposés par les livres et les revues spécialisés, et parvenir à les transformer. Si vous suivez ces conseils, vous vous retrouverez rapidement en train d'écrire vos propres programmes !

# Comment fonctionne un ordinateur

Vous pouvez réaliser toutes sortes de choses avec un ordinateur : dessiner, écrire d'amusantes poésies, esquisser des graphiques, combiner des chiffres et des lettres de multiples façons dans un but pratique ou simplement pour jouer. On définit parfois l'ordinateur comme une machine de traitement de l'information. Effectivement, son rôle consiste à saisir l'information fournie, à la traiter selon vos instructions et à vous donner les résultats.



L'ordinateur est un outil. Pour qu'il travaille, il faut lui donner des instructions extrêmement précises. La liste des instructions s'appelle un *programme*, les informations que vous

lui fournissez, des *données*. Le programme doit être écrit dans un langage que l'ordinateur comprend, ainsi le BASIC, et en respecter toutes les règles.

## Les micro-ordinateurs

La plupart se présentent sous la forme d'une console qui ressemble au clavier d'une machine à écrire, à raccorder à un écran. Instructions et informations sont transmises à l'ordinateur en tapant sur le clavier de la console. Tout ce qui est frappé sur le clavier, ainsi que les résultats fournis par l'ordinateur, s'affichent à l'écran.

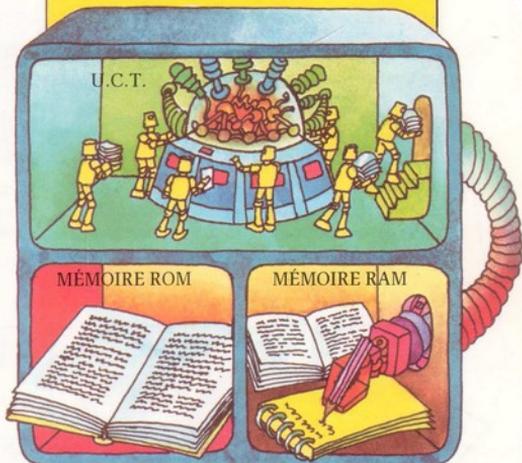
Certains micro-ordinateurs ont de petits écrans incorporés, tout comme les calculateurs de poche. D'autres peuvent être reliés à des écrans spéciaux : les *moniteurs* qui se présentent comme des téléviseurs



privés de leurs circuits de réception des émissions TV. Le clavier d'un micro-ordinateur ressemble à celui d'une machine à écrire, avec quelques touches supplémentaires. Sur certains appareils, chaque touche correspond à

## A l'intérieur d'un ordinateur

Un micro-ordinateur est constitué de deux « organes » essentiels : l'*unité centrale de traitement*, le cœur de l'ordinateur, où s'effectue tout le travail, et la *mémoire* où programmes et données sont stockés.



En réalité, l'ordinateur a deux mémoires. L'une, appelée ROM (Read Only Memory = mémoire en lecture seulement), contient un programme permanent. L'autre, appelée RAM (Random Access Memory = mémoire à accès direct), dont le contenu s'efface à chaque fois qu'on débranche l'ordinateur.



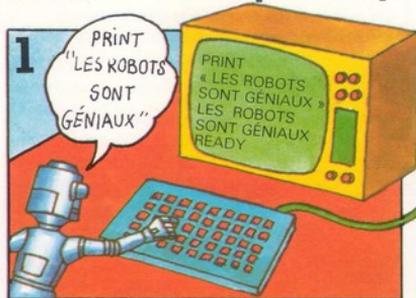
une instruction en BASIC, que vous n'aurez donc pas besoin d'entrer lettre à lettre.

L'affichage des informations s'effectue sur un écran TV ou sur un moniteur. Si l'on veut conserver une trace écrite des programmes et des données, il faut utiliser une imprimante

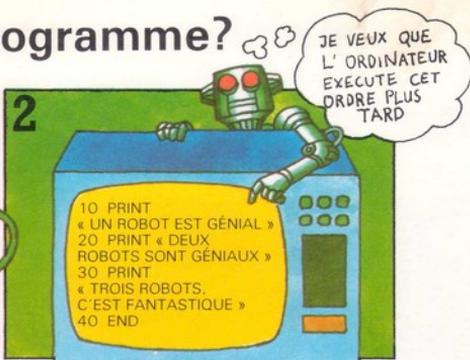


pour les transférer sur papier. On peut également les enregistrer sur une cassette de magnétophone. Il suffira de charger celle-ci sur l'ordinateur pour retrouver les informations.

# Qu'est-ce qu'un programme?



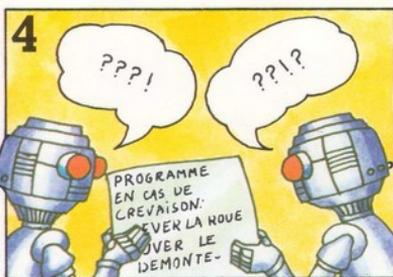
Pour faire agir l'ordinateur, il faut lui donner un ordre qu'il comprenne. Cette instruction peut être un ordre direct qu'il exécute immédiatement, ou un



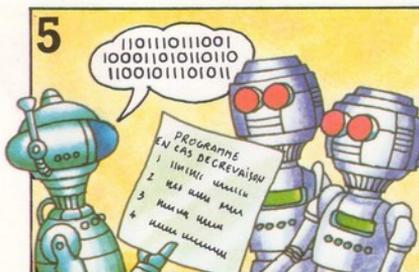
programme d'instructions qu'il garde en mémoire et n'exécutera que lorsque vous lui en donnerez le signal.



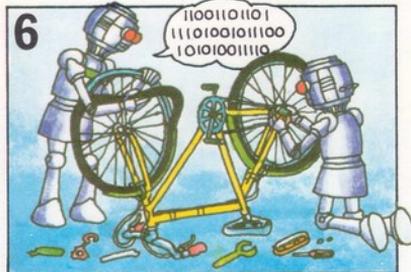
Les instructions d'un programme doivent être très soigneusement formulées. L'ordinateur essaiera d'exécuter vos instructions avec précision, même si elles sont fausses.



L'ordinateur ne comprend pas les ordres donnés dans notre langue. Il faut donc les traduire dans un des nombreux langages informatiques. Vous en trouverez quelques exemples page ci-contre.



À l'intérieur de l'ordinateur, toutes les informations circulent sous la forme d'impulsions électriques codées. Les instructions que vous donnez sont traduites dans le code de l'ordinateur par un programme spécial : l'interpréteur intégré à l'ordinateur.

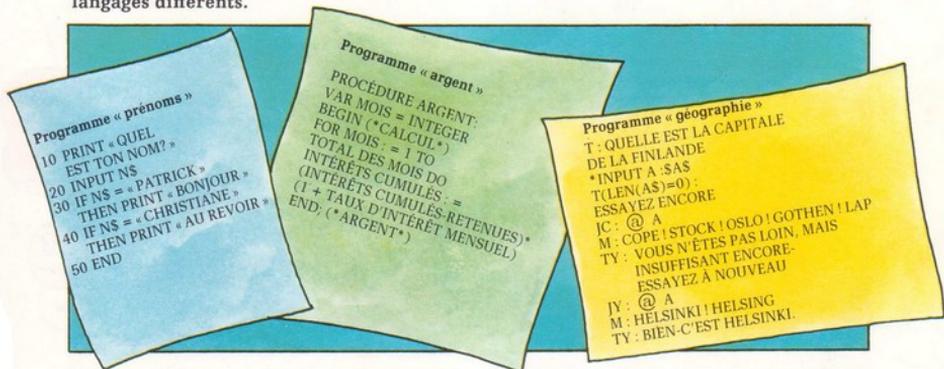


Dans ce code, chaque instruction correspond à une suite d'impulsions particulières : la valeur 1 représente une impulsion alors que la valeur 0 en marque l'absence.

## Les langages de programmation

Vous pourriez écrire vos programmes directement dans le code de l'ordinateur. Mais ce serait très difficile. En revanche, il existe des langages de programmation, dits *langages évolués*, que l'ordinateur peut traduire dans sa propre langue.

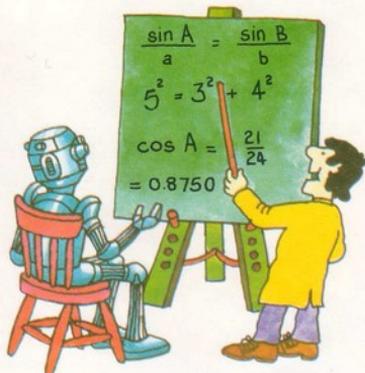
On trouve des centaines de langages évolués. La plupart ont une application dans un domaine particulier : sciences et techniques, administration ou commercial... Le BASIC est un des langages les plus courants. Le sigle BASIC signifie : Beginner's All-purpose Symbolic Instruction Code (langage général de programmation pour débutants). Mais il n'y a pas que les débutants qui l'utilisent. Vous trouverez ci-dessous des exemples de trois langages différents.



Voici un programme très court rédigé en BASIC. La ligne 10 dit à l'ordinateur d'afficher à l'écran « Quel est ton nom ? ». Puis l'ordinateur entre votre réponse dans sa mémoire et, si votre nom est Patrick ou Christiane, vous transmet un message.

Ce programme est écrit en Pascal, du nom du célèbre mathématicien français. Certains pensent qu'il est plus facile d'écrire de bons programmes en Pascal qu'en BASIC.

Voici le langage PILOT. On l'utilise pour écrire des programmes pédagogiques. Avec ce langage, l'ordinateur peut reconnaître une réponse, même si elle n'est pas parfaitement exacte.



11. Bb3, Ne5
12. 0-0-0, Nc4
13. Bxc4, Rxc4
14. h5, Nxf5



\*Moins 15 °C, je pense.

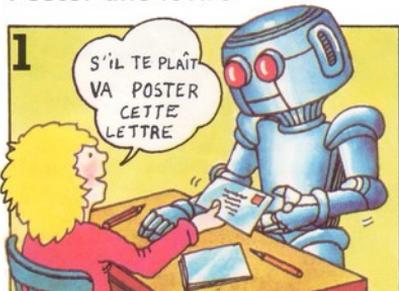
A première vue, les langages de programmation semblent très compliqués, tout comme des langues étrangères tel le chinois ou le finnois que vous pouvez lire sur l'image de droite. Mais il suffit d'apprendre à les connaître. Dans beaucoup d'autres domaines aussi, on se

sert de codes spéciaux : par exemple, en mathématiques on a recours à des signes particuliers pour désigner des idées et des formules et éviter une terminologie fastidieuse; le jeu d'échecs et la musique utilisent également leurs propres signes.

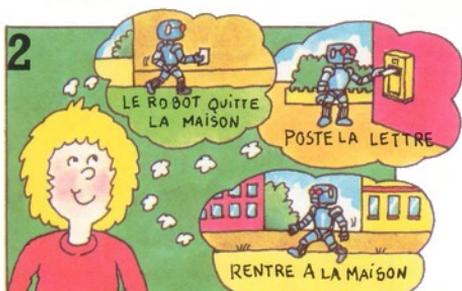
# Écrire un programme

Un programme, c'est comme une règle du jeu ou une recette de cuisine. S'il y a une erreur dans la règle ou dans la recette, on ne pourra ni bien jouer, ni faire un bon gâteau. De même, les résultats qu'on obtient de l'ordinateur dépendent de la qualité des instructions qu'on lui a données. Avant d'écrire un programme, il faut d'abord étudier soigneusement ce que l'on veut faire, déterminer les étapes principales qui conduiront au résultat souhaité.

## Poster une lettre



Imaginez que vous essayez d'écrire un programme qui demande à un robot de poster une lettre. Un ordre simple, comme celui exprimé dans la bulle, serait trop difficile à comprendre pour un robot.



Vous devez relever avec précision toutes les actions que le robot doit exécuter pour poster la lettre. L'ordinateur placé à l'intérieur du robot réclame des instructions pour savoir quoi faire à chaque étape.

## 3

Quitter la maison  
aller jusqu'à la porte  
ouvrir la porte  
sortir et fermer la porte  
trouver la boîte aux lettres

Poster la lettre  
introduire la lettre  
dans la boîte  
lâcher la lettre

Rentrer à la maison  
se retourner  
revenir sur ses pas  
ouvrir la porte  
entrer  
fermer la porte

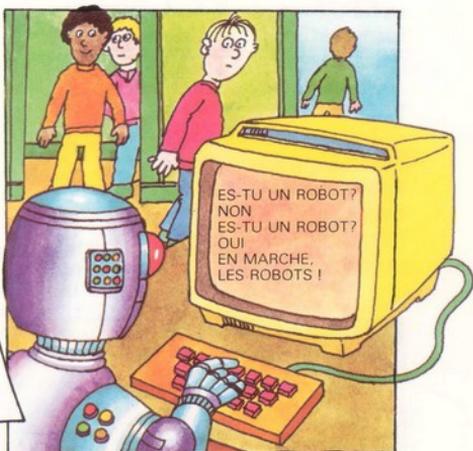
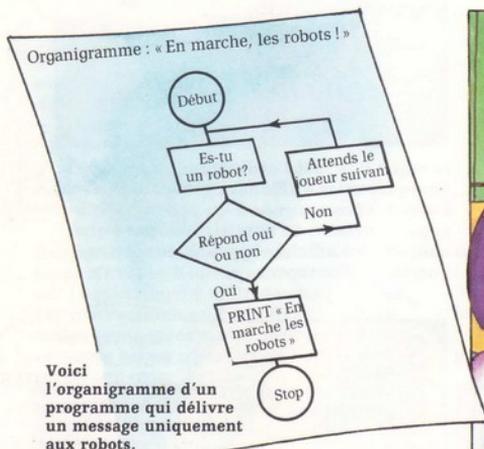
JE DOIS  
POSTER CETTE  
LETTRE

Pour programmer, il faut découper à chaque étape les instructions en ordres simples, immédiatement traduisibles dans la langue du robot. Le robot essaiera d'exécuter vos instructions, même si elles sont mauvaises

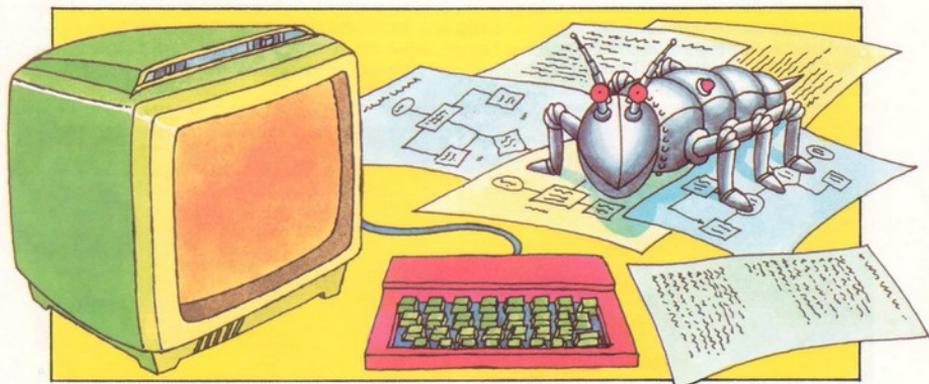
ou incomplètes. On appelle « bugs » les erreurs de programmation qui peuvent parfois conduire à des résultats inattendus de la part de l'ordinateur. Passer à travers un mur, par exemple !

## Les organigrammes

Quand on écrit un programme, il est parfois utile d'établir un schéma, ou organigramme, indiquant les principales étapes nécessaires à la résolution du problème envisagé. On y situe très exactement chaque étape du déroulement du programme.



Ces diverses étapes sont indiquées par des encadrés de formes différentes. Début et fin s'inscrivent dans des formes rondes; les ordres d'exécution dans des rectangles; les prises de décision (là où l'ordinateur agit différemment selon les réponses qu'il reçoit) dans des losanges. Les traits et les flèches indiquent les itinéraires que l'ordinateur peut suivre.



Après avoir mis en évidence tous les détails du programme, on peut le convertir en BASIC. Toutefois, il ne fonctionnera certainement pas du premier coup, car quelques bugs s'y seront sans doute glissés : erreurs de frappe faites au moment où l'on a entré le programme,

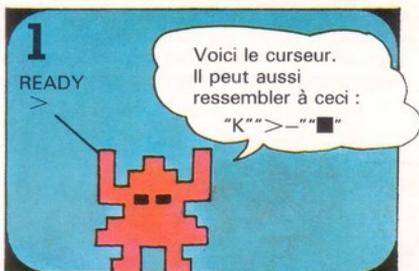
erreurs de syntaxe ou erreurs de logique. Avant de faire tourner le programme, il faut détecter tous les bugs et les corriger. Parfois, un bug peut opérer une légère transformation et aboutir à un résultat que l'on aurait préféré. De tels bugs, utiles, sont appelés « pugs ».

\*Vous trouverez quelques trucs pages 90 et 91 pour vous aider à trouver les bugs.

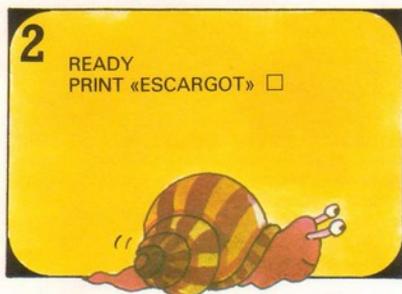
# Premiers pas en BASIC

La plupart des termes utilisés en BASIC viennent de l'anglais. Ainsi « PRINT » signifie « afficher sur l'écran », « RUN » donne l'ordre de mise en route du programme, « INPUT » permet de donner une information à l'ordinateur. Ces deux pages vous expliquent comment utiliser la fonction PRINT.

La plupart des ordinateurs domestiques possèdent un interpréteur BASIC qui leur est intégré. Aussi, une fois branchés, ils sont immédiatement prêts à être programmés dans ce langage\*.



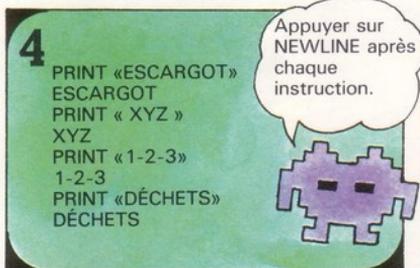
Lorsque l'on met l'ordinateur en marche, quelques mots s'affichent généralement à l'écran, accompagnés d'un petit symbole appelé curseur. Ce curseur indique à quel endroit va s'afficher la prochaine lettre que vous allez taper.



Pour demander à l'ordinateur d'afficher un mot à l'écran, on utilise la fonction PRINT, suivie du mot souhaité entre guillemets. Par exemple, PRINT « ESCARGOT » indique à l'ordinateur d'afficher le mot ESCARGOT à l'écran.



L'ordinateur n'exécutera pas votre instruction si l'on n'appuie pas sur la touche NEWLINE, RETURN ou ENTER – selon les appareils –, ce qui lui montre que l'ordre est terminé.



L'ordinateur affiche à l'écran tout ce que vous aurez placé entre guillemets, que ce soit des lettres, des chiffres, des mots ou des symboles. Remarquez qu'il n'affiche pas les guillemets eux-mêmes.



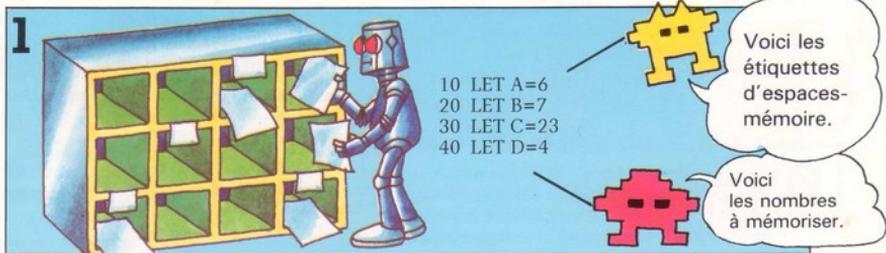
Si l'on a à afficher des nombres, il n'est pas nécessaire d'utiliser de guillemets. Pour vider l'écran, il suffit, sur la plupart des ordinateurs, de taper CLS (regardez dans votre mode d'emploi, si vous avez un ordinateur).



# Entrer des données

Pour que l'ordinateur accomplisse des tâches plus intéressantes que d'afficher simplement des signes à l'écran, il faut lui fournir des informations qu'il puisse utiliser en les mettant en mémoire jusqu'à ce que vous lui donniez l'ordre d'aller les chercher.

**1**



```
10 LET A=6
20 LET B=7
30 LET C=23
40 LET D=4
```

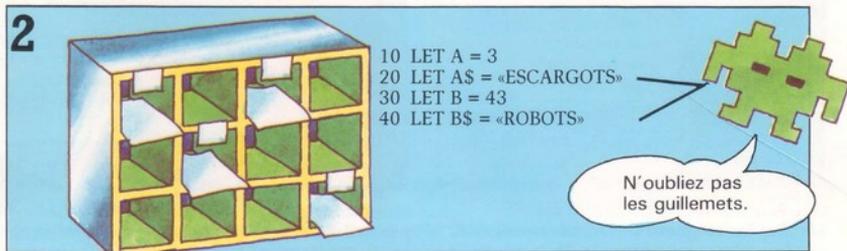
Voici les étiquettes d'espaces-mémoire.

Voici les nombres à mémoriser.

Lorsque l'on introduit une donnée dans l'ordinateur, il faut lui attribuer une étiquette pour pouvoir la retrouver. A cette fin, on peut utiliser des lettres de l'alphabet. Pour repérer une case-mémoire et y inscrire un nombre, on

se sert de l'instruction LET. Un espace-mémoire réservé s'appelle une *variable*, car il peut contenir des données diverses à des moments différents du programme.

**2**



```
10 LET A = 3
20 LET A$ = «ESCARGOTS»
30 LET B = 43
40 LET B$ = «ROBOTS»
```

N'oubliez pas les guillemets.

Il faut utiliser des étiquettes particulières pour mémoriser les lettres et les symboles. Ceux-ci forment une succession de caractères qu'on appelle des *chaînes alphanumériques*.

Pour les étiqueter, on a

recours à une lettre de l'alphabet suivie du signe \$. L'instruction LET permet de mettre une chaîne en mémoire, tout comme une variable, si ce n'est que lettres et symboles doivent être écrits entre guillemets.

**3**

```
10 LET B = 365
20 LET D$ = « JOURS DANS
UNE ANNÉE »
30 LET L$ = « SAUF LES ANNÉES
BISSEXTILES »
40 PRINT B
50 PRINT D$
60 PRINT L$
70 END
```

Beaucoup d'ordinateurs n'ont pas besoin de la ligne END.

Pour afficher l'information à l'écran, utilisez PRINT suivi du nom de la variable, par exemple PRINT A\$. Ce court programme permet d'afficher le contenu des variables B, D\$ et L\$.

**4**

```
RUN
365
JOURS DANS L'ANNÉE
SAUF LES ANNÉES BISSEXTILES
```

Vous pouvez relancer le programme autant de fois que vous le souhaitez. Jusqu'à ce que vous changiez les données en variable, l'ordinateur reproduira les mêmes informations.

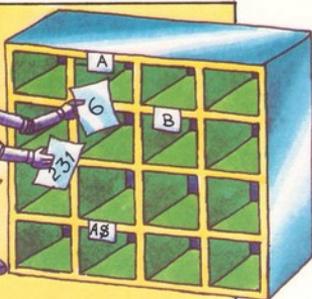
## Une autre méthode

```
10 READ A
20 READ B
30 READ AS
40 DATA 6,231,VENDREDI
```

Il faut utiliser les bonnes étiquettes pour les nombres et les lettres.  
Virgules



Sur certains ordinateurs, il faut mettre les données entre guillemets.



On peut aussi mémoriser l'information avec les instructions READ ou DATA. Les lignes READ indiquent à l'ordinateur d'étiqueter des espaces-mémoire, et la ligne DATA contient les informations.

Quand on lance le programme, l'ordinateur place chaque donnée dans l'espace-mémoire correspondant. Les articles des données doivent être séparés par des virgules afin que la machine les situe\*.

## Quelques programmes

```
1 10 READ Q
  20 READ X$
  30 DATA 24,
  40 PRINT Q
  50 PRINT X$
  60 END
  RUN
  24
  ÉCRANS TV
```

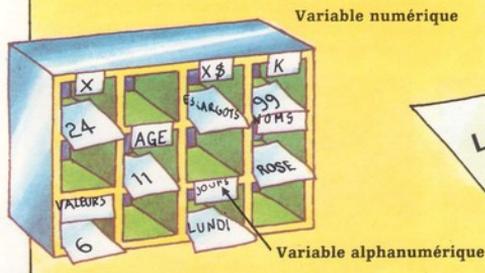
Virgule  
ÉCRANS TV  
Article de DATA, y compris l'intervalle.

Voici deux programmes utilisant l'un READ et DATA, l'autre LET, pour stocker l'information.

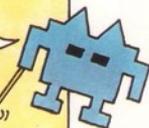
```
2 10 LET AS$ = «LES ROBOTS
   SONT GÉNIAUX»
  20 LET BS$ = «SI TU VEUX»
  30 LET CS$ = «GÉNIAUX EN BÊTISE»
  40 PRINT AS$
  50 PRINT BS$
  60 PRINT CS$
  70 END
  RUN
  LES ROBOTS SONT GÉNIAUX
  SI TU VEUX
  GÉNIAUX EN BÊTISE
```

Attention aux guillemets

## Des précisions sur les variables



Vous ne pouvez pas utiliser ces mots comme variables, car ils contiennent des instructions de BASIC.



LIST\$

IF

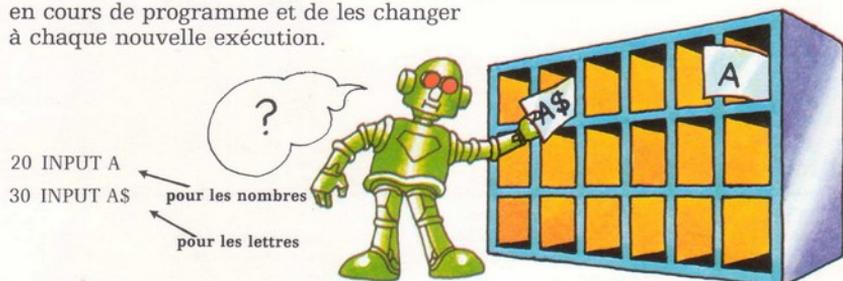
DIM

Les variables sont des espaces-mémoire étiquetés où l'information est stockée et dont le contenu peut changer en cours de programme. Une variable contenant un chiffre s'appelle une *variable numérique*, une variable comportant des lettres et des

symboles, une *variable alphanumérique*. Certains ordinateurs se servent de mots comme « étiquettes de variables », à l'exception des termes de BASIC, car il y aurait alors confusion pour l'ordinateur.

# Comment utiliser INPUT

On peut encore fournir des informations à l'ordinateur par l'instruction INPUT. Ce qui permet d'introduire des données en cours de programme et de les changer à chaque nouvelle exécution.



On utilise INPUT avec une étiquette comme A pour un nombre et A\$ pour une lettre. Lorsque l'ordinateur rencontre l'instruction INPUT dans le déroulement d'un programme, il place l'étiquette dans un espace-mémoire et demande quelle est

la donnée, en inscrivant un point d'interrogation, ou tout autre symbole, sur l'écran. On entre alors la donnée, que l'ordinateur mémorise, avant de continuer à exécuter le programme.

**1 Programmes INPUT**

```
10 INPUT G
20 INPUT B$
30 PRINT G
40 PRINT B$
50 END
```

Questions posées par l'ordinateur

**2** RUN  
756  
?CINQUANTE-SIX  
56  
CINQUANTE-SIX

Votre chiffre et votre mot

Appuyer sur NEWLINE après chaque INPUT.

Ne pas entrer une chaîne alphanumérique dans une variable numérique.

DONNEZ-MOI UN NOMBRE

Soixante-deux

INTERDIT

La Figure 2 montre ce qui se passe lorsque vous lancez ce programme. Quand l'ordinateur rencontre INPUT à la ligne 10, il inscrit un point d'interrogation et attend qu'on lui indique la valeur de G. Puis il affiche à

nouveau un point d'interrogation pour l'INPUT de la ligne 40. Cette fois, il faut entrer des mots ou des symboles, car l'étiquette B\$ indique à l'ordinateur qu'il doit attendre une chaîne de caractères.

**3**

```
10 PRINT « QUEL EST TON NOM? »
20 INPUT N$
30 PRINT « QUEL EST TON ÂGE? »
40 INPUT A
50 PRINT N$
60 PRINT « A »
70 PRINT A; « ANS »
80 END
```

Tapez votre nom, puis appuyez sur NEWLINE.

**4**

```
RUN
QUEL EST TON NOM
? MAX-LE-ROBOT
QUEL EST TON ÂGE
? 77
MAX-LE-ROBOT
A
77 ANS
```

Si vous avez un ordinateur, essayez d'entrer ce programme, puis tapez RUN pour le lancer. Lorsque l'ordinateur vous demandera des informations, donnez-lui votre nom et votre âge, ou n'importe quel nom et chiffre fantaisistes comme dans

l'exemple donné ci-dessus. Renouvelez l'expérience plusieurs fois en introduisant des données différentes. L'ordinateur affichera toujours ce que vous venez d'entrer en N\$ et A.

## Un peu de poésie !

Vous savez maintenant assez de BASIC pour écrire un poème sur un ordinateur. Voici un programme de poésie qui utilise les instructions PRINT et INPUT.

```
10 PRINT «QUEL EST TON NOM»
20 INPUT NS
30 PRINT «POÈME ÉCRIT PAR»; s'inscrit votre nom.
40 PRINT NS
50 PRINT «TAPE UN MOT»
60 PRINT «QUI RIME AVEC PEUR»
70 INPUT AS
80 PRINT «VOICI LE POÈME»
90 PRINT «DANS LE TEMPS LES ORDINATEURS
ME FAISAIENT PEUR»;
100 PRINT «MAIS MAINTENANT JE SUIS
AUSSI HEUREUX QU'UN»
110 PRINT AS
120 END
```

A cette ligne s'inscrit le mot que vous avez choisi.



En exécutant le programme, l'ordinateur vous demande votre nom, mémorise votre réponse en NS, puis l'affiche à la ligne 40. Il stocke le mot que vous avez choisi en AS, et l'écrit à la ligne 110, comme s'il

```
RUN
QUEL EST TON NOM
? JEAN
POÈME ÉCRIT PAR JEAN
TAPE UN MOT
QUI RIME AVEC PEUR
? RATON LAVEUR ←
VOICI LE POÈME
DANS LE TEMPS LES
ORDINATEURS ME FAISAIENT PEUR
MAIS MAINTENANT JE SUIS AUSSI
HEUREUX QU'UN RATON LAVEUR
```

Votre nom et le mot.



Tapez RUN pour recommencer avec un autre mot.

était intégré au poème. Si vous avez un ordinateur, testez ce programme plusieurs fois en entrant des mots différents à la ligne 70.

## Casse-tête

Écrivez un programme dans lequel l'ordinateur vous demande votre nom, puis écrit Bonjour, suivi de votre nom et d'un message à votre intention.

26 œdæ uoiojnos

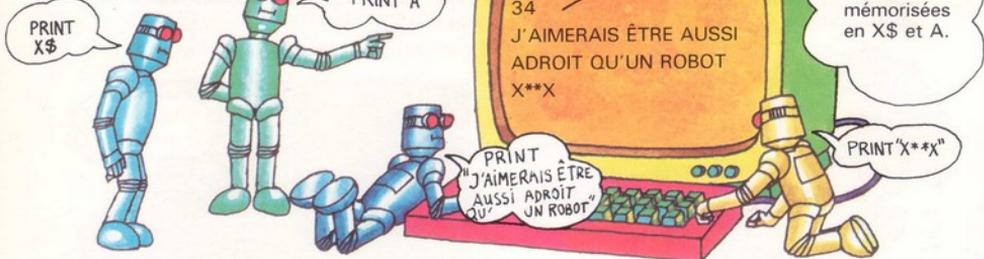
## Ce qu'il faut faire avant d'entrer un programme

1. Avant d'entrer un nouveau programme, tapez NEW. Vous effacerez ainsi tous les listings ou variables restant en mémoire.
2. Pendant que vous entrez le programme, n'oubliez pas de taper sur la touche NEWLINE (ou la touche correspondante de votre ordinateur) à la fin de chaque ligne d'instructions.
3. Après avoir entré le programme, vérifiez chaque ligne pour contrôler qu'il n'y a pas de fautes de frappe ni de ligne oubliée.
4. Vous pouvez alors taper CLS (ou l'instruction propre à votre ordinateur) pour vider l'écran. Lancez ensuite le programme en tapant RUN.
5. Pour faire réapparaître le programme, que ce soit pour le vérifier ou le modifier, tapez LIST. Pour visualiser une ligne en particulier, vous pouvez taper LIST suivi du numéro de la ligne d'instructions. Contrôlez sur la notice d'utilisation de votre micro-ordinateur.
6. Pour arrêter le programme en cours d'exécution, tapez BREAK ou ESCAPE. Là aussi, vérifiez dans votre notice. Attention sur certains appareils, ESCAPE élimine le programme de la mémoire de l'ordinateur. Pour relancer le programme, tapez RUN.

Voir pages 90-91 comment éviter quelques bugs.



# Que faire avec PRINT



Jusqu'à présent, on a vu comment utiliser PRINT pour afficher des chiffres et des lettres à l'écran, ou pour visualiser des variables. Nous allons maintenant apprendre à jouer avec les virgules et les

points-virgules pour situer l'affichage sur l'écran. Avec PRINT, on peut aussi effectuer des opérations. La page ci-contre donne d'autres exemples d'utilisation des variables.

## Virgules et points-virgules

10 PRINT « C'EST »  
20 PRINT « ESPACÉ » ← virgule

10 PRINT « C'EST »;  
20 PRINT « ATTACHÉ » ← point-virgule

10 PRINT « C'EST VRAIMENT »  
20 PRINT  
30 PRINT « ESPACÉ »

C'EST ESPACÉ

C'EST ATTACHÉ

C'EST VRAIMENT

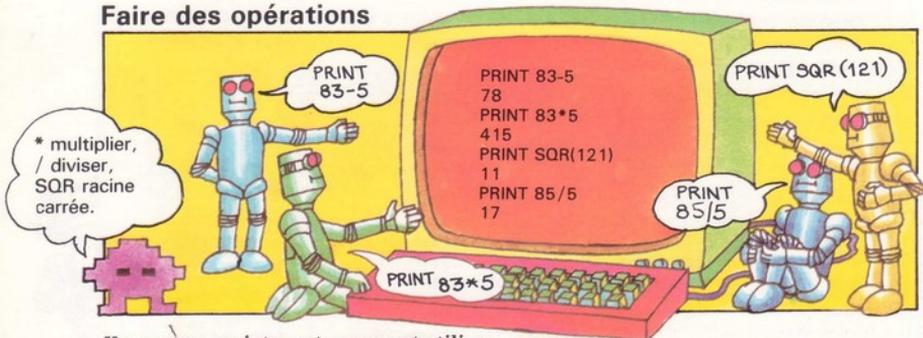
ESPACÉ

PRINT employé seul fait sauter une ligne

Voici comment utiliser les virgules et les points-virgules pour indiquer à l'ordinateur où il doit afficher la donnée. Une virgule lui indique de laisser un espace; un point-virgule signifie qu'il faut

enchaîner directement. L'image montre comment s'afficheraient les lignes sur l'écran selon le signe de ponctuation choisi. L'instruction PRINT employée seule fait sauter une ligne.

## Faire des opérations

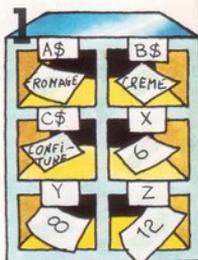


Vous savez, maintenant, comment utiliser PRINT. Pour l'addition et la soustraction, servez-vous des signes + et - habituels; pour multiplier, tapez \* (pour ne pas

confondre ce signe avec la lettre x), et pour diviser /. L'ordinateur peut réaliser des opérations plus sophistiquées comme le calcul des sinus, cosinus, racines carrées, etc.

## Pour en savoir plus sur les variables

Sur la plupart des ordinateurs, il faut laisser un espace de chaque côté de la variable entre les guillemets.



2

Espaces

PRINT « J'AI MANGÉ »; X; « SAUCISSES ET DES SANDWICHES »; A\$;  
J'AI MANGÉ 6 SAUCISSES ET DES SANDWICHES AU FROMAGE

PRINT « J'AI MANGÉ »; Z; « SAUCISSES ET DES SANDWICHES »; C\$;  
J'AI MANGÉ 12 SAUCISSES ET DES SANDWICHES À LA CONFITURE

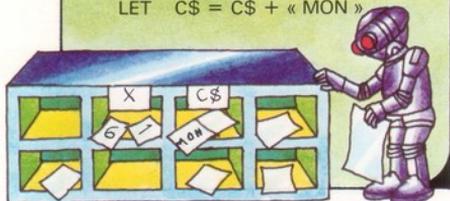
Afficher des variables n'est pas, en soi, très intéressant. Il faut souvent les intégrer dans une phrase. Pour afficher des mots et une variable ensemble, on doit placer les mots entre guillemets et

mettre un point-virgule de part et d'autre de la variable. Si l'on veut que l'information se détache, il suffit d'utiliser des virgules à la place de points-virgules.

3

LET X = X + 1

LET C\$ = C\$ + « MON »



En cours de programme, on peut changer le contenu des espaces-mémoire. Pour l'ordinateur, l'exemple cité dans ce dessin signifie qu'il faut ajouter 1 au chiffre étiqueté X et « mon » aux lettres étiquetées C\$.

4

Espaces

PRINT « J'AI MANGÉ »; X; « SAUCISSE  
ET »; C\$; « SANDWICHES »

J'AI MANGÉ 7 SAUCISSES ET DES  
SANDWICHES

Chaque fois que l'on demande à l'ordinateur d'afficher les variables, il restitue les dernières données enregistrées.

5

10 LET A=9

20 LET B=7

30 PRINT A\*B

40 PRINT A/B

50 END

RUN

63

1.28571

Multiplier

Diviser

On peut également effectuer des opérations avec des variables. L'ordinateur va chercher les nombres dans les espaces-mémoire, puis donne le résultat.

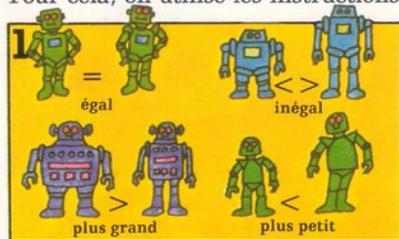
### Casse-tête

- Écrivez un programme qui permette d'ajouter des nombres aux variables du programme situé à gauche de telle sorte que les réponses 100 et 1 s'affichent sur une même ligne, tout en étant séparées par un espace.
- Transformez les lignes 30 et 40 pour qu'elles inscrivent les nombres, les opérations effectuées et les résultats : par exemple « 7 fois 9 font 63 ».
- Répondez au casse-tête de la page 63 de telle sorte que votre nom et le message s'inscrivent sur une même ligne.

# Comment les ordinateurs comparent

Une des fonctions de l'ordinateur : être capable de comparer des informations et d'en tirer les conséquences. Pour cela, on utilise les instructions IF ... THEN (si ... alors).

SI (IF) IL FAIT PLUS FROID DEMAIN ALORS (THEN) JE METTRAI MON ÉCHARPE



2 IF A = B THEN PRINT « ILS SONT ÉGAUX »  
IF A > B THEN PRINT « A EST PLUS GROS »  
IF A < B THEN PRINT « A EST PLUS PETIT »  
IF A <> B THEN PRINT « ILS NE SONT PAS ÉGAUX »

L'ordinateur peut réaliser plusieurs types de mesures pour comparer des informations. Vous voyez les symboles utilisés ci-dessus. Le micro-ordinateur peut déterminer si deux éléments sont identiques ou différents, si l'un est plus grand ou plus petit que l'autre.

Ces lignes montrent comment combiner les symboles avec les instructions IF et THEN pour que l'ordinateur effectue des comparaisons entre éléments. On peut comparer n'importe quels éléments : mots, chiffres ou variables.

## 3 Programme météo

```
10 PRINT « QUEL TEMPS FAIT-IL AUJOURD' HUI »
20 INPUT T$
30 IF T$ = « PLUVIEUX » THEN
  PRINT « PENSEZ AU PARAPLUIE »
40 IF T$ = « ENSOLEILLÉ » THEN
  PRINT « C'EST BON »
50 END
```

Si vous entrez ces mots, rien ne se passera.

NEIGE

CHAUD

```
4 RUN
QUEL TEMPS FAIT-IL
AUJOURD' HUI
? ENSOLEILLÉ
C'EST BON
RUN
QUEL TEMPS FAIT-IL
AUJOURD' HUI
? PLUVIEUX
PENSEZ AU
PARAPLUIE
```

Voici un programme qui utilise IF et THEN. A la ligne 20, l'ordinateur met en mémoire le mot entré en variable T\$. Puis, aux lignes 30 et 40, il cherche à savoir si le mot entré en T\$ est

« pluvieux » ou « ensoleillé ». Si c'est le cas, il affiche une des réponses indiquées. Sinon, rien ne se passe; il faudra alors modifier les mots des lignes 30 et 40.

## 5 Question d'âge

```
10 PRINT « QUEL ÂGE AS-TU ? »
20 INPUT A
30 IF A > 16 THEN PRINT « VIEUX »
40 IF A < 16 THEN PRINT « JEUNE »
50 IF A = 16 THEN PRINT « C'EST BON »
RUN
QUEL ÂGE AS-TU
? 16
C'EST BON
```

Dans ce programme « Quel âge as-tu? », l'ordinateur compare l'INPUT A avec le chiffre 16. S'il est plus petit que 16, il affiche « jeune »; plus grand, il affichera « vieux »; s'il est égal à 16, il affiche

## 6 Leçon d'anglais

```
10 PRINT « COMMENT DIT-ON ROUGE
EN ANGLAIS »
20 INPUT A$
30 IF A$ = « RED » THEN PRINT « JUSTE »
40 IF A$ <> « RED » THEN PRINT
« NON, RED »
RUN
COMMENT DIT-ON ROUGE EN ANGLAIS
? BLUE
NON, RED
```

« c'est bon ». Dans le programme de droite, l'ordinateur affiche l'une ou l'autre réponse, selon que A est identique à « red » ou non.

## Branchements de programmes

**1** IF A=6 THEN LET A\$="SIX"  
IF X=Y-2 THEN LET Z=0  
IF S=T THEN STOP  
IF R<10 THEN GOTO 30

L'ordinateur est envoyé à la ligne 30.



On peut donner presque n'importe quel ordre à l'ordinateur après l'instruction THEN. Ce qui s'avère utile, c'est par exemple de l'envoyer à une autre ligne du programme (sur la plupart des

**2** 10 PRINT K\$  
20 IF K\$ = « OUI » THEN  
GOTO 100  
30 IF K\$ = « NON » THEN  
GOTO 200  
100 PRINT « VOUS AVEZ TAPÉ OUI »  
110 STOP  
200 PRINT « VOUS AVEZ TAPÉ NON »  
210 END

Ces deux lignes envoient l'ordinateur à d'autres parties du programme.



ordinateurs, hormis le SINCLAIR ZX81, on peut se dispenser de l'instruction GOTO). Il faut introduire un STOP après GOTO, sinon l'ordinateur continuerait à répéter sans fin le même programme.

## Programme de mathématiques

```
10 PRINT « TAPEZ UN NOMBRE »
20 INPUT A
30 PRINT « TAPEZ UN AUTRE NOMBRE »
40 INPUT B
50 PRINT « QUE VOULEZ-VOUS FAIRE »
60 PRINT « ADDITIONNER, SOUSTRAIRE,
    MULTIPLIER »
70 PRINT « DIVISER OU ARRÊTER »
80 INPUT CS
90 IF CS = « ADDITIONNER » THEN
    PRINT A + B
100 IF CS = « SOUSTRAIRE » THEN
    PRINT A - B
110 IF CS = « MULTIPLIER » THEN
    PRINT A*B
120 IF CS = « DIVISER » THEN PRINT A/B
130 IF CS = « STOP » THEN STOP
140 GOTO 10
```

```
RUN
TAPEZ UN NOMBRE
? 17
TAPEZ UN AUTRE NOMBRE
? 184
QUE VOULEZ-VOUS FAIRE
ADDITIONNER, SOUSTRAIRE, MULTIPLIER
DIVISER OU STOPPER
? ADDITIONNER
201 ← Réponse de l'ordinateur
TAPEZ UN NOMBRE
?
```

LE PROGRAMME SE S'ARRÊTE QU'EN ENTRANT L'INSTRUCTION STOP



Dans ce programme, les nombres entrés sont stockés en A et B et vos ordres en CS. Des lignes 80 à 130, l'ordinateur compare C avec cinq mots différents. Quand il reconnaît le bon mot, il exécute l'instruction. Il saute toutes les lignes qui ne le concernent pas.

## Programme « Quel est ton âge »

**1** 10 PRINT  
« QUEL EST MON ÂGE »  
20 INPUT A  
30 IF A<<14 THEN PRINT  
« ESSAYE ENCORE »  
40 IF A<>14 THEN  
GOTO 20  
50 PRINT « JUSTE »  
60 END

**2** RUN  
QUEL EST MON ÂGE  
? 15  
ESSAYE ENCORE  
? 14  
JUSTE

**3** QUEL EST MON ÂGE  
? 15  
PLUS JEUNE QUE ÇA  
? 13  
PLUS VIEUX QUE ÇA  
? 14  
JUSTE

SURIEZ-VOUS PROGRAMMER CEÇI ?



Ce programme se répétera jusqu'à ce que A = 14. Quand A = 14, l'ordinateur sautera les lignes 30 et 40 et affichera « bonne réponse ». Pouvez-vous modifier

le programme de telle sorte qu'il vous fournisse quelques indications, comme le montre le dessin de droite?

# Programmes en BASIC

Les programmes présentés sur ces deux pages utilisent la plupart des notions de BASIC que nous avons rencontrées. Le premier programme propose un jeu spatial sur ordinateur pour deux joueurs. Si vous n'avez pas d'ordinateur, regardez bien ces programmes et essayez de comprendre leur fonctionnement.



## Commando de l'espace

```
10 PRINT « POSITION DE L'ENNEMI : LATITUDE »
20 INPUT A
30 PRINT « POSITION DE L'ENNEMI : LONGITUDE »
40 INPUT B
50 CLS
60 PRINT « POSITION DU COMMANDO : LATITUDE »
70 INPUT C
80 PRINT « POSITION DU COMMANDO : LONGITUDE »
90 INPUT D
100 CLS
110 LET X=SQR((A-C)*(A-C)+(B-D)*(B-D))
120 PRINT « VOUS ÊTES MAINTENANT A »
130 PRINT X; « CASES L'UN DE L'AUTRE »
140 IF X<1.5 THEN PRINT « ENNEMI LOCALISÉ »
150 IF X<1.5 THEN STOP
155 PRINT « INDIQUEZ VOS NOUVELLES POSITIONS »
160 GOTO 10
170 END
```

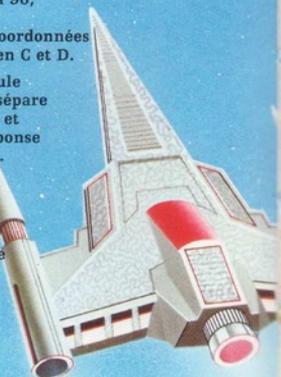
De la ligne 10 à 40, l'ordinateur mémorise les coordonnées ennemies en A et B.

La ligne 50 fait disparaître les coordonnées ennemies.

De la ligne 60 à 90, l'ordinateur mémorise les coordonnées du commando en C et D.

Cette ligne calcule la distance qui sépare les combattants et mémorise la réponse en X (ligne 110).

Si X est plus petit que 1,5 le jeu s'arrête. Si X est plus grand que 1,5 le programme recommence.

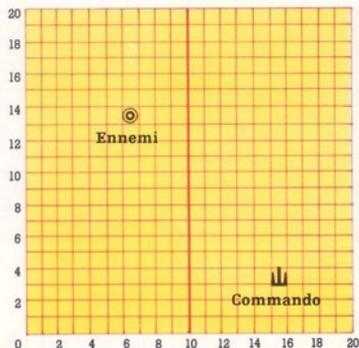


Dans ce jeu, un commando essaye d'attraper l'ennemi. Chaque joueur dessine une carte secrète sur laquelle il inscrit ses positions (on vous indique ci-dessous comment procéder).

Chacun transmet ses coordonnées à l'ordinateur qui détermine quelle distance sépare les adversaires. Les calculs de l'ordinateur aident les joueurs à choisir leur prochain coup.

## Règle du jeu

La carte secrète : chaque joueur dessine une grille de 20 carrés sur 20 carrés et les numérote comme indiqué ci-contre. Les ennemis démarrent sur la gauche de la grille, le commando sur la droite. A chaque coup, on peut se déplacer de deux cases, dans n'importe quelle direction. Puis on indique ses nouvelles coordonnées à l'ordinateur. Quand le commando parvient à moins d'1,5 cases de l'ennemi, on estime qu'il l'a attrapé.



## Comment donner l'air intelligent à l'ordinateur

Dans ce programme, l'ordinateur donne l'impression de dialoguer et d'établir une véritable conversation avec vous. Le fonctionnement de ce programme figure dans les dessins en bas de page. On utilise l'instruction INPUT d'une manière un peu différente de celle que nous avons décrite jusqu'ici. Ce qui rend le programme plus court et plus facile à lire.

**1**

```
10 INPUT « TAPE UN NOMBRE »; N
20 INPUT « UN AUTRE »; M
30 PRINT N; « FOIS »; M;
  « FONT »; N*M
```

Le BBC n'a pas besoin de point-virgule.



**2**

```
RUN
TAPE UN NOMBRE? 10
UN AUTRE? 8
10 FOIS 8 FONT 80
```

Sur le SINCLAIR ZX81, il faut taper  
10 PRINT « TAPE UN NOMBRE »  
15 INPUT N



Sur la plupart des ordinateurs (sauf le SINCLAIR ZX81), on peut rendre la ligne INPUT plus explicite en plaçant quelques mots entre guillemets avant la variable.

Dans le déroulement du programme, le point d'interrogation apparaît à la fin de la chaîne de caractères.

## Programme

Voici la nouvelle façon d'utiliser INPUT. Votre réponse est mémorisée en AS.

A la ligne 30, l'ordinateur recherche la première apparition de DATA, prend le premier article et l'inscrit en BS.

La variable C aux lignes 60 et 70 compte le nombre de tours effectués : elle agit comme un contrôleur de boucle. Lorsque C = 6, tous les articles en DATA ont été utilisés et l'ordinateur passe à la ligne 100.

La ligne 80 fait retourner l'ordinateur à la ligne 30; c'est l'article suivant dans la liste de DATA qui apparaîtra.

Les blancs laissés aux lignes 110 et 130 permettent de libérer un espace à l'écran avant que ne s'affiche votre réponse. Peu importe le nombre d'espacements laissés.

```
5 LET C=0
10 PRINT « J'AIMERAIS QU'ON PARLE ENSEMBLE »
20 INPUT « RACONTE-MOI QUELQUE CHOSE DE DRÔLE QUI
  TE SOIT ARRIVÉ CETTE SEMAINE »; AS
30 READ BS
40 PRINT BS ← L'ordinateur reste à la même ligne.
50 INPUT CS ← Votre réponse est mémorisée en CS.
60 LET C=C+1
70 IF C=6 THEN GOTO 100
80 GOTO 30
90 DATA POURQUOI, POURQUOI DONC
95 DATA EXPLIQUE-MOI, POURQUOI
98 DATA DIS-MOI POURQUOI, POUR QUELLE RAISON
100 PRINT « AINSI, LA VÉRITABLE RAISON POUR LAQUELLE »
110 PRINT « »; AS
120 PRINT « C'EST QUE »
130 PRINT « »; CS
140 PRINT « COMME C'EST BIZARRE »
150 PRINT « TAPE RUN, QUE NOUS CONTINUIONS
  À NOUS AMUSER »
160 END
```

## Comment ça marche

**1**

RUN  
J'AIMERAIS QU'ON PARLE ENSEMBLE  
RACONTE-MOI QUELQUE CHOSE DE DRÔLE QUI TE SOIT ARRIVÉ CETTE SEMAINE

JE SUIS TOMBÉ DANS UN TROU



**2**

POURQUOI

PARCE QUE JE NE REGARDAIS PAS OÙ JE METTAIS LES PIEDS.



**3**

POURQUOI DONC

JE MANGEAIS UNE GLACE.

EXPLIQUE-MOI



**4**

JE ME LÉCHAIS LES DOIGTS

DIS-MOI POURQUOI

PARCE QUE ÇA COULAIT

POURQUOI



**5**

PARCE QUE JE LA CACHAIS

POUR QUELLE RAISON

JE NE VOUAIS PAS QUE MON COPAIN LA VOIT.

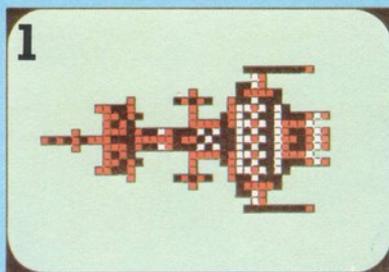


AINSI, LA VÉRITABLE RAISON POUR LAQUELLE JE SUIS TOMBÉ DANS UN TROU C'EST QUE JE NE VOUAIS PAS QUE MON COPAIN VOIE QUE J'AVAIS UNE GLACE COMME C'EST BIZARRE TAPE RUN, QUE NOUS CONTINUIONS À NOUS AMUSER

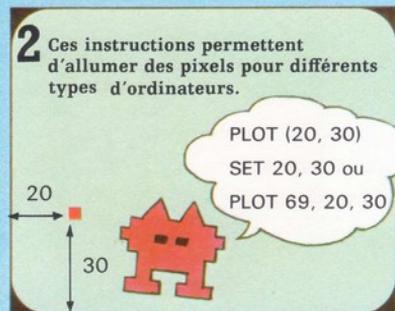
# Dessins

Un ordinateur compose des dessins en allumant des petits rectangles sur l'écran. Chaque rectangle, pixel en anglais, qu'on peut appeler un témoin, répond à une instruction particulière. Dans ces deux pages, on découvre comment se servir du BASIC

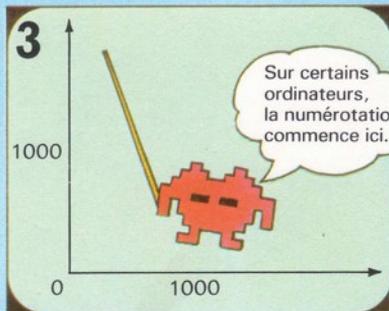
pour dessiner sur l'écran. Beaucoup d'ordinateurs permettent de dessiner en couleurs : reportez-vous donc à votre notice explicative pour profiter de cet avantage.



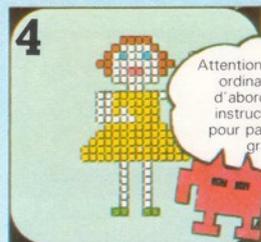
Souvent, on distingue les témoins graphiques sur le dessin. Cependant un ordinateur à grande mémoire peut réaliser des dessins avec des milliers de tout petits points. Ils sont dits dessins de très haute résolution.



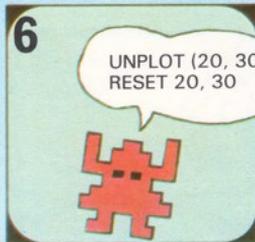
Ces instructions permettent d'allumer des pixels pour différents types d'ordinateurs. L'instruction qui permet d'allumer un pixel varie selon les appareils, mais elle se présente généralement sous la forme de PLOT (X, Y). X et Y sont les coordonnées en abscisse et en ordonnée des points témoins.



Un ordinateur à haute résolution graphique offre jusqu'à 1 000 points en abscisses et en ordonnées. Un ordinateur moins puissant présente une grille d'environ 60 x 40. (Vérifiez les dimensions de votre écran, car vous risquez de situer votre point en dehors des limites définies.)



Attention : pour certains ordinateurs, il faut d'abord donner une instruction générale pour passer en mode graphique.



On appelle graphiques les dessins réalisés par un ordinateur. Sur certains appareils, il faut donner une instruction spéciale pour passer en mode graphique. Par exemple, sur le BBC, on doit taper MODE plus un nombre\*.

Pour éteindre un pixel, on utilise l'instruction UNPLOT (X, Y). Vérifiez cette commande dans la notice.

\* Pour les programmes de ce livre, utilisez MODE 5, puis PLOT 69, X, Y. Pour éteindre, PLOT 71, X, Y.

## Programmes de tracés

### 1

```
10 PRINT
« ENTREZ DEUX NOMBRES »
20 INPUT X
30 INPUT Y
40 PLOT (X, Y)
50 GOTO 10
```

Les instructions de tracé varient selon les ordinateurs.



Il faut taper NEWLINE ou RETURN après avoir entré chaque nombre.

Ce court programme réclame deux nombres, puis allume deux pixels aux coordonnées indiquées. Vérifiez que les nombres que vous entrez sont compris dans les limites de votre ordinateur.

### 2

```
RUN
ENTREZ DEUX NOMBRES
?24
?24
ENTREZ DEUX NOMBRES
?30
?15
```

premier point

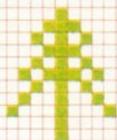
deuxième point

La ligne 50 permet au programme de se répéter sans fin. La seule façon de l'arrêter, c'est de taper sur la touche BREAK (ou la touche correspondante de votre ordinateur). Sauriez-vous introduire un contrôleur de boucle (cf. page 69) qui ne fasse tourner le programme que six fois ?

## Dessins

```
10 LET Y=10
20 LET Y=10
30 PLOT (X, Y)
40 LET X=X+1
50 LET Y=Y-1
60 IF X<14 THEN GOTO 30
```

Vous dessinez ainsi une diagonale descendante.



```
100 LET Y=Y+1
110 LET X=X+1
120 PLOT (X, Y)
130 IF X<20 THEN GOTO 100
```

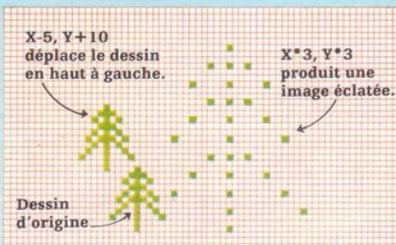
Vous dessinez ainsi une diagonale ascendante.

En ajoutant 1 à X, et pas à Y, on obtient une ligne horizontale.

En ajoutant 1 à Y, et pas à X, on obtient une ligne verticale.

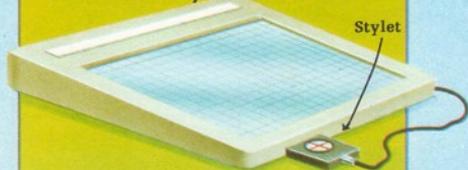
Il faut d'abord tracer le dessin sur une feuille quadrillée et déterminer les coordonnées de la figure.

On peut alors utiliser le programme pour reporter les coordonnées de chaque case. A partir des valeurs d'origine de X et Y, en les augmentant ou en les diminuant, et en faisant se répéter certaines parties du programme, l'ordinateur trace des séries de points.



Une fois le programme écrit, il est facile de transformer le dessin en changeant la valeur des nombres. On peut le déplacer sur l'écran, en modifiant les valeurs d'origine, ou produire une « image élargie » en multipliant tous les nombres par trois.

## Une autre façon de dessiner



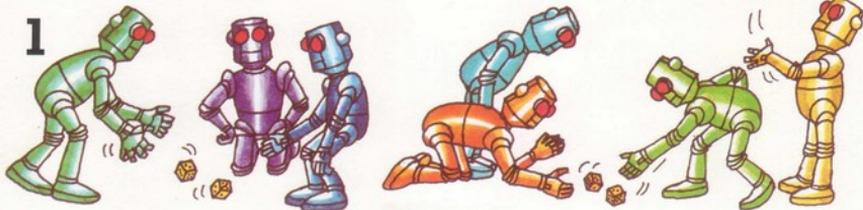
Avec la fonction PLOT, on n'obtient que des dessins rudimentaires. Pour faire mieux, il faut utiliser un matériel spécial, comme une tablette graphique. On place un dessin sur la tablette, puis on en suit le tracé avec un stylet. Les coordonnées s'enregistrent automatiquement dans l'ordinateur.



Casse-tête. Écrivez un programme qui fasse se dessiner vos initiales à l'écran. Vous en trouverez un exemple page 92.

# Jouons

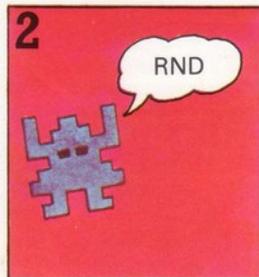
1



Quand on lance les dés, on ne sait jamais quels numéros sortiront. N'importe quel chiffre entre 1 et 6 a des chances égales. De la même manière, l'ordinateur peut générer des nombres au hasard : ce sont des *nombres aléatoires*. La fonction

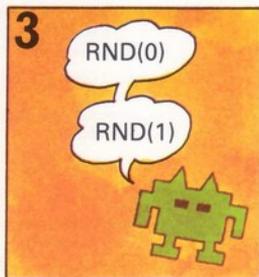
RANDOM (RND) permet à l'ordinateur de les sélectionner. Parfois des chiffres se répètent; mais, dans des séries de nombres aléatoires, la probabilité de ressortir plusieurs fois la même valeur est à peu près nulle.

2



Pour que l'ordinateur génère un nombre aléatoire, on utilise l'instruction RND (RANDOM). Sur certains modèles, il faut ajouter 1 ou 0 entre parenthèses. Consultez votre notice pour confirmation.

3

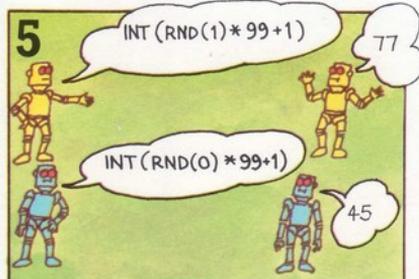


4



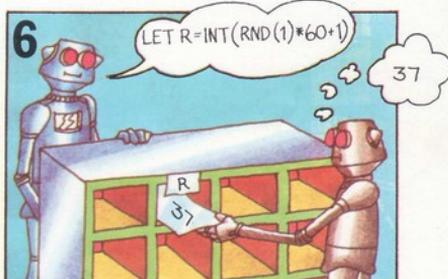
L'instruction RND propose toujours un nombre inférieur à 1. Sur certains ordinateurs, on peut écrire un nombre entre parenthèses après RND. On obtient alors une valeur située entre 1 et le nombre placé entre parenthèses.

5



Sur d'autres ordinateurs, on utilise l'instruction INT (abréviation de "integer" = nombre entier) avant RND [ou RND(1) ou RND(0)]. Il faut ensuite multiplier par le nombre le plus fort que vous souhaitiez et ajouter 1, car vous ne retiendrez que des valeurs supérieures à 1.

6



Cette instruction signifie : l'ordinateur tire un nombre aléatoire et le mémorise en R. Les programmes proposés dans ce livre utilisent la formulation suivante : INT(RND(1)\*60+1). Il se peut que vous deviez la transformer pour votre ordinateur.



## L'attaque spatiale

Ce programme de jeu utilise les nombres aléatoires. Dans ce jeu, votre vaisseau spatial est attaqué par une nuée d'extra-terrestres. L'ordinateur du vaisseau repère l'ennemi et vous donne ses positions. Pour détruire chaque extra-terrestre, on doit déterminer l'angle de tir en multipliant les coordonnées qui vous sont communiquées et en tapant le résultat.



```
10 LET C=0
20 LET A=INT(RND(1)*20+1)
30 LET B=INT(RND(1)*20+1)
40 PRINT « POSITION DES EXTRA-TERRESTRES »
45 PRINT A, B; « FEU »
50 INPUT X
60 LET C=C+1
70 IF X=A*B THEN PRINT
  « VAISSEAU ENNEMI DÉTRUIT »
80 IF X<>A*B THEN PRINT « RATÉ »
90 IF C<6 THEN GOTO 20
100 END
```

C permet de compter le nombre de répétitions du programme. A chaque passage, la ligne 60 ajoute 1 à C.

Ces deux lignes génèrent des nombres aléatoires qui déterminent les positions extra-terrestres. Les coordonnées sont mémorisées en A et B

Votre réponse est enregistrée en X.

Aux lignes 70 et 80, l'ordinateur compare votre réponse à ses propres résultats.

Cette ligne relance le programme si C est inférieur à 6.

## Le programme tourne

L'encadré à droite montre ce qui se passe quand le programme est lancé. Si vous donnez la bonne réponse (le résultat de la multiplication des deux coordonnées), l'ordinateur affichera « vaisseau ennemi détruit ». Si vous vous êtes trompé, vous lirez « raté ».

```
RUN
POSITION DES EXTRA-TERRESTRES
17 3 FEU           La ponctuation
741               de la ligne 45
RATÉ              établit ces espaces.
POSITION DES EXTRA-TERRESTRES
11 5 FEU
755
VAISSEAU ENNEMI DÉTRUIT
POSITION DES EXTRA-TERRESTRES
13 6 FEU
```

## Casse-tête

Sauriez-vous ajouter un autre élément au programme pour comptabiliser le nombre d'ennemis abattus et voir s'afficher votre score? Vous devez introduire une variable, S, lui donner au départ la valeur 0, puis l'augmenter de 1 à chaque réussite.

## Dessin aléatoire

```
5 CLS
10 LET X=INT(RND(1)*30+1)
20 LET Y=INT(RND(1)*30+1)
30 PLOT (X,Y)
40 GOTO 10
```

Cette instruction vide l'écran.

Contrôlez que la valeur donnée aux nombres aléatoires correspond aux capacités de votre ordinateur.

Cette instruction permet au programme de se répéter indéfiniment.

Dans ce programme, on utilise des nombres aléatoires pour afficher des pixels à l'écran. Les lignes 10 et 20 tirent des nombres entre 1 et 30 et les mémorisent en X et Y. La ligne 30

affiche le point dont les coordonnées viennent d'être sorties. Pour arrêter le programme, on doit taper BREAK, ESCAPE, ou le mot qui a cette fonction sur votre ordinateur.

Les instructions CLS, RND, PLOT varient d'un ordinateur à l'autre. Parfois, on devra passer par le mode graphique.



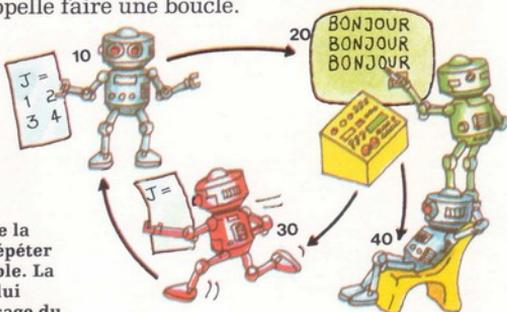
# Faisons des boucles

Dans un programme, on a souvent besoin que l'ordinateur répète plusieurs fois la même chose. Vous avez vu, à la page 69, qu'on peut obtenir ce résultat en utilisant GOTO et une variable qui fait fonction de compteur. On peut aussi passer par les instructions FOR, TO et NEXT. C'est ce qu'on appelle faire une boucle.

## 1 La boucle du Bonjour

```
10 FOR J=1 TO 6  
20 PRINT « BONJOUR » Boucle  
30 NEXT J  
40 END
```

Ce programme forme une boucle de la ligne 10 à la ligne 30, qui fera se répéter six fois la ligne 20. J est une variable. La ligne 10 indique à l'ordinateur de lui donner la valeur 1 au premier passage du programme, 2 à la suivante et ainsi de suite jusqu'à 6. Lorsque J=6, l'ordinateur passe à la ligne 40.



## 2 Opérations fantaisistes

```
10 FOR J=1 TO 8  
20 PRINT « 2 PLUS 2 FONT 5 » Boucle  
30 NEXT J  
40 PRINT  
50 PRINT « C'EST POUR RIRE ! »  
60 END
```

Certains ordinateurs n'ont pas de point d'exclamation. On peut s'en passer.

```
2 PLUS 2 FONT 5  
2 PLUS 2 FONT 5
```

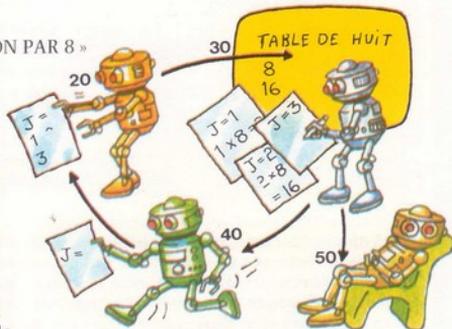
C'EST POUR RIRE !

Dans ce programme, la boucle des lignes 10 à 30 fait exécuter 8 fois la ligne 20 à l'ordinateur. A chaque fois, il affiche cette même opération fantaisiste. Au bout de

huit passages, l'ordinateur continue; la ligne 40 ne sert qu'à lui faire sauter une ligne.

## 3 Table de multiplication par 8

```
10 PRINT « TABLE DE MULTIPLICATION PAR 8 »  
20 FOR J=1 TO 12  
30 PRINT J*8 Boucle  
40 NEXT J  
50 END
```



Cette fois, J sert à compter le nombre de boucles. Mais il intervient aussi comme élément de l'opération  $J*8$ . La ligne 20 indique à l'ordinateur de donner à J la valeur 1, puis 2, 3, ainsi de suite jusqu'à 12. La ligne 30 prend la valeur donnée à J, la multiplie par 8 et affiche le résultat. Enfin, la ligne 40 renvoie l'ordinateur à la ligne 20 où J prend une nouvelle valeur.

## Pour les dessins

On utilise les boucles FOR... NEXT pour réaliser des dessins simples et répétitifs. Le programme qui correspond à cette figure serait trop long à écrire, mais il ressemblerait à ceci :

```
10 FOR I=1 TO 45
20 Dessine un rectangle et change un tout
   petit peu ses coordonnées à chaque fois.
30 NEXT I
40 END
```

## Pas de boucle

Parfois on veut changer la valeur de J, non plus de 1 en 1, mais de 3 en 3 ou de 7 en 7. Pour cela, on utilise l'instruction STEP (pas). Dans le programme ci-dessous, STEP-1 fait décroître de 1 la valeur de J à chaque passage du programme de la ligne 10 à la ligne 40.

## L'ordinateur gourmand

Le chiffre 2 arrête le programme après J=2 (quand il ne reste plus qu'un gâteau).

```
5 CLS
10 FOR J=7 TO 2 STEP-1
20 PRINT « IL RESTE » ; J ; « GÂTEAUX »
30 NEXT J
40 PRINT
50 PRINT « JE VAIS EXPLOSER »
60 FOR K=1 TO 1000
70 REM: NE FAIS RIEN
80 NEXT K
90 PRINT
100 PRINT "SPLATCH"
```

Il y a deux boucles dans ce programme. Celle de la ligne 10 à 30 fait s'afficher 6 fois la ligne 20, avec une valeur de J minorée de 1 à chaque fois. Dans la boucle des lignes 60 à 80, l'ordinateur n'a rien à faire; il passe uniquement en revue toutes les valeurs de K comprises entre 1

```
IL RESTE 7 GÂTEAUX
IL RESTE 6 GÂTEAUX
IL RESTE 5 GÂTEAUX
IL RESTE 4 GÂTEAUX
IL RESTE 3 GÂTEAUX
IL RESTE 2 GÂTEAUX

JE VAIS EXPLOSER
SPLATCH
```

Certains ordinateurs sont plus lents que d'autres, et il faut remplacer 1000 par 500 ou 250 à la ligne 60.



et 1 000, ce qui provoque une pause dans le déroulement du programme. Les lignes commençant par REM (abréviation de "remark") ne sont pas prises en compte par l'ordinateur; elles ne servent qu'à vous rappeler ce qui se passe dans le programme.

## Casse-tête

1. Pouvez-vous transformer le programme de la table de 8 pour que s'affiche "1×8=" et le résultat.
2. Sauriez-vous écrire un programme de la table de N, que vous pourriez utiliser pour n'importe quel nombre que vous entrez sur l'ordinateur?

Celui-ci doit tout d'abord vous demander la valeur de N. Puis, au moyen d'une boucle, il calcule et affiche la table de multiplication. Vous pouvez ajouter quelques lignes à la fin du programme pour qu'il vous demande si vous désirez une autre table.

# Quelques astuces

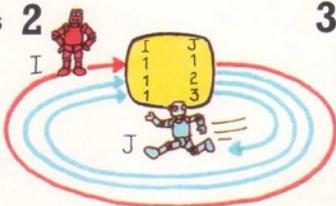
Voici d'autres programmes avec des boucles. On y découvre comment utiliser des boucles à l'intérieur d'autres boucles pour répéter plusieurs données à des rythmes différents. Ce sont des *boucles imbriquées*.

## 1 Boucles imbriquées

```

5 PRINT "I", "J"
10 FOR I=1 TO 3
20 FOR J=1 TO 3
30 PRINT I, J
40 NEXT J
50 NEXT I
60 END
    
```

Boucle J  
Boucle I



## 3

1	J
1	1
1	2
1	3
2	1
2	2
2	3
3	1
3	2
3	3

Le programme comporte une boucle I et une boucle J. La boucle J est imbriquée dans la boucle I. Pour chaque passage de la boucle I, la boucle J se répète trois fois,

affichant à chaque tour une nouvelle valeur de J. Vous voyez dans l'encadré ci-dessus les résultats de ce programme. Les espaces sont dus aux virgules.

### L'horloge de l'ordinateur

```

5 CLS
10 LET M=0
20 LET S=0
30 FOR M=0 TO 59
40 FOR S=0 TO 59
50 PRINT M; " "; S
60 CLS
70 NEXT S * des secondes
80 NEXT M ← Boucle des minutes
90 END
    
```

0:45

Pour mettre à l'heure, utilisez la boucle d'attente :

```

54 FOR Z = 1 TO 100
58 NEXT Z
    
```

### Des bugs dans les boucles

```

10 FOR I = 1 TO 4
20 FOR J = 1 TO 4
30 PRINT I
40 PRINT J
50 NEXT I
60 NEXT J
    
```

Les deux parties d'une boucle imbriquée doivent se trouver à l'intérieur de l'autre boucle.

Dans un ordinateur se trouve une horloge électronique qui imprime la fréquence de son travail. L'horloge émet entre 1 et 4 millions d'impulsions à la seconde. Ce programme indique à l'ordinateur de se comporter comme une horloge numérique. On utilise des boucles imbriquées, l'une pour compter les secondes, l'autre pour les minutes. La

boucle imbriquée tourne 59 fois pour un seul passage de la boucle des minutes. Quand vous testerez ce programme sur votre appareil, il se peut qu'il tourne trop vite. Ajoutez alors une boucle d'attente et adaptez-la en changeant le chiffre de telle sorte que votre ordinateur affiche au même rythme qu'une véritable horloge.

## Test des nombres aléatoires

```

10 FOR I=1 TO 1000
20 LET R=INT(RND(1)*6+1)
30 IF R=1 THEN LET A=A+1
40 IF R=2 THEN LET B=B+1
50 IF R=3 THEN LET C=C+1
60 IF R=4 THEN LET D=D+1
70 IF R=5 THEN LET E=E+1
80 IF R=6 THEN LET F=F+1
90 NEXT I
100 PRINT
« TERMINE »
110 PRINT A, B, C
120 PRINT D, E, F
130 END
    
```

Le déroulement de ce programme est très, très long. Vous pouvez le raccourcir en remplaçant à la ligne 10 le nombre 1 000 par 500 ou même 250.

RUN			
TERMINÉ			
162	168	167	
160	187	156	

Ce programme vérifie le fonctionnement de RANDOM. La boucle de la ligne 10 à 90 fait tirer à l'ordinateur un chiffre entre 1 et 6 un millier de fois. L'ordinateur totalise combien de fois chaque nombre a été mémorisé en variable A, B, C, D, E ou F, puis affiche les résultats\*.

\* Sur certains ordinateurs, comme le SINCLAIR ZX81 ou le BBC, il faut écrire quelques lignes en début de programme pour initialiser les variables.

## Répétition de dessins

Ce programme utilise des boucles imbriquées pour répéter un même petit dessin partout sur l'écran. Il semble ardu, mais si on le regarde soigneusement, si l'on découvre l'utilité de chaque ligne, on comprendra rapidement son mode de fonctionnement. C'est le tirage qui décidera de la forme du dessin; il sera donc différent à chaque fois que vous lancerez le programme.

Pour les ordinateurs à haute résolution graphique, prenez des nombres aléatoires plus importants. Ainsi, pour le BBC, changez les chiffres des lignes 10 à 40.



```

5 CLS
10 LET A=INT(RND(1)*6+1)
20 LET B=INT(RND(1)*7+1)
30 LET C=INT(RND(1)*6+1)
40 LET D=INT(RND(1)*4+1)
50 INPUT « QUELLE EST LA DÉFINITION
DE VOTRE ÉCRAN : LARGEUR »; L
60 INPUT « HAUTEUR »; H
65 CLS
70 FOR I=0 TO H STEP H/6
80 FOR J=0 TO L STEP L/6
90 PLOT (J+A,I+B)
100 PLOT (J+A,I+C)
110 PLOT (J+C,I+D)
120 PLOT (J+B,I+D)
130 NEXT J
140 NEXT I
150 END
    
```

Ces lignes tirent les nombres aléatoires et les mémorisent en A, B, C ou D.

Les lignes 50 et 60 demandent la largeur (L) et la hauteur (H) de votre écran.

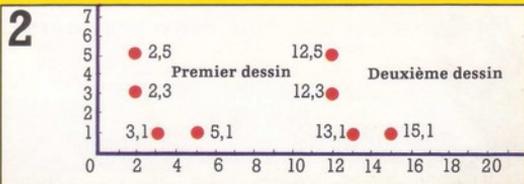
La boucle I compte le nombre de fois où le dessin se répète à l'écran. A chaque fois s'ajoute à I la hauteur de l'écran divisé par 6, si bien que le dessin s'affiche 6 fois en hauteur sur l'écran.

A chaque fois que les boucles se répètent, les lignes 90 à 120 font tracer à l'ordinateur 4 points correspondant aux valeurs de I et de J augmentées du nombre aléatoire.

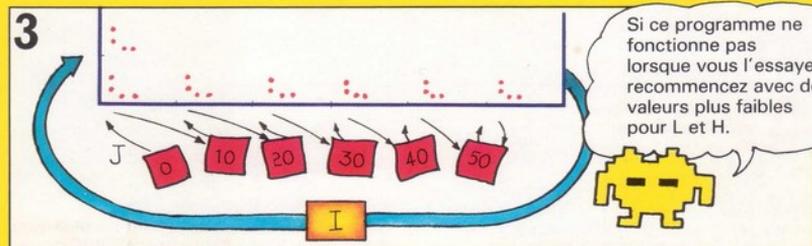
La boucle J compte le nombre d'apparitions du dessin sur la largeur de l'écran. Elle fonctionne comme la boucle I.



Imaginons que l'ordinateur ait tiré les nombres 2, 5, 3 et 1 et que la largeur et la hauteur de l'écran soient toutes deux de 60.



Au premier tour, I et J valent 0; l'ordinateur trace donc le premier dessin uniquement en fonction du nombre tiré. La ligne 130 le renvoie chercher la valeur suivante de J, qui est  $J+60/6$ , c'est-à-dire 10. Il trace alors le second dessin en utilisant le nombre aléatoire, plus 10 pour J. Le dessin se répète sur l'écran.



Si ce programme ne fonctionne pas lorsque vous l'essayez, recommencez avec des valeurs plus faibles pour L et H.



L'ordinateur répète 6 fois la boucle J, en ajoutant à chaque fois 10 à la valeur de J. Il revient ensuite prendre la valeur suivante de I qui est 10. J est de nouveau

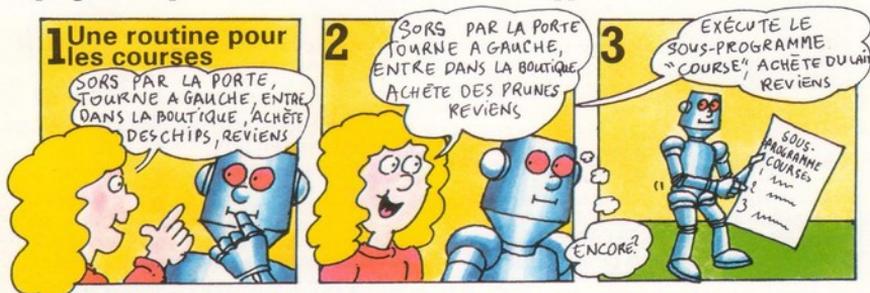
égal à 0 et l'ordinateur trace une nouvelle ligne de dessins en prenant 10 comme valeur de I et en augmentant à nouveau J de 10 en 10.



Casse-tête. Inventez un programme dans lequel la forme qui se répète serait un envahisseur extra-terrestre. Vous trouverez en page 93 quelques trucs pour vous aider.

# Sous-programmes

Un sous-programme est une petite routine qui se trouve à l'intérieur d'un autre programme. Sa fonction est d'exécuter des tâches particulières comme additionner des nombres ou tenir un score. On peut y envoyer l'ordinateur dès que l'on veut que cette tâche soit effectuée, ce qui évite d'écrire plusieurs fois les mêmes lignes et rend le programme plus facile à lire et à entrer dans l'appareil.



Imaginez que vous ayez un robot pour vous aider, et que vous puissiez programmer ses déplacements. Si vous voulez qu'il aille acheter quelque chose dans une boutique, il faut lui donner des indications précises sur la façon de s'y

rendre. A chaque fois que vous voudrez qu'il fasse des courses, vous devrez lui répéter les mêmes ordres. Il serait beaucoup plus facile de programmer une routine de courses et de lui dire de s'y référer.

## 4 Programme pour faire ses courses

```
10 PRINT « DE QUOI AVEZ-VOUS  
BESOIN AU MAGASIN? »
```

```
20 INPUT XS
```

```
30 GOSUB 100 ]
```

```
40 PRINT « AUTRE CHOSE? »
```

```
50 INPUT MS
```

```
60 IF MS=« OUI » THEN GOTO 10
```

```
70 STOP ]
```

```
100 REM : ROUTINE COURSES ]
```

```
110 PRINT « SORS, TOURNE À GAUCHE »
```

```
120 PRINT « ENCORE À GAUCHE,  
ENTRE DANS LA BOUTIQUE »
```

```
130 PRINT « ACHÈTE »; XS; « REVIENS »
```

```
140 RETURN ]
```

La ligne 30 envoie l'ordinateur à la première ligne de la routine.

STOP, à la fin du programme principal, empêche l'ordinateur d'enchaîner sur la routine.

Étiqueter une routine avec REM permet de savoir à quoi elle sert.

L'ordinateur est renvoyé à la ligne 40, juste après GOSUB.

Si vous oubliez la ligne RETURN, c'est un bug.



En BASIC, pour dire à l'ordinateur de se rendre à un sous-programme, on utilise les instructions GOSUB et RETURN. GOSUB doit être suivi du numéro de la première ligne d'instructions du sous-programme. RETURN, placé à la fin de la routine, n'a pas besoin d'être suivi

d'un numéro : il renvoie automatiquement à la ligne d'instructions située en dessous de celle qui a fait quitter le programme principal. On peut envoyer l'ordinateur à une routine autant de fois que l'on veut et d'où l'on veut dans le programme.

## Des programmes avec gosub

On utilise des sous-programmes chaque fois que l'on veut exécuter plusieurs fois une même tâche. Voici quelques programmes recourant à des routines.

### Programme numérique

```
50 INPUT A
60 INPUT B
70 GOSUB 250
80 PRINT « A DIVISÉ PAR B = »; A/B
90 GOTO 50
250 REM: SOUS-PROGRAMME D'ARRÊT
260 IF A=0 AND B=0 THEN STOP
70 RETURN
```

Ce sous-programme permet de sortir du programme principal. Si vous voulez arrêter de diviser, entrez 0 en INPUT aux lignes 50 et 60. On ne met pas de STOP dans ce programme, puisque la ligne 90 renvoie à la ligne 50.

### Programme de conversion

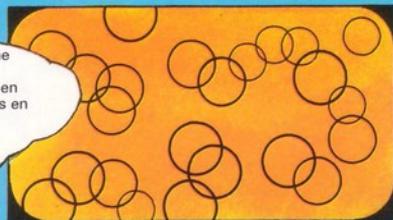
```
100 INPUT « DISTANCE »; M
110 INPUT « TEMPS »; T
120 GOSUB 200
130 PRINT « LA VITESSE MOYENNE ÉTAIT DE »;
140 PRINT M/T; "MPH ET"; K/M; "KMH"
150 STOP
200 REM: SOUS-PROGRAMME DE CONVERSION
DES MILES
210 LET KM=M*1609
220 RETURN
```

Voici un sous-programme dont le rôle est de convertir les miles en kilomètres. Il peut servir dans de nombreux programmes. Vérifiez seulement le nom des variables.

### Programme de tracé de cercles

```
1 Centre du cercle = X, Y
2 Rayon du cercle = R
3 Couleur = X
4 GOSUB 10
5 GOTO 1
10 REM: sous-programme pour tracer
des cercles
11 Tracé d'un cercle de centre X, Y,
de rayon R et de couleur
12 RETURN
```

Ce programme est écrit en français, non en BASIC. Il vous en donne l'idée générale.



On utilise les sous-programmes dans les applications graphiques : ils permettent de tracer des diagrammes à partir des nombres trouvés dans le programme

principal. Dans ce programme, on pourrait dessiner de nombreux cercles différents, simplement en entrant dans l'ordinateur des informations aux lignes 1 à 5.

## Programme de QUIZ

```
5 LET C=0
10 PRINT « QUAND CELA A-T-IL ÉTÉ INVENTÉ? »
20 READ CS, F
30 PRINT CS
40 INPUT A
50 LET C=C+1
60 IF C=3 THEN STOP
70 GOSUB 100
80 GOTO 10
100 REM: SOUS-PROGRAMME DE RÉPONSES
110 IF ABS(A-F)<10 THEN PRINT « OUI »
120 IF ABS(A-F)>10 THEN PRINT « NON »
130 PRINT « NOUVEL ESSAI »
140 RETURN
200 DATA TÉLÉPHONE, 1876, IMPRIMERIE, 1450, BICYCLETTE, 1791
```

Ce programme a recours à une routine pour contrôler les réponses. Les bonnes réponses sont mémorisées en F et les propositions du joueur en A. Aux lignes 100 et 110 de la routine,

A la ligne 20, l'ordinateur se rend à la ligne DATA et prend le premier article pour le placer en CS et F.

Le mot mémorisé en CS s'affiche ici. Le compteur C arrête le programme une fois qu'il s'est déroulé 3 fois, car il n'y a que 3 articles en CS et F.

Voilà le sous-programme.

A chaque nouveau passage du programme, les données en CS et F sont remplacées par les suivantes.

L'ordinateur compare A et F. ABS veut dire valeur absolue : l'appareil compare A et F sans tenir compte des signes négatifs. Si la différence est inférieure à 10, il affiche « OUI », autrement « NON ».

# Jouons avec les mots

La plupart des ordinateurs savent analyser les mots mémorisés en variable. Cette analyse les rend capables de diverses applications : ils peuvent détailler le contenu d'une variable et déterminer si un mot ou une lettre particulière s'y trouve, ce qui permet de vérifier les mots introduits par un utilisateur. Ils savent aussi réorganiser les lettres et les mots en un ordre différent et combiner plusieurs variables. Voici comment procéder en BASIC.

1

```
10 A$ = « JE SUIS UN IDIOT »
20 B$ = « SEUL UN FOU PEUT
PENSER QUE »
30 C$ = B$ + " " + A$
RUN
SEUL UN FOU PEUT PENSER QUE JE SUIS
UN IDIOT
```

Sur la plupart des ordinateurs, mais pas le SINCLAIR ZX81, l'instruction

LET n'est pas indispensable.

2

```
PRINT LEFT$(B$,4)
SEUL
```

```
PRINT LEFT$(B$,4) + " " + A$
SEUL JE SUIS UN IDIOT
```

Vous pouvez ajouter le contenu de deux variables. Il faut ménager un espace entre guillemets pour laisser un intervalle entre les mots

Vous pouvez aussi rapprocher des éléments de différentes variables. LEFT\$(B\$,4) signifie que vous prenez les quatre premières lettres à gauche de B\$.

3

```
PRINT RIGHT$(A$,5)
IDIOT
```

Pour dire à l'ordinateur de prendre des lettres sur la droite, utilisez RIGHT\$ suivi du nom de la chaîne et du nombre de lettres que vous demandez.

4

```
PRINT MID$(B$,6,6)
UN FOU
```

Voilà comment indiquer à l'ordinateur d'afficher les lettres du milieu. Le premier nombre lui indique où commencer, le second combien de lettres prendre.

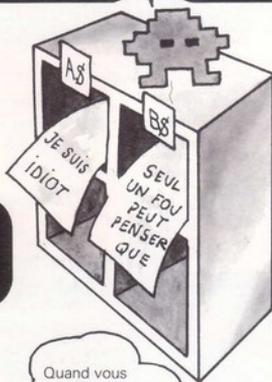
5

```
10 K$ = "DING-DONG"
20 PRINT LEN(K$)
RUN
10
```

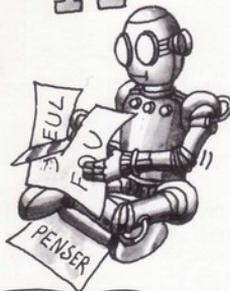
Vous pouvez aussi déterminer la longueur d'une chaîne : nombre de lettres, d'espaces, de symboles. Pour cela, on utilise la fonction LEN, abréviation de "length" (longueur).



Si A\$ = « LE LIVRE DE L'ORDINATEUR »  
que sera LEFT\$(A\$,8)?  
RIGHT\$(A\$,10)?  
MID\$(A\$,10,2)?



Quand vous comptez les lettres, pensez aussi à compter les espaces et la ponctuation.



Pour ceux qui utilisent un SINCLAIR

```
PRINT A$(12 TO 16)
IDIOT
PRINT B$(13 TO 16)
PEUT
```

L'instruction indique à l'ordinateur de prendre les lettres de 9 à 13.



Le SINCLAIR n'utilise pas les fonctions LEFT\$, RIGHT\$ ET MID\$, mais on peut lui demander d'afficher certaines lettres.

## Codons des messages

Ce programme code les mots automatiquement. Ce sont des programmes similaires, mais bien plus compliqués, qu'utilisent les services secrets pour écrire et décoder des messages.

Pour comprendre le fonctionnement de ce programme, écrivez un message sur une feuille de papier, entrez le programme sur l'ordinateur. Vous vous rendez compte, au fur et à mesure, des transformations apportées à votre message initial.

```
5 LET CS=""
7 LET DS=""
```

Réserve des emplacements pour des variables alphanumériques.

```
10 PRINT « ENTREZ UN COURT MESSAGE »
20 INPUT MS
30 PRINT « MAINTENANT ENTREZ UN NOMBRE
   SECRET COMPRIS ENTRE 2 ET »; LEN(MS)-1
40 INPUT N
```

Cela correspond à la longueur du message moins 1. N, votre nombre secret détermine le nombre de lettres prises sur la droite.

```
50 LET AS=RIGHT$(MS,N)
60 LET BS=LEFT$(MS,LEN(MS)-N)
70 LET MS=AS+BS
```

La longueur de MS-N nombre de lettres prises à gauche du message.

```
80 FOR I=1 TO LEN(MS) STEP 2
90 LET CS=CS+MID$(MS,I,1)
100 NEXT I
```

Remplace la lettre en MS avec AS+BS. De 1 au nombre total de lettres de votre message, de deux en deux, par exemple 1, 3, 5... A chaque fois que la ligne 90 repasse, l'ordinateur prend une lettre de MS dans la boucle I et la met en CS.

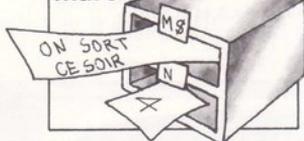
```
110 FOR J=2 TO LEN(MS) STEP 2
120 LET DS=DS+MID$(MS,J,1)
130 NEXT J
```

De 2 au nombre total de lettres de votre message, de deux en deux (2, 4, 6...). Même processus que pour la boucle I.

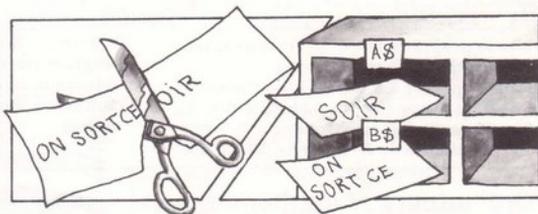
```
140 LET MS=CS+DS
150 PRINT « VOILÀ LE MESSAGE CODÉ »
160 PRINT MS
170 END
```

Réintroduit à nouveau les lettres en MS.

### Comment ça marche



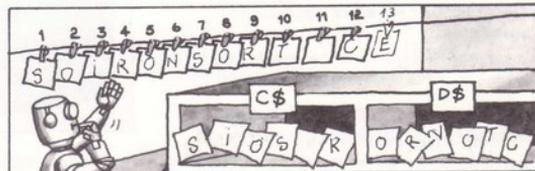
Imaginons que le message soit « On sort ce soir » et le chiffre secret 4. Ces données sont mémorisées en MS et N.



Aux lignes 50 et 60, l'ordinateur utilise le chiffre secret pour segmenter le message. A la ligne 50, il prend 4 lettres à la droite du message et les met en AS. A la ligne 60, il met le reste du message en BS.



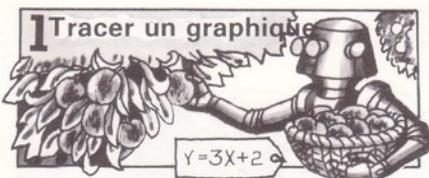
A la ligne 70, il réunit AS et BS. Ainsi les dernières lettres du message se retrouvent-elles à l'avant.



Chaque fois que la boucle I se répète, elle place une lettre différente en CS (comme S, I, O). De même, à chaque fois que la boucle J se répète, elle place une lettre en DS (comme O, R, N). Puis l'ordinateur réunit CS et DS pour établir le message codé.

# Graphiques et symboles

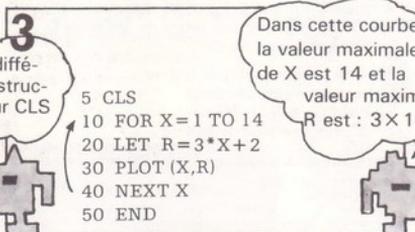
Vous pouvez programmer un ordinateur pour qu'il donne les informations de différentes façons : mots, nombres, dessins, graphiques. On facilite ainsi la compréhension de sujets complexes en les illustrant par des graphiques, des images ou des symboles.



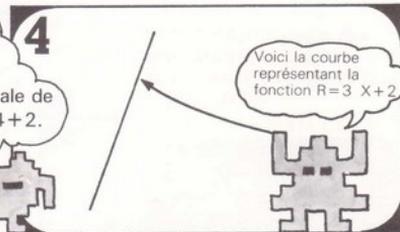
Prenons un pêcher dont le rendement en fruits croît chaque année en fonction de son âge. On peut poser ce fait en équation, par exemple  $R=3X+2$  (R représente le rendement et X l'âge). Telle quelle, la notion est difficile à saisir. Un graphique sera le bienvenu.



Avec un ordinateur, il est très facile de tracer la courbe de déplacement de R en fonction de X. Pour tracer cette courbe, déterminer la valeur de R pour chaque valeur de X. Vous obtiendrez ce résultat en programmant l'instruction `LET R=3*X+2`.



Voilà le programme qui permet de tracer cette courbe. La boucle donne successivement à X toutes les valeurs comprises entre 1 et 14; à chaque fois, la ligne 20 en déduit la valeur correspondante de R. La ligne 30 affiche



le point X, R à l'écran. Dans ce type de programme, on doit vérifier que les valeurs maximales de X et R sont contenues dans l'écran. Autrement, c'est un bug.

## Ordinateurs et mathématiques

Dans les calculs multiples, comme  $3*X+2$ , l'ordinateur effectue toujours les multiplications et les divisions avant d'opérer les additions et les soustractions. Il donnera donc le même résultat pour les deux opérations suivantes :

```

PRINT 4*6+8      PRINT 8+4*6
32                32
    
```

Si vous voulez modifier l'ordre des opérations, utilisez des parenthèses :

```

PRINT (8+4)*6
72
    
```

L'ordinateur a d'abord additionné 8 et 4, puis il a multiplié cette somme par 6.

## Casse-tête

PENSEZ À UN NOMBRE  
MULTIPLIEZ-LE PAR 2, AJOUTEZ 4  
DIVISEZ PAR 2, AJOUTEZ 7  
MULTIPLIEZ PAR 8, ENLEVEZ 12  
DIVISEZ PAR 4, ENLEVEZ 11  
DITES-MOI COMBIEN IL RESTE.  
LE NOMBRE AUQUEL VOUS AVIEZ  
PENSÉ ÉTAIT :

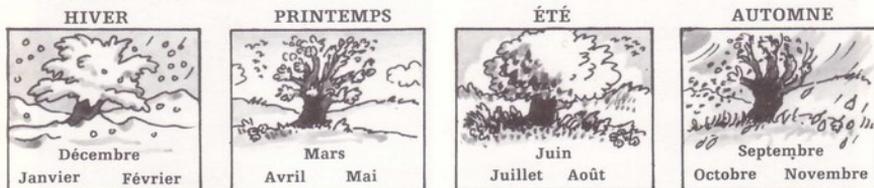
Sauriez-vous écrire un programme qui traite ce jeu bien connu? (Pour retrouver le premier chiffre, vous enlevez 4 au « reste », puis vous divisez par 2.)

## Programme d'anniversaire

Voici une autre méthode pour afficher des informations à l'écran. On utilise des signes pour comparer le nombre de personnes nées en chaque saison de l'année. Un programme de ce type peut servir, par exemple, à comparer les apparitions d'une espèce d'oiseaux selon les saisons, ou le nombre de buts marqués par une équipe de football. Avant de se lancer dans la rédaction d'une liste aussi longue, il est bon d'établir un plan de programme.

### Plan du programme

But : comparer le nombre des personnes dont l'anniversaire tombe en hiver, printemps, été, automne.



1. Entrer les données dans l'ordinateur (saison de la naissance) pour un échantillon de 20 personnes.
2. Mémoriser les données.
3. Afficher les données à l'écran.

### Le programme

```
5 LET A=0
6 LET B=0
7 LET C=0
8 LET D=0
10 FOR I=1 TO 20
20 PRINT « LA PERSONNE N° »; I; « EST NÉE EN »
30 PRINT « HIVER, PRINTEMPS, ÉTÉ, AUTOMNE »
40 PRINT « TAPÉZ H, P, E, OU A »
50 INPUT BS
60 IF BS=« H » THEN LET A=A+1
70 IF BS=« P » THEN LET B=B+1
80 IF BS=« E » THEN LET C=C+1
90 IF BS=« A » THEN LET D=D+1
100 NEXT I
110 PRINT « TOTAL HIVER »;
115 LET N=A
120 GOSUB 200
130 PRINT « TOTAL PRINTEMPS »;
135 LET N=B
140 GOSUB 200
150 PRINT « TOTAL ÉTÉ »;
155 LET N=C
160 GOSUB 200
170 PRINT « TOTAL AUTOMNE »;
175 LET N=D
180 GOSUB 200
190 STOP
200 REM : SOUS-PROGRAMME
D'AFFICHAGE DES ÉTOILES
210 IF N=0 THEN GOTO 250
220 FOR I=1 TO N
230 PRINT « * »;
240 NEXT I
250 PRINT
260 RETURN
```

Variables vierges prêtes à totaliser les totaux de chaque saison.

Cette boucle oblige l'ordinateur à poser la question pour chaque personne de l'échantillon.

De la ligne 60 à la ligne 100, l'ordinateur compare la réponse en BS avec les données et ajoute 1 à la variable correspondante.

Renvoie l'ordinateur à la ligne 10.

Le sous-programme permet d'afficher un nombre d'étoiles correspondant à la valeur de chacune des variables.

En transposant le total en N, le programme peut se servir de la même routine pour toutes les saisons.

Cette ligne oblige l'ordinateur à afficher les étoiles sur une même ligne.

La ligne 210 est utile au cas où personne ne serait né en une quelconque saison.

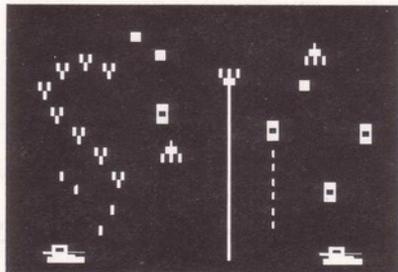
Le programme principal utilise N comme total pour A, B, C ou D. L'ordinateur reproduit la ligne 230 « N » fois.

### Exemple de fonctionnement

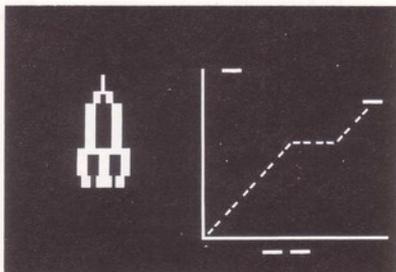
```
RUN
TOTAL HIVER*****
TOTAL PRINTEMPS***
TOTAL ÉTÉ*****
TOTAL AUTOMNE****
```

# Encore des graphiques

Voici deux pages qui montrent comment utiliser PLOT et UNPLOT pour créer des dessins sur l'écran. Ces graphiques animés servent dans les programmes de jeu ou pour illustrer des trajectoires, des principes de gravité ou de balistique.



Dans les jeux vidéo à usage personnel ou public, les dessins sont contrôlés par un petit ordinateur programmé uniquement pour ces jeux. Les instructions sont délivrées en langage machine et non en BASIC.



Un micro-ordinateur multi-tâches programmé en BASIC ne peut réaliser que des dessins plus lents et plus simples. Il n'est pas capable en effet de traiter assez rapidement toutes les instructions concernant l'écran.

## 1 Plot/unplot

```
10 LET X=1
20 LET Y=1
30 PLOT (X,Y)
40 UNPLOT (X,Y)
50 LET X=X+1
60 LET Y=Y+1
70 GOTO 30
```

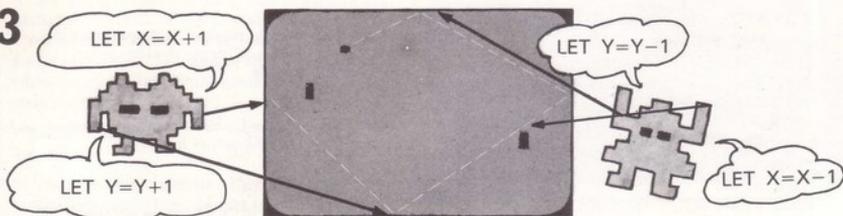
Certains ordinateurs doivent être mis en mode graphique.



Par ce court programme, un point lumineux se déplace à travers l'écran. N'oubliez pas que les instructions PLOT et UNPLOT varient selon les ordinateurs.

Lorsque le point atteint le bord de l'écran, le programme s'arrête : les valeurs de X et Y se trouvent alors au-delà des limites de l'écran de l'ordinateur.

## 3

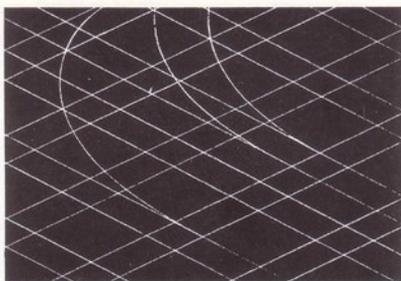


Les jeux de balles vidéo utilisent des programmes du type de celui que nous venons de voir. Il existe des règles simples à appliquer pour que le jeu se poursuive lorsque la balle touche les bords de l'écran.

Lorsque la balle touche le haut de l'écran, la valeur qui aurait dû être ajoutée à Y se trouve au contraire retranchée. De même, lorsqu'elle atteint le bord droit de l'écran, la valeur de X est minorée.

## Tracé de ligne

Ce programme permet de tracer une ligne continue sur l'écran. Lorsqu'elle se heurte à un bord de l'écran, le programme la renvoie dans une autre direction. Comme on n'utilise pas l'instruction UNPLOT, on obtient un dessin. Vous voyez sur la figure de droite ce qui se passe quand vous lancez le programme. A la ligne 100, l'ordinateur est programmé pour tracer jusqu'à 10 000 pixels. Vous pouvez changer ce chiffre pour raccourcir le programme, ou l'arrêter en cours d'exécution lorsque le dessin vous plaît.



```
10 REM: PASSER EN MODE GRAPHIQUE SI NÉCESSAIRE
20 PRINT "DÉFINITION DE L'ÉCRAN,
LARGEUR";
30 INPUT L
40 PRINT « HAUTEUR »;
50 INPUT H
```

Les lignes 20 à 50 vous demandent le nombre de points de votre écran. Les points-virgules permettent à la réponse de s'afficher à la même ligne que la question.

```
55 CLS
60 LET X=L/2
70 LET Y=H/2
80 LET S=1
90 LET T=1
```

Ainsi X et Y démarrent du centre de l'écran. S et T sont les valeurs qui seront ajoutées à X et Y pour les faire se déplacer.

```
100 FOR I=1 TO 10000
110 LET S=S+(INT(RND(1)*10+1)-5)/50
120 LET X=X+S
130 LET Y=Y+T
```

La boucle des lignes 100 à 190 se répète 10 000 fois. A chaque passage, X et Y changent légèrement de valeur.

```
140 IF X<5 THEN LET S=-S
150 IF X>L-5 THEN LET S=-S
160 IF Y<5 THEN LET T=-T
170 IF Y>H-5 THEN LET T=-T
```

Cette instruction donne une très petite valeur à ajouter à X.

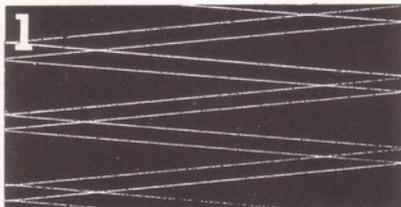
Ces lignes contrôlent les bords de l'écran et inversent S et T lorsque X et Y arrivent à moins de cinq points du bord.

```
180 GOSUB 300
190 NEXT I
200 STOP
300 REM: PLOT LINE
310 PLOT (X,Y)
320 RETURN
```

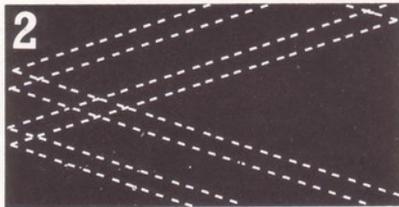
Envoie l'ordinateur au sous-programme qui affiche le point.

Trace le pixel correspondant aux valeurs données de X et Y.

## Essais



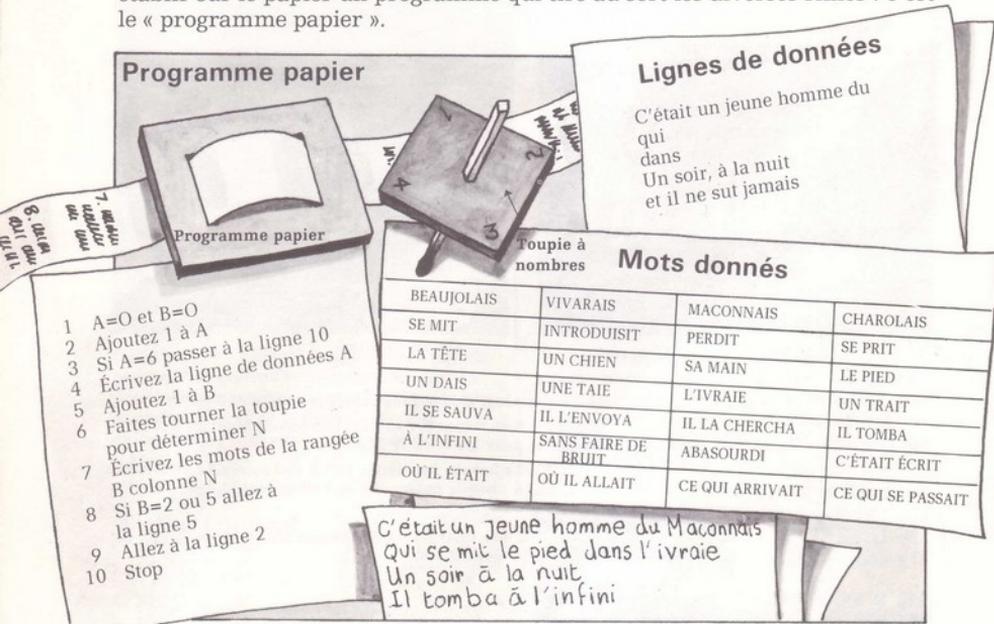
La ligne 110 ajoute un très petit nombre aléatoire à X et à chaque passage de boucle, ce qui fait osciller la ligne à travers l'écran. Si vous avez un ordinateur, amusez-vous à effacer cette instruction : les lignes deviendront



parallèles. Essayez de changer les nombres aux lignes 80 et 90, disons 5 ou 10 (ou plus sur un ordinateur à très haute résolution). Vous obtiendrez des lignes pointillées.

# Programmmons des poèmes drôles

Les quelques pages suivantes vous indiquent comment écrire un programme capable de composer des dizaines de poésies. Vous en aviez déjà trouvé une version dans l'*Introduction aux ordinateurs*. Vous y appreniez comment établir sur le papier un programme qui tire au sort les diverses rimes : c'est le « programme papier ».



- 1 A=O et B=O
- 2 Ajoutez 1 à A
- 3 Si A=6 passer à la ligne 10
- 4 Écrivez la ligne de données A
- 5 Ajoutez 1 à B
- 6 Faites tourner la toupie pour déterminer N
- 7 Écrivez les mots de la rangée B colonne N
- 8 Si B=2 ou 5 allez à la ligne 5
- 9 Allez à la ligne 2
- 10 Stop

BEAUJOLAIS	VIVARAIS	MACONNAIS	CHAROLAIS
SE MIT	INTRODUISIT	PERDIT	SE PRIT
LA TÊTE	UN CHIEN	SA MAIN	LE PIED
UN DAIS	UNE TAIE	L'IVRAIE	UN TRAIT
IL SE SAUVA	IL L'ENVOYA	IL LA CHERCHA	IL TOMBA
À L'INFINI	SANS FAIRE DE BRUIT	ABASOURDI	C'ÉTAIT ÉCRIT
OÙ IL ÉTAIT	OÙ IL ALLAIT	CE QUI ARRIVAIT	CE QUI SE PASSAIT

Voilà le programme papier. Il ressemble à du BASIC, mais ne fonctionnerait pas sur un véritable ordinateur. Mots et phrases du poème sont conservés sur des

morceaux de papier et le programme indique lequel sélectionner. La toupie donne un nombre aléatoire entre 1 et 4.

## 1 Traduction du programme en BASIC

```

10 LET A=O
20 LET B=O
30 LET A=A+1
40 IF A=6 THEN STOP
50 Écrire la ligne de données A
60 LET B=B+1
70 LET N=INT(RND(1)*4+1)
80 Écrire les données
   de la rangée B colonne N
90 IF B=2 THEN GOTO 60
100 IF B=5 THEN GOTO 60
110 GOTO 30
120 END
    
```

Ce programme ne peut pas encore fonctionner sur un ordinateur.



- Ces lignes libèrent des variables.
- Les lignes 30 et 40 tiennent compte du nombre de données que l'ordinateur a sélectionnées.
- Les lignes 50 et 80 ne sont pas en BASIC.
- La ligne 60 compte le nombre de mots en données.
- Détermine un nombre aléatoire entre 1 et 4.
- Les lignes 90 et 100 envoient l'ordinateur sélectionner une autre ligne de données.

La plus grande partie du programme se traduit sans difficulté en BASIC. Toutefois les lignes 50 et 80 posent problème. Il faut donner à l'ordinateur une méthode

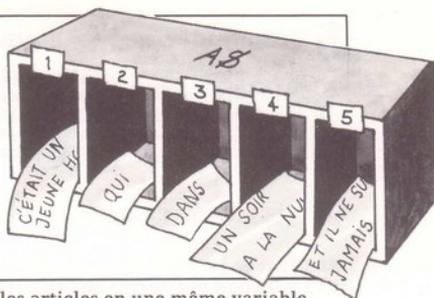
de stockage et de tirage au sort des lignes et des mots qui lui permette de les sélectionner au bon moment dans le poème.

## 2 Entrer les données

50 READ AS

180 DATA C'ÉTAIT UN JEUNE HOMME DU,  
QUI, DANS

190 DATA UN SOIR À LA NUIT,  
ET IL NE SUT JAMAIS



Pour entrer les données, on peut utiliser les instructions READ et DATA. Chaque fois que l'ordinateur rencontre l'instruction READ, il prend un nouvel article à la ligne DATA et le mémorise en variable. Il est possible de mémoriser

tous les articles en une même variable AS. Toute variable contenant plus d'un article s'appelle un tableau et chaque article est repéré par un numéro, par exemple READ AS(3) donne « dans »\*.

## 3

	1	2	B\$	3	4
1	BEUJOLAIS	VIVARAIS	MACONNAIS	CHAROLAIS	
2	SE MIT	INTRODUISIT	PERDIT	SE PRIT	
3	LA TÊTE	UN CHIEN	SA MAIN	LE PIED	
4	UN DAIS	UNE TAIE	L'IVRAIE	UN TRAIT	
5	IL SE SAUVA	IL L'ENVOYA	IL LA CHERCHA	IL TOMBA	
6	A L'INFINI	SANS FAIRE DE BRUIT	ABASOURDI	C'ÉTAIT ÉCRIT	
7	OÙ IL ÉTAIT	OÙ IL ALLAIT	CE QUI ARRIVA	CE QUI SE PASSAIT	

LIRE  
B\$(2,3)



LIRE  
B\$(7,1)



Une variable peut aussi regrouper plusieurs rangées de données. C'est alors un tableau à deux dimensions. Vous pouvez ainsi mémoriser tous les mots du poème. Repérez chaque article par le numéro de la rangée et le numéro de la

colonne dans lesquelles il se trouve. Ainsi READ B\$(4,2) donne « une taie » et READ B\$(6,3) « abasourdi ».

En utilisant une variable numérique, vous pouvez aussi stocker des nombres dans un tableau.

## 4 Mettons les données en variables

10 FOR I=1 TO 7 ]- I renvoie au n° de la rangée

20 FOR J=1 TO 4 ]- J renvoie au n° de la colonne

30 READ BS(I,J)

40 NEXT J

50 NEXT I

60 DATA BEUJOLAIS, VIVARAIS,  
MACONNAIS, CHAROLAIS

70 DATA SE MIT, INTRODUISIT, PERDIT, SE PRIT

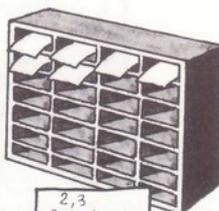
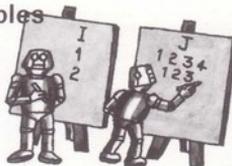
80 DATA LA TÊTE, UN CHIEN, SA MAIN, LE PIED

90 DATA UN DAIS, UNE TAIE, L'IVRAIE, UN TRAIT

100 DATA IL SE SAUVA, IL L'ENVOYA, IL LA CHERCHA, IL TOMBA

110 DATA À L'INFINI, SANS FAIRE DE BRUIT, ABASOURDI,  
C'ÉTAIT ÉCRIT

120 DATA OÙ IL ÉTAIT, OÙ IL ALLAIT, CE QUI ARRIVAIT,  
CE QUI SE PASSAIT



Pour que chaque article passe en variable, il faut modifier les nombres entre parenthèses après READ. Des boucles peuvent le faire : on utilise des

boucles imbriquées pour BS, la boucle J (numéro de colonne) se répétant 4 fois pour chaque passage de la boucle I (numéro de rangée).

\*Ce programme ne fonctionnera pas sur un SINCLAIR qui traite les variables de façon différente. Vous en saurez plus à la page suivante.

## 5 Créons des espaces pour les variables

```
5 DIM K$(5)
10 FOR I=1 TO 5
20 READ K$(I)
30 NEXT I
40 STOP
```

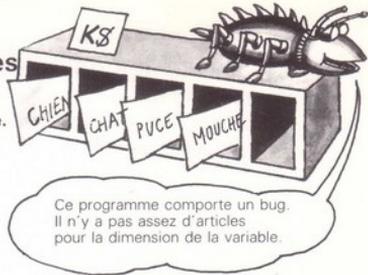
Dimension de la variable.  
Exemple : 5 articles dans une rangée.

Cette ligne inscrit la donnée en K\$ à chaque passage de la boucle.

```
60 DATA CHIEN, CHAT, PUCE, MOUCHE
```

En début de programme, on doit indiquer à l'ordinateur quelle sera la dimension de la variable. Pour cela, utilisez le mot DIM suivi du nom de la variable et du nombre d'articles qu'elle comporte. Exemple : DIM K\$(5). Dans un tableau à deux

dimensions, on indique à l'ordinateur le nombre de rangées et le nombre de colonnes de la variable : DIM CS(5,3). Il faut toujours que le nombre d'articles et le nombre indiqué coïncident, ou c'est un bug.



## 6 Affichons les données

```
200 LET A=0
210 LET B=0
220 LET A=A+1
230 IF A=6 THEN STOP
240 PRINT AS(A)
250 LET B=B+1
260 LET N=INT(RND(1)*4+1)
270 PRINT BS (B,N)
280 IF B=2 THEN GO TO 250
290 IF B=5 THEN GO TO 250
300 GO TO 220
310 END
```

A compte le nombre de fois où cette partie du programme est répétée.

B compte les rangées de mots et vérifie qu'on utilise bien la bonne rangée pour chaque ligne de données.

Les lignes 280 et 290 font afficher à l'ordinateur les mots sélectionnés dans une rangée avant de visualiser la ligne de données suivante.

Donne l'ordre à l'ordinateur d'aller afficher la ligne de données suivante.

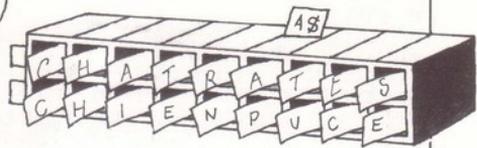
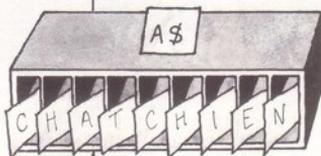
Ce programme est nécessaire pour que l'ordinateur affiche mots et données dans le bon ordre. Cette partie du programme se répète 5 fois. À chaque passage,

l'ordinateur affiche la ligne de données A et un mot de la rangée B. La sélection des mots est faite par un nombre aléatoire N.

### L'ordinateur SINCLAIR et les variables

Ce programme ne fonctionne pas tel quel sur les ordinateurs SINCLAIR, car ils traitent les chaînes différemment.

Il faut donner à l'ordinateur le nombre de rangées et le nombre de caractères. Par exemple AS(2,6 TO 9) c'est « PUCE ».



Pour indiquer à un SINCLAIR de sélectionner une donnée en particulier, il faut lui fournir la position du premier et du dernier caractère du mot retenu. C'est en somme la même méthode que pour LEFT\$ et RIGHT\$ page 80. Pour les tableaux à deux dimensions, il

En début de programme, vous devez indiquer à l'ordinateur le nombre de rangées et le nombre de caractères par rangée. DIM AS(2,9) explique qu'il y a deux rangées de neuf caractères chacune. Toutes les rangées du tableau doivent comporter le même nombre de caractères.

## Programme complet de poésie

On peut maintenant rassembler tous ces éléments pour reconstituer enfin le programme de poésie in extenso. La première partie enregistre les données (lignes 10 à 90), la seconde affiche le poème (lignes 200 à 310). À chaque fois que vous lancez le programme, vous obtenez une nouvelle version du poème, puisque le nombre aléatoire N fait sélectionner des mots différents à l'ordinateur.

```
10 DIM AS(5)
20 DIM BS (7,4)
30 FOR I=1 TO 7
40 FOR J=1 TO 4
50 READ BS(I,J)
60 NEXT J
70 NEXT I
80 DATA BEAUJOLAIS, VIVARAIS, MACONNAIS, CHAROLAIS
90 DATA SE MIT, INTRODUISIT, PERDIT, SE PRIT
100 DATA LA TÊTE, UN CHIEN, SA MAIN, LE PIED
110 DATA UN DAIS, UNE TAIE, L'IVRAIE, UN TRAIT
120 DATA IL SE SAUVA, IL L'ENVOYA, IL LA CHERCHA, IL TOMBA
130 DATA À L'INFINI, SANS FAIRE DE BRUIT, ABASOURDI, C'ÉTAIT ÉCRIT
140 DATA OÙ IL ÉTAIT, OÙ IL ALLAIT, CE QUI ARRIVAIT, CE QUI SE PASSAIT
150 FOR I=1 TO 5
160 READ AS(I)
170 NEXT I
180 DATA C'ÉTAIT UN JEUNE HOMME DU, QUI, DANS
190 DATA UN SOIR À LA NUIT, IL NE SUT JAMAIS
200 LET A=0
210 LET B=0
220 LET A=A+1
230 IF A=6 THEN 310
240 PRINT AS(A)
250 LET B=B+1
260 LET N=(RND(1)*4+1)
270 PRINT BS(B,N)
280 IF B=2 THEN GOTO 250
290 IF B=5 THEN GOTO 250
300 GOTO 220
310 END
```

Les lignes 10 et 20 réservent l'espace pour les variables.

Boucles imbriquées pour mettre les données en BS.

Les lignes 80 à 140 contiennent tous les mots à entrer en BS.

Boucle servant à mettre les données en AS.

Les lignes 180 et 190 contiennent toutes les données à entrer en AS.

Cette instruction fait s'afficher la donnée mémorisée en AS, rangée A.

Cette instruction fait s'afficher le mot mémorisé en BS, rangée B, colonne N.

Le programme s'arrête à la ligne 230 quand A=6. Mais certains ordinateurs réclament tout de même un END.

## Exemples

```
C'ÉTAIT UN JEUNE HOMME DU
VIVARAIS
QUI
SE MIT
LA TÊTE
DANS
UN DAIS
UN SOIR À LA NUIT
IL SE SAUVA
SANS FAIRE DE BRUIT
ET IL NE SUT JAMAIS
CE QUI ARRIVAIT
```

```
C'ÉTAIT UN JEUNE HOMME DU
CHAROLAIS
QUI
PERDIT
LE PIED
DANS
UNE TAIE
UN SOIR À LA NUIT
IL TOMBA
C'ÉTAIT ÉCRIT
ET IL NE SUT JAMAIS
OÙ IL ALLAIT
```

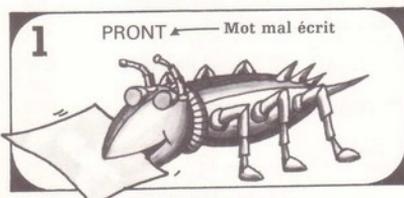
Voilà deux des 16 384 versions possibles de la poésie ! Si, en lançant le programme, vous obtenez toujours les mêmes données, regardez dans votre manuel ce que vous devez faire pour que votre ordinateur tire des nombres

différents. Sur certains modèles, les nombres aléatoires sont les mêmes à chaque fois qu'on met en marche l'ordinateur. En utilisant la fonction GOTO, par exemple, vous éliminerez cet inconvénient.

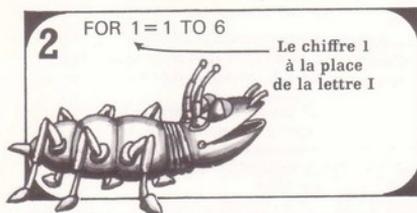
# Quelques trucs de programmation

Dans ces deux pages, vous trouverez quelques trucs pour vous aider à écrire vos propres programmes, ainsi qu'une liste des bugs les plus courants. On vous explique le pourquoi des bugs en partant des plus évidents jusqu'aux plus retors.

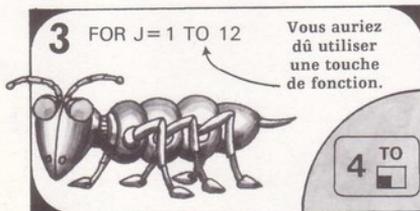
## Trouver le bug



Cherchez d'abord les fautes de frappe dans les instructions de BASIC. Si un des termes est mal écrit, l'ordinateur ne le reconnaîtra pas.



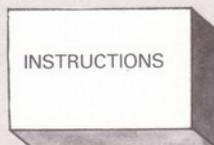
Vérifiez les O et les 0, les 1 et les I pour être certain qu'il n'y a pas eu de confusion.



Si vous avez un SINCLAIR, vérifiez que vous n'avez pas tapé lettre à lettre une instruction qui s'effectue en appuyant sur une touche spéciale.

## Programmer

Quand on écrit des programmes, il faut se rappeler qu'un ordinateur accomplit trois tâches principales : instructions simples, répétitions et prises de décision. Ce sont les pierres angulaires des programmes.



```
LET A=3  
LET N=N+1  
PRINT A/T  
PLOT (X,Y)
```



```
FOR J=1 TO 6  
20 LET A=1  
30 IFA < 10 THEN  
GOTO 100
```



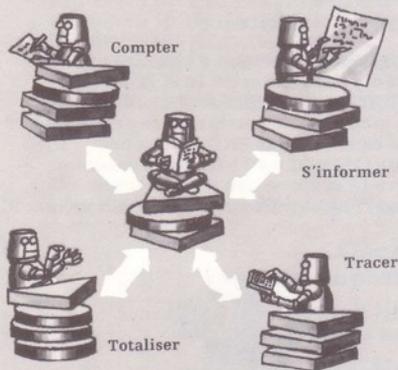
```
IF X=Y THEN STOP  
IF K$=" BONJOUR "  
THEN PRINT A
```

Ce livre vous a expliqué toutes les instructions essentielles du BASIC, qui permettent d'exécuter ces fonctions. Lorsque vous écrivez un programme, déterminez ce dont l'ordinateur a besoin à chaque étape, puis appliquez les instructions correspondantes.



Vérifiez que vous n'avez pas oublié guillemets, virgules ou autres. Contrôlez soigneusement les lignes de programme très chargées en signes.

On peut écrire un même programme de plusieurs façons; certaines sont plus claires et rapides que d'autres. Quand vous devez établir un long programme, il est bon de le diviser en plusieurs parties et d'utiliser des sous-programmes pour chaque type de fonction. Le corps central peut parfois se résumer à quelques instructions, décisions et répétitions qui commandent les sous-programmes.



Morceler ainsi un programme le rend beaucoup plus facile à tester. On peut rechercher les erreurs, portion par portion, sans avoir à lancer le programme en entier. N'oubliez pas d'étiqueter chaque partie avec une ligne REM, - pour vous y retrouver.



Vérifiez que vous utilisez la bonne instruction pour RND, PLOT, et CLS. N'oubliez pas de passer en mode graphique.

## Message d'erreurs

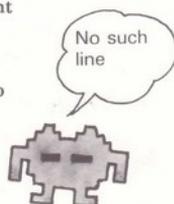
Tous les ordinateurs affichent un message lorsqu'il y a un bug. Ces messages sont expliqués dans votre notice.

Voici les messages les plus courants :



◀ Il n'y a pas assez de rubriques en DATA. Peut-être avez-vous oublié une virgule entre deux articles que l'ordinateur a alors pris pour un seul.

▶ GO TO ou GOSUB envoient l'ordinateur à une ligne qui n'existe pas. Vous avez effacé cette ligne en donnant le même numéro à une autre ligne ou, tout simplement, vous vous êtes trompé en tapant le nombre.



◀ Vous rencontrerez ce message sur le BBC ou le SINCLAIR. Il indique généralement que vous n'avez pas initialisé une variable avant de l'utiliser avec une ligne comme LET C=0 ou Let C=« ». ».

▶ Dans votre boucle, la ligne NEXT manque. Ou vous vous êtes trompé dans le nom de la variable, ou vous avez écrit le chiffre 1 à la place de la lettre I.



## Un dernier mot

Certains bugs sont très difficiles à détecter. Pourtant, si le programme ne tourne pas, c'est qu'il y a une erreur quelque part... Si vous ne la trouvez pas, tapez à nouveau les lignes particulièrement délicates ou compliquées. Peut-être que le programme se mettra à fonctionner sans que vous puissiez vraiment déterminer d'où provenait l'erreur.

# Réponses aux casse-tête

## Page 63

### Un message à votre nom

```
10 PRINT « QUEL EST TON NOM »
20 INPUT N$
30 PRINT « BONJOUR »
40 PRINT N$
50 PRINT « COMMENT VAS-TU »
```

## Page 65

### 1. Programme d'addition

```
10 LET A=9
20 LET B=7
30 PRINT A*B
40 PRINT A/B
50 LET A=A+1
60 LET B=B+3
70 PRINT A*B, A/B
80 END
```

La virgule  
laisse un espace.

### 2. Table de multiplication

```
30 PRINT A; « FOIS »; B; « FONT »; A*B
40 PRINT A; « DIVISÉ PAR »; B; « ÉGALE »; A/B
```

Espaces

### 3. Sur une même ligne

```
10 PRINT « QUEL EST TON NOM »
20 INPUT N$
30 PRINT « BONJOUR »; N$;
  « COMMENT VAS-TU »
```

## Page 66

### Opérations

```
10 PRINT « COMBIEN FONT 7 FOIS 7 »
20 INPUT A
30 IF A=49 THEN PRINT « JUSTE »
40 IF A<>49 THEN PRINT « NON »; 7*7
```

N'oubliez pas  
les points-virgules.

## Page 67

### Quel est mon âge?

Il faut transformer la ligne  
30 et ajouter la ligne 35 :

```
30 IF G<14 THEN PRINT
  « PLUS VIEUX QUE ÇA »
35 IF G>14 THEN PRINT
  « PLUS JEUNE QUE ÇA »
```

## Page 71

### Contrôleur de boucle

```
5 LET C=0
45 LET C=C+1
50 IF C<6 THEN GOTO 10
```

## L'initiale

Voici un exemple pour la lettre L.

```
10 LET X=15
20 LET Y=30
30 PLOT (X,Y)
40 LET Y=Y-1
50 IF Y>5 THEN GOTO 30
60 LET X=X+1
70 PLOT (X,Y)
80 IF X<45 THEN GOTO 60
90 END
```

## Page 72

### Nombre aléatoire

La formule permettant d'obtenir un nombre aléatoire entre 10 et 20 est :  $INT(RND(1)*11+9)$ . Pour les ordinateurs qui acceptent un nombre entre parenthèses tout de suite après RND, cela donne  $RND(11)+9$ . Il y a 11 nombres possibles entre 10 et 20; on doit donc tirer un nombre aléatoire entre 1 et 11, puis ajouter 9.

## Page 73

### Attaque spatiale

Voici les lignes qu'il faut ajouter pour obtenir ce résultat :

```
15 LET S=0
75 IF X=A*B THEN LET S=S+1
95 PRINT « VOUS AVEZ ABATTU »
  ; S; « EXTRA-TERRESTRES »
```

## Page 75

### 1. Table de 8

```
10 PRINT « LA TABLE DE MULTIPLICATION
  PAR HUIT »
20 FOR J=1 TO 12
30 PRINT J; « x8= »; J*8
40 NEXT J
```

### 2. Table de N

```
10 INPUT « TAPER UN NOMBRE »; N
20 PRINT « VOICI LA TABLE
  DE MULTIPLICATION PAR »; N
30 FOR I=1 TO 12
40 PRINT I; « FOIS »; N; « FONT »; I*N
50 NEXT I
60 INPUT « UN AUTRE NOMBRE
  (O OU N) »; M$
70 IF M$=« O » THEN GOTO 10
```

Sur le SINCLAIR ZX81, il faut séparer les PRINT et les INPUT.

## Page 80 Le livre de l'ordinateur

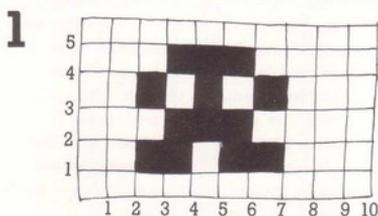
LEFT\$(A\$,8): LE LIVRE  
RIGHT\$(A\$,10): ORDINATEUR  
MID\$(A\$,10,2): DE

## Page 82 Le nombre truqué

```
10 PRINT « PENSEZ À UN NOMBRE »
20 PRINT « MULTIPLIEZ-LE PAR 2, AJOUTEZ 4 »
30 PRINT « DIVISEZ PAR 2, AJOUTEZ 7 »
40 PRINT « MULTIPLIEZ PAR 8, ENLEVEZ 12 »
50 PRINT « DIVISEZ PAR 4, ENLEVEZ 11 »
60 PRINT « DITES-MOI COMBIEN IL RESTE »
70 INPUT N
80 PRINT « LE NOMBRE AUQUEL VOUS AVIEZ PENSÉ ÉTAIT »; (N-4)/2
```

Il ne faut pas oublier les parenthèses pour que l'ordinateur exécute d'abord la soustraction.

## L'envahisseur extra-terrestre



Dessinez une forme simple d'envahisseur extra-terrestre sur une feuille de papier quadrillé.

**2**

4,5; 5,5; 6,5  
3,4; 5,4; 7,4  
4,3; 5,3; 6,3  
3,2; 4,2; 6,2; 7,2

Puis relevez toutes les coordonnées des cases qui constituent la forme de l'envahisseur.

**3**

```
5 CLS
50 INPUT « DÉFINITION DE L'ÉCRAN, LARGEUR »; L
60 INPUT « HAUTEUR »; H
65 CLS
70 FOR I=0 TO H STEP H/6
80 FOR J=0 TO L STEP L/6
130 NEXT J
140 NEXT I
150 END
```

Si vous voulez que l'envahisseur apparaisse plus souvent à l'écran, remplacez 6 par un chiffre plus fort. (Si vous choisissez un nombre trop fort, c'est un bug.)

Insérez ici les lignes PLOT:  
90 PLOT (J+3,I+2)  
92 PLOT (J+4,I+2)

pour réaliser les deux cases en bas à gauche de la forme dessinée ici. Il faut une ligne de programme pour chaque case.

Recopiez le programme de répétition de dessins en remplaçant les lignes 10 à 40 par celles ci-dessus (vous n'avez pas besoin des lignes aléatoires qu'elles génèrent).  
Il faut introduire vos instructions de

tracé entre les lignes 80 et 140 (vous pouvez établir une nouvelle numérotation si besoin est). Pour chaque coordonnée, vous devez ajouter J au premier chiffre et I au second pour que le dessin se répète à l'écran.

# Comment charger et sauvegarder des programmes

Charger et sauvegarder des programmes n'est pas si simple qu'il y paraît, et il peut vous arriver de perdre des heures à essayer vainement de charger ce que vous venez de sauvegarder. Voici un panorama des difficultés les plus fréquentes et quelques conseils pour les résoudre au mieux.

## Chargement des programmes

Pour réussir le chargement d'un programme conservé sur cassette, il faut avant tout sélectionner sur votre magnétophone le volume sonore approprié. Si le message émis est trop fort ou trop faible, l'ordinateur ne parviendra pas à déchiffrer les signaux.

Vous trouverez le bon niveau sonore par tâtonnements... Si vous disposez d'un réglage de tonalité, mettez-le au plus aigu et réglez le volume de façon moyenne. Suivez alors les indications de chargement de programme propres à votre ordinateur et appuyez sur la touche « Play » du magnétophone. Généralement l'ordinateur vous signale le moment où le programme est chargé. Mais attention, même si l'opération semble avoir réussi, le programme chargé pourra parfois se révéler plein de bugs et d'erreurs, à cause d'une déformation du message émis par le magnétophone.

Si vous ne parvenez pas à charger un programme, ramenez la cassette à zéro, tapez NEW sur l'ordinateur pour vider la mémoire, puis essayez à nouveau avec une autre intensité sonore. Après plusieurs essais, vous devriez pouvoir déterminer quel est le volume convenant à votre ordinateur. Mais si vos problèmes ne se résolvent pas, vous pouvez toujours tenter de nettoyer les têtes de lecture du magnétophone. En cas d'échec, c'est alors sûrement l'enregistrement du programme qui est en cause. Si vous l'avez sauvegardé vous-même, reportez-vous à la colonne de droite et tâchez de trouver ce qui a pu se passer. S'il s'agit d'un programme que vous avez acheté, demandez à un ami de l'essayer sur son matériel et en cas d'insuccès, retournez-le au vendeur.

Une fois que vous aurez déterminé le volume

idone, il conviendra pour tous les programmes que vous aurez sauvegardés vous-même, mais les cassettes du commerce ou celles enregistrées sur d'autres ordinateurs risquent de ne pas s'adapter tout à fait au même réglage.

## Sauvegarde des programmes

Pour sauvegarder un programme, vous n'avez pas besoin de vous préoccuper du volume du magnétophone. Contentez-vous de suivre les indications propres à votre ordinateur. Toutefois, une sage précaution consiste à sauvegarder plusieurs fois le même programme.

Une fois que le programme est enregistré, vous pouvez le faire défiler comme une cassette audio classique. Si vous mettez le magnétophone en lecture, en libérant l'écoute, vous entendrez une suite de bruits très aigus, signe que votre programme a bien été sauvegardé. Pour vous assurer que tout fonctionne, chargez le programme dans l'ordinateur. Si l'enregistrement a été mal réalisé, soit la manœuvre se révélera impossible, soit le résultat sera bourré d'erreurs. Un mauvais enregistrement peut être dû à la vétusté de la cassette utilisée ou à l'encrassement des têtes de lecture; procurez-vous une cassette de nettoyage, prenez une autre bande et recommencez. Entre-temps, il est probable que le programme ne sera plus en mémoire sur votre micro et qu'il vous faudra l'entrer à nouveau !

Si malgré tout, vous n'arrivez pas encore à sauvegarder le programme, il se peut que les signaux émis par l'ordinateur soient trop forts ou trop faibles pour le magnétophone. La plupart des magnétophones bon marché ont un niveau d'enregistrement déterminé automatiquement; il n'y a donc rien que vous puissiez faire s'ils ne correspondent pas à votre ordinateur? Essayez d'autres marques, demandez conseil à des clubs d'utilisateurs ou à des revendeurs spécialisés. Un bon professionnel saura même modifier les réglages pour rendre ordinateurs et magnétophones compatibles. Si vous pouvez régler le niveau d'enregistrement de votre magnétophone, testez plusieurs réglages pour déterminer le point optimum.

Tableau ASCII

Numéro de code	Caractère ASCII	Numéro de code	Caractère ASCII
32	espace	62	>
33	!	63	?
34	"	64	((
35	#	65	A
36	\$	66	B
37	%	67	C
38	&	68	D
39	'	69	E
40	(	70	F
41	)	71	G
42	*	72	H
43	+	73	I
44	,	74	J
45	-	75	K
46	.	76	L
47	/	77	M
48	0	78	N
49	1	79	O
50	2	80	P
51	3	81	Q
52	4	82	R
53	5	83	S
54	6	84	T
55	7	85	U
56	8	86	V
57	9	87	W
58	:	88	X
59	;	89	Y
60	<	90	Z
61	=		

Tableau ZX81

Numéro de code	Caractère ZX81	Numéro de code	Caractère ZX81
11	"	41	D
12	£	42	E
13	\$	43	F
14	:	44	G
15	?	45	H
16	(	46	I
17	)	47	J
18	>	48	K
19	<	49	L
20	=	50	M
21	+	51	N
22	-	52	O
23	*	53	P
24	/	54	Q
25	;	55	R
26	,	56	S
27	.	57	T
28	0	58	U
29	1	59	V
30	2	60	W
31	3	61	X
32	4	62	Y
33	5	63	Z
34	6		
35	7		
36	8		
37	9		
38	A		
39	B		
40	C		

## Dimension des écrans

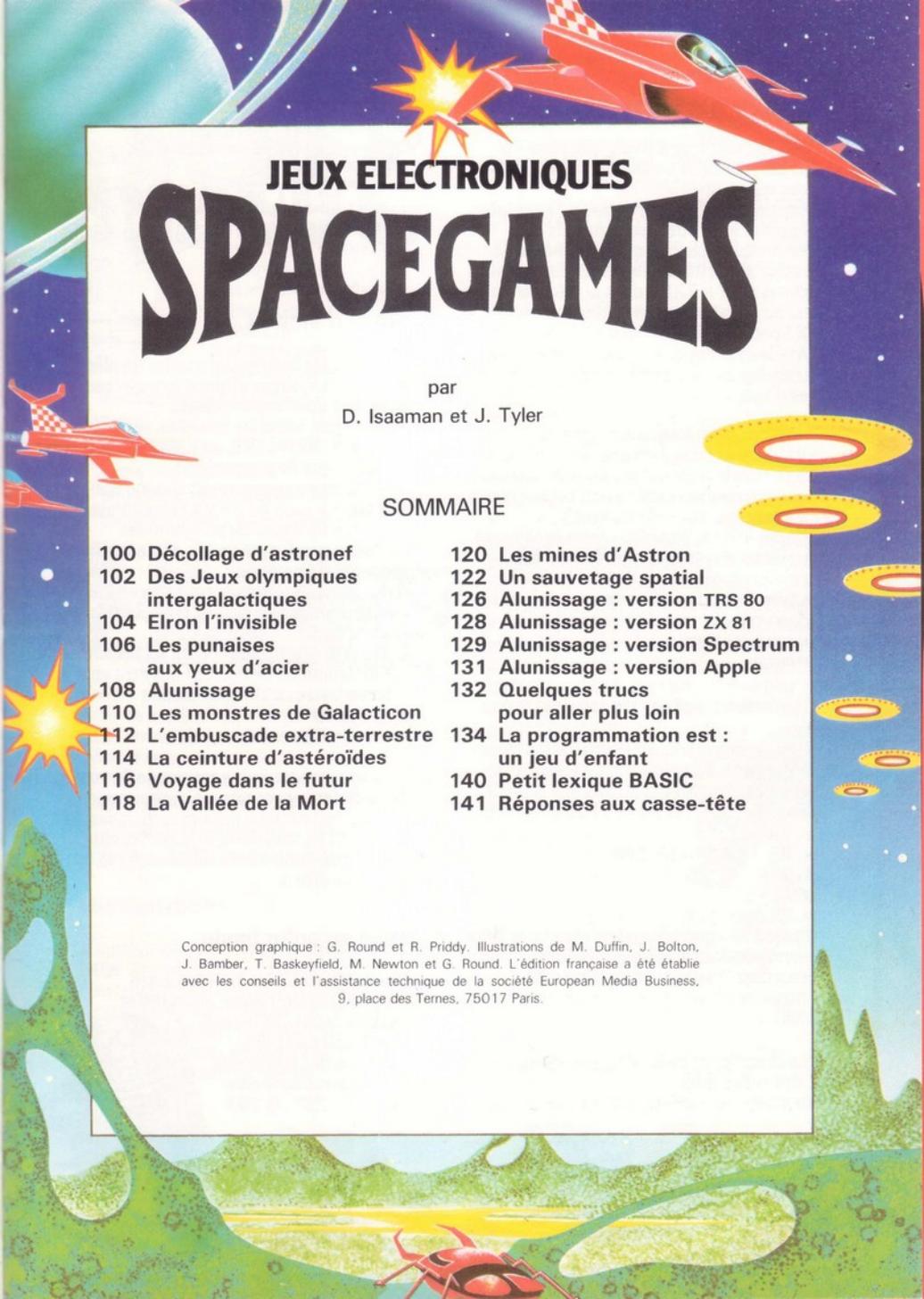
	Nombre maxi. d'éléments stockés en largeur (ou nombre de colonnes)	Nombre maxi. de lignes verticales (ou nombre de rangées)
VIC 20	22	23
TRS-80	64	16
BBC	20/40/80	16/24/32
ZX81	32	22
ZX Spectrum	32	22
Apple	40	25

# Tableau de conversion

Ce tableau montre quelques-unes des variations de langage BASIC pour les ordinateurs cités dans cet ouvrage. Il ne prend pas en compte les instructions graphiques, ni celles produisant des effets sonores ou des couleurs ; les différences sont trop importantes. Constatez également que si la majorité des ordinateurs (sauf le BBC) utilisent des instructions PEEK et POKE, par contre, les systèmes d'adressage en mémoire, c'est-à-dire les nombres placés après celles-ci et permettant de les retrouver, doivent être modifiés pour chacun d'entre eux.

Micro-ordinateur	BBC	VIC/Pet	Apple	TRS-80	ZX Spectrum	ZX81
Le programme						
Tire au sort un nombre compris entre 0 et 0,999999999	RND(1)	RND(1)	RND(1)	RND(0)	RND	RND
Tire au sort un nombre compris entre 1 et N	RND(N)	RND(1)*N + 1	RND(1)*N + 1	RND(N)	RND*N + 1	RND*N + 1
Tire au sort une lettre comprise entre A et Z	CHR\$(RND(26) + 64)	CHR\$(INT(RND(1)*26 + 65))	CHR\$(INT(RND(1)*26 + 65))	CHR\$(RND(26) + 64)	CHR\$(INT(RND*26 + 65))	CHR\$(INT(RND*26 + 38))
Nettoie l'écran	CLS	PRINT CHR\$(147)	HOME	CLS	CLS	CLS
Vérifie si une touche du clavier est pressée	INKEY\$(N)	GET XS	XS = "" IF PEEK(-16384) > 127 THEN GET XS	INKEY\$	INKEY\$	INKEY\$
Convertit les caractères en nombre code	ASC("X") (utilise le code ASCII)	ASC("X") (utilise le code ASCII)	ASC("X") (utilise le code ASCII)	ASC("X") (utilise le code ASCII)	CODE("X") (utilise le code ASCII)	CODE("X") (utilise le code ZX81)
Déplace le curseur vers le haut	PRINT CHR\$(1)	PRINT CHR\$(145)	CALL -998	PRINT CHR\$(27)	PRINT CHR\$(11)	PRINT CHR\$(112)
Déplace le curseur vers le bas	PRINT CHR\$(10)	PRINT CHR\$(17)	PRINT CHR\$(10)	PRINT CHR\$(26)	PRINT CHR\$(10)	PRINT CHR\$(113)
Déplace le curseur vers la gauche	PRINT CHR\$(8)	PRINT CHR\$(157)	PRINT CHR\$(8)	PRINT CHR\$(24)	PRINT CHR\$(8)	PRINT CHR\$(114)
Déplace le curseur vers la droite	PRINT CHR\$(9)	PRINT CHR\$(29)	PRINT CHR\$(21)	PRINT CHR\$(25)	PRINT CHR\$(9)	PRINT CHR\$(115)
Prend les N premiers caractères*	LEFT\$(A,N)	LEFT\$(A,N)	LEFT\$(A,N)	LEFT\$(A,N)	AS1 TO N)	AS1 TO N)
Prend les N derniers caractères*	RIGHT\$(A,N)	RIGHT\$(A,N)	RIGHT\$(A,N)	RIGHT\$(A,N)	ASIN TO )	ASIN TO )
Prend N caractères au milieu*	MID\$(A,N1,N2)	MID\$(A,N1,N2)	MID\$(A,N1,N2)	MID\$(A,N1,N2)	ASIN1 TO N2)	ASIN1 TO N2)

\* de la chaîne de caractères (variable alphanumérique).



# JEUX ELECTRONIQUES SPACEGAMES

par  
D. Isaaman et J. Tyler

## SOMMAIRE

- |   |   |
|---|---|
| 100 Décollage d'astronef                    | 120 Les mines d'Astron                        |
| 102 Des Jeux olympiques<br>intergalactiques | 122 Un sauvetage spatial                      |
| 104 Elron l'invisible                       | 126 Alunissage : version TRS 80               |
| 106 Les punaises<br>aux yeux d'acier        | 128 Alunissage : version ZX 81                |
| 108 Alunissage                              | 129 Alunissage : version Spectrum             |
| 110 Les monstres de Galacticon              | 131 Alunissage : version Apple                |
| 112 L'embuscade extra-terrestre             | 132 Quelques trucs<br>pour aller plus loin    |
| 114 La ceinture d'astéroïdes                | 134 La programmation est :<br>un jeu d'enfant |
| 116 Voyage dans le futur                    | 140 Petit lexique BASIC                       |
| 118 La Vallée de la Mort                    | 141 Réponses aux casse-tête                   |

Conception graphique : G. Round et R. Priddy. Illustrations de M. Duffin, J. Bolton, J. Bamber, T. Baskeyfield, M. Newton et G. Round. L'édition française a été établie avec les conseils et l'assistance technique de la société European Media Business, 9, place des Ternes, 75017 Paris.

## Avant-propos

Cette dernière partie vous propose des programmes de jeux simples pour micro-ordinateurs.

Écrits en BASIC (abréviation de l'anglais : *Beginner's All purpose Symbolic Instruction Code*), langage utilisé par la très grande majorité de ces matériels, ils sont prévus pour le ZX81, dont la mémoire n'est que de 1K\*. Ils s'adaptent facilement sur le ZX Spectrum, le BBC, le VIC 20, le TRS-80, l'Apple et le Pet grâce aux modifications signalées par des symboles renvoyant en bas de page.

Conçus pour tous, ces programmes n'exploitent peut-être que partiellement les possibilités de votre micro-ordinateur. Vous pourrez les modifier en conséquence, en vous reportant à la rubrique « Casse-Tête ». Vous trouverez également des idées, des trucs pour réaliser vos propres programmes, une table de conversion (page 96) pour vous aider à les adapter sur votre machine ainsi qu'un lexique des termes BASIC les plus courants.

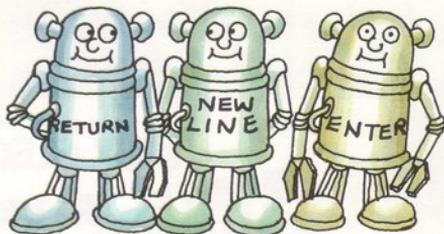
### Comment entrer un programme

Lorsque des modifications dans les lignes de programme s'avéreront nécessaires pour des ordinateurs autres que le ZX81, vous rencontrerez les symboles suivants :

- ▲ VIC et Pet
- ★ BBC et Acorn Electron
- TRS-80
- Apple
- ◻ ZX Spectrum.

Chaque fois que vous rencontrerez le signe correspondant à votre ordinateur, consultez l'instruction indiquée et entrez-la à la place de celle prévue pour le ZX81.

*\*La capacité de mémoire correspondant à 1 024 caractères.  
En moyenne, elle est de l'ordre de 16 à 64K.*



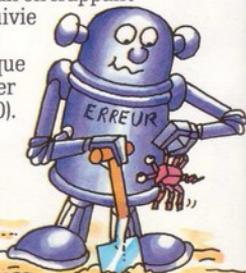
### Mode d'emploi

- 1 Entrez les instructions telles qu'elles sont écrites ; soyez attentif à respecter ponctuation et espacements.
- 2 Appuyez, selon les modèles, sur la touche RETURN, NEWLINE, ou ENTER, à la fin de chaque ligne de programme.
- 3 Vérifiez soigneusement chaque ligne.
- 4 Assurez-vous qu'il n'y a ni confusion, ni omission de lignes de programmes ; n'hésitez pas à vous munir d'une feuille de papier et d'une règle.
- 5 Repérez les symboles et assurez-vous que vous enregistrez bien l'instruction correspondant à votre matériel.
- 6 Dans le cas du ZX81 ou du ZX Spectrum, les instructions ne doivent pas être entrées lettre à lettre ; utilisez la touche appropriée.

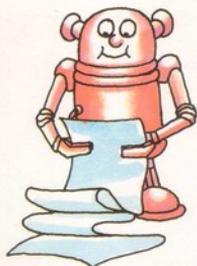
Si vous trouvez plus commode de vous faire dicter le programme, recommandez bien au lecteur de n'oublier ni virgules, ni points, ni parenthèses, ni espacements... Faites attention à bien différencier la lettre O et le chiffre 0.

### Mise au point finale

Une fois entré, faites défiler votre programme sur l'écran en frappant l'instruction LIST, suivie éventuellement des numéros des lignes que vous désirez consulter (par ex. : LIST 20-100).

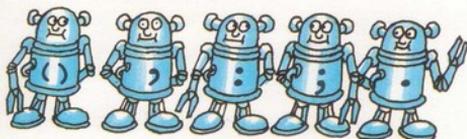


Vérifiez que tout a été convenablement tapé ; néanmoins il restera certainement quelques erreurs. Le mode d'emploi, fourni par le fabricant de votre ordinateur, pourra vous aider à en corriger quelques-unes ; autrement, réécrivez la ligne en entier, l'ordinateur la mettra automatiquement à la place de l'ancienne.



Voici une liste des principales sources d'erreur :

- 1 Ligne oubliée.
- 2 Ligne mal numérotée.
- 3 Vous avez oublié d'aller à la ligne.
- 4 Virgules, points, parenthèses, deux

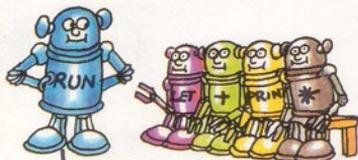


points, points-virgules, espacements... omis ou confondus, tout particulièrement dans les cas de programmes longs et complexes.

- 5 Vous avez utilisé une instruction inconnue de votre ordinateur.
- 6 Lettre O confondue avec 0 (zéro).
- 7 Erreur de chiffre ou nombre.

## Mise en route

Pour lancer un jeu frappez l'instruction RUN. Attention, sur certains micros l'instruction RUN ne doit pas être utilisée dans les jeux faisant appel au hasard (Random number) ; faites GOTO à la place (GOTO le premier numéro du listing).  
S'il subsiste une erreur de



programmation, le jeu s'arrêtera en cours d'exécution et l'ordinateur vous signalera, quelquefois, l'origine de cette interruption. Faites défiler à nouveau le programme et comparez-le scrupuleusement au listing de référence.

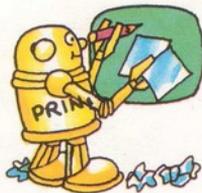
Si BREAK IN LINE 200 ou STOPPED AT 200 apparaît à la fin du jeu, relancez l'instruction de départ pour rejouer.

## Un peu d'entraînement

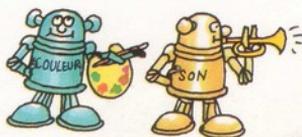
Vous trouverez dans cette partie des suggestions pour modifier ou compléter les jeux ; tentez également de les changer par vous-même. N'ayez pas peur, en aucun cas vous ne risquez d'endommager l'ordinateur ; et le programme d'origine reste disponible en cas d'échec.

Après quelques utilisations, vous souhaiterez modifier la vitesse d'exécution du jeu ; vous trouverez un commentaire concernant la localisation et les modalités de cette intervention en bas de page\*.

Là où figure l'instruction PRINT, vous avez la possibilité de modifier le texte entre guillemets. Sauf si votre appareil est un ZX81, vous pouvez même rajouter des messages. Choisissez un nombre (par exemple 105 pour glisser un texte entre 100 et 110), puis frappez PRINT et votre texte entre guillemets. Si votre ordinateur fait appel à des couleurs et des sons, aidez-vous de son mode d'emploi pour rendre encore plus intéressants les jeux présentés dans ce livre.



\*Voir page 133 la note particulière concernant BBC et ZX Spectrum.



# Décollage d'astronef

Vous êtes le commandant d'un astronef qui s'est écrasé sur une planète étrangère et vous devez repartir au plus vite dans l'engin extra-terrestre que vous venez de capturer. L'ordinateur de bord du vaisseau vous indique la gravitation de la planète. Il vous reste à deviner la poussée nécessaire au décollage. Si vous l'avez sous-estimée, le mécanisme de sécurité du vaisseau empêchera que vous ne soyez réduit en cendres. Si, après dix essais, vous êtes toujours sur la planète, les extra-terrestres vous captureront.





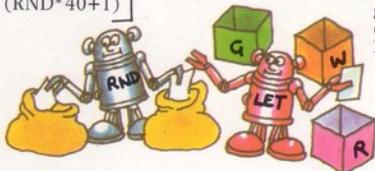
# Programme

▲●10 CLS  
20 PRINT « DÉCOLLAGE DE L'ASTRONEF »

Vide l'écran.

★■▲●30 LET G=INT(RND\*20+1)  
★■▲●40 LET M=INT(RND\*40+1)

L'ordinateur tire au hasard deux nombres : G gravitation entre 1 et 20, M masse du vaisseau entre 1 et 40.



50 LET R=G\*M

Multiplication de G par M donnant R (réponse).

60 PRINT « GRAVITATION= »;G

Écrit GRAVITATION suivi de sa valeur G.

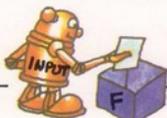
70 PRINT « TAPEZ LA POUSSÉE »

Demande l'introduction de la poussée P.

80 FOR C=1 TO 10

Contrôle d'une boucle telle que votre ordinateur recommande dix fois les instructions suivantes pour vous donner dix essais.

90 INPUT P



Entrée et stockage de la valeur choisie pour la poussée.

100 IF P>R THEN PRINT « TROP GRAND »;  
110 IF P<R THEN PRINT « TROP PETIT »;  
120 IF P=R THEN GOTO 190

Compare la valeur de la poussée avec celle de la réponse R, inscrit le bon message ou bien part en 190.

130 IF C<>10 THEN PRINT «, ESSAYEZ À NOUVEAU »

Demande un nouvel essai si vous y avez encore droit.

140 NEXT C

Retour de la boucle en 80 pour un nouveau tour.

150 PRINT

160 PRINT « VOUS AVEZ ÉCHOUÉ- »

170 PRINT « LES EXTRA-TERRESTRES VOUS ONT EU »

S'écrit après dix essais infructueux.

180 STOP

190 PRINT « C'EST BON DÉCOLLEZ »

200 END

La liste ci-dessus fonctionne sur un ZX81. Pour un autre ordinateur faire les modifications suivantes :

- 10 HOME
- ▲10 PRINT CHR\$(147)
- ★▲●30 LET G=INT(RND(1)\*20)
- 30 LET G=INT(RND(0)\*20)
- ★▲●40 LET M=INT(RND(1)\*40)
- 40 LET M=INT(RND(0)\*40)

Sauf sur le ZX81 et le Spectrum, on peut remplacer STOP par END (remarquez alors ce qui se passe).



## Comment rendre le jeu plus difficile

Vous pouvez modifier le programme de façon à ce qu'il vous autorise moins de dix essais, en agissant sur la dernière valeur de la ligne 80 et sur celle de la ligne 130 (elles doivent être égales).

## Casse-tête

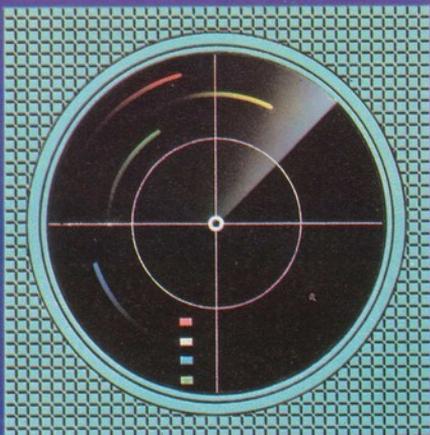
Vous pouvez modifier l'intervalle des valeurs possibles pour la poussée. Comment le feriez-vous?



Solution page 141.

# Des Jeux olympiques intergalactiques

Une compétition acharnée oppose les grandes compagnies mondiales de télévision pour obtenir la couverture exclusive des premiers Jeux olympiques intergalactiques. La compagnie qui l'emportera sera celle qui placera la première un satellite de télécommunication en orbite à la bonne altitude. Vous êtes l'ingénieur chargé du lancement pour le compte de la chaîne « Ère nouvelle ». Les décisions quant à l'angle et à la vitesse de la fusée de lancement reposent entièrement sur vos épaules. Réussirez-vous ?



## Programme

10 PRINT « JEUX OLYMPIQUES INTERGALACTIQUES »	}	Le programme choisit l'altitude à laquelle vous devez satelliser l'engin, la stocke en H et l'affiche.
20 LET H=INT(RND*100+1)		
30 PRINT « LE SATELLITE DOIT ÊTRE MIS SUR ORBITE »	}	Contrôle d'une boucle vous donnant 8 essais.
40 PRINT « À UNE ALTITUDE DE »;H		
50 FOR E=1 TO 8	}	Demande un angle et le stocke en A.
60 PRINT « ENTREZ L'ANGLE (0-90) »		
70 INPUT A	}	Demande une vitesse et la stocke en V.
80 PRINT « ENTREZ LA VITESSE (0-40000) »		
90 INPUT V	}	Utilise H pour calculer ce que l'angle devrait être, soustrait celui que vous avez entré, donnant ainsi votre erreur.
100 LET A=A-ATN(H/3)*180/3.14159		
110 LET V=V-3000*SQR(H+1/H)	}	Calcule la valeur théorique de la vitesse et la retranche de celle introduite.
120 IF ABS(A)<2 AND ABS(V)<100 THEN		
GOTO 210	}	Vérifie si vous êtes assez proche pour gagner, et si oui va en 210.
130 IF A<-2 THEN PRINT « TROP PENCHÉ »		
140 IF A>2 THEN PRINT « TROP VERTICAL »	}	Inscrit un commentaire approprié pour vous aider à votre prochain essai.
150 IF V<-100 THEN PRINT « TROP LENT »		
160 IF V>100 THEN PRINT « TROP RAPIDE »	}	Retour de la boucle pour un prochain essai.
170 NEXT E		
180 PRINT « VOUS AVEZ ÉCHOUÉ »	}	Affiche en cas d'échec après huit essais infructueux.
190 PRINT « VOUS ÊTES LICENCIÉ »		
200 STOP	}	Message de victoire.
210 PRINT « VOUS AVEZ RÉUSSI »		
220 PRINT « ÈRE NOUVELLE TV GAGNÉ-FÉLICITATIONS »	}	
230 STOP		

La liste ci-dessus fonctionne sur ZX81. Pour les autres ordinateurs, procédez aux modifications suivantes :

■ 20 LET H=INT(RND(0)\*100+1)

★▲● 20 LET H=INT(RND(1)\*100+1)

## Ajouts au programme

Ces trois lignes supplémentaires permettront à l'ordinateur de vous donner des points de bonus suivant la rapidité avec laquelle vous avez réussi le lancement.

```
222 LET B=INT(1000/E)
```

```
225 PRINT « VOUS AVEZ GAGNÉ UN »
```

```
227 PRINT « BONUS DE »; B; « CRÉDITS »
```

(Crédits : monnaie intergalactique bien connue.)

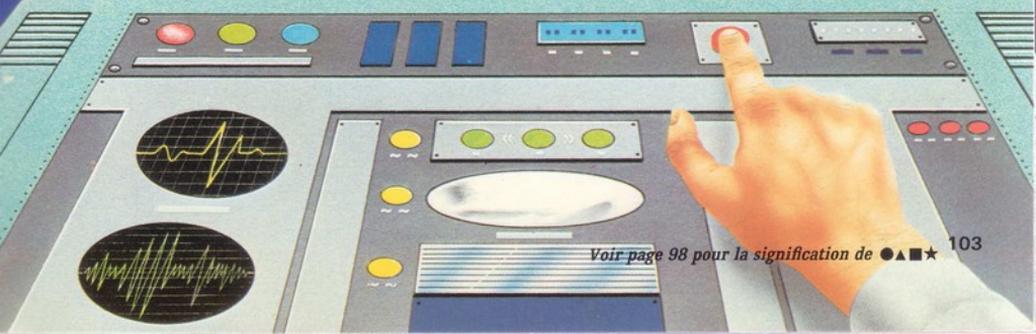
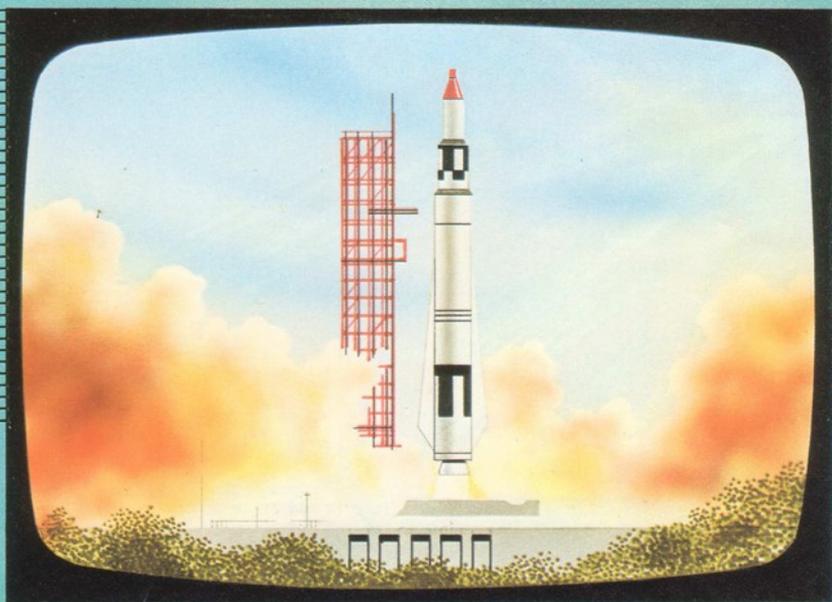


## Casse-tête

(Conseil : vous devez changer deux lignes et en ajouter une.)

Testez combien de temps vous tiendrez avant que la chaîne ne vous licencie... Pouvez-vous modifier le programme pour qu'il vous propose une nouvelle partie à chaque fois que vous gagnez et additionnez un nouveau bonus?

*Solution page 141.*



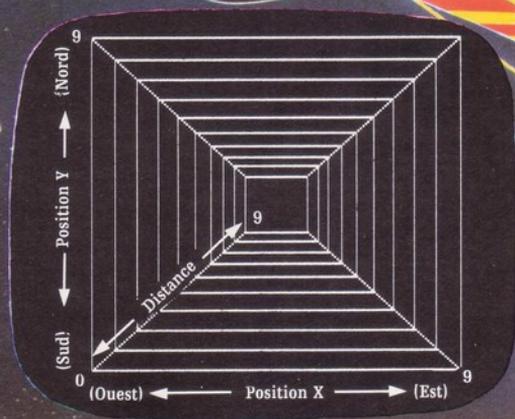
# Elron l'invisible

Quelque part en dessous de vous, dans la profondeur des ténèbres spatiales, se dissimule Elron, la créature hostile. Vous avez réussi à neutraliser toutes ses armes, sauf celles à court rayon d'action. Et Elron peut toujours rendre son vaisseau invisible. Vous savez qu'il rôde à l'intérieur de la grille tridimensionnelle que vous fournit l'ordinateur de bord de votre

chasseur (voir ci-dessous).

Mais où se trouve-t-il exactement? Vous disposez de quatre missiles spatiaux : chacun peut exploser à une position spécifiée par trois chiffres, de 0 à 9, que vous demandera votre ordinateur.

Pourrez-vous pulvériser le terrible Elron avant qu'il ne s'approche furtivement et ne vous capture?



▲●5 CLS

10 PRINT « CRÉATURE HOSTILE »

20 LET T=10

30 LET E=4



## Programme

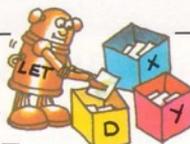
Donne la taille de la grille.

Donne le nombre d'essais alloués au joueur.

★▲●40 LET X=INT(RND)\*T

★▲●50 LET Y=INT(RND)\*T

★▲●60 LET D=INT(RND)\*T



La position d'Elron est donnée par ces trois lignes sélectionnant trois nombres entre 0 et la taille de la grille.

Début d'une boucle contrôlant la répétition des 15 lignes suivantes.

70 FOR I=1 TO E

80 PRINT « POSITION X(0 à 9)? »

85 INPUT X1

90 PRINT « POSITION Y (0 à 9)? »

100 INPUT Y1

110 PRINT « DISTANCE (0 à 9)? »

120 INPUT D1

Ces instructions vous demandent d'introduire trois nombres et les stockent en X1, Y1 et D1.

130 IF X=X1 AND Y=Y1 AND D=D1 THEN GOTO 300

Vérifie si vos nombres sont justes, et dans ce cas va en 300.

140 PRINT « LE TIR ÉTAIT »;

150 IF Y1>Y THEN PRINT « AU NORD »;

160 IF Y1<Y THEN PRINT « AU SUD »;

170 IF X1>X THEN PRINT « A L'EST »;

180 IF X1<X THEN PRINT « A L'OUEST »;

190 PRINT

200 IF D1>D THEN PRINT « TROP LONG »

210 IF D1<D THEN PRINT « TROP COURT »

Votre jeu est comparé à la position d'Elron et le résultat est affiché sur l'écran.

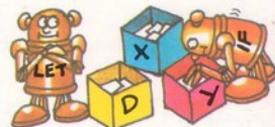
220 NEXT I

230 PRINT « VOTRE FIN EST PROCHE !! »

240 STOP

300 PRINT « \*BOUM\* VOUS L'AVEZ EU ! »

310 STOP



Retour de la boucle pour un nouvel essai.

Apparaît si vous avez utilisé tous vos essais.

Message de victoire.

(Un END final peut être nécessaire à certains ordinateurs.)

La liste du programme ci-dessus fonctionne sur un ZX81. Pour les autres ordinateurs, procédez aux modifications suivantes :

●5 HOME

▲5 PRINT CHR\$(147)

★▲●40 LET X=INT(RND(1)\*T)

■40 LET X=INT(RND(0)\*T)

★▲●50 LET Y=INT(RND(1)\*T)

■50 LET Y=INT(RND(0)\*T)

★▲●60 LET D=INT(RND(1)\*T)

■60 LET D=INT(RND(0)\*T)

## Comment rendre le jeu plus difficile

Ce programme a été écrit pour que vous puissiez accentuer la difficulté du jeu en modifiant la taille de la grille. Pour cela, donnez une valeur différente à la ligne 20 pour T. Si vous augmentez la taille de la grille, le nombre de missiles nécessaires sera plus grand. Modifiez alors la valeur de E en ligne 30.

## Casse-tête

Pouvez-vous trouver comment modifier le programme de façon à ce que l'ordinateur vous demande de choisir un niveau de difficulté qu'il prendrait comme taille de la grille T. (Tuyau : limitez les valeurs de T à l'intervalle 6-30, et prenez INT(T/3) comme valeur de E dans la ligne 30.)



Solution page 141.

Voir page 98 pour la signification de ●▲★

## Les punaises aux yeux d'acier

Vous êtes piégé ! De tous côtés vous apercevez les lueurs froides comme l'acier des yeux des punaises de l'espace avant qu'elles ne s'éclipsent derrière un rocher.

Lentement, les monstres se rapprochent de vous sans bruit, vous encerclent, attendant la moindre occasion pour vous agripper par leurs pattes crochues. Par chance, vous avez pu conserver votre canon à protons.

Les yeux apparaissent en quatre endroits différents sur votre écran, et correspondent aux touches 1 à 4. Si vous pressez la bonne touche pendant que les yeux sont affichés sur l'écran, vous pulvériserez la bête. Il y a dix punaises spatiales en tout. Plus vous en tuez, plus grandes seront vos chances de survivre...

### Programme

10 PRINT « PUNAISES DE L'ESPACE »

20 LET S=0

30 FOR T=1 TO 10

▲●40 CLS

★■▲●50 FOR I=1 TO INT(RND\*30+20)  
60 NEXT I

★■▲●70 LET R=INT(RND\*4+1)

★■▲●80 GOSUB 210+30\*R

90 PRINT « 00 »

★■▲●100 FOR I=1 TO 20

★▲●110 LET RS=INKEYS

120 IF RS<>« » THEN GOTO 140

130 NEXT I

140 IF VAL(« 0 »+RS)<>R THEN GOTO 210

150 LET S=S+1

Met le score à zéro pour le début du jeu.

Contrôle de la boucle vous donnant 10 tours de jeu.

Vide l'écran.

Boucle de retardement permettant un temps de latence différent dépendant de la valeur issue du RND.

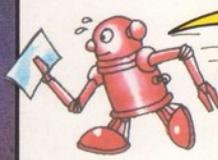
Tire un nombre de 1 à 4 et le stocke dans R.

Saut à un des quatre sous-programmes, dépendant de la valeur issue du RND. Ce qui permet de rechercher deux valeurs correspondant à une position sur l'écran : « A » espaces du bord gauche et « D » lignes du haut, puis saut en 350 pour placer le curseur dans l'espace choisi.

Affiche les yeux à la position choisie.

Boucle qui vous permet de savoir si vous appuyez sur une touche. Si oui, l'ordinateur va en 140 et teste si c'est la bonne.

Augmente le score de 1.



GOSUB branche l'ordinateur sur un sous-programme en dehors du programme principal (voir page suivante). RETURN, à la fin du sous-programme, renvoie au programme principal.

▲●160 CLS

170 GOSUB 350  
180 PRINT « \* »

★▲●190 FOR J=1 TO 40  
200 NEXT J

210 NEXT T

220 PRINT « VOUS AVEZ PULVÉRISÉ »;S;« /10 PUNAISES »

230 STOP

240 LET D=5

250 LET A=1

260 GOTO 350

270 LET D=1

280 LET A=9

290 GOTO 350

300 LET D=5

310 LET A=18

320 GOTO 350

330 LET D=10

340 LET A=7

350 FOR I=1 TO D

360 PRINT

370 NEXT I

380 PRINT TAB(A);

390 RETURN

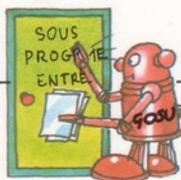
Vide l'écran.

Envoie le curseur dans la même position et affiche une étoile.

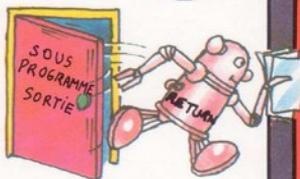
Boucle d'attente laissant l'étoile sur l'écran pour que vous puissiez la voir.

Retour de boucle pour un prochain tour.

Écrit le score.



Sous-programmes



## Comment changer la vitesse

Vous pouvez accélérer le jeu en remplaçant la dernière valeur de la ligne 100 par une autre, plus faible.

## Plus de place sur l'écran

Ce programme a été écrit pour le VIC 20 qui possède la plus petite taille d'écran. Pour les autres ordinateurs, vous pouvez augmenter les valeurs de A aux lignes 250,280,310 et 340. Consultez votre notice pour connaître la largeur de votre écran.



La liste ci-dessus fonctionne sur le ZX81. Pour les autres ordinateurs, procédez aux modifications suivantes :

●40,160 HOME

▲40,160 PRINT CHR\$(147)

■50 FOR I=1 TO INT (RND(0)\*300+200)

★▲●50 FOR I=1 TO INT (RND(1)\*300+200)

■70 LET R=INT(RND(0)\*4+1)

★▲●70 LET R=INT(RND(1)\*4+1)

★▲●80 ON R GOSUB 240,270,300,330

★▲●100 FOR I=1 TO 150

●105 R\$=« »

▲110 GET R\$

★110 R\$=INKEY\$(1)

●110 IF PEEK(-16384)>127 THEN GET R\$

★▲●190 FOR J=1 TO 300

## Casse-tête

Pourriez-vous modifier le programme pour que les yeux apparaissent en plus de quatre endroits? Comment ajoutez-vous d'autres punaises?

Voir page 98 pour la signification de ●▲■★

# Alunissage

Vous êtes aux commandes d'un module lunaire qui amène une petite équipe d'astronautes à la surface de la Lune. Pour alunir sain et sauf, vous devez ralentir lors de l'approche. Ce qui brûle du carburant, et vous n'en n'avez qu'une réserve limitée.

L'ordinateur de bord vous indique l'altitude, la vitesse et la réserve de carburant initiales. Il vous demande la quantité de carburant que vous comptez utiliser. Le programme donnera alors les nouvelles vitesse et altitude de l'appareil. Une consommation de 5 unités maintiendra une vitesse constante; plus forte, elle freinera la progression du module. Pour alunir sans casse, il vous faut amener cette vitesse au plus près de zéro.

```
▲●10 CLS
20 PRINT « ALUNISSAGE »
30 LET T=0
40 LET H=500
50 LET V=50
60 LET F=120
70 PRINT « CHRONO »;T,« ALTITUDE »;H
80 PRINT « VITESSE »;V,« FUEL »;F
```

```
90 IF F=0 THEN GOTO 140
```

```
100 PRINT « CONSUMMATION? (0-30) »
110 INPUT C
120 IF C<0 THEN LET C=0
130 IF C>30 THEN LET C=30
140 IF C>F THEN LET C=F
```

```
150 LET V1=V-C+5
160 LET F=F-C
170 IF (V1+V)/2>=H THEN GOTO 220
```

```
180 LET H=H-(V1+V)/2
```

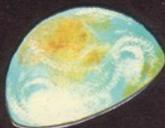
```
190 LET T=T+1
```

```
200 LET V=V1
```

```
210 GOTO 70
```

```
220 LET V1=V+(5-C)*H/V
230 IF V1>5 THEN PRINT « CRASH-AUCUN SURVIVANT »
240 IF V1>1 AND V1<=5 THEN PRINT « OK-MAIS QUELQUES BLESSÉS »
250 IF V1<=1 THEN PRINT « ALUNISSAGE PARFAIT »
260 STOP
```

Remarquez les virgules et points-virgules aux lignes 70 et 80. Essayez de voir ce qui se passe si vous ne les mettez pas.



La liste ci-dessus fonctionne sur un ZX81. Pour les autres ordinateurs, procédez aux modifications suivantes :

- 10 HOME
- ▲10 PRINT CHR\$(147)

Voir page 98 pour la signification de ●▲



## Programme

Donne et affiche les valeurs initiales du chrono, de l'altitude, de la vitesse et de la réserve en carburant.

Si vous n'avez plus de carburant, l'ordinateur saute la partie du programme où il vous demande combien vous désirez consommer et affiche jusqu'à l'arrivée au sol les commentaires concernant votre descente.

Demande votre consommation, la modifications pour que la consommation entre dans les limites imposées.

stocke en C et procède aux tests et Calcule la nouvelle vitesse V1.

Calcule votre nouvelle réserve de combustible.

Teste si la distance parcourue pendant votre dernier essai est suffisante pour avoir aluni. Dans ce cas, l'ordinateur va en 220 pour savoir de quelle façon vous l'avez fait.

Calcule votre nouvelle altitude.

Augmente le chrono d'une unité.

Stocke la nouvelle vitesse en V pour pouvoir la traiter ultérieurement.

Retour de la boucle pour le prochain tour.

Calcule la vitesse à l'arrivée au sol, commente le type d'alunissage.

## Ajouts au programme

Si vous complétez le programme avec les lignes suivantes, vous verrez une étoile s'afficher à chaque tour. La distance entre l'étoile et le bord gauche de l'écran correspond à votre altitude au-dessus de la surface de la Lune.

```
85 FOR I=2 TO H/500*nn
```

```
86 PRINT « »;
```

```
87 NEXT I
```

```
88 PRINT «*»
```

nn est la largeur de votre écran.

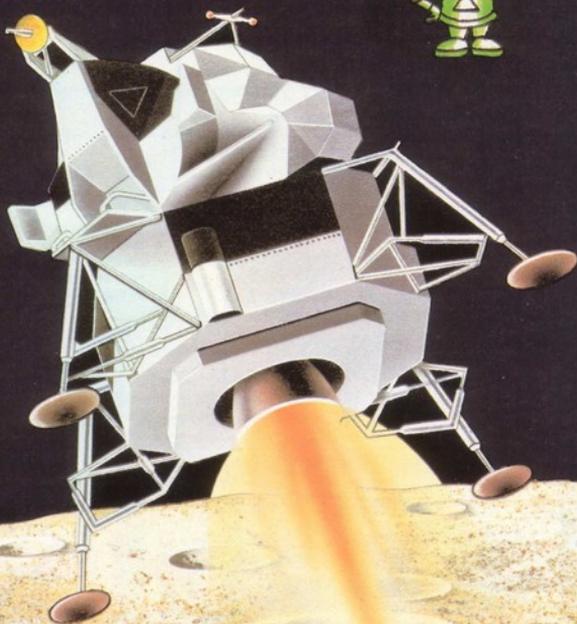
## Essayez de changer

Essayez de changer les valeurs de H, V et F des lignes 40 à 60, et voyez ce qui arrive.

## Casse-tête

Vous pourriez rendre le jeu plus facile en augmentant la vitesse maximale possible pour un alunissage sans danger. Comment feriez-vous cette modification?

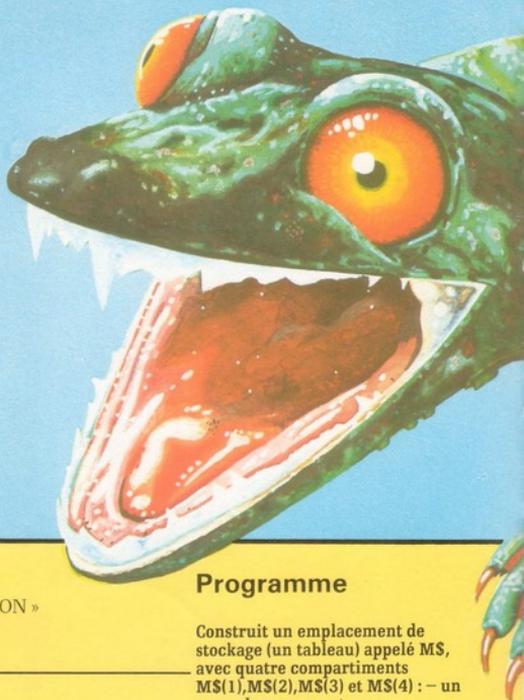
*Solution page 141.*



# Les monstres de Galacticon

Atterrir sur Galacticon semblait facile. Mais personne ne vous avait prévenu que cette planète abritait les monstres les plus terrifiants de l'univers.

A chaque fois qu'un monstre vient vous menacer, vous devez choisir une de vos trois armes : un pistolet à rayons (R), un sabre-laser (S) ou un troueur-dispersant (T). Ferez-vous le bon choix? Si vous survivez, la conquête de Galacticon est à votre portée.



## Programme

10 PRINT « LES MONSTRES DE GALACTICON »

20 DIM M\$(4)

30 LET N=4

40 LET M=5

50 LET M\$(1)=« SULFACIDOR »

60 LET M\$(2)=« FLAMGONDAR »

70 LET M\$(3)=« BALNOLOTIN »

80 LET M\$(4)=« GOLANDOR »

90 FOR I=1 TO N

★▲●100 LET A=INT(RND\*N+1)

★▲●110 LET B=INT(RND\*N+1)

120 LET T\$(A)=M\$(B)

130 LET M\$(A)=M\$(B)

140 LET M\$(B)=T\$(A)

150 NEXT I

160 FOR T=1 TO 8

▲●170 CLS

★▲●180 LET R=INT(RND\*N+1)

190 PRINT « UN MONSTRE APPROCHE... »

200 PRINT « C'EST UN »;M\$(R)

210 PRINT « QUELLE ARME? (R,S OU T) »

220 INPUT R\$

s★▲●230 LET W=CODE(R\$)-54+R

★▲●240 LET W=W-3\*(W>3)-3\*(W>6)

250 IF W=2 THEN GOTO 300

Construit un emplacement de stockage (un tableau) appelé **M\$**, avec quatre compartiments **M\$(1)**, **M\$(2)**, **M\$(3)** et **M\$(4)** : un pour chaque monstre.

Remet le nombre de monstres à 4.

Donne l'effectif de votre groupe : 5.

Place les quatre noms de monstres dans le tableau.

Vous pouvez utiliser cette routine dans un autre programme principal où l'on veut mélanger des chaînes de caractères.

Ces lignes mélangent les noms des monstres. L'ordinateur exécute **N** fois la boucle; à chaque passage il tire 2 nombres entre 1 et **N** et échange les deux noms correspondant à ces numéros. **T\$** est une chaîne intermédiaire utilisée en cours de traitement.

Contrôle de la boucle pour 8 tours.

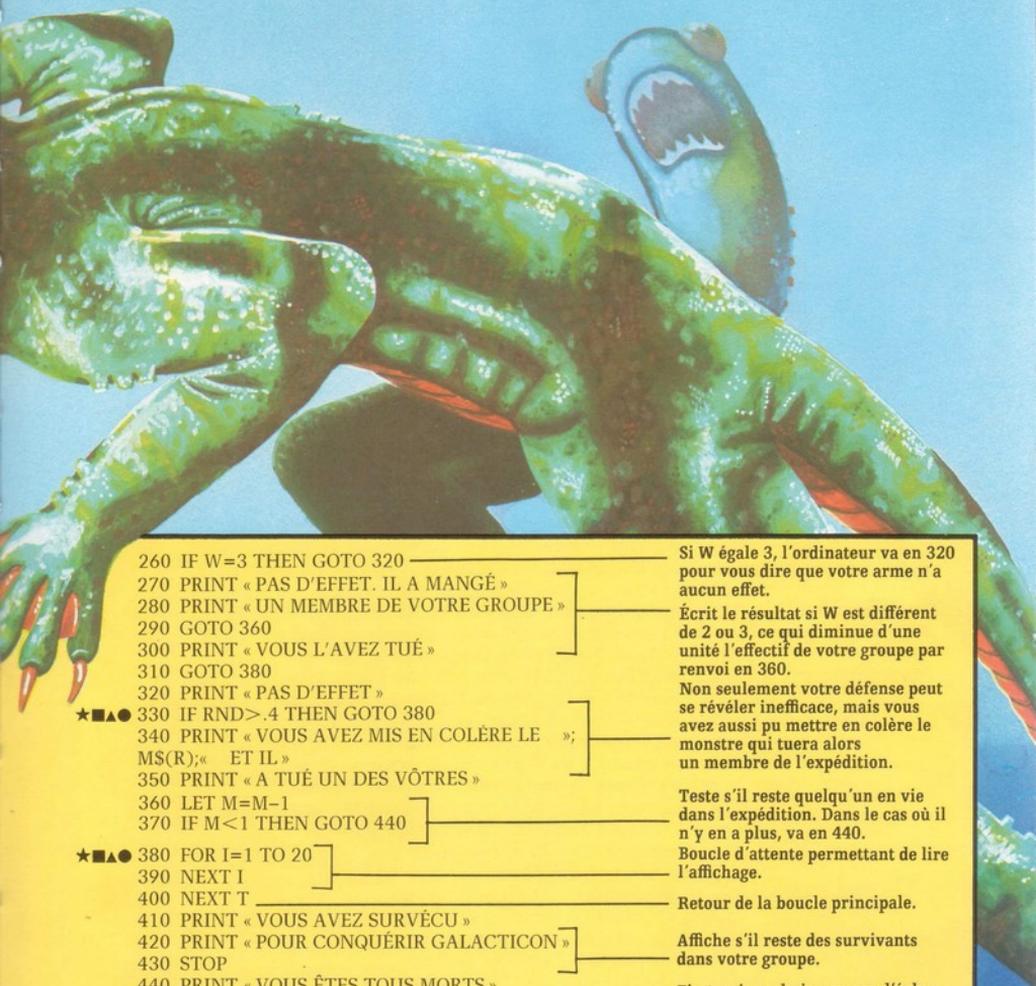
Vide l'écran.

Choisit un des monstres et affiche son nom.

L'ordinateur détermine l'arme choisie en utilisant les valeurs de **R** et **R\$**, en passant par l'intermédiaire du code ASCII.

Si **W** égale 2, l'ordinateur va en 300 pour vous informer que vous avez tué le monstre.





```

260 IF W=3 THEN GOTO 320
270 PRINT « PAS D'EFFET. IL A MANGÉ »
280 PRINT « UN MEMBRE DE VOTRE GROUPE »
290 GOTO 360
300 PRINT « VOUS L'AVEZ TUÉ »
310 GOTO 380
320 PRINT « PAS D'EFFET »
★■▲ 330 IF RND>.4 THEN GOTO 380
340 PRINT « VOUS AVEZ MIS EN COLÈRE LE
MS(R);« ET IL »
350 PRINT « A TUÉ UN DES VÔTRES »
360 LET M=M-1
370 IF M<1 THEN GOTO 440
★■▲ 380 FOR I=1 TO 20
390 NEXT I
400 NEXT T
410 PRINT « VOUS AVEZ SURVÉCU »
420 PRINT « POUR CONQUÉRIR GALACTICON »
430 STOP
440 PRINT « VOUS ÊTES TOUS MORTS »
450 STOP

```

Si W égale 3, l'ordinateur va en 320 pour vous dire que votre arme n'a aucun effet.

Écrit le résultat si W est différent de 2 ou 3, ce qui diminue d'une unité l'effectif de votre groupe par renvoi en 360.

Non seulement votre défense peut se révéler inefficace, mais vous avez aussi pu mettre en colère le monstre qui tuera alors un membre de l'expédition.

Teste s'il reste quelqu'un en vie dans l'expédition. Dans le cas où il n'y en a plus, va en 440.

Boucle d'attente permettant de lire l'affichage.

Retour de la boucle principale.

Affiche s'il reste des survivants dans votre groupe.

Fin tragique du jeu en cas d'échec..

La liste ci-dessus fonctionne sur un ZX81. Pour les autres ordinateurs, procédez aux modifications suivantes :

- 100 LET A=INT(RND(0)\*N+1)
- ★▲ 100 LET A=INT(RND(1)\*N+1)
- 110 LET B=INT(RND(0)\*N+1)
- ★▲ 110 LET B=INT(RND(1)\*N+1)
- 170 HOME
- ▲ 170 PRINT CHR\$(147)
- 180 LET R=INT(RND(0)\*N+1)
- ★▲ 180 LET R=INT(RND(1)\*N+1)
- ★■▲ 230 LET W=ASC(R\$)-81+R
- s230 LET W=CODE(R\$)-81+R

- ★■▲ 240 LET W=W+3\*(W>3)+3\*(W>6)
- ★▲ 330 IF RND(1)>.4 THEN GOTO 380
- 330 IF RND(0)>.4 THEN GOTO 380
- ★■▲ 380 FOR I=1 TO 300

## Casse-tête

Il existe au moins quatre moyens de rendre le jeu plus difficile. Pourriez-vous trouver lesquels?



Solution page 141.

# L'embuscade extra-terrestre



Vous êtes le capitaine d'un croiseur interstellaire qui, à la suite d'un incident technique sur un de ses moteurs, s'est égaré dans une zone interdite. Des chasseurs-commandos extra-terrestres vous attaquent et, pour rendre l'événement encore plus critique, ils utilisent un système de brouillage-radar qui vous fournit de fausses indications.

Heureusement, votre ordinateur possède un système de décodage dont vous pouvez vous servir pour calculer la position des vaisseaux ennemis; mais vous devez être rapide, car ils ne restent pas longtemps en place!

L'ordinateur vous indique la position erronée par une lettre accompagnée d'un nombre de code qui correspond au nombre de lettres séparant, dans l'alphabet, la lettre de position fournie par le radar (la fausse) de celle réellement valable. Si l'on vous propose de décoder M4, vous devez entrer O,

de même E pour C2 ou T pour S1. L'introduction de la lettre oriente et déclenche automatiquement votre canon laser. Si la réponse était la

bonne, le score augmente d'un coup. Vous

pouvez choisir la difficulté du jeu en entrant un nombre de 1 à 10

correspondant au maximum de lettres à ajouter. Il y a 10 chasseurs extra-terrestres à abattre à chaque partie.

Combien en toucherez-vous?

## Programme



```

10 CLS
20 PRINT « EMBUSCADE EXTRA-TERRESTRE »
30 PRINT
40 PRINT « DIFFICULTÉ (1-10) »
50 INPUT D
60 IF D<1 OR D>10 THEN GOTO 50
70 LET S=0
80 FOR G=1 TO 10

```

Demande le niveau de difficulté, le stocke en D et vérifie s'il est compris dans les bonnes limites.

Remet le score à zéro.

Contrôle de la boucle vous donnant 10 tours.

```

90 LET L$=CHR$(INT(RND*(26-D)+38))

```

Tire une lettre entre A et celle qui représente votre niveau de difficulté

```

100 LET N=INT(RND*D+1)

```

Tire un nombre entre 1 et D.

```

110 CLS
120 PRINT
130 PRINT L$,N

```

Affiche la lettre et le nombre.

```

140 FOR I=1 TO 20+D*5

```

Teste si vous êtes en train d'appuyer sur une touche. Dans ce cas, va en 190.

```

150 LET IS=INKEY$
160 IF IS<><> THEN GOTO 190
170 NEXT I

```

```

180 GOTO 200

```

Si vous n'avez pas appuyé sur une touche, l'ordinateur va en 200, ce qui renvoie la boucle pour un nouvel essai.



Teste si vous avez appuyé sur la bonne touche. Dans ce cas, augmente votre score de 1.

```

190 IF IS=CHR$(CODE(L$)+N) THEN LET S=S+1

```

Retour de la boucle, pour un nouveau tour.

```

200 NEXT G

```

```

210 PRINT « VOUS AVEZ ABATTU »;S;« /10 »

```

Écrit le score après 10 tours.

La liste ci-dessus fonctionne sur ZX81. Pour les autres ordinateurs, procédez aux modifications suivantes :

- 10, 110 HOME
- ▲ 10, 110 PRINT CHR\$(147)
- 90 LET L\$=CHR\$(INT(RND(0)\*(26-D)+65))
- ★▲ 90 LET L\$=CHR\$(INT(RND(1)\*(26-D)+65))
- s90 LET L\$=CHR\$(INT(RND\*(26-D)+65))
- 100 LET N=INT(RND(0)\*D+1)
- ★▲ 100 LET N=INT(RND(1)\*D+1)
- ★▲ 140 FOR I=1 TO 200+D\*50
- 140 FOR I=1 TO 100+D\*50
- 145 IS=« »
- ★ 150 LET IS=INKEY\$(1)
- ▲ 150 GET IS
- 150 IF PEEK(-16384)>127 THEN GET IS
- ★▲ 190 IF IS=CHR\$(ASC(L\$)+N) THEN LET S=S+1

## Comment modifier la rapidité du jeu

Si le jeu est trop rapide pour vous, placez un nombre plus fort au milieu de la ligne 140 (remplacez 20 par 200, par exemple). Vous pouvez de même accélérer l'allure en le réduisant.



## Comment rendre le jeu plus difficile

Vous pouvez changer le 1 des lignes 40 et 60 en 3 par exemple (ou plus).

## Casse-tête

Modifiez le programme pour que le score s'adapte à la difficulté du jeu : un point par succès au niveau 1; deux points au niveau deux...

# La ceinture d'astéroïdes

Vous devez traverser une ceinture d'astéroïdes. Pour éviter de vous y écraser, la seule solution est de les détruire par une puissance de feu calculée en fonction de la taille de l'objet céleste.

Les astéroïdes apparaissent sur l'écran de votre ordinateur comme un groupe d'étoiles, le nombre des étoiles correspondant à la masse. Tapez alors le nombre d'étoiles représentées pour déclencher le tir; mais faites vite, car les astéroïdes viennent vers vous en amas denses et rapides.

```
10 PRINT « CEINTURE D'ASTÉROÏDES »
```

```
20 LET S=0
```

```
30 FOR G=1 TO 10
```

```
▲● 40 CLS
```

```
★■▲● 50 LET A=INT(RND*18+1)
```

```
★■▲● 60 LET D=INT(RND*12+1)
```

```
★■▲● 70 LET N=INT(RND*9+1)
```

```
80 FOR I=1 TO D  
90 PRINT  
100 NEXT I
```

```
110 FOR I=1 TO N  
120 IF I<>1 AND I<>4 AND I<>7 THEN GOTO 150  
130 PRINT  
140 PRINT TAB(A);  
150 PRINT « * »;  
160 NEXT I
```

```
170 PRINT
```

```
★■▲● 180 FOR I=1 TO 10
```

```
★▲● 190 LET Q=VAL(« 0 »+INKEYS)
```

```
200 IF Q<>0 THEN GOTO 240
```

```
★ 210 NEXT I
```

```
220 PRINT « ÉCRASÉ SUR L'ASTÉROÏDE »
```

```
230 GOTO 270
```

```
240 IF Q<>N THEN GOTO 270
```

## Programme

Remet le score à zéro.

Contrôle d'une boucle vous donnant dix tours.

Tire un nombre pour le positionnement de l'astéroïde en largeur sur l'écran. L'affiche en A.

Tire un nombre (de 1 à 12) pour le positionnement de l'astéroïde en hauteur sur l'écran. L'affiche en D.

Tire le nombre d'étoiles qui représenteront l'astéroïde (1 à 9).

Déplace le curseur de D lignes vers le bas de l'écran.

Boucle tournant N fois pour afficher à chaque passage une étoile en position appropriée.

Boucle permettant à l'ordinateur de savoir si vous appuyez sur une touche. Si oui, de stocker la valeur en Q, puis d'aller en 240.

S'affiche si vous avez dépassé le temps imparti pour l'introduction d'une réponse.

Teste si votre nombre est différent de N (nombre réel). Dans ce cas, va en 270.



- 250 PRINT « VOUS L'AVEZ DÉTRUIT » \_\_\_\_\_ Affiche si vous avez entré le bon nombre.
- 260 LET S=S+1 \_\_\_\_\_ Augmente votre score de 1.
- 270 IF Q<N THEN PRINT « TIR TROP FAIBLE »  
280 IF Q>N THEN PRINT « TIR TROP PUISSANT » \_\_\_\_\_ Compare votre valeur avec N et commente l'erreur.
- ★▲● 290 FOR I=1 TO 50  
300 NEXT I \_\_\_\_\_ Boucle d'attente pour la visualisation des messages.
- 310 NEXT G \_\_\_\_\_ Retour de la boucle pour un nouveau tour.
- 320 PRINT « DESTRUCTION  
DE »;S;« SUR 10 » \_\_\_\_\_ Affiche votre score après dix tours.
- 330 STOP

La liste ci-dessus fonctionne sur ZX81. Pour les autres ordinateurs, procédez aux modifications suivantes :

- 40 HOME
- ▲ 40 PRINT CHR\$(147)
- 50 LET A=INT(RND(0)\*18+1)
- ★▲● 50 LET A=INT(RND(1)\*18+1)
- 60 LET D=INT(RND(0)\*12+1)
- ★▲● 60 LET D=INT(RND(1)\*12+1)
- 70 LET N=INT(RND(0)\*9+1)
- ★▲● 70 LET N=INT(RND(1)\*9+1)
- 175 Q=0
- ★ 180
- ▲● 180 FOR I=1 TO 100
- ▲ 190 GET Q
- 190 IF PEEK(-16384)>127 THEN GET Q
- ★ 190 Q=INKEY(100)-48
- ★ 210
- ★■ 290 FOR I=1 TO 500
- ▲● 290 FOR I=1 TO 250

### Modifier la vitesse du jeu

La ligne 180 (190 pour le BBC ACORN) détermine le temps pour appuyer sur la touche. Changez la dernière valeur de cette ligne (ou la valeur entre parenthèses pour le BBC) par une plus faible pour accélérer le jeu.

### Casse-tête

Pourriez-vous adapter le programme de façon à ce qu'il ajoute au score le nombre d'étoiles que comprenait l'astéroïde détruit?



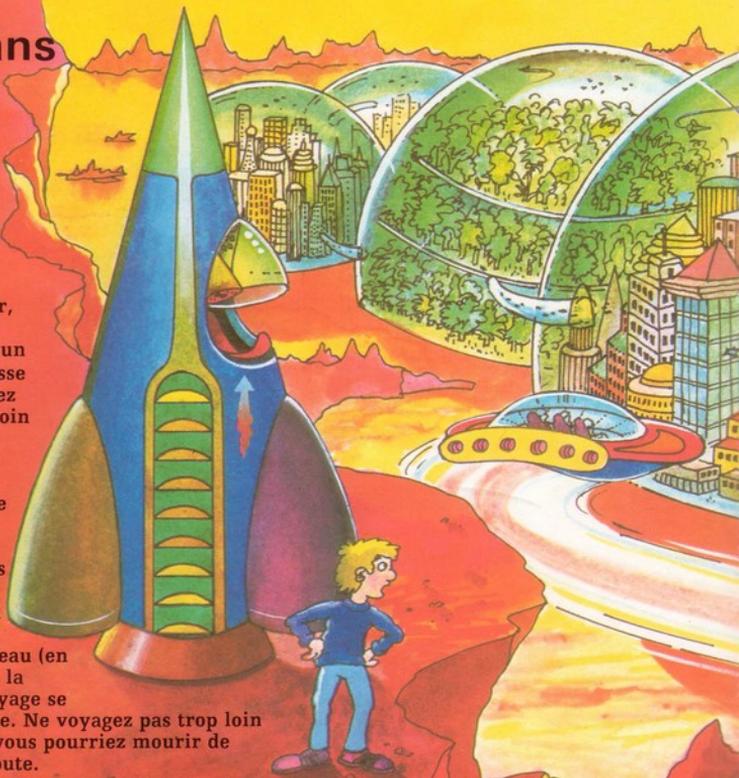
Solution page 142.

# Voyage dans le futur

Imaginez que vous voyagez dans un vaisseau spatial à une vitesse proche de celle de la lumière.

Étrangement, le temps se met à passer plus lentement dans votre vaisseau qu'à l'extérieur, suivant les théories d'Einstein. Ainsi, après un long voyage à cette vitesse fulgurante, vous pourriez revenir sur Terre plus loin dans le futur que ne l'indiqueraient vos horloges de bord.

L'ordinateur vous donne le temps en années qui s'écouleront sur Terre avant votre retour. Vous décidez alors de la distance à parcourir (en années-lumière) et de la vitesse de votre vaisseau (en fraction de la vitesse de la lumière) pour que le voyage se termine à la date désirée. Ne voyagez pas trop loin ou trop lentement, car vous pourriez mourir de vieillesse en cours de route.



```

▲● 10 CLS
20 PRINT « VOYAGE DANS LE FUTUR »
★■▲● 30 LET T=INT(RND*100+25)
40 PRINT « VOUS DÉSIREZ ALLER »;T
50 PRINT « ANNÉES DANS LE FUTUR. »
60 PRINT

70 PRINT « VITESSE DU VAISSEAU (ENTRE 0 et 1) »
80 INPUT V
90 IF V>=1 OR V<=0 THEN GOTO 70

100 PRINT « DISTANCE À PARCOURIR »
110 INPUT D
120 LET T1=D/V
    
```



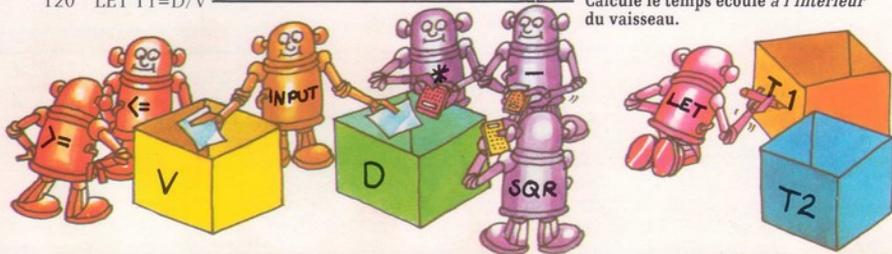
## Programme

Choisit le nombre d'années que vous aurez à laisser écouler sur Terre (entre 25 et 124) avant votre retour et l'affiche.

Demande la vitesse et vérifie si elle est bonne.

Demande une distance à parcourir et la stocke en D.

Calcule le temps écoulé à l'intérieur du vaisseau.





130 LET T2=T1/SQR(1-V\*V)

Calcule le temps écoulé à l'extérieur du vaisseau.

140 PRINT « VOUS AVEZ MIS »;T1;« ANS »

150 PRINT « ET ÊTES ARRIVÉ »;T2;« ANNÉES ».

160 PRINT « DANS LE FUTUR. »

Affiche les deux durées écoulées après le voyage.

170 IF T1>50 THEN GOTO 210

Teste si vous avez mis plus de temps que vous n'aviez à vivre (50 ans). Va en 210 dans ce triste cas.

180 IF ABS(T-T2)<=5 THEN PRINT « VOUS ÊTES ARRIVÉ À TEMPS »

190 IF ABS(T-T2)>5 THEN PRINT « MÊME À 5 ANS PRÈS »

200 STOP

210 PRINT « VOUS ÊTES DÉCÉDÉ EN COURS DE ROUTE »

220 STOP

Teste si vous êtes arrivé, à 5 ans près, à destination et l'affiche.

La liste ci-dessus fonctionne sur un ZX81. Pour les autres ordinateurs, procédez aux modifications suivantes :

- 10 HOME
- ▲ 10 PRINT CHR\$(147)
- 30. LET T=INT(RND(0)\*100+25)
- ★▲ 30 LET T=INT(RND(1)\*100+25)

### Casse-tête

Trouvez comment modifier le programme pour prendre en compte les propositions suivantes :

- 1) Augmenter l'intervalle possible de temps à laisser écouler sur Terre.
- 2) Réduire la précision d'arrivée de cinq à deux ans.
- 3) Allonger ou écourter votre espérance de vie.



# La Vallée de la Mort

Il n'y a qu'une solution pour échapper aux implacables dépeceurs : garder votre sang-froid pour piloter votre « dard rapide monoplace » à travers la gorge déchiquetée et sans fond connue sous le nom de Vallée de la Mort. Votre ordinateur vous demandera la largeur de la vallée. Essayez 15 (10 si vous utilisez un VIC 20) pour commencer; puis, à vous de jouer - 8 est déjà difficile (6 pour le VIC 20). Dirigez votre appareil en appuyant sur P pour aller à droite et Q pour la gauche. Accrochez-vous si vous voulez sortir sain et sauf de la Vallée de la Mort !

## Programme

10 PRINT « VALLÉE DE LA MORT »

20 LET S=0

30 LET M=200

40 PRINT « LARGEUR? »

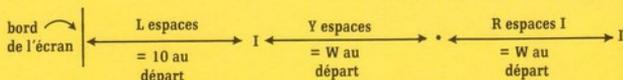
50 INPUT W

60 LET W=INT(W/2)

▲ 70 LET L=10

80 LET Y=W

90 LET R=W



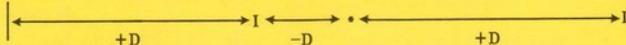
★▲● 100 LET D=INT(RND\*3-1)

110 IF L+D<0 OR L+D>20 THEN GOTO 100

120 LET L=L+D

130 LET Y=Y-D

140 LET R=R+D



s★▲● 145 SCROLL

150 LET N=L

160 GOSUB 1000

170 PRINT « I »;

180 LET N=Y

190 GOSUB 1000

200 PRINT « \* »;

210 LET N=R

220 GOSUB 1000

230 PRINT « I »

→ ★▲● 240 LET IS=INKEYS

250 IF IS<>« Q » THEN GOTO 280

260 LET Y=Y-1

270 LET R=R+1

280 IF IS<>« P » THEN GOTO 310

Remet à zéro le nombre de tours passés depuis le début du programme.

M est le nombre de tours maximal alloués.

Demande une valeur pour la largeur (W), la divise par deux, en prend la valeur entière pour éliminer toute partie fractionnaire et stocke en W.

R, Y et L sont les distances entre votre vaisseau et les parois du ravin.

Tire une valeur égale à -1,0 ou +1 et la stocke en D.

Teste si L+D n'est ni trop grand ni trop petit, ce qui ferait disparaître les parois du ravin des bords de l'écran.

Modifie la répartition des espaces entre les parois et le vaisseau.

Déplace le curseur de L espaces du bord de l'écran et affiche une paroi (I).

(Le point-virgule permet au curseur de ne pas passer à la ligne suivante.)

Déplace le curseur de Y espaces et affiche une étoile.

Déplace le curseur de R espaces et affiche une paroi. (Pas de point-virgule, pour que le curseur passe à la ligne suivante.)

Teste si vous avez appuyé sur une touche.

Si la touche sur laquelle vous appuyez n'est pas Q, l'ordinateur va en 280.

Si Q a été enfoncé, alors Y est diminué de 1 et R augmente de 1. Ainsi l'étoile se déplace vers la gauche.

Teste si la touche pressée est P.

```

290 LET Y=Y+1
300 LET R=R-1
310 IF Y<1 OR R<1 THEN GOTO 370
320 LET S=S+1
330 IF S<M THEN GOTO 100
340 PRINT « SUPERBE-VOUS AVEZ RÉUSSI »
350 PRINT « LA VALLÉE DE LA MORT EST PASSÉE »
360 STOP
370 PRINT « VOUS VOUS ÊTES ÉCRASÉ SUR LA PAROI »
380 PRINT « DÉSINTÉGRÉ »
390 STOP
1000 IF N=0 THEN RETURN
1010 FOR I=1 TO N
1020 PRINT « »;
1030 NEXT I
1040 RETURN

```

Si c'est P, l'étoile est déplacée vers la droite en augmentant de 1 la valeur Y et en diminuant la valeur R.

Teste si vous vous êtes écrasé contre une paroi. Dans ce cas, va en 370.

Augmente le nombre de tours effectués de 1.

Retour pour un nouveau tour si vous y avez encore droit.

Affiche, si vous ne vous êtes pas écrasé pendant les M tours.

Affiche en cas d'échec.

Sous-programme pour déplacer le curseur à l'endroit approprié.

La liste ci-dessus fonctionne sur ZX81. Pour les autres ordinateurs, procédez aux modifications suivantes :

- ▲ 70 LET L=4
- 100 LET D=INT(RND(0)\*3-1)
- ★▲ 100 LET D=INT(RND(1)\*3-1)
- S★■▲ 145
- 235 IS=« »
- ▲ 240 GET IS
- ★ 240 LET IS=INKEY\$(1)
- 240 IF PEEK(-16384)>127 THEN GET IS

## Ralentir le jeu

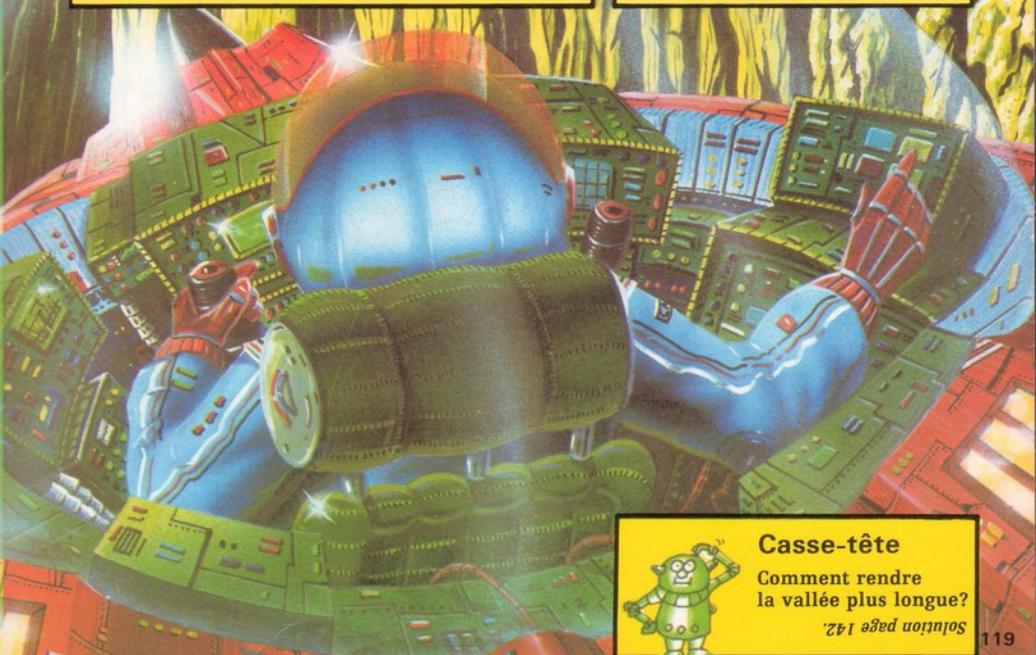
Si le jeu est trop rapide pour vous, vous pouvez ajouter une boucle d'attente aux lignes 141 et 142 :

```

141 FOR J=1 TO 100
142 NEXT J

```

Ajoutez la valeur limite de la boucle à la ligne 141. Plus la valeur est faible plus le jeu est rapide.



## Casse-tête

Comment rendre la vallée plus longue?

Solution page 142.



# Les mines d'Astron

Vous venez d'être élu chef d'une colonie minière sur la planète Astron. Toutes les décisions concernant la vente de minerai aux marchands intergalactiques, les achats et ventes de réserve de nourriture et les recherches de nouvelles mines, vous incombent. Pourrez-vous satisfaire votre peuple pendant votre mandat de dix ans, ou la vie dans la colonie disparaîtra-t-elle dans un terrible désastre sous votre gouvernement?

## Programme

```

10 LET M=INT(RND*3+5)
20 LET P=INT(RND*60+40)
30 LET A=INT(RND*50+10)*P
40 LET PN=INT(RND*40+80)
50 LET QM=INT(RND*40+80)
  
```

Ces lignes décident du nombre de mines M, de l'effectif de la population P, de l'argent dont vous disposez A, du prix de la nourriture PN, du montant total de minerai produit par mine QM pour le début du jeu.

```
60 LET S=0
```

Remet le stock S de minerai à zéro.

```
70 LET C=1
```

Met la cote de popularité C à 1.

```
80 LET Y=1
```

Met l'année du mandat à 1.

```
★▲● 90 LET PM=INT(RND*2000+2000)
```

Fixe le prix de commercialisation d'une mine.

```
★▲● 100 LET VM=INT(RND*12+7)
```

Fixe le prix de vente du minerai.

```
▲● 110 CLS
```

```
120 PRINT « ANNÉE »;Y
```

```
130 PRINT
```

```
140 PRINT « L'EFFECTIF DE LA COLONIE EST DE »;P;
```

```
150 PRINT « VOUS AVEZ »;M;« MINES ET » $ A
```

```
160 PRINT « COTE DE POPULARITÉ »;C
```

```
170 PRINT
```

```
180 PRINT « VOS MINES PRODUISENT »;QM;« TONNES CHACUNE »
```

```
190 LET S=S+QM*M
```

```
200 PRINT « STOCK DE MINERAI= »;S;« TONNES »
```

```
210 PRINT « VENTES »
```

```
220 PRINT « PRIX DE VENTE MINERAI= »;VM
```

```
230 PRINT « PRIX DE VENTE MINE= »;PM;« /MINE »
```

```
240 PRINT « QUANTITÉ DE MINERAI À VENDRE? »
```

```
250 INPUT MV
```

```
260 IF MV<0 OR MV>S THEN GOTO 240
```

```
270 LET S=S-MV
```

État des affaires en cours dans la colonie.

Demande la quantité de minerai que vous désirez vendre et vérifie que c'est possible.

Défalque la vente du stock de minerai.

Calcule ce que rapporte la vente de minerai et en augmente la caisse.

```
280 LET A=A+MV*VM
```

```
290 PRINT « COMBIEN DE MINES VOULEZ-VOUS VENDRE? »
```

```
300 INPUT MS
```

```
310 IF MS<0 OR MS>M THEN GOTO 290
```

```
320 LET M=M-MS
```

```
330 LET A=A+MS*PM
```

```
340 PRINT
```

```
350 PRINT « VOUS AVEZ $ »;A
```

```
360 PRINT
```

Même principe pour la vente des mines.

Affiche l'état de vos liquidités.

```
370 PRINT « ACHATS »
```

```
380 PRINT « COMBIEN VOULEZ-VOUS CONSACRER À L'ALIMENTATION? »
```

```
385 PRINT « ENVIRON 100$ PAR TÊTE »
```

```
390 INPUT AN
```

```
400 IF AN<0 OR AN>A THEN GOTO 380
```

Demande combien vous comptez consacrer à l'achat de nourriture et le porte en AN.

Vérifie si vous avez assez d'argent pour payer vos achats.

Calcule votre nouvel état en liquidités.



420 IF AN/P>120 THEN LET C=C+.1	}	Ajuste la cote de popularité selon les achats de vivres.
430 IF AN/P<80 THEN LET C=C-.2		
440 PRINT « COMBIEN DE MINES VOULEZ-VOUS ACHETER? »	}	Demande combien de mines vous voulez acheter et vérifie que vous en avez les moyens.
450 INPUT AM		
460 IF AM<0 OR AM*PM>A THEN GOTO 440		
470 LET M=M+AM		Augmente le nombre de mines si nécessaire.
480 LET A=A-AM*PM		Liquidités disponibles.
490 IF C<.6 THEN GOTO 660		Teste la cote de popularité. Si celle-ci s'est effondrée, l'ordinateur va en 660 pour la fin du jeu.
★■▲● 500 IF C>1.1 THEN LET QM=QM+INT(RND*20+1)		Si votre popularité est bonne, la production par mine augmente.
★■▲● 510 IF C<.9 THEN LET QM=QM-INT(RND*20+1)		Si la cote est faible, la productivité baisse.
520 IF P/M<10 THEN GOTO 680		S'il y a moins de 10 personnes par mine, le jeu est fini.
★■▲● 530 IF C>1.1 THEN LET P=P+INT(RND*10+1)		Si la cote est bonne, de nouveaux colons arrivent.
★■▲● 540 IF C<.9 THEN LET P=P-INT(RND*10+1)		Si la cote est faible, des colons partent.
550 IF P<30 THEN GOTO 700		S'il reste moins de 30 personnes, le jeu est fini.
★■▲● 560 IF RND>.01 THEN GOTO 590	}	Introduit avec une faible probabilité le risque de perte de la moitié de la population.
570 PRINT « FUITE RADIOACTIVE... NOMBREUX DÉCÈS »		
580 LET P=INT(P/2)	}	Si la production est trop importante, les prix sont diminués de moitié.
590 IF QM<150 THEN GOTO 620		
600 PRINT « MARCHÉ ENGORGÉ-LES PRIX CHUTENT »		
610 LET QM=INT(QM/2)		Une année de plus comptabilisée.
620 LET Y=Y+1		Si le mandat n'expire pas, retour en 90.
630 IF Y<11 THEN GOTO 90		
640 PRINT « VOUS AVEZ SURVÉCU À VOTRE MANDAT »		Affiche si l'ordinateur arrive là après dix ans.
650 STOP		
660 PRINT « LE PEUPLE SE RÉVOLTE »		
670 STOP		
680 PRINT « VOUS AVEZ ÉPUISÉ TOUT LE MONDE »		
690 STOP		
700 PRINT « IL NE RESTE PAS ASSEZ DE MONDE »		
710 STOP		

La liste ci-dessus fonctionne sur un ZX81. Pour les autres ordinateurs, procédez aux modifications suivantes :

- 110 HOME
- ▲ 110 PRINT CHR\$(147)
- ★▲● 10,20,30,40,50,90,100,500,510,530,540,560, remplacez RND par RND(1)
- 10,20,30,40,50,90,100,500,510,530,540,560, remplacez RND par RND(0)



### Casse-tête

Pourriez-vous modifier le programme de façon à ce qu'il vous demande si vous désirez un autre mandat, tout en reportant au début de la nouvelle partie les valeurs avec lesquelles vous aviez terminé votre précédent mandat.

# Un sauvetage spatial

Vous devez effectuer un voyage urgent à travers un bras spiralé de la Galaxie pour rejoindre une planète en développement qui a besoin d'aide médicale. Le voyage est tellement long que vous serez la plupart du temps en état d'hibernation. Mais avant de tomber dans ce sommeil profond, vous programmez le vaisseau pour le trajet désiré. L'ordinateur de bord vous demandera quelles quantités d'énergie vous comptez donner aux moteurs, au système de survie et aux boucliers, puis vous vous endormirez. Quand vous vous réveillerez, il vous indiquera ce qui ce sera passé pendant le voyage et si tout s'est bien déroulé. Votre vaisseau orbitera autour de la planète. Vous devrez alors commander l'atterrissage pour effectuer une bonne arrivée au sol.

Si vous accomplissez la mission sans encombres, vous aurez une bonne chance d'être promu amiral de la flotte spatiale. Bonne chance !

```
▲● 10 CLS
    20 PRINT «SAUVETAGE SPATIAL »
    30 PRINT
    40 PRINT « DÉSIREZ-VOUS DES RENSEIGNEMENTS?(O-N) »
    50 INPUT IS
★▲● 60 IF IS(1) = « O » THEN GOSUB 1000
★▲● 70 LET D=INT(RND*800+101)
★▲● 80 LET E=INT(RND*400+401)
    90 LET T=INT(D/SQR(E/5)+.5)
    100 PRINT « LA PLANÈTE EST À »;D;« ANNÉES-LUMIÈRE »
    110 PRINT « VOUS DISPOSEZ DE »;E;« UNITÉS D'ÉNERGIE »
    120 PRINT « ET UNE DURÉE LIMITE DE »;T;« JOURS »
    130 PRINT
    140 PRINT « RÉPARTITION DE L'ÉNERGIE : »
    150 PRINT « POUR LES MOTEURS? »
    160 INPUT P
    170 PRINT « AU SYSTÈME DE SURVIE? »
    180 INPUT L
    190 PRINT « AUX BOUCLIER? »
    200 INPUT S
    210 IF P+L+S>E THEN GOTO 140
    220 LET X=E-P-L-S
    230 LET V=INT(SQR(P))
    240 LET T1=INT(D/V)
▲● 250 CLS
    260 PRINT « VOTRE VITESSE EST DE »;V
    270 PRINT « UNE ÉTAPE DE »;T1;« JOURS »
    280 PRINT
★▲● 290 FOR I=1 TO INT(RND*5+6)
★▲● 300 IF RND>.5 THEN GOTO 430
★▲● 310 GOTO 320+INT(RND*4)*30
    320 PRINT « PLUIE D'ASTÉROÏDES - BOUCLIER ENDOMMAGÉS »
```

```

*■▲● 330 LET S=S-20-INT(RND*40+1)
340 GOTO 430
350 PRINT « PANNE D'ORDINATEUR -
RETARD POUR RÉPARATIONS »
*■▲● 360 LET D=D+INT(RND*20+1)
370 GOTO 430
380 PRINT « PROBLÈME MOTEUR -
OBLIGÉ DE RALENTIR »
390 LET V=V-.5
400 GOTO 430
410 PRINT « IRRADIATION X-SYSTÈME DE SURVIE ATTEINT »
420 LET L=L-20-INT(RND*40+1)
*■▲● 430 FOR J=1 TO 50
440 NEXT J
450 NEXT I
460 LET T1=INT(D/V)
●▲ 470 CLS
480 PRINT « ARRIVÉE EN »,T1,« JOURS »
490 IF S<0 THEN PRINT « BOUCLIER DÉTRUIT »
495 IF S<0 THEN PRINT « VOUS AVEZ ÉTÉ PULVÉRISÉS »
500 IF L<=0 THEN PRINT « SYSTÈME DE SURVIE HORS D'USAGE »
505 IF L<=0 THEN PRINT « VOUS ÊTES MORT »
510 IF V<=0 THEN PRINT « MOTEURS ARRÊTÉS »
520 IF T1>T THEN PRINT « VOUS ÊTES ARRIVÉS TROP TARD »
530 IF S<0 OR L<=0 OR V<=0 OR T1>T THEN STOP
*■▲● 540 LET G=INT(RND*10+5)
550 LET G$=« FORTE »
560 IF G<12 THEN LET G$=« MOYENNE »
570 IF G<8 THEN LET G$=« FAIBLE »
*■▲● 580 LET A=INT(RND*10+5)
590 LET A$=« DENSE »
600 IF A<12 THEN LET A$=« NORMALE »
610 IF A<8 THEN LET A$=« LÉGÈRE »
620 PRINT
630 PRINT « VOUS ORBITEZ AUTOUR DE LA PLANÈTE »
640 PRINT « RÉSERVE ÉNERGIE= »,X
650 PRINT « LA GRAVITÉ EST »,G$
660 PRINT « L'ATMOSPHÈRE EST »,A$
670 PRINT
680 PRINT « QUANTITÉ ÉNERGIE POUR LES RÉACTEURS? »
690 INPUT B
700 PRINT « QUANTITÉ ÉNERGIE POUR LES BOUCLIER À CHALEUR? »
710 INPUT S
720 IF B+S>X THEN GOTO 680
●▲ 730 CLS
▲● 740 IF B>=G*10 THEN GOTO 770

```

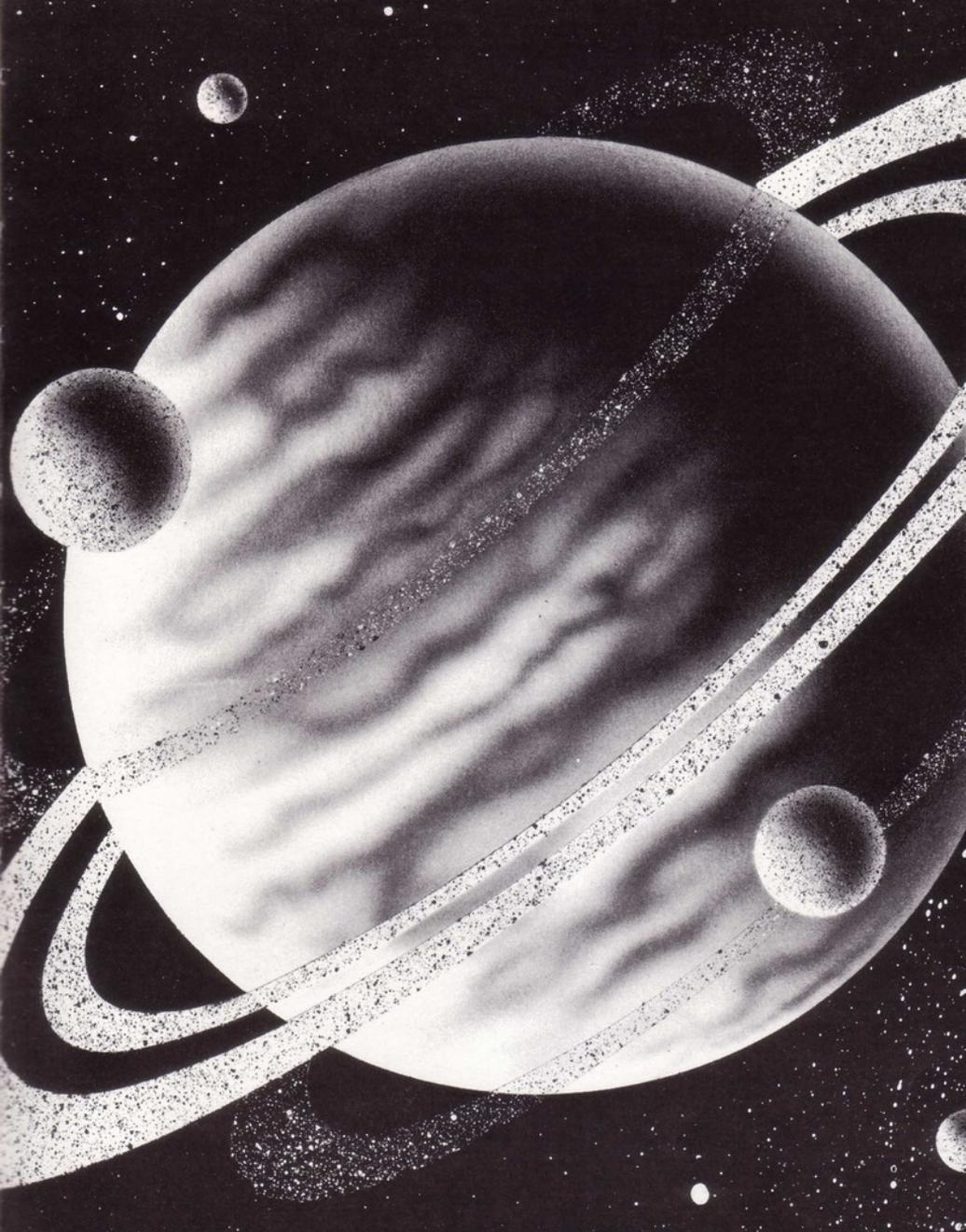
*Le programme continue à la page suivante.*

## Suite du programme Sauvetage Spatial

```
750 PRINT « VOUS AVEZ CREUSÉ UN NOUVEAU CRATÈRE »
760 GOTO 840
770 IF S<=A*10 THEN GOTO 800
780 PRINT « VOUS FAITES UNE TRÈS BELLE COMÈTE »
790 GOTO 840
800 PRINT « VOUS AVEZ RÉUSSI À ATERRIR - BRAVO ! »
810 IF X-S-B>25 THEN GOTO 840
820 PRINT « DOMMAGE : VOUS N'AVEZ PAS ASSEZ D'ÉNERGIE »
830 PRINT « POUR OUVRIR LA PORTE »
840 STOP
1000 PRINT
1010 PRINT « VOUS ÊTES SUR LE POINT D'EMBARQUER POUR »
1020 PRINT « UNE MISSION SUR UNE PLANÈTE LOINTAINE »
1030 PRINT « QUI A UN BESOIN URGENT D'AIDE MÉDICALE »
1040 PRINT « VOUS DEVEZ TOUT D'ABORD PROGRAMMER »
1050 PRINT « VOTRE VAISSEAU POUR LE VOYAGE EN DESTINANT »
1060 PRINT « UNE PARTIE DES RÉSERVES ÉNERGÉTIQUES DE »
1070 PRINT « L'APPAREIL AUX MOTEURS, AUX BOUCLIERS ET »
1080 PRINT « AU SYSTÈME DE SURVIE. VOUS HIBERNEZ ENSUITE »
1090 PRINT « PENDANT LA MAJEURE PARTIE DU TRAJET »
1100 PRINT « A L'ARRIVÉE L'ORDINATEUR VOUS COMMUNIQUE »
1110 PRINT « SON RAPPORT SUR LES PROBLÈMES DE ROUTE »
1120 PRINT « IL NE VOUS RESTE PLUS QU'À VOUS POSER »
1130 PRINT « SUR LE SOL DE LA PLANÈTE... »
1140 PRINT « APPUYEZ SUR UNE TOUCHE »
★▲● 1150 IF INKEY$= « » THEN GOTO 1150
▲● 1160 CLS
1170 RETURN
```

La liste ci-dessus fonctionne sur un ZX81.  
Pour les autres ordinateurs, procédez aux  
modifications suivantes :

- RND devient RND(0)
- ▲★ RND devient RND(1)
  - 10,250,470,730,1160 HOME
  - ▲ 10,250,470,1160 PRINT CHR\$(147)
- ▲★ 60 IF LEFT\$(I\$,1) = « O » THEN GOSUB 1000
- ▲★ 310 ON INT(RND\*4+1) GOTO 320,350,380;410
- ▲ 430 FOR J=1 TO 500
  - ★ 430 FOR J=1 TO 1000
- ★ 1150 I=GET
- 1150 GET IS
- ▲ 1150 GET IS: IF IS=« » THEN GOTO 1150



### **Ajouts au programme**

Ce jeu est en fait composé de deux parties. La première, où vous programmez votre voyage pour parvenir à orbiter autour de la planète. La seconde, où vous essayez d'atterrir. Vous pourriez peut-être en ajouter une troisième où la tâche du joueur consisterait à rallier, en traversant une région traître, le quartier général de la Croix Rouge intergalactique.

# Alunissage: version TRS 80

Ce jeu diffère des précédents, car il utilise des instructions graphiques. Comme les ordinateurs sont très différents dans leurs modes graphiques, un programme est proposé pour chaque type de matériel. Lisez tout d'abord les renseignements de cette page, puis cherchez plus loin la version qui correspond à votre machine.

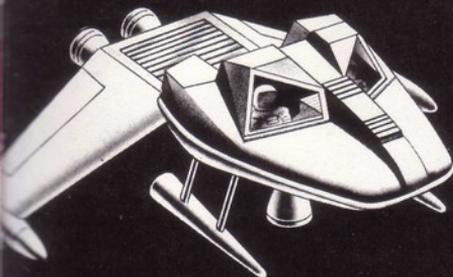
## Comment prendre contact avec le sol

L'as de l'espace, le capitaine Flash, est assis à côté de vous pour vous faire passer le troisième degré de votre certificat de pilotage d'engins spatiaux. Le module lunaire biplace d'atterrissage s'approche rapidement de la surface de la Lune. La vitesse doit être aussi proche que possible de zéro à l'arrivée au sol. Vous devez contrôler avec dextérité la poussée en appuyant sur A pour l'augmenter et sur D pour la diminuer\*, tout en observant à chaque instant votre progression sur l'écran. Si vous donnez trop de puissance, vous remonterez; pas assez, il y aura un nouveau cratère à la surface de la Lune. Pourrez-vous impressionner le capitaine Flash par votre maîtrise?

\* Pour le VIC, utilisez la touche de descente du curseur pour augmenter la valeur de la poussée et celle de déplacement du même curseur pour la diminuer.

## Version du programme pour TRS-80

```
20 CLS
30 CLEAR 200
31 BS=STRING$(25,131)
33 M1$=« »+STRING$(2,176)
34 M2$=« »+STRING$(4,191)
35 M3$=CHRS(131)+CHRS(135)+STRING$(2,131)+CHRS(139)+CHRS(131)
40 GOSUB 250
50 GOSUB 300
60 C=1:GOSUB 390
70 A=1:B=F:GOSUB 460
80 A=2:B=ABS(V):GOSUB 460
90 A=3:B=H:GOSUB 460
100 A=4:B=T:GOSUB 460
110 GOSUB 530
120 V1=V-T/20+G : F=F-T/10
130 H1=H-(V+V1)/10
140 C=0:GOSUB 390
150 IF H1<0 THEN 200
160 H=H1:V=V1
170 IF H<=100 THEN 60
180 GOSUB 590
190 GOTO 220
200 H=0:C=1:GOSUB 390
210 GOSUB 660
220 END
250 H=100:F=100:T=0
260 V=INT(RND(0)*10+6)
270 G=INT(RND(0)*40+41)/100
280 RETURN
300 FOR X=80 TO 127
320 SET (X,47-INT(RND(0)*5))
330 NEXT
340 PRINT « GRAVITÉ= »:G
350 PRINT@192,« FUEL: »
355 PRINT@384,« VITESSE: »
360 PRINT@576,« ALTITUDE: »
365 PRINT@768,« POUSSÉE: »
370 RETURN
390 Y=818-64*INT(H/8)
400 PRINT Y,;:IF C=1 THEN PRINT M1$;ELSE
PRINT STRING$(6,32);
410 PRINT Y+64,; :IF C=1 THEN PRINT M2$
;ELSE PRINT STRING$(6,32);
420 PRINT Y+128,; :IF C=1 THEN PRINT M3$
;ELSE PRINT STRING$(6,32);
440 RETURN
460 Y=(A*3+1)*64
470 PRINT Y,STRING$(25,32);
480 PRINT@Y,LEFT$(BS,B/4);
510 RETURN
530 IS=INKEYS
540 IF IS="A" THEN LET T=T+4:IF T>100 THEN
LET T=100
550 IF IS="D" THEN LET T=T-4 :IF
T<0 THEN LET T=0
560 IF T>F THEN LET T=F
570 RETURN
590 CLS
600 FOR I=1 TO 20
610 PRINT@INT(RND(0)*1024),« * »
620 NEXT
630 PRINT 470, "PERDU DANS L'ESPACE!!"
640 RETURN
650 CLS
660 PRINT « VITESSE AU SOL »:
INT((V+V1)*5)/10
670 IF (V+V1)<8 THEN PRINT « SAIN ET SAUF »
ELSE PRINT « MORT! »
680 RETURN
```



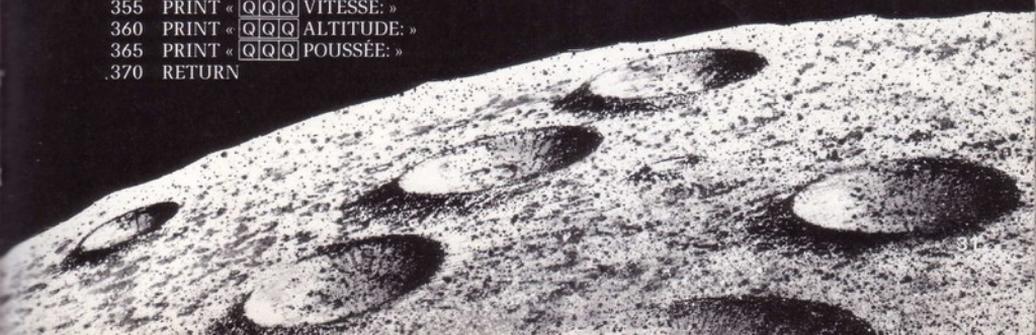
## Version du programme pour VIC 20

```
20 PRINT CHR$(147)CHR$(5);
25 POKE 36879,8
30 DEF FNR(X)=INT(RND(1)*X+1)
40 GOSUB 250
50 GOSUB 300
60 C=1:GOSUB 390
70 A=1:B=F:GOSUB 460
80 A=2:B=ABS(V):GOSUB 460
90 A=3:B=H:GOSUB 460
100 A=4:B=T:GOSUB 460
110 GOSUB 530
120 V1=V-T/20+G: F=F-T/10
130 H1=H-(V+V1)/10
140 C=0:GOSUB 390
150 IF H1<0 THEN 200
160 H=H1:V=V1
170 IF H<=100 THEN 60
180 GOSUB 590
190 GOTO 220
200 H=0:C=1:GOSUB 390
210 GOSUB 660
220 END
250 H=100:F=100:T=0
260 V=5+FNR(10)
270 G=(FNR(40)+40)/100
280 RETURN
300 FOR X=8178 TO 8185
320 POKE X,98+2*FNR(3)
330 NEXT
340 PRINT « GRAVITÉ= »;G
350 PRINT «  FUEL: »
355 PRINT «  VITESSE: »
360 PRINT «  ALTITUDE: »
365 PRINT «  POUSSÉE: »
370 RETURN
```

```
390 Y=8137-22*INT(H/5)
400 IF C=0 THEN 425
405 POKE Y,108 : POKE Y+1,123
410 POKE Y+22,160 : POKE Y+23,160
415 POKE Y+44,75 : POKE Y+45,74
420 GOTO 440
425 FOR Z=0 TO 44 STEP 22
430 POKE Y+Z,32 : POKE Y+Z+1,32
435 NEXT
440 RETURN
460 FOR X=0 TO 9
470 Y=A*88+X+7724
480 IF X<B/10 THEN POKE Y,102 :
GOTO 500
485 IF X<B/10+.5 THEN POKE Y,92 :
GOTO 500
490 POKE Y,32
500 NEXT
510 RETURN
530 GET IS
540 IF IS="  " THEN T=T+4 : IF T>100
THEN T=100
550 IF IS="  " THEN T=T-4 : IF T<0
THEN T=0
560 IF T>F THEN T=F
570 RETURN
590 PRINT CHR$(147)
600 FOR I=1 TO 20
610 POKE 7679+FNR(506),42
620 NEXT
630 PRINT « PERDUS DANS L'ESPACE!! »
640 RETURN
650 PRINT CHR$(147) « ALUNI »
660 PRINT « VITESSE AU SOL »;INT((V+V1)*5)/10
670 IF (V+V1)<8 THEN PRINT « SAIN ET SAUF »;
RETURN
680 PRINT « MORT... »;RETURN
```

 est la touche de descente du curseur.

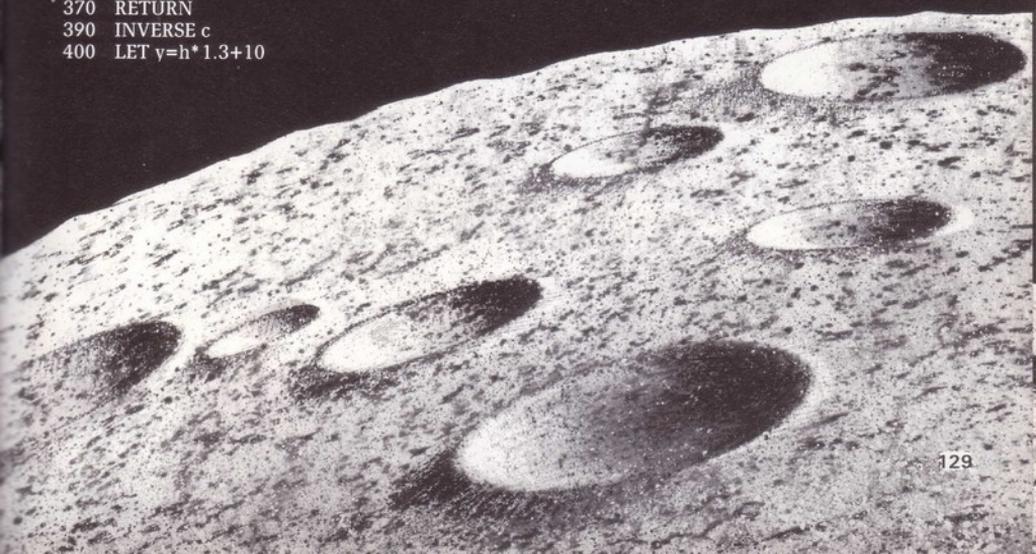
 est la touche de déplacement  
du curseur vers la droite.





## Version du programme pour ZX spectrum

```
20 CLS
30 DEF FNr (x)=INT(RND*x+1)
40 GOSUB 250
50 GOSUB 300
60 LET c=0: GOSUB 390
70 LET a=1: LET b=f: LET c=2*(f<25)
75 GOSUB 460
80 LET a=2: LET b=ABS v:
  LET c=4*(v<0)
85 GOSUB 460
90 LET a=3: LET b=h: LET c=2*(h<25)
95 GOSUB 460
100 LET a=4: LET b=t: LET c=0
105 GOSUB 460
110 GOSUB 530
120 LET v1=v-t/20+g: LET f=f-t/10
130 LET h1=h-(v+v1)/10
140 LET c=1: GOSUB 390
150 IF h1<0 THEN GOTO 200
160 LET h=h1: LET v=v1
170 IF h<=100 THEN GOTO 60
180 GOSUB 590
190 GOTO 220
200 LET h=0: LET c=0: GOSUB 390
210 GOSUB 660
220 STOP
250 LET h=100: LET f=100: LET t=0
260 LET v=5+FNr(10)
270 LET G=(FNr(40)+40)/100
280 RETURN
300 PLOT 180,8
310 FOR x=1 TO 15
320 DRAW 5, FNr(3)-2
330 NEXT x
340 PRINT « GRAVITÉ= »;g
350 PRINT" « FUEL: »"" « VITESSE: »
360 PRINT" « ALTITUDE: »"" « POUSSÉE: »
370 RETURN
390 INVERSE c
400 LET v=h*1.3+10
410 PLOT 200,y: DRAW 34,0
420 DRAW -4,20: DRAW -13,10
430 DRAW -13,-10: DRAW -4,20
440 RETURN
460 LET y=172-a*32
470 INK c
480 PLOT 0,y
490 DRAW b,0
500 DRAW INVERSE 1,100-b,0
510 RETURN
530 LET i$=INKEYS
540 IF i$=« a » THEN LET t=t+4 :
  IF t>100 THEN LET t=100
550 IF i$=« d » THEN LET t=t-4 :
  IF t<0 THEN LET t=0
560 IF t>f THEN LET t=f
570 RETURN
590 CLS
600 FOR i=1 TO 20
610 PRINT AT FNr(21),FNr(31);« * »
620 NEXT i
630 PRINT « PERDU DANS L'ESPACE!!! »
640 RETURN
650 PRINT AT 0,0;« ALUNI A »;
  INT((v+v1)*5)/10'
660 IF (v+v1)<8 THEN GOTO 680
670 PRINT « MORT... »: RETURN
680 PRINT « SAIN ET SAUF »: RETURN
```

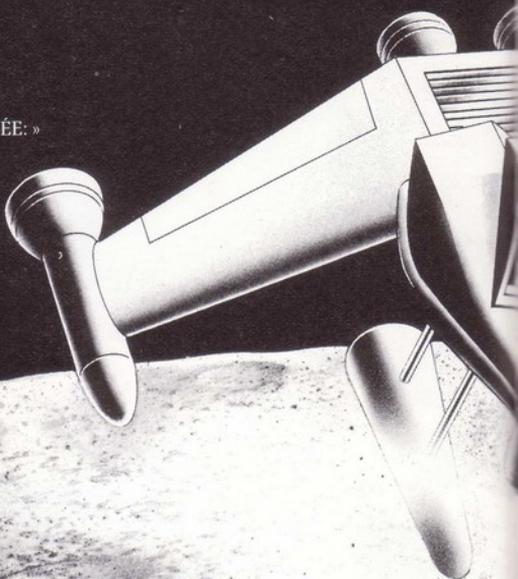


## Version du programme pour BBC-ACORN

```

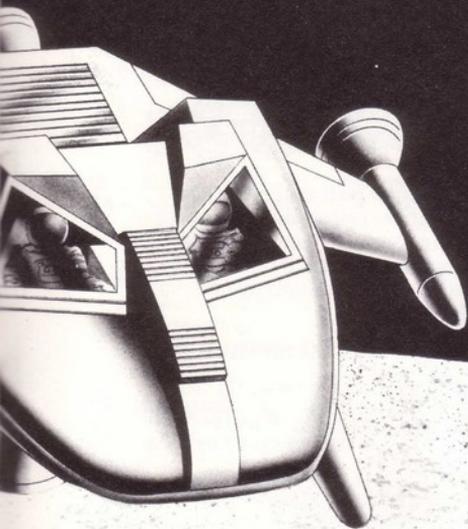
20  MODE 5
30  *FX 12,1
40  PROCSETVAR
50  PROCAFFICHAGE
60  PROCMODULE (H,3)
70  PROCBAR (1,F,3+2* (F<25))
80  PROCBAR (2,ABSV,3+(V<0))
90  PROCBAR (3,H,3+2* (H<25))
100 PROCBAR (4,T,3)
110 PROCPOUSSEE
120 V1=V-T/20+G : F=F-T/10
130 H1=H-(V+V1)/10
140 PROCMODULE (H,0)
150 IF H1<0 THEN 200
160 H=H1:V=V1
170 IF H<=100 THEN 60
180 PROCERDU
190 GOTO 220
200 PROCMODULE (0,2)
210 PROCALUNI
220 *FX 12
230 END
240 DEF PROCSETVAR
250 H=100:F=100:T=0
260 V=5+RND(10)
270 G=(RND(40)+40)/100
280 ENDPROC
290 DEF PROCAFFICHAGE
300 MOVE 800,30
310 FOR X=800 TO 1280 STEP 16
320 DRAW X,10+RND (40)
330 NEXT
340 PRINT « GRAVITÉ= »;G
350 PRINT' ' « FUEL: »' ' « VITESSE: »
360 PRINT' ' « ALTITUDE: »' ' « POUSSÉE: »
370 ENDPROC
380 DEF PROCMODULE (H,C)
390 GCOL 0,C
400 Y=H*8.5+150
410 MOVE 1040,Y : PLOT 1,-40,-40
420 PLOT 1,-8,-60 : PLOT 1,96,0
430 PLOT 1,-8,60 : PLOT 1,-40,40
440 ENDPROC
450 DEF PROCBAR(N,V,C)
460 Y=1000-192*N
470 GCOL 0,C
480 MOVE 0,Y : MOVE 0,Y-16
490 PLOT 85,V*4,Y : PLOT 85,V*4,Y-16
500 PLOT 87,400,Y : PLOT 87,400,Y-16
510 ENDPROC
520 DEF PROCPOUSSEE
530 *FX 15,1
540 IF INKEY(-194) THEN T=T+4 :
    IF T>100 THEN T=100
550 IF INKEY(-179) THEN T=T-4 :
    IF T<0 THEN T=0
560 IF T>F THEN T=F
570 ENDPROC
580 DEF PROCERDU
590 CLS
600 FOR I=1 TO 20
610 VDU 31,RND(19),RND(31),42
620 NEXT
630 PRINT TAB(4,16) « PERDU DANS L'ESPACE!! »
640 ENDPROC
650 DEF PROCALUNI
660 VDU 28,0,31,11,0,12
670 PRINT' « ALUNI »
680 PRINT' « VITESSE »;INT ((V+V1)*5)/10
690 IF (V+V1)<8 THEN PRINT' « SAIN ET SAUF »
    ELSE PRINT' « MORT! »
700 ENDPROC

```



## Version du programme pour APPLE

```
15 HOME
20 HGR
30 DEF FNR(X)=INT(RND(1)*X+1)
40 GOSUB 250
50 GOSUB 300
60 C=3:GOSUB 390
70 A=1:B=F:GOSUB 460
80 A=2:B=ABS(V):GOSUB 460
90 A=3:B=H:GOSUB 460
100 A=4:B=T:GOSUB 460
110 GOSUB 530
120 V1=V-T/20+G : F=F-T/10
130 H1=H-(V+V1)/10
140 C=0:GOSUB 390
150 IF H1<0 THEN 200
160 H=H1:V=V1
170 IF H<=100 THEN 60
180 GOSUB 590
190 GOTO 220
200 H=0:C=3:GOSUB 390
210 GOSUB 660
220 END
250 H=100:F=100:T=0
260 V=5+FNR(10)
270 G=(FNR(40)+40)/100
280 RETURN
300 HCOLOR=3
305 HPLLOT 0,155
310 FOR X=0 TO 279 STEP 5
320 HPLLOT TO X,159-FNR(10)
330 NEXT
335 FOR I=1 TO 30 :HPLLOT FNR(279),
FNR(150)
337 NEXT
340 VTAB 21 : PRINT TAB(34);«G=»;G
350 VTAB 21:PRINT « FUEL: »: PRINT « VITESSE: »
360 PRINT « ALTITUDE: » : PRINT « POUSSÉE: »;
370 RETURN
390 HCOLOR=C
400 Y=(100-H)*1.3
410 HPLLOT 140,Y TO 120,Y+10
420 HPLLOT TO 120,Y+20 :
HPLLOT TO 160,Y+20
430 HPLLOT TO 160,Y+10 : HPLLOT TO 140,Y
435 HPLLOT 155,Y+20 TO 160,Y+25
437 HPLLOT 125,Y+20 TO 120,Y+25
440 RETURN
460 VTAB (20+A) : HTAB 8
470 INVERSE
480 PRINT SPC(B/4);
490 NORMAL
500 PRINT SPC(26-B/4);
510 RETURN
530 IS=« »: IF PEEK(-16384)>127
THEN GET IS
540 IF IS="A" THEN T=T+4 :
IF T>100 THEN T=100
550 IF IS="D" THEN T=T-4 :
IF T<0 THEN T=0
560 IF T>F THEN T=F
570 RETURN
590 HOME : VTAB 23
600 PRINT « PERDU DANS L'ESPACE!! »
640 RETURN
660 HOME : VTAB 22
670 PRINT « ALUNI À LA VITESSE DE »;
INT((V+V1)*5)/10
680 IF (V+V1)<8 THEN 700
690 PRINT « MORT... »: RETURN
700 PRINT « SAIN ET SAUF » : RETURN
```



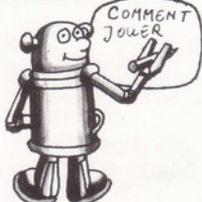
# Quelques trucs pour aller plus loin



Voici quelques idées pour compléter les programmes de ce livre ou vos propres programmes. Dans la plupart des cas vous ne pourrez pas les utiliser pour le ZX81, puisque les jeux occupent à eux seuls la totalité de sa mémoire (1K) ; mais ce n'est guère le cas pour les autres micro-ordinateurs.

Attention : vous serez peut-être limité par les espacements laissés libres entre les numéros de lignes de programmes. N'hésitez pas à renuméroter la liste sans oublier de modifier également les numéros de lignes auxquels renvoient les instructions GOTO et GOSUB.

Pour chacun des programmes, l'ordinateur peut afficher une série d'informations



vous rappelant le mode d'emploi du jeu. Il suffit de rajouter quelques lignes au début de la liste et de placer le sous-programme à la fin.

```
10 PRINT "TITRE DU JEU"  
11 PRINT "VOULEZ-VOUS CONNAITRE"  
12 PRINT "LE MODE D'EMPLOI ?"  
15 INPUT IS
```

```
S ZX17 IF IS (1) = "0" THEN GOSUB 1000  
* ■▲●17 IF LEFT$(IS,1) = "0" THEN GOSUB 1000
```

Emplacement du programme principal

```
1000 PRINT "MODE D'EMPLOI"  
1010 PRINT "....."  
1999 RETURN
```

*A priori*, vous n'êtes pas limité par le nombre des écritures. Numérotez bien chaque ligne, frappez PRINT et vérifiez que le nombre de caractères entre guillemets ne dépasse pas les possibilités d'affichage de votre ordinateur.

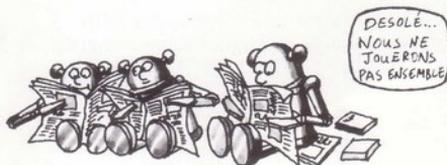
L'instruction RETURN doit obligatoirement figurer à la fin du sous-programme. Dans le cas contraire, il ne fonctionnerait pas.

## L'ordinateur peut s'arrêter et vous attendre

Si vos lignes d'instructions sont particulièrement nombreuses et que vous ne voulez pas les voir disparaître prématurément de l'écran, le sous-programme que nous vous proposons maintenant présente un réel intérêt : il arrête le déroulement du programme en un point défini tant qu'une touche du clavier n'a pas été frappée. Insérez l'instruction GOSUB à l'endroit choisi et écrivez ce sous-programme :

```
1000 PRINT "POUR CONTINUER"  
1005 PRINT "FRAPPEZ UNE TOUCHE"  
S ZX1010 IF INKEY$="" THEN GOTO 1010  
★1010 GET IS  
●1010 GET IS : IF IS : "" THEN GOTO 1010  
▲1020 PRINT  
1030 RETURN
```

## L'ordinateur vous parle



L'ordinateur peut vous poser des questions et agir, ensuite, en fonction de votre réponse. Dans le premier exemple ci-dessous, il refusera de jouer avec ceux dont l'initiale du prénom ne sera pas J.

```
1 PRINT "QUEL EST TON PRENOM ?"  
2 INPUT IS  
S ZX3 IF IS(1) <> "J" THEN GOTO 1000  
* ■▲●3 IF LEFT$(IS,1) <> "J" THEN GOTO 1000  
4 PRINT "VOUS POUVEZ JOUER"  
5 PRINT "ETES-VOUS PRET ?"  
6 INPUT JS  
S ZX7 IF JS(1) <> "0" THEN GOTO 5  
* ■▲●7 IF LEFT$(JS,1) <> "0" THEN GOTO 5
```

Programme principal

```
1000 PRINT "DESOLE, CE JEU"  
1010 PRINT "EST RESERVE A CEUX"  
1020 PRINT "DONT LE PRENOM"  
1030 PRINT "COMMENCE PAR J"  
1040 STOP
```

Dans ce deuxième exemple, l'ordinateur vous met au défi de le braver.

```

10 PRINT "CE JEU EST EFFRAYANT"
12 PRINT "SEREZ-VOUS ASSEZ BRAVE"
14 PRINT "POUR ATTAQUER LE MONSTRE"
15 PRINT "VERT ET POILU ?"
16 INPUT IS
S ZX17 IF IS (1) = "0" THEN GOTO 20
★ ■ ▲ ● 17 IF LEFT$(IS, 1) = "0" THEN GOTO 20
18 PRINT "LACHE"
19 STOP

```

Pour combiner ces deux exemples, il vous faudra renuméroter la série d'instructions données aux pages précédentes (lignes 11 à 17 à transformer en lignes 20-26). Le programme principal démarrera en 30 et sera suivi du ou des sous-programmes existants.

## Voulez-vous jouer encore une fois ?

Pour ne pas avoir à relancer le programme après chaque jeu (RUN), si vous ajoutez les lignes suivantes juste avant l'instruction de fin de programme (STOP ou END), l'ordinateur vous proposera de rejouer avec lui.

```

1000 PRINT "UN AUTRE TOUR ?"
1010 INPUT IS
S ZX1020 IF IS (1) = "0" THEN RUN
★ ■ ▲ ● 1020 IF LEFT$(IS, 1) = "0" THEN RUN
1030 PRINT "Bon - Salut..."
1040 STOP

```

Modifiez la numérotation de cet exemple en fonction de votre programme.



## Pour produire des effets sonores

BBC, VIC20, ZX Spectrum et certains Apples peuvent produire des effets sonores.

Placez les instructions correspondantes aux endroits appropriés pour recréer le bruit d'une explosion ou bien jouer un « petit air » en cas de victoire. Dans la mesure où

chaque ordinateur fait appel à des instructions différentes, reportez-vous au manuel de votre appareil. Dans certains cas une seule ligne d'instructions suffira\*, dans d'autres cas il faudra créer un sous-programme. Placez GOSUB à l'endroit voulu, numérotez chaque ligne du sous-programme et placez RETURN à la fin. Dans les dernières pages du manuel VIC vous trouverez quelques exemples destinés à produire des effets sonores tels que le « rayon laser », l'« explosion » et l'« alerte rouge ».

\* Voici, par exemple, l'effet « coup de feu » sur BBC ; trouvez le meilleur emplacement possible dans votre programme et numérotez la ligne d'instructions :

SOUND 0, - 15, 5, 10

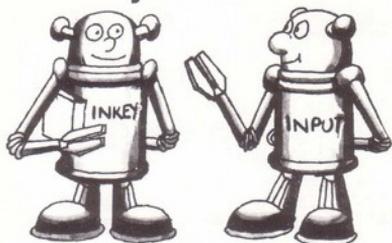


## Note aux utilisateurs de BBC et ZX Spectrum

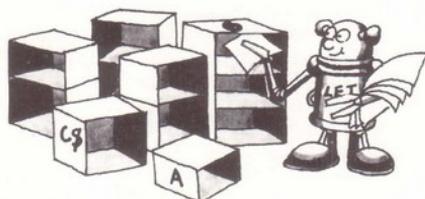
Utilisateurs de BBC et ZX Spectrum, vous trouverez que certains des jeux présentés dans cet ouvrage ont un déroulement trop rapide. C'est pourquoi des encadrés vous indiquent à chaque fois comment en modifier la vitesse d'exécution. Pour ralentir, utilisez des nombres plus grands. Les derniers modèles BBC, deux fois plus rapides que leurs aînés, rendent ces jeux impossibles à utiliser tels quels. Il faudra intervenir de façon importante sur le nombre conditionnant la vitesse d'exécution des programmes.

# La programmation est un jeu d'enfant

Après avoir épuisé toutes les possibilités de modification de programme que nous vous proposons (ou que vous aurez imaginées par vous-même), vous serez probablement lassé des jeux figurant dans ce livre. Ce sera le moment de commencer à écrire vos propres jeux. Ces deux pages vous y aideront. Mais, avant toute chose, ayez présentes à l'esprit les principales possibilités de votre ordinateur.



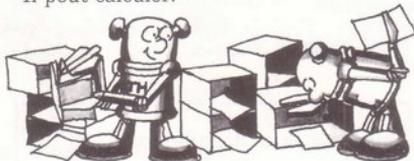
Il peut poser des questions.



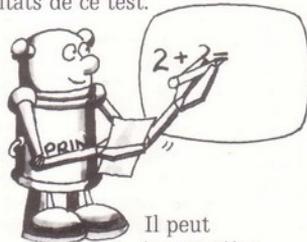
Il peut stocker des informations.



Il peut calculer.



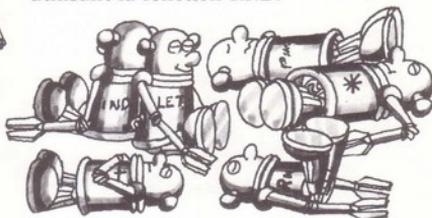
Il peut réaliser un test de comparaison d'informations et agir en fonction des résultats de ce test.



Il peut transmettre les résultats de calculs, tests de comparaison et également le contenu de sa mémoire.



Il peut tirer un nombre au hasard en utilisant la fonction RND.



Il ne peut rien faire que vous ne lui ayez communiqué.



A condition d'employer le bon langage, il fait très exactement ce qu'on lui demande, même si cela est absurde.

Se rappeler, lorsqu'on écrit un jeu, qu'il ne faut rien prévoir que l'ordinateur ne sache faire.

## Planifiez un jeu

Avant de communiquer quoi que ce soit à votre micro-ordinateur, il vous faudra parfaitement maîtriser le déroulement du jeu et de ses règles. Votre ordinateur a besoin d'instructions simples et logiques. Élaborez d'abord le jeu dans votre tête, ou mieux, sur une feuille de papier, puis décomposez-le en grandes étapes.

Réalisez ensuite le plan (en français, pas encore en BASIC) des différentes phases du jeu.

Voici, à titre d'exemple, le plan d'un jeu de tir, où il peut s'agir d'envoyer des boulets sur un bateau pirate ou des rayons laser sur un envahisseur.

### PLAN

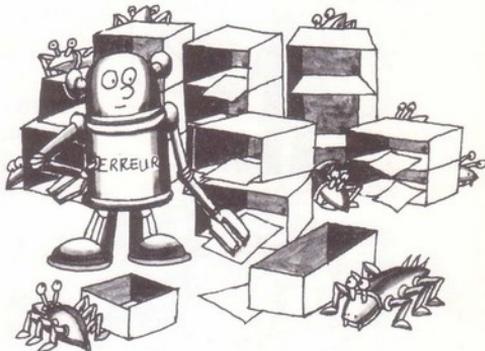
- 1) Donner un titre et les principales informations du jeu.
- 2) Définir une cible, un objet.
- 3) Début d'une boucle pour donner N coups au joueur.
- 4) Le joueur tire.
- 5) Vérifier si le tir était dans la cible.
- 6) Afficher un message correspondant à la précision du tir.
- 7) Nouvel essai en cas d'échec.

## Écrivez votre programme en BASIC

Convertissez ensuite ce plan en BASIC, chacune des étapes nécessitant plusieurs lignes d'instructions.

Numérotez systématiquement ces lignes en veillant à laisser des espaces suffisants pour permettre corrections et additions ultérieures.

Une fois le premier brouillon réalisé sur le papier vous entrerez progressivement le jeu dans votre ordinateur. Celui-ci repérera les erreurs beaucoup plus rapidement que vous ne le feriez vous-même et vous indiquera dans de nombreux cas leur origine. Rappelez-vous que la mise au point d'un programme est un travail long et fastidieux, n'espérez pas réussir du

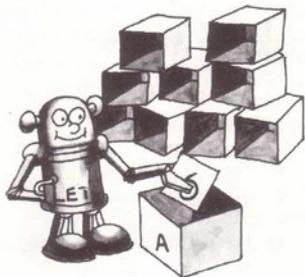


premier coup. Quand la structure de base du programme fonctionnera convenablement vous pourrez y ajouter : le score, des instructions et informations complémentaires, d'autres cibles ou bien des éléments de programmes issus de ce livre. Tout ceci se fera peu à peu.

Les premiers programmes que vous réaliserez ne seront ni très originaux, ni très excitants ; certaines idées, *a priori* très banales, s'avéreront hors du champ d'application de votre appareil. Restez simples et adaptez vos programmes au fur et à mesure que croîtra votre expérience, c'est-à-dire votre maîtrise des possibilités réelles de l'appareil. Vous réaliserez alors à quel point la programmation est un jeu d'enfant !

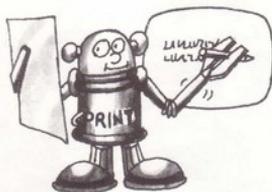
# Petit lexique BASIC

Ce lexique rassemble quelques-unes des instructions les plus communes au langage BASIC et décrit plus particulièrement leurs rôles et utilisations. La plupart d'entre elles ont été utilisées dans les programmes de jeu, ce qui vous permettra de vérifier leur fonction. Tous les ordinateurs ne se réfèrent pas à la même terminologie : une table de conversion page 96 établit les principales équivalences. Nous indiquons, entre parenthèses, une traduction libre des termes présentés.



**LET (SOIT)** indique à l'ordinateur qu'il lui faut repérer une section de sa mémoire et y placer une valeur particulière - LET A=6 signifie qu'une section définie de la mémoire a été repérée, nommée, « A » et que la valeur 6 y a été placée. A s'appelle *variable* ; on assigne une valeur à cette variable en y plaçant un nombre.

Certaines variables sont suivies du signe dollar (\$). Cela signifie que l'on peut leur assigner une chaîne de caractères, incluant des lettres, des nombres et des symboles.



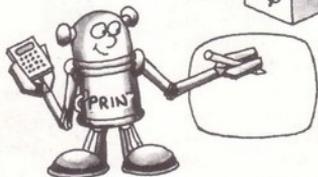
**PRINT (ÉCRIRE)** permet l'affichage à l'écran : cette instruction peut être utilisée de différentes façons.

- Un message entre guillemets précédé de PRINT sera écrit tel quel. La partie comprise entre les guillemets sera écrite à votre convenance, en BASIC ou n'importe quel type de langage ou signe.

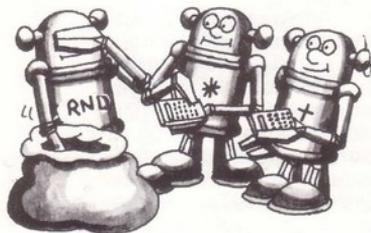
- PRINT suivi d'une variable (PRINT A ou PRINT A\$) fait afficher à l'écran le contenu de cette variable.

- L'instruction PRINT peut également réaliser des calculs et afficher ensuite leur résultat (PRINT 6 \* 4 affiche 24 à l'écran).

- Pour sauter une ligne (ligne vide) utiliser PRINT seul.

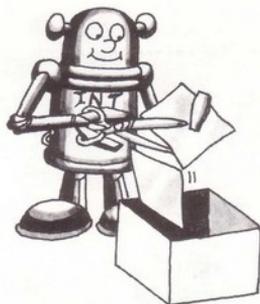


**RND (RANDOM = ÉTABLI AU HASARD)** - Cette instruction permet de tirer un nombre au hasard. Les ordinateurs utilisent différentes formes d'instructions RND comme vous pourrez le constater page 96 (table de conversion). Pour les Sinclair, RND seul donne le nombre compris entre 0 et 0,99999999. Vous pourrez ensuite faire varier ces limites en multipliant RND ou en y ajoutant des valeurs. RND \*20 donne un nombre compris entre 1 et 0,99999999. Voyez INT pour l'obtention de nombres entiers et CHR\$ en ce qui concerne les lettres et autres caractères, tirés au hasard.



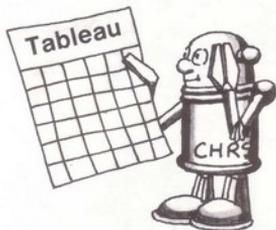
**INT** est l'abréviation de **INTEGER** (NOMBRE ENTIER) – Pour les nombres positifs tous les chiffres placés après la virgule sont ignorés. Ex.  $\text{INT}(20.999) = 20$ . Pour les nombres négatifs, même abandon des chiffres après la virgule mais la valeur de l'entier est arrondie à l'unité supérieure :  $\text{INT}(-3.6) = -4$ . **INT** est presque toujours utilisé avec **RND** de la façon suivante :

$\text{INT}(\text{RND} * 20 + 1)$ , ce qui signifie que vous tirez au hasard un nombre entier entre 1 et 20.



**CHR\$** (CARACTÈRE) convertit des nombres en lettres. Sauf en ce qui concerne le ZX81, tous les ordinateurs présentés dans cet ouvrage se réfèrent au code ASCII\* dans lequel chaque caractère correspond à un nombre défini. Par exemple, le nombre-code 65 et l'instruction **PRINT CHR\$(65)** afficheront A à l'écran. On peut également utiliser **CHR\$** avec **INT** et **RND** pour générer des lettres de façon aléatoire. Ainsi :  $\text{CHR}(\text{INT}(\text{RND} * 26 + 65))$

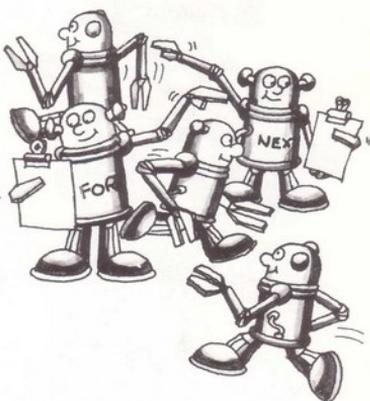
Cette instruction opère un tirage aléatoire de lettre sur le ZX Spectrum (voir table de conversion pour les autres appareils).



**FOR** (POUR) sert à déclencher une boucle qui fera se répéter une partie du programme un certain nombre de fois. **FOR** sera suivi d'une variable – lettre G, par exemple, pour définir le nombre de tours autorisés – à laquelle on assignera une valeur de départ et une valeur d'arrivée, telle que 1 TO 10 (1 à 10).

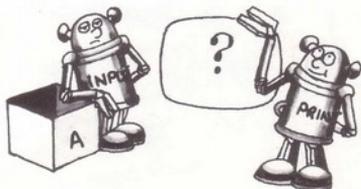
La fin de la boucle est matérialisée par l'instruction **NEXT** (**NEXT G** dans cet exemple) qui augmente la valeur de la variable de 1 unité à chaque tour et renvoie l'exécution du programme à l'instruction **FOR**.

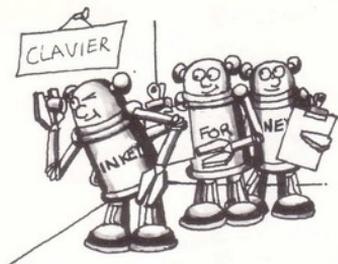
Lorsque la variable atteint sa valeur maximale, l'ordinateur ignore l'instruction **NEXT** et passe à la ligne suivante. Chaque **FOR** doit être suivi d'un **NEXT** ; dans le cas contraire il y a erreur.



**INPUT** (INFORMATION FOURNIE) réserve un espace dans la mémoire de l'ordinateur, affiche un point d'interrogation et attend que vous frappiez sur le clavier l'information à placer dans l'espace mémoire préalablement indiqué. L'exécution du programme sera arrêté jusqu'à ce que la touche **RETURN**, **ENTER** ou **NEWLINE** soit pressée.

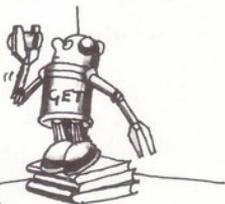
Vous pouvez faire suivre **INPUT** de variables A ou A\$ avec assignation de nombres ou de chaîne de caractères ; mais attention, si vous optez pour une variable numérique (A) votre ordinateur n'acceptera aucune lettre.





**INKEY\$** (voir GET) contrôle le clavier pour savoir si vous avez appuyé sur une touche, et si oui, laquelle. Contrairement à l'instruction INPUT, l'exécution du programme n'est pas, pour autant, arrêtée. INKEY\$ est habituellement placé dans une boucle et l'ordinateur exécute plusieurs contrôles de clavier afin de vous laisser le temps de presser la touche choisie.

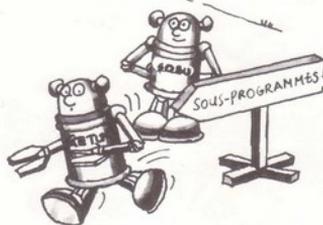
Si d'aventure vous n'y parvenez pas, le programme continue avec une chaîne de caractères vide (variable à valeur nulle). NB Apple et VIC n'utilisent pas l'instruction INKEY\$.



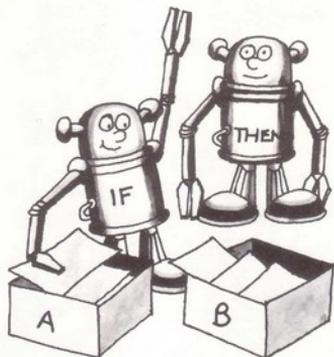
**GET** (ALLER CHERCHER) est utilisé à la place de INKEY\$ pour les ordinateurs VIC et PET.



**GOTO** (ALLER A). L'ordinateur saute directement à la ligne de programme indiquée (en amont ou en aval) en ignorant les instructions intermédiaires. Ne pas oublier d'indiquer le numéro de la ligne choisie après l'instruction GOTO.



**GOSUB** (Gosub, sub-routine = ALLER AU SOUS-PROGRAMME). Quittant le programme principal, l'ordinateur va vers un sous-programme. GOSUB doit être suivi du numéro de la première ligne de ce sous-programme qui s'achèvera obligatoirement par l'instruction RETURN ; celle-ci renvoie au programme principal, à la ligne située immédiatement derrière l'instruction GOSUB d'origine. GOSUB sans RETURN provoque une erreur.



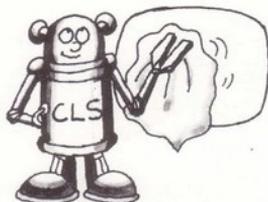
**IF... THEN** (SI... ALORS) fait réaliser à l'ordinateur un test de comparaison, et selon la réponse, exécuter telle ou telle instruction.

IF...THEN, qui peut se combiner avec AND (ET) et OR (OU), s'utilise avec les signes suivants :

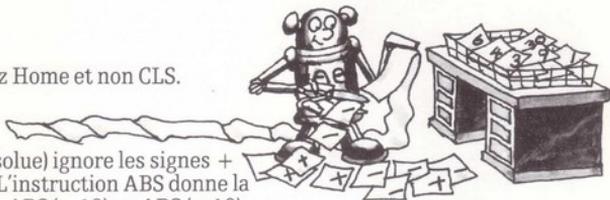
- = égal à
- < plus petit que
- > plus grand que
- < = plus petit ou égal à
- > = plus grand ou égal à
- < > différent de.

Si le test de comparaison est positif, l'instruction suivant THEN est exécutée ; dans le cas contraire, elle est ignorée et l'ordinateur passe à la ligne suivante.

**CLS** est utilisée pour effacer (CLEAR = NETTOYER) l'écran sans enlever ou modifier quoi que ce soit dans la mémoire. C'est utile pour retirer de l'écran la liste du programme avant de commencer un jeu, surtout dans ceux qui font appel à des messages ou caractères apparaissant très peu de temps. NB : Apple et VIC n'utilisent pas cette instruction : voir tableau de conversion.

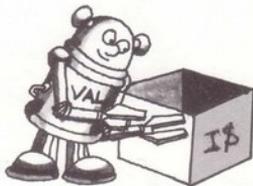


**HOME** : sur Apple, employez Home et non CLS.

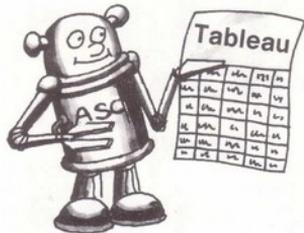


**ABS** (Absolute = Valeur absolue) ignore les signes + et - précédant un nombre. L'instruction ABS donne la valeur absolue. Par exemple, ABS (-10) ou ABS (+10) = 10.

**VAL** (Valeur numérique) prend la valeur numérique d'un nombre préalablement écrit comme une chaîne ou suite de caractères (variable alphanumérique). Dans les faits, l'ordinateur ignore le signe \$ (dollar) et traite la chaîne de caractères comme une variable numérique ordinaire. Si I\$ = « 60 » alors VAL (I\$) donne le nombre 60.



**ASC** (début de ASCII) donne à un caractère sa valeur-nombre dans le système de codage ASCII (American Standard Code for Information). L'expression entre guillemets doit obligatoirement être une chaîne de caractères (variables alphanumériques). Par exemple, ASC (« 3 ») donne 51. NB : ZX81 et ZX Spectrum n'utilisent pas cette instruction puisqu'ils ne se réfèrent pas au code ASCII.



**CODE** remplace ASC pour ZX81 et ZX Spectrum. Se souvenir que ces ordinateurs utilisent un code différent.

**TAB** (TABULATOR) déplace le curseur sur l'écran jusqu'à une colonne définie. Cette instruction accompagne généralement PRINT pour afficher un message au milieu de l'écran. Le nombre d'espaces pour le déplacement de curseur est fixé par le nombre entre parenthèses suivant TAB - PRINT TAB (20). Sa valeur dépend de la largeur de l'écran de votre ordinateur.

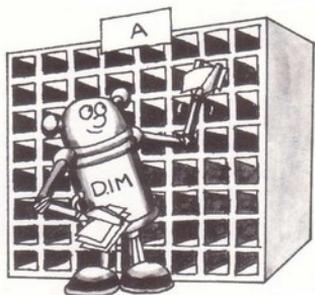




**SGN (SIGNE)** permet à l'ordinateur de repérer le signe précédant un nombre. Cela donne  $-1$  pour un nombre négatif,  $0$  pour zéro et  $+1$  pour un nombre positif.

Exemples :

$SGN(-30)$  donne  $-1$ ,  $SGN(+7)$  donne  $+1$  et  $SGN(0)$  donne  $0$ .



**DIM (DIMENSION)** indique à l'ordinateur quel espace il doit réserver en mémoire pour un tableau (rangée ou grille). Par exemple :  $DIM X(6)$  fait réserver à l'ordinateur une zone suffisante pour contenir une rangée de 6 éléments et la nomme X.

$DIM A(8,8)$  signifie un espace mémoire appelé A et suffisamment grand pour stocker 8 éléments en largeur et 8 éléments en hauteur. Le nombre d'éléments stockés utilisés par le programme doit correspondre aux nombres entre parenthèses suivant l'instruction DIM. En cas contraire, il y a une erreur.



**SQR (Square root = RACINE CARRÉE)** extrait la racine carrée de nombres. Exemple :  $SQR(16)$  donne 4.

**SIN (SINUS)** calcule le sinus d'un angle. Lorsque vous utilisez SIN dans un programme, la mesure de l'angle doit être effectuée en radians et non en degrés. Consultez un livre de maths si cette notion vous est étrangère.

**ATN (ARC TANGENT)** est l'une des fonctions trigonométriques que les ordinateurs peuvent calculer. Cette instruction détermine l'arc tangent et il est important de se rappeler que le résultat est donné en radians.

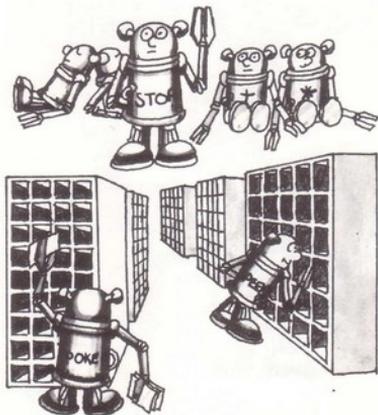
**STOP** arrête l'exécution du programme. Les ordinateurs autres que le ZX81 utilisent également END (FIN).

**PEEK (JETER UN COUP D'ŒIL)** : une manière de vérifier ce qui est stocké en une zone spécifique de la mémoire de l'ordinateur. Il vous faudra utiliser un nombre qui spécifiera l'adresse en mémoire.

NB : Instruction non utilisée sur BBC.

**POKE (FOURRER, PLACER)** : une façon particulière de placer des informations dans la mémoire d'un ordinateur en utilisant une adresse spécifique.

NB : instruction non utilisée sur BBC.





# Réponses aux casse-tête

Il se peut que votre solution soit différente de celle proposée ici. A partir du moment où elle fonctionne sur votre ordinateur, cela n'a pas d'importance. Mais comparez tout de même pour voir si cette solution est aussi simple et claire que celles présentées ci-dessous.

## Page 101 Décollage d'astronef

Les lignes 30 et 40 définissent les valeurs qui détermineront la poussée. Pour augmenter l'intervalle de valeurs possibles pour la poussée, augmentez soit l'une, soit l'autre des valeurs 20 ou 40 (vous pouvez augmenter les deux). L'élargissement de l'intervalle des valeurs de la poussée accroît obligatoirement la difficulté du jeu.

## Page 103 Des Jeux olympiques intergalactiques

Modifiez les lignes 222 et 230 comme suit :

```
220 LET B=B+INT(1000/G)
230 GOTO 20
```

Ajoutez cette ligne supplémentaire :

```
15 LET B=0
```

## Page 105 Elron l'invisible

Modifiez les lignes 20 et 30 et ajoutez la ligne 25 comme suit :

```
20 PRINT « NIVEAU DE DIFFICULTÉ?
(6-30) »
25 INPUT T
30 LET G=INT(T/3)
```

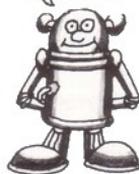
## Page 107 Les punaises aux yeux d'acier

Pour faire apparaître les yeux en plus de quatre endroits sur l'écran, vous devez placer une valeur supérieure à 4 au milieu de la ligne 70, modifier la ligne 80 et ajouter des sous-programmes à la fin du programme – un pour chaque nouvelle position.

Voici les changements pour faire apparaître les yeux en 5 endroits :

```
70 LET R=INT(RND*5+1)
SZX80 GOSUB 220+20*R
*■▲80 ON R GOSUB 240,260,
280,300,320
240 LET D=5
245 LET A=1
250 GOTO 350
260 LET D=1
265 LET A=9
270 GOTO 350
280 LET D=5
285 LET A=18
290 GOTO 350
300 LET D=10
305 LET A=7
310 GOTO 350
320 LET D=15
325 LET A=15
330 GOTO 350
```

Vous pouvez utiliser les valeurs que vous désirez pour A et D, pourvu qu'elles entrent dans votre écran.



Pour ajouter plus de punaises spatiales : modifiez la valeur 10 des lignes 30 et 220 par une plus importante, donnant le nombre de paires d'yeux. (Faites attention à ce que la valeur soit la même aux deux lignes.)

## Page 109 Alunissage

Pour augmenter la vitesse possible correspondant à un alunissage réussi, modifiez les lignes 230, 240, 250. Vous pouvez y placer n'importe quelle valeur – le jeu sera d'autant plus facile que cette valeur sera élevée.

Dans cet exemple, une vitesse égale au plus à 2 permet un bon alunissage, et l'arrivée demeure possible jusqu'à une vitesse de 7.

```
230 IF V1>7 THEN PRINT « CRASH-AUCUN
SURVIVANT »
240 IF V1>2 AND V1<=7 THEN PRINT
« OK-MAIS QUELQUES BLESSÉS »
250 IF V1<=2 THEN PRINT « ALUNISSAGE
PARFAIT »
```

## Page 111 Les monstres de Galacticon

Quatre moyens de rendre le jeu plus difficile :

1) Débuter le jeu avec un effectif réduit. En plaçant une valeur plus faible que 5 à la ligne 40.

2) Augmenter le nombre de monstres en modifiant le 4 des lignes 20 et 30 par un chiffre supérieur. Ajoutez des noms de monstres de la ligne 81 à 89 en utilisant M\$(5) et M\$(6).

3) Réduire le nombre de tours (essais) par la diminution du 8 de la ligne 160.

4) Augmenter les chances de mettre en colère le monstre en augmentant de un le 4 de la ligne 330.

## Page 113

### L'embuscade extra-terrestre

Dans ce jeu, N est le numéro de code qui permet de déterminer la position réelle. Pour adapter le score à la difficulté du jeu, vous devez augmenter ce dernier de la valeur N à chaque réussite. Pour cela, modifiez la ligne 190 comme suit :

```
S ZX 190 IF I$=CHRS(CODE(LS)+N)
      THEN LET S=S+N
★■● 190 IF I$=CHRS(ASC(LS)+N) THEN
      LET S=S+N
```

## Page 115

### La ceinture d'astéroïdes

Vous devez modifier la ligne 260 pour que l'ordinateur ajoute au score le nombre d'étoiles à la place d'une unité. Le nombre d'étoiles est contrôlé par la valeur tirée et stockée en N à la ligne 70; ce sera la valeur à ajouter au score. Vous devez aussi modifier la ligne 320 d'écriture du résultat.

```
260 LET S=S+N
320 PRINT « VOUS AVEZ MARQUÉ »;S;
      « POINTS »
```

## Page 117

### Voyage dans le futur

1) Pour augmenter le nombre d'années pouvant s'écouler durant votre voyage sur

la Terre, il vous faut remplacer le 100 de la ligne 30 par un plus grand nombre : 150 par exemple.

```
30 LET T=INT(RND*150+25)
```

2) Pour augmenter la précision de l'arrivée de 5 à 2 ans, il faut changer les 5 en 2 dans les lignes 180 et 190 :

```
180 IF ABS(T-T2)<=2 THEN PRINT
      « VOUS ÊTES ARRIVÉ À TEMPS »
190 IF ABS(T-T2)>2 THEN PRINT
      « MÊME PAS À 2 ANS PRÈS »
```

3) La ligne 170 contient la valeur qui détermine la durée de votre espérance de vie. Transformez le 50 en un nombre plus petit ou plus grand selon votre choix.

## Page 119

### La Vallée de la Mort

Vous pouvez augmenter la longueur de la vallée en remplaçant la valeur proposée à la ligne 30 par une autre plus importante.

## Page 121

### Les mines d'Astron

Ajoutez ces lignes pour que l'ordinateur vous demande si vous désirez un nouveau mandat :

```
645 PRINT « UN AUTRE MANDAT? (O OU N) »
646 INPUT AS
647 IF AS=« O » THEN GOTO 10
```

Vous devez par ailleurs ajouter une nouvelle ligne en 5 et modifier la ligne 30 pour additionner l'argent restant dans les caisses après votre précédent mandat (selon le même principe, vous pouvez reporter les autres valeurs).

```
5 LET A=0
30 LET A=A+INT(RND*50+10)*P
```

(Faites attention de bien adapter le RND à votre ordinateur.)

# Index

- ABS, 79, 139  
Amplificateur, 21  
Animation, 18, 19, 46  
Architecture, 36, 37  
ASCII, Code, 28, 46  
Automate, 34-35  
Automobile, 37
- Bandes données, 16  
Banque de données, 23  
Barre d'espace, 8  
BASIC, 3, 6, 7, 8, 9, 10, 12, 13, 15, 28, 46, 51, 52, 53, 55, 57, 58-59, 61, 68-69, 70, 78, 80, 84, 86, 90, 98, 135, 136-140  
BBC, 69, 70, 91  
Bit, 26, 29, 46  
Boucle, 74-75, 76-77  
BREAK, 63, 71, 73, 85, 99  
Bruitage, 20-21  
Bug, 14, 15, 16, 46, 57, 59, 76, 90, 91, 135  
Bus, 24, 46  
Byte, 7, 26, 28, 29, 46
- Calculateur, 23, 30, 31, 35  
Capteur, 34, 35, 46  
Cartouche de programme, 10  
Cassette, 10, 11, 16, 17  
Chaîne alphanumérique, 60  
Chargement, 46  
Circuit, 24, 26-27  
- imprimé, 22, 23  
- intégré, 22, 30, 46  
Clavier, 4, 8, 9, 40  
CLS, 58, 63, 68, 73, 75, 76, 77, 82, 85, 91, 139  
Club, 11  
Code, 139  
- binaire, 26, 27, 46  
- hexadécimal, 29, 46  
- machine, 46  
- mnémonique, 29, 47  
Compatibilité, 46  
Connexion, 16, 34  
Console, 4, 8, 22-23, 34  
Copie papier, 17, 46  
Courrier électronique, 33  
Crayon optique, 3, 18, 19, 20, 24, 37, 39  
Curseur, 9, 58, 126
- DATA (données), 4, 7, 46, 61, 69, 79, 87, 88, 89, 91  
DELETE, 59  
Dessin, 18, 19, 39, 70-73  
DIM, 88, 89, 90  
Disquette, 10, 11, 16, 17, 38, 40  
Données v. DATA
- Écran, 5, 18, 19, 40  
EDIT, 59  
END, 59, 101, 133  
ENIAC, 31  
ENTER, 58  
Entrée, 4, 46, 60-61  
ESCAPE, 63, 73
- Fibre optique, 32  
FOR ... NEXT, 74-77, 91, 137  
FORTRAN, 46
- GOSUB, 78-79, 83, 85, 91, 106, 132, 133, 138  
GOTO, 67, 68, 69, 74, 91, 99, 132, 138  
Graphiques, 52, 82-83, 84-85  
Graphisme, 18, 19, 40, 46
- Haut-parleur, 20-21
- IF ... THEN, 66-67, 68, 69, 71, 73, 76, 79, 83, 85, 88, 89, 138  
Imprimante, 16, 17, 38, 39, 53  
Informatique, 33, 40  
INPUT, 58, 62-63, 66, 69, 137  
INT, 72, 105, 137  
Interface, 16, 17, 34, 38, 40, 46  
Interpréteur, 28, 29, 46, 54
- Jeux, 29, 38, 98-142  
Joystick, 38
- Kilobyte, 7, 46
- Lampe, 30-31  
Langage, 55  
Langage de programmation v. BASIC, FORTRAN, PASCAL  
Lecteur-enregistreur, 17, 38  
LEFT\$, 80-81  
LEN, 80-81  
LET, 60, 61, 65, 80, 136  
LIST, 59, 63, 98  
Liste, 10, 14, 47  
Logiciel, 6, 10, 12, 40
- Magnétophone, 5, 10, 14, 16, 17, 38, 53, 58  
Matériel, 6, 47  
Mathématiques, 64, 65, 82  
Médecine, 36  
Mémoire, 4, 7, 13, 19, 25, 30, 32, 38, 53, 60, 61, 65, 132
- morte, 7, 22, 23, 24, 25, 28, 47  
- de secours, 29  
- vive, 7, 8, 16, 22, 23, 25, 29, 38, 39, 47  
Météorologie, 36  
Micro-ordinateur, ALICE (Matra-Hachette), 45  
APPLE II Plus (APPLE), 45, 98, 131  
Atari 400, 40, 43  
Atari 800, 40  
BBC Micro (Acorn), 44, 98, 115, 130, 133  
DRAGON (Dragon Data), 43  
Newbrain, 42  
PC 1500 (Sharp), 42  
PET (Commodore), 45, 98  
SPECTRUM (Sinclair), 41, 101  
TI-99/4 (Texas Instruments), 43  
TO 7 (THOMSON), 44  
TRS-80, 44, 98, 126-127  
VIC 20 (Commodore), 42, 98, 107, 118, 126, 133  
ZX 81 Sinclair, 41, 98, 99, 101, 108, 113, 117, 119, 121, 124, 128, 132  
Microprocesseur, 23, 24, 25, 28, 31, 34, 37, 40, 47  
MIDS, 80-81  
MODE, 70  
MODEM, 32, 40, 47  
Modulateur, 23  
Moniteur, 4, 47, 52  
Musique, 20-21
- Navette spatiale, 35  
NEW, 63  
NEWLINE, 58, 59, 62  
Nombres aléatoires, 72
- Onde sonore, 20, 21  
Organigramme, 57
- Paddle, 38  
PASCAL, 12, 47, 55  
Pastille v. puce  
Pédagogie, 11, 21, 33, 37  
Périphériques, 38-39  
Pile, 29  
PILOT, 55, 84  
Pixel, 18, 19, 39, 47, 70-71, 73, 77, 85  
PLOT, 70-71, 73, 77, 82, 84, 85, 91  
PRINT, 58, 59, 60, 61, 64-65, 99, 132, 136  
Programmation, 3, 6, 9, 10-11, 12-13, 14-15,

16-17, 20, 23, 25, 28,  
29, 32, 34, 38, 134-135  
Programme, 52, 53, 54, 55,  
56, 57, 63, 67, 68-69,  
82, 86-87, 90-91  
Puce, 3, 22, 23, 24-25,  
26-27, 28, 30, 31  
Pug, 57

RAM v. mémoire vive  
RANDOM, 72, 76, 77, 85,  
86, 88, 89, 91  
READ, 61, 69, 79, 87, 88,  
89, 91  
READY, 58  
Régulateur de tension, 22  
REM, 75, 78, 79, 83, 85, 91  
Réseaux, 32-33  
Résistance, 22  
Résolution v. pixel  
RETURN, 58, 78, 79, 106,  
132, 133  
RIGHT\$, 80  
RND, 134, 136  
Robot, 6, 34-35, 36

ROM v. mémoire morte  
Routine, 78-79  
RUN, 58-59, 62, 63, 71, 74,  
83, 85, 87, 88, 90, 91,  
99, 133

Satellite, 32, 33  
Sauvegarde, 47  
Silicium, 22, 24, 25, 30  
SINCLAIR, 80, 88, 90  
Sonde spatiale, 35  
Sortie, 4, 47  
Sous-programme, 78-79  
STEP, 75, 77, 81  
STOP, 67, 78, 79, 83, 86, 88,  
89, 101, 133  
Synthétiseur, 20, 21, 47  
Système variable, 29, 47

TAB, 139  
Table  
– graphique, 18, 38, 39  
– traçante, 38, 39  
Téléphone, 32, 33  
Télérel, 10, 33

Télévision, 5, 18, 23, 32, 38  
Terminal, 27  
THEN v. IF  
Touches  
– alphanumériques, 8  
– d'effacement, 9  
– programmable, 9  
– retour chariot, 9  
– shift, 8  
Transformateur, 5  
Transistor, 22, 26, 30, 47  
Transpac, 10  
Unité  
– centrale, 4, 23, 47, 53  
– logique et  
arithmétique, 23, 25  
– de visualisation, 47  
UNPLOT, 70, 84, 85

VAL, 139  
Variable, 60-63, 65, 66, 69,  
75, 80, 88  
Vitesse de transmission,  
47  
ZX81, 61, 67, 69, 80

# GUIDE DU MICRO ORDINATEUR

Comment fonctionne un ordinateur? ■ Un coup d'œil sur la console  
■ Qu'est-ce qu'un programme? ■ Premiers pas en BASIC ■ Entrée  
des données ■ Comment les ordinateurs comparent ■ Écrivez vos  
propres programmes ■ Faites-les fonctionner sans peine ■  
Comment conserver un programme ■ Dessins, graphismes et ani-  
mation ■ Bruitages et musique ■ Quelques astuces et trucs de  
programmation ■ Histoire de la micro-informatique ■ Réseaux  
d'ordinateurs ■ Robots et automates ■ D'autres utilisations de la  
micro-informatique ■ Des périphériques pour votre micro-ordina-  
teur ■ Petit lexique du BASIC ■ Le guide des micro-ordinateurs ■  
Et 13 programmes de jeux.