

# MICROCOMPUTADOR CURSO PRÁTICO

TIRE O MÁXIMO DO MICRO EM 30 SEMANAS

PUBLICAÇÃO SEMANAL ILUSTRADA

Cr\$ 10.250

24



**MSX: padrão internacional • O jogo da centopéia**  
**Máquinas sob comando da mente**

*rioGráfica*

Depois de ler com atenção todo o fascículo, responda a estas perguntas. Veja a solução no próximo número, junto com um novo teste.

- 1) Em que fenômeno orgânico se baseiam o detector de mentiras e o "acoplamento mental" do homem ao computador?
- 2) Como se chamam os motores que produzem movimentos extremamente precisos, utilizados nos robôs e nas tartarugas mecânicas?
- 3) Qual o microprocessador adotado pelos micros padrão MSX?
- 4) Quais os dois primeiros micros padrão MSX introduzidos no Brasil?

## RESPOSTAS DO TESTE ANTERIOR

- 1) Numa comparação da visão humana com a robótica, os olhos são os sensores e o cérebro é o processador.
- 2) O dispositivo mais usado como "olho" robótico é a câmara de vídeo.
- 3) O processador onde se executam as operações aritméticas e lógicas é a ALU (Arithmetic and Logic Unit, "unidade lógica e aritmética").
- 4) Um sistema eficiente de mala direta deve ser utilizado em conjunto com um processador de texto e um gerenciador de banco de dados.



## APLICAÇÕES

Comando mental	477
O passo do robô	490



## PROJETOS DE PROGRAMAS

Centopéia faminta	480
-------------------	-----



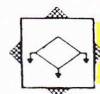
## CHIPS & BYTES

Solução para compatibilidade	482
Força bruta	488



## RAIO X

Padrão MSX	485
------------	-----



## LINGUAGENS

Mudança de endereço	493
---------------------	-----



## PERFIL

Labo	496
------	-----

## PLANO DA OBRA

MICROCOMPUTADOR - CURSO PRÁTICO é uma coleção de 30 fascículos de periodicidade semanal. Os 29 primeiros terão, cada um, vinte páginas internas (miolo) e quatro capas, além de uma introdução de dezesseis páginas acompanhando o fascículo 1. O fascículo 30 incluirá o índice geral da obra. Os miolos, encadernados, constituem dois volumes de um curso prático de microcomputação, com seções indicadas por tarjas de diferentes cores. As duas primeiras capas são descartáveis; as 3.<sup>as</sup> e 4.<sup>as</sup> trazem programas de jogos. Encadernadas, formam um volume de cem páginas.

## COMO ENCADERNAR

As capas duras, incluindo as guardas, estarão à venda simultaneamente com os fascículos 10 (**Volume 1**) e 30 (**Volume 2 e Jogos**).

**Volume 1** - Deve ser encadernado com os elementos dispostos nesta ordem: guardas; frontispício e introdução (publicados com o fascículo 1); miolos dos fascículos de 1 a 15; e guardas.

**Volume 2** - Guardas; frontispício (que virá com o fascículo 30); miolos dos fascículos de 16 a 30; e guardas.

**Jogos** - Guardas; frontispício e sumário (que constituem as 3.<sup>as</sup> e 4.<sup>as</sup> capas do fascículo 30); 3.<sup>as</sup> e 4.<sup>as</sup> capas dos fascículos de 1 a 29; e guardas.

**rioGráfica**

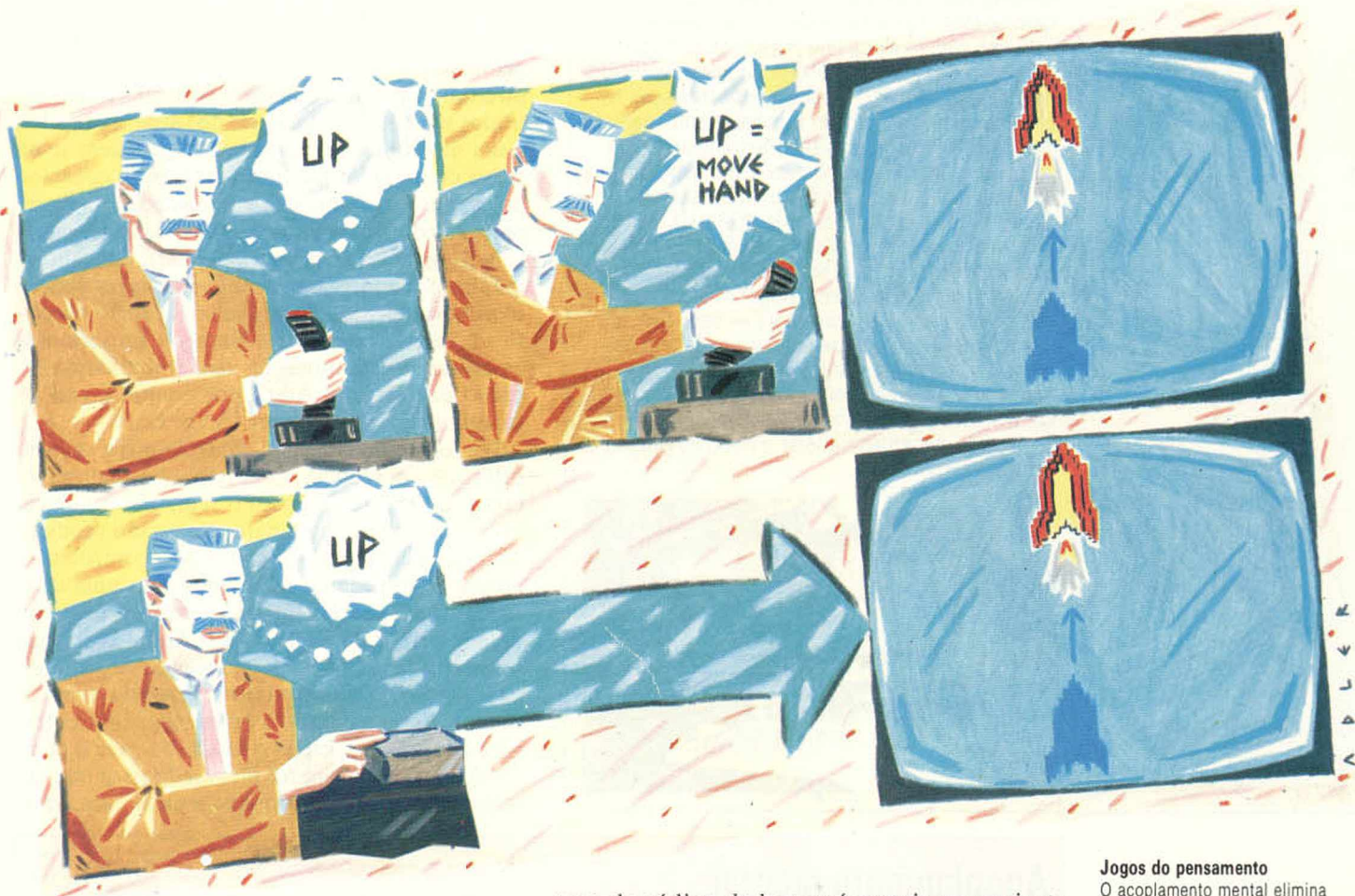
**CONSELHO DE ADMINISTRAÇÃO:** Roberto Irineu Marinho; João Roberto Marinho; José Roberto Marinho; Oscar Neves; **Diretoria:** Oscar Neves; Danilo Esteves Costa; João J. Noro; Marcos Marchi; Nilo Sérgio de Almeida; **GRUPO EDUCAÇÃO E CULTURA - Divisão Comercial - Diretor:** Jaime Rodrigues; **Gerente de Planejamento:** João de Azeredo Coutinho; **Gerente de Vendas:** Rubens Barbosa; **Coordenadores de Promoção:** Paulo Cesar M. Seixas; Edgard Mello M. Neto; **Gerente de Circulação:** Norberto Martin; **Supervisor de Operações:** Abel Pereira Pinto; **Editorial - Diretor:** João J. Noro; **Editor-chefe:** Maurício Rittner; **Diretor de Arte:** Adalberto Cornavaca; **Administração Editorial:** Heitor de Souza Paixão; **Editor:** Jesse Navarro; **Editor-assistente:** Claudio Marcondes; **Coordenadora de Revisão:** Isis Augusta Loyolla; **Revisores:** Cacilda Guerra; Luiz Vicente Vieira Filho; Maria da Graça Mendonça Couto; Sérgio Roberto Torres; **Chefe de Arte:** Bonifácio Duardes Miranda; **Chefe de Estúdio:** José Yuji Kuribayashi; **Assistentes de Arte:** Miguel Luis Escámez Simón; Edemilson Ribeiro da Silva; Wagner Celestino; **Colaboradores - Editores-assistentes:** Isa Mara Lando; Luiz Carlos Pizarro Marin; **Texto:** Carlos Eduardo Silveira Matos; José Antônio Arantes; Jussara Maturu; Paulo Brito; Solange A. Menezes; **Preparação de Texto:** Luiz Carlos Cardoso; **Tradução:** Luiz Roberto de Godoy Vidal; Maria Clara Cascato; Maria Ester Menezes Marinho; Fernando Dorfman Knijnik; Newton Roberval Eichenberg; **Pesquisa:** Stella Maria Quentel; **Assessoria Técnica:** Geraldo Coen; Gustavo José Ferreira Grubba; José Rubens Salles Toledo; Lauro de Lauro; Pierluigi Piazzì. As fotos não creditadas pertencem à obra original. Segue-se o crédito das demais: Labo; © APSIF Copenhagen, 1984; © Orbis Publishing Ltd., 1984; © Editora Rio Gráfica Ltda., 1985, para a língua portuguesa. **Composição:** Editora Rio Gráfica Ltda. e AM Produções Gráficas Ltda. **Impressão:** JBIG.

**NÚMEROS ATRASADOS** - Você poderá comprar os exemplares que faltam em sua coleção, pelo preço do último fascículo posto à venda, em sua banca de jornal preferida. Caso não consiga obtê-los, faça sua solicitação por carta endereçada diretamente à Editora Rio Gráfica Ltda., rua Itapiru, 1209, Rio Comprido, Rio de Janeiro, CEP 20251. O atendimento será feito por via postal. Nas cidades do Rio de Janeiro e de São Paulo, as compras poderão ser efetuadas pessoalmente nos seguintes endereços: Rio de Janeiro - Rua Itapiru, 1209, Rio Comprido; São Paulo - Rua Frei Caneca, 1152, Consolação.

Distribuidor exclusivo para todo o Brasil: Fernando Chinaglia - Rua Teodoro da Silva, 907, Rio de Janeiro. Distribuidor para Portugal: Electrolibr Lda. - Rua Prof. Reinaldo dos Santos, 1488, Lisboa.



# COMANDO MENTAL



**O controle de um computador pelo pensamento deixou de ser ficção científica. Conheça o fenômeno que fundamenta essa tecnologia e o dispositivo que acopla o sistema nervoso ao computador.**

Desde que surgiram os primeiros computadores, o teclado tem sido o principal dispositivo de entrada de informações. Talvez seja o método mais eficaz de se introduzir textos e, em menor grau, números — pelo menos até que se desenvolvam sistemas confiáveis de reconhecimento de voz e destinados a objetivos gerais. Entretanto, por mais adequado que seja para se fornecer textos à máquina, o teclado não é, necessariamente, o melhor dispositivo para se introduzirem outros tipos de informação. Se precisamos inserir dados direcionais com rapidez, como em vários jogos, damos preferência ao joystick ou ao paddle; no caso de dados posicionais, como numa seleção de opções num menu, o mouse, a caneta óptica ou a tela sensível ao toque mostram-se em geral mais apropriados; e a lei-

tora de código de barras é o mais conveniente para longos números ou códigos seriais.

Mas todos esses dispositivos apresentam um inconveniente: são indiretos. Nos jogos, o joystick é mais adequado que o teclado (para fazer a nave subir, por exemplo, empurramos a empunhadura para a frente), mas ele ainda representa um hiato entre o que pretendemos e o que de fato acontece. Por exemplo, pensamos “para cima”, traduzimos isso mentalmente como “empurrar para a frente” e só então empurramos. Para uma interação mais rápida e direta entre nós e o computador, seria preciso eliminar essa mediação: por que não apenas pensar “para cima” e obter do aparelho pronta resposta ao nosso pensamento?

Assim, jogaríamos Invasores do Espaço apenas ordenando mentalmente “Para cima”, “Para baixo”, “Vire”, “Fogo”, e assim por diante; ou, então, escreveríamos uma carta apenas pensando nas palavras. Claro que o processador de textos telepático ainda não existe, mas jogar Invasores do Espaço por controle mental já é uma realidade, graças à noção de “acoplamento mental”. Segundo esta, empregamos o pensamento — mais precisamente, as alterações fisiológicas resultantes das mudanças nos padrões de pensa-

## Jogos do pensamento

O acoplamento mental elimina o estágio mecânico intermediário da conversão dos impulsos do jogador em sinais significativos para o computador. Colocando usuário e máquina em relação direta, o acoplamento mental possibilita programas mais interativos e entradas mais rápidas.

## Notas mentais

A foto ilustra uma demonstração de um protótipo do sistema RGP funcionando com um computador Apple. Sistemas semelhantes já foram desenvolvidos para uso no IBM PC e no Commodore, mas é bastante provável que o software efetivo ainda leve mais tempo. Roger Dilts, presidente da Behavioral Engineering e autor de software para RGP, tem especial interesse na possibilidade de combinar o acoplamento mental com a NLP (Neuro-Linguistic Programming, "programação neurolíngüística"), um ramo da psicologia dedicado ao estudo do aprendizado. Com a utilização da técnica de RGP para monitorar o estado mental do usuário, o software pode medir o grau de tensão do mesmo e ajustar o andamento do programa. Roger Dilts acredita que os programas educacionais — para citar um exemplo — serão capazes de avaliar as reações emocionais dos estudantes ao material apresentado. Se este for demasiado complexo, o computador captará a tensão emocional provocada, permitindo alterar o ritmo do programa ou simplificar o tema.



mento — para controlar um dispositivo eletrônico, que funciona como uma interface para um computador, tal como o joystick ou outro dispositivo de entrada.

O acoplamento mental baseia-se no conhecido fenômeno RGP (Reação Galvânica da Pele), pelo qual mudanças no estado emocional se manifestam na condutividade elétrica da pele. Eletrodos são colocados na pele do usuário e ligados a um medidor de resistência. Introduzem-se os sinais do medidor no micro através do conector de expansões, e um software especial os interpreta e processa. A empresa Behavioral Engineering (Engenharia Comportamental), da Califórnia, baseou tanto jogos como softwares aplicativos nessa técnica, inclusive uma versão simplificada do Invasores do Espaço. Importantes empresas de computação vêm trabalhando há anos nesse setor, a Atari, por exemplo, mas só recentemente alcançaram algum êxito.

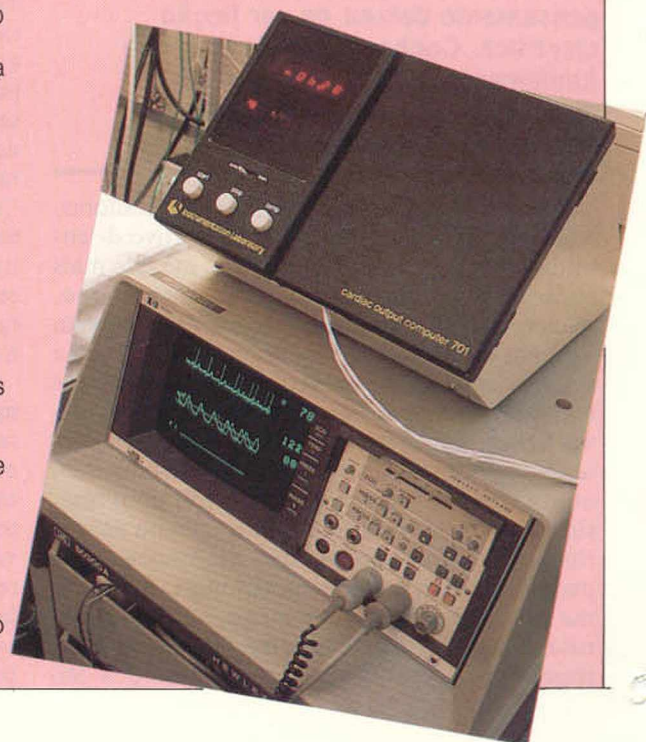
No jogo convencional, controlamos a nave na órbita de um planeta. O objetivo é atingir naves inimigas sem que a nossa o seja, e sem que ela se espatife contra o planeta. Na versão produzida pela Behavioral Engineering, controlamos apenas a altitude da nave, mas com uma diferença: pelo pensamento! A empresa projetou uma interface de RGP para o Apple II em que o usuário tem apenas de colocar os dedos indicador e médio num aparelho parecido com um mouse. Este, medindo a resistência elétrica entre os dois dedos, envia o resultado para o computador. O software foi planejado para que um aumento na tensão comande a nave para cima,

## Acoplamento mental

Esse princípio baseia-se num fenômeno conhecido por três nomes alternativos: RGP (Reação Galvânica da Pele); RPG (Reflexo Psicogalvânico); e RED (Reflexo Eletrodérmico). Preferimos adotar o primeiro termo. A RGP refere-se a mudanças na condutividade elétrica da pele que correspondem a alterações no estado emocional de uma pessoa. Experimentos mostram que, quanto mais tensa uma pessoa, mais baixa se torna a resistência elétrica de sua pele. A aplicação mais conhecida da RGP está nos polígrafos, os chamados detectores de mentiras.

Embora se observe o fenômeno da RGP em animais e seres humanos desde o século XIX, pouco se conhece sobre sua causa. Segundo a teoria inicial, o suor produzido pela excitação ou ansiedade age como condutor eletrolítico, baixando assim a resistência da pele. Experiências mais recentes, porém, lançaram dúvidas sobre essa teoria tão simplista. Seja qual for a razão do efeito, sabe-se que a RGP relaciona-se diretamente ao grau de tensão no sistema nervoso simpático. Este depende do sistema nervoso central e, portanto, do cérebro. Daí ser possível o controle de um computador pelo pensamento. Alterações na atividade cerebral acarretam mudanças no estado do sistema nervoso central; isso modifica o estado do sistema nervoso simpático, resultando

em variação na resistência elétrica da pele. E a alteração numa corrente elétrica é a base do funcionamento de qualquer dispositivo de entrada.





e uma redução a leve para baixo. O objetivo é controlar esses movimentos e alinhar a nave com os alvos que se aproximam.

O termo “acoplamento mental” talvez seja enganoso, pois essa ligação se dá não pela mente, mas pelo sistema nervoso. Eis a questão: se ambos são interdependentes, faz sentido separá-los? Muitas pessoas mostram-se capazes, após cerca de 20 minutos, de exercer um razoável grau de controle, quase sempre sem saber como o conseguem. Umas, de forma consciente, tensionam e relaxam levemente o corpo; outras simplesmente pensam “Para cima” ou “Para baixo” e deixam o sistema nervoso fazer o resto.

O acoplamento mental tem outras aplicações mais práticas. Uma pessoa totalmente parálitica ainda conta com o recurso de gerar, de modo consciente, efeitos de RGP, embora não controle os músculos. Um dispositivo de RGP ligado a um paralítico foi acoplado com sucesso a um robô Topo, através de um computador Apple, dando a essa pessoa a possibilidade de controlar o robô. Uma outra aplicação prática estaria no espaço sideral, sob gravidade zero. É extremamente difícil operar controles mecânicos sem a gravidade para contrabalançar a energia aplicada. Um dispositivo de RGP poderia ser o substituto perfeito para muitos controles mecânicos.

## Um software que capta o ânimo

Entre as aplicações exóticas está a captação da disposição de ânimo. Os softwares educacionais, em particular, poderiam se beneficiar com a retroalimentação do estado interior do usuário por meio da RGP. O indivíduo usaria em torno do pulso uma faixa com eletrodos, que o software calibraria no estado de relaxamento normal. Se, num dado momento, o aparelho registrasse acentuado aumento de tensão, significando estar o usuário em dificuldades, ele ajustaria o andamento do programa. Softwares comerciais seriam capazes de detectar a tensão mental de usuários — e de sugerir um descanso.

A rapidez do desenvolvimento dessa tecnologia depende não só da criação de dispositivos mais precisos e de respostas mais rápidas, como também de nossa habilidade em interpretar os efeitos. O primeiro problema está em que a RGP ocorre 2 segundos após o evento e leva de 2 a 10 para desaparecer. Os dispositivos atuais superam em parte o problema, medindo o ritmo de mudança da reação e não sua intensidade. O segundo problema está na imprecisão das nossas deduções a partir dos dados fornecidos pela RGP. Uma queda brusca na resistência elétrica da pele indica tensão ou estresse, mas não se pode precisar se a causa é agradável ou dolorosa.

Como se vê, apesar de todos os problemas, a RGP descortina possibilidades entusiasmantes. E a Atari não é a única a levá-la a sério: a Commodore ofereceu a respeitável soma de 2 milhões de dólares à Behavioral Engineering pelos direitos do dispositivo e do software de RGP — devidamente recusados!



## O detector de mentiras

Numa investigação ou julgamento, um dos maiores problemas enfrentados pela polícia e sistema judicial é saber se um suspeito ou uma testemunha está mentindo. Embora um perito possa detectar pistas mínimas (como alterações na coloração da pele e na respiração, por exemplo), a medição desses fatores não está definida nem catalogada — e muito menos é admissível como prova. Por isso se pesquisa no sentido de desenvolver meios objetivos de distinguir entre declarações falsas e verdadeiras. Um dos resultados nesse sentido foi o polígrafo, ou detector de mentiras.

O polígrafo, na verdade, consiste apenas num medidor de resistência elétrica. Supõe-se que o indivíduo sofrerá aumentos significativos de tensão ao dizer uma mentira ou ouvir palavras associadas ao crime, e que isso se revelará por uma queda na resistência elétrica da pele.

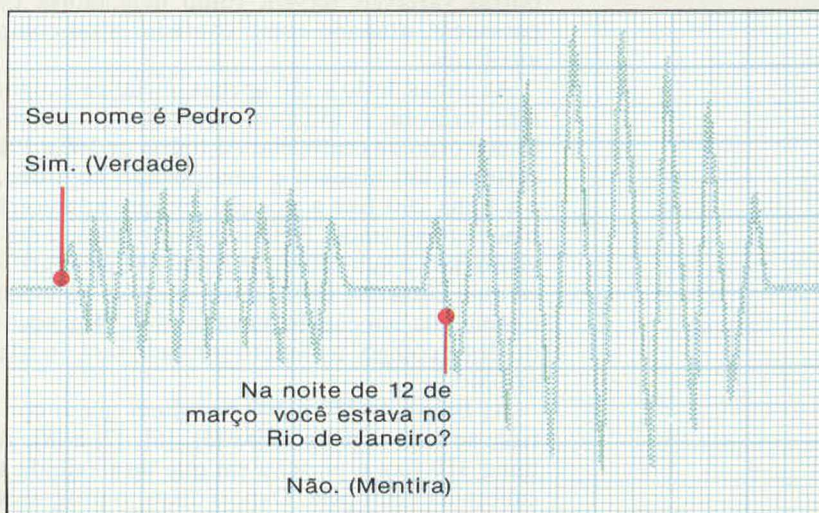
O polígrafo vem sendo objeto de muita controvérsia. É defendido pela polícia, nos EUA em especial, mas seus críticos alegam que o nosso conhecimento sobre o fenômeno RGP é insuficiente para uma interpretação segura e coerente. Além do fato de indivíduos diferentes reagirem de forma também diferente, sejam eles culpados ou inocentes.

### Na conquista de um emprego

Às vezes se usa a RGP na avaliação de candidatos a empregos, alimentando um detector de mentiras na análise das respostas do entrevistado. Na foto acima, o interessado recebe os resultados do teste que acabou de fazer.

### Mentiras inofensivas

O detector de mentiras mede mudanças de RGP causadas pelas reações emocionais do indivíduo. Mas é pouco provável que esse aparelho forneça respostas tão precisas quanto as que vemos abaixo. Reações emocionais variam conforme a sensibilidade do indivíduo a certos temas (se os acha constrangedores, por exemplo) e não dependem apenas da falsidade ou veracidade das respostas.



# CENTOPÉIA FAMINTA

**Diversos programas criados para o TK 85 são adaptáveis ao TK 90X. Analisamos aqui duas versões de um mesmo jogo para esses micros. Aproveite para fazer suas próprias modificações.**

Nosso programa é um jogo de ação onde você dirige uma centopéia esfomeada, em busca de comida, dentro de um jardim. O controle se faz com as teclas [5], [6], [7] e [8] (as setas de controle do cursor), e você deve evitar que a centopéia “morda” a si mesma ou os tijolos que delimitam o jardim. Se isso acontecer, ela morrerá.

Os alimentos são pequenas larvas (números) que surgem ao acaso na superfície do jardim. Podem ser mais ou menos gordas: quanto mais gorda, maior o número por ela representado, e cada número que a centopéia devora a faz crescer proporcionalmente em segmentos.

Mas o jardim permite que a centopéia tenha no máximo quatrocentos segmentos. Como ela já “nasce” com um, significa que poderá crescer até 399. Seu objetivo final, então, consiste em fazer com que todo o jardim seja ocupado pela centopéia.

## A versão para o TK 85

As linhas 20, 30 e 40 desenham na tela os “tijolos”. As variáveis L e C armazenam a posição do último segmento da centopéia, e as variáveis Y e X armazenam a posição da cabeça. O string A\$ armazena os movimentos que a cauda deve realizar para chegar à posição da cabeça. As outras variáveis servem como indicadores de passagem, atuando como contadores ou como auxiliares.

As linhas de 40 a 130 apenas inicializam algumas variáveis. A 150 mostra na tela um segmento do corpo da centopéia. As linhas de 160 a 190 sorteiam um número e uma posição no vídeo, e a linha 200 fixa a posição em que aparecerá o número sorteado. Feito isso, o programa é desviado para a sub-rotina da linha 490, que verifica qual caractere existe nessa posição do vídeo e armazena seu código na variável E. Dependendo desse código, o número sorteado será ou não exibido.

As linhas de 250 a 300 permitem que o controle se faça com as teclas [5], [6], [7] e [8]. A linha 310 fixa a posição em que será exibida a cabeça da centopéia. Depois disso, o programa se desvia para a sub-rotina da linha 490.

As linhas 330 e 340 verificam se a centopéia simplesmente andou, se ela mordeu um tijolo, se devorou a si mesma ou a uma larva. Em cada caso, algumas das linhas entre 350 e 480 são executadas.

As linhas 350, 355 e 360 são executadas se a centopéia morre. Já as 370, 380 e 390 são executadas no caso de ela devorar alguma larva. A linha 400 exhibe a cabeça no vídeo e a 420 apaga a cauda. O movimento da cauda é produzido nas linhas 430 e 440.

A variável B indica se vai ou não haver crescimento, Z conta quantos segmentos a centopéia aumentou, e W armazena o total de alimentos na superfície do canteiro.

As linhas 410 e 470 estão relacionadas com o conteúdo de B, e as 460 e 480 desviam incondicionalmente o programa para a linha 150, onde um outro segmento do corpo da centopéia é exibido.

O programa deve ser executado no modo slow, extremamente lento quando o micro tem mais de 2 Kbytes de RAM. Isso ocorre porque, quando o aparelho possui menos de 3,25 Kbytes de RAM disponíveis, gerencia o armazenamento do vídeo de uma forma mais econômica.

Caso seu micro tenha 16 Kbytes, para aumentar-lhe um pouco mais a velocidade, você poderá “enganá-lo” alterando o valor da variável do sistema que armazena o endereço do último byte existente na RAM: a RAMTOP ou RTP. Para isso, basta comandar diretamente (antes de digitar o programa):

```
POKE 16388,200
```

e

```
POKE 16389,11
```

Isso fará o micro “pensar” que está apenas com cerca de 3 Kbytes de RAM, tornando-se um pouco mais rápido. Quanto mais à esquerda estiver a imagem no vídeo, menos memória será necessária para o armazenamento. Devido a isso, o programa para o TK 85 (e similares) constrói a moldura descentralizada, mais à esquerda.

Devido à lentidão, impõe-se que os números sorteados na linha 170 (larvas) sejam sempre superiores a 3, para que o crescimento se acelere mais.

## A versão para o TK 90X

O TK 90X possui muitos recursos visuais e sonoros que podem ser usados em quase todos os tipos de programas. Em nosso caso específico, esses recursos são essenciais para transformar o programa Centopéia (lento e um pouco pobre no



TK 85) num programa ao nível de qualquer bom videogame.

Uma grande vantagem do TK 90X sobre o TK 85 é patente: a velocidade de execução do programa. O TK 85 possui duas velocidades de processamento. No modo slow, o vídeo se apresenta permanentemente ao usuário, o que torna a execução lenta. No modo fast, a execução acontece mais rápida, porém o vídeo não é mostrado.

Num jogo de ação, precisa-se utilizar o modo slow. O TK 90X funciona sempre mostrando o vídeo, porém com uma velocidade de processamento próxima à do fast.

Outro fator que também torna o programa lento no TK 85 é a utilização da sub-rotina da linha 490, necessária para checar a posição do vídeo em que ocorrerá a próxima impressão. No TK 90X existe uma instrução extremamente simples, a SCREEN\$, utilizada nas linhas 200 e 310, que substitui com vantagem a sub-rotina usada no TK 85.

O uso das cores também é muito simples. Os comandos BORDER, PAPER e INK são usados na linha 10 de modo a gerar uma tela branca com caracteres pretos e uma moldura vermelha.

Acrescentamos som ao programa com o comando SOUND. A linha 250 gera um sinal cada vez que o teclado é verificado, e a 350 gera um sinal de "morte" da centopéia (sua marcha fúnebre!).

Outro grande melhoramento possibilitado pelo TK 90X é o uso de caracteres determinados pelo usuário. Foram definidos três deles: um para os "tijolos" do jardim (linha 30), um para o corpo da centopéia (linha 150) e outro para sua cabeça (linha 400). Os três estão ilustrados em detalhes na figura para que possam ser reproduzidos.

Por fim, como fica evidente, muitas vezes utilizamos linhas com multiinstrução, isto é, várias instruções numa mesma linha do programa.

Como no TK 90X não temos problemas com a velocidade de processamento, as larvas são representadas por qualquer número entre 1 e 9. Além de muito mais rápido, mais estético e mais interessante, o programa no TK 90X também fica mais compacto.

## Outras opções

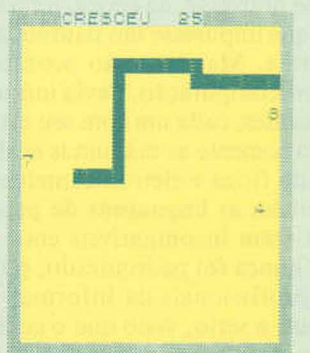
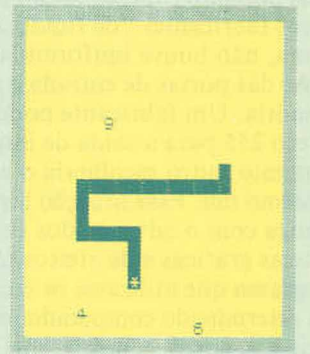
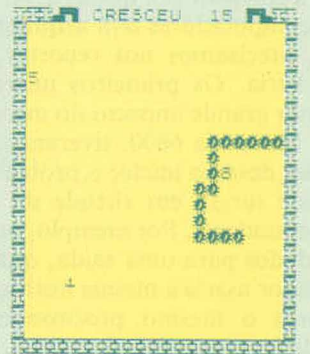
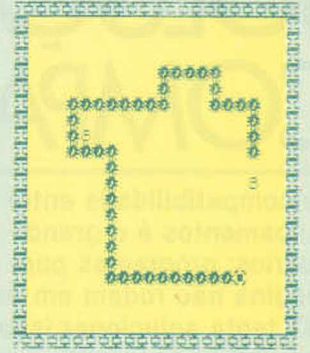
Existem várias maneiras de se produzir um programa como este; utilizamos apenas uma delas. No caso específico do TK 85, certamente não é a melhor (seria muito mais rápido imprimir no vídeo, colocando, por meio do comando POKE, caracteres diretamente na memória de vídeo).

No TK 90X, entretanto, a lógica usada é bastante viável (nesse caso, acabaria sendo bem mais complicado utilizar o POKE para colocar caracteres diretamente no vídeo).

Constituiria um exercício bastante instrutivo tentar tornar o programa para o TK 85 mais rápido usando POKE ao invés de PRINT. Ainda mais instrutivo seria adaptar esse programa para o TK 90X, dada a sua complexidade. Fica como sugestão para um fim de semana chuvoso.

```

5 REM PARA O TK90X
6 BORDER 2: PAPER 7: INK 0
7 FOR A=0 TO 21
8 PRINT AT A,0;" ";TAB 21;" "
9 AT 0,A;" " AT 21,A;" "
10 NEXT A
11 LET L=11: LET C=L: LET Y=L
12 LET X=C: LET A$="" : LET C$=""
13 LET Z=1: LET B=0: LET W=B
14 PRINT AT Y,X;" "
15 IF W>15 THEN GOTO 250
16 LET S=INT (1+9*RAND)
17 LET T=1+20*AND
18 LET U=1+20*AND
19 GOTO 200
20 IF SCREEN$ (T,U)<>" " THEN
21 GOTO 130
22 PRINT AT T,U;S
23 LET W=W+S
24 LET D$=INKEY$: SOUND .01,25
25 IF D$<>" " THEN LET C$=D$
26 IF C$<>"5" AND C$<>"8" AND
27 C$<>"7" AND C$<>"8" THEN GOTO 25
28
29 LET A$=A$+C$
30 LET Y=Y+(C$="6")-(C$="7")
31 LET X=X+(C$="8")-(C$="5")
32 PRINT AT Y,X;
33 GOSUB 490
34 IF E=0 THEN GOTO 400
35 IF E<>8 AND E<>128 THEN GOT
36 PRINT AT 0,4;"CRESCEU ";Z-
37
38 PAUSE 200
39 RUN
40 LET B=B+E-20
41 LET Z=Z+E-20
42 LET U=W-E+20
43 PRINT " "
44 IF B>0 THEN GOTO 470
45 PRINT AT L,C;" "
46 LET L=L+(A$(1)="6")-(A$(1)="7")
47 LET C=C+(A$(1)="6")-(A$(1)="5")
48 IF LEN A$>0 THEN LET A$=A$(
49 TO
50 GOTO 150
51 LET B=B-1
52 GOTO 150
53
54 REM PARA O TK85
55
56 CLS
57 FOR A=0 TO 21
58 PRINT AT A,0;" ";TAB 21;" "
59 AT 0,A;" " AT 21,A;" "
60 NEXT A
61 LET L=11
62 LET C=L
63 LET Y=L
64 LET X=C
65 LET A$=""
66 LET Z=1
67 LET B=0
68 LET W=B
69 PRINT AT Y,X;" "
70 IF W>15 THEN GOTO 250
71 LET S=INT (4+6*AND)
72 LET T=1+20*AND
73 LET U=1+20*AND
74 PRINT AT T,U;S
75 GOSUB 490
76 IF E<>0 THEN GOTO 180
77 PRINT " "
78 LET W=W+S
79 IF D$<>" " THEN LET C$=D$
80 IF C$<>"5" AND C$<>"8" AND
81 C$<>"7" AND C$<>"8" THEN GOTO 25
82
83 LET A$=A$+C$
84 LET Y=Y+(C$="6")-(C$="7")
85 LET X=X+(C$="8")-(C$="5")
86 PRINT AT Y,X;
87 GOSUB 490
88 IF E=0 THEN GOTO 400
89 IF E<>8 AND E<>128 THEN GOT
90 PRINT AT 0,4;"CRESCEU ";Z-
91
92 PAUSE 200
93 RUN
94 LET B=B+E-20
95 LET Z=Z+E-20
96 LET U=W-E+20
97 PRINT " "
98 IF B>0 THEN GOTO 470
99 PRINT AT L,C;" "
100 LET L=L+(A$(1)="6")-(A$(1)="7")
101 LET C=C+(A$(1)="6")-(A$(1)="5")
102 IF LEN A$>0 THEN LET A$=A$(
103 TO
104 GOTO 150
105 LET B=B-1
106 GOTO 150
107
108 LET E=PEEK (PEEK 16398+256+
109 PEEK 16399)
110 RETURN
    
```





# SOLUÇÃO PARA COMPATIBILIDADE

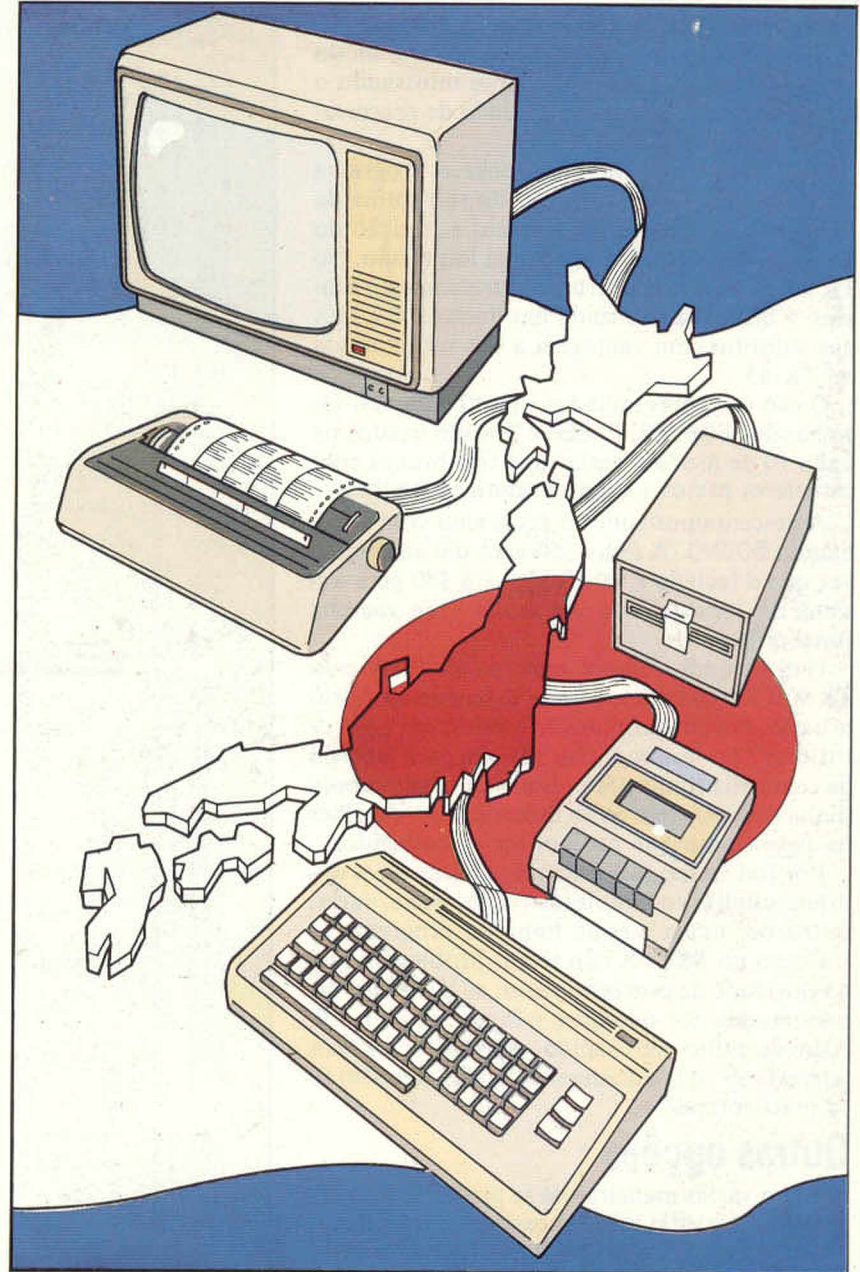
**A incompatibilidade entre os diversos equipamentos é o grande problema dos usuários: programas para determinada máquina não rodam em outra. O padrão MSX tenta solucionar isso.**

Para entender por que as diversas marcas de microcomputadores têm arquiteturas tão diferentes, precisamos nos reportar à história dessa indústria. Os primeiros microprocessadores a causar grande impacto no mercado, o Intel 8080 e o Motorola 6800, tiveram suas instruções fixadas desde o início; o problema da compatibilidade surgiu em virtude da versatilidade dos processadores. Por exemplo, para enviar um byte de dados para uma saída, qualquer microcomputador usaria a mesma instrução, desde que utilizasse o mesmo processador. Mas a saída poderia ter qualquer endereço, entre centenas ou milhares.

Os primeiros micros foram produzidos por diversos fabricantes "de fundo de quintal"; dessa forma, não houve uniformidade sobre a localização das portas de entrada e saída no mapa da memória. Um fabricante poderia escolher o endereço 255 para a saída de impressora paralela, enquanto outro escolheria o endereço 254 para o mesmo fim. Essa situação tornou-se ainda mais caótica com o advento dos chips controladores de telas gráficas e de efeitos sonoros. Qualquer programa que utilizasse os recursos especiais de um determinado computador certamente não rodaria em outro, a menos que sofresse consideráveis modificações.

Se houvesse, na época, uma indústria mais forte, que impusesse um padrão, a situação seria diferente. Mas isso não aconteceu: no início da microcomputação, havia inúmeros pequenos fabricantes, cada um com seu estilo e seus padrões. Não somente as máquinas acabavam se apresentando física e eletronicamente diferentes, como também as linguagens de programação utilizadas eram incompatíveis entre si. Mesmo o BASIC nunca foi padronizado, pois na década de 70 os profissionais da informática não o levaram muito a sério, visto que o consideravam linguagem para principiantes.

No final da década de 70 e início da de 80, os microcomputadores se desenvolveram a enorme velocidade. Os projetos pioneiros — o Apple, por exemplo — incorporaram refinamentos como gráficos e cores; mas para pôr em prática essas inovações, os fabricantes desenvolveram suas próprias versões do BASIC, que assim proliferaram. Se por um lado essa expansão possibilitou



ampla liberdade de escolha ao consumidor, por outro gerou muitas frustrações aos proprietários, aos fabricantes de micros e aos projetistas de software. O orgulhoso proprietário de um SORD ficaria desesperado para obter o último jogo da moda; mas, se o programa fosse escrito para o Sinclair, ele teria de esperar muito pela versão compatível com sua máquina.

O fabricante que tem um novo micro a lançar sabe que existe pouquíssimo software para seu equipamento até que haja um número substancial de usuários. Mas, com pouco ou nenhum

## Rumo ao sucesso

O padrão MSX é a rota do Japão para entrar nos mercados mundiais de computação. Se obtiver sucesso, poderá dominar esse mercado, da mesma maneira que dominou as áreas de aparelhos de som e das câmaras fotográficas.





software disponível, o potencial de vendas de sua máquina é muito limitado, pondo em risco todo o investimento feito no desenvolvimento do produto.

Se você cria software comercial, suas vendas limitam-se (na melhor das hipóteses) às pessoas que possuem o modelo de computador para o qual se destina o programa. Suponhamos que você crie um jogo de aventuras para o TK 90X chamado Masmorras do Inferno, com personagens como o Fantasma Invisível, o Carcereiro Sádico e incontáveis Antros do Terror para apanhar os incautos. A equipe de pesquisa de mercado afirma ser imenso o potencial de vendas, desde que se lance o produto a um certo preço e se vendam, no mínimo, 65.000 cópias. Entretanto, o número de proprietários do TK 90X não basta para garantir a vendagem; é necessário pelo menos mais uma versão. O custo de produção de novas versões, para, digamos, o CP 400 e o Apple, aumentaria o valor unitário em 20%; a esse preço, as vendas ficariam abaixo de um nível economicamente viável. É esse o dilema que frustra muitos projetistas de software em potencial.

O problema da compatibilidade de software não foi ignorado pela indústria de computadores. A óptica individualista do Ocidente não favorece a padronização; mas, no Extremo Oriente, os fabricantes preferem que as coisas sejam mais sistemáticas, organizadas e uniformes, especialmente quando isso se transforma em lucro.

A ASCII/Microsoft está tentando organizar o caos. A companhia resulta de uma fusão da Microsoft Corporation com a ASCII, uma bem-sucedida editora japonesa de revistas populares. Como linha secundária, a ASCII também publica software comercial e, quando a Microsoft se impõe o desafio de entrar no "impenetrável" mercado japonês, a ASCII foi a primeira opção de parceria num empreendimento conjunto: a Microsoft possuía a tecnologia e a ASCII a experiência de marketing.

A Microsoft Corporation construiu sua reputação sobre a versão do BASIC mais aceita como padrão, o MBASIC, adotado por fabricantes de micros do mundo todo. Mesmo assim, não havia garantias de que um programa em MBASIC rodasse em todos os computadores que usassem essa linguagem, pois sempre que houvesse recursos especiais haveria incompatibilidade de hardware.

O BASIC da Microsoft foi vendido para numerosos fabricantes japoneses através da ASCII/Microsoft. Mas isso também não resolveu o problema de compatibilidade de hardware e software. Como solução, a ASCII/Microsoft criou, em cooperação com os principais fabricantes japoneses, um padrão de aceitação internacional. O resultado desses esforços foi o chamado padrão MSX, que inclui requisitos básicos de hardware (centrados no microprocessador Z80 e alguns outros chips) e também uma linguagem padronizada. No final de 1985, os mi-

cro MSX já faziam bastante sucesso no mercado internacional; no Brasil, os pioneiros foram o Gradiente Expert e o Sharp Hot-Bit.

## Especificações do MSX

O BASIC MSX assemelha-se muito ao MBASIC (da Microsoft), mas com várias melhorias que aproveitam as modernas possibilidades gráficas e sonoras. Seguem-se os novos comandos: SCREEN, para especificar o modo de tela, o tamanho dos sprites, o bip das teclas, a velocidade de transmissão de cassete e as opções da impressora; LOCATE, que posiciona os caracteres na tela; COLOUR seleciona uma das dezesseis cores de linha e de fundo; PUT SPRITE estabelece atributos dos sprites; CIRCLE faz círculos e elipses; DRAW executa desenhos; LINE traça retas entre coordenadas especificadas; e PAINT preenche figuras com uma determinada cor. Há também o comando KEY, para alocar strings a teclas de função. Outros comandos permitem introduzir valores na RAM de vídeo (VPOKE), inscrever valores nos registradores do chip de efeitos sonoros (SOUND) e controlar o motor do gravador cassete (MOTOR).

O MSX, entretanto, envolve não apenas a padronização do software, mas também especificações quanto ao hardware. A CPU é um processador Z80 funcionando a 3,58 MHz. Deve haver no mínimo 32 Kbytes de ROM para armazenar o software MSX, e no mínimo 8 Kbytes de RAM. Não há limite máximo para a capacidade de ROM e de RAM. Um computador MSX deve incorporar um chip controlador de vídeo TMS9918A, da Texas Instruments (ou equivalente), e um AY38910—um chip gerador de som com três vozes. O vídeo deve apresentar 24 linhas, com 32 ou 40 colunas. No momento, não há recursos para apresentação em 80 colunas. Quanto à resolução, exige-se 256 x 192 pixels.

Escolheu-se a fita cassete como meio básico de armazenamento de programas e dados; ela deve usar o sistema de codificação FSK, com velocidade de transmissão de 1.200 ou 2.400 bits por segundo. O teclado também é padronizado, e inclui teclas de função e caracteres gráficos. No Brasil, a Gradiente optou por dotar o Expert com o gerador internacional de caracteres, que permite grafar todas as letras e acentos da língua portuguesa da mesma forma que nas máquinas de escrever.

Uma entrada para cartuchos de ROM permite a utilização de softwares aplicativos. Há um conector de cinquenta pinos para entrada e saída, e interface para dois joysticks.

Os formatos dos discos também são padronizados, assim como o sistema operacional em disco, o MSX-DOS, que equivale funcionalmente ao MS-DOS e permite a leitura de arquivos de dados gravados com esse sistema. Parece ser compatível, ainda, com o popular sistema operacional CP/M 2.2. Também se estabeleceram formatos para os disquetes de 3 1/2 (Sony e Gradiente), 5 1/4 e 8 polegadas.



# Arquitetura do MSX

**CPU**  
O popular microprocessador Z80 é a base de todas as máquinas MSX.

**Dispositivos adicionais**  
As máquinas MSX podem incorporar qualquer dispositivo extra que o fabricante queira incluir.

**Chip de som**  
Um 8910 proporciona som com efeitos especiais.

**Gravador cassette**  
As máquinas MSX operam com gravadores comuns em alta velocidade: 1.200 ou 2.400 bauds.

**Teclado**  
Todas as máquinas MSX têm as mesmas teclas, embora o posicionamento possa variar.

**Entrada para cartucho**  
Padronizada, permite a inserção de qualquer cartucho MSX.

**Unidade de disco**  
O MSX inclui especificações para discos que utilizam um formato padrão.

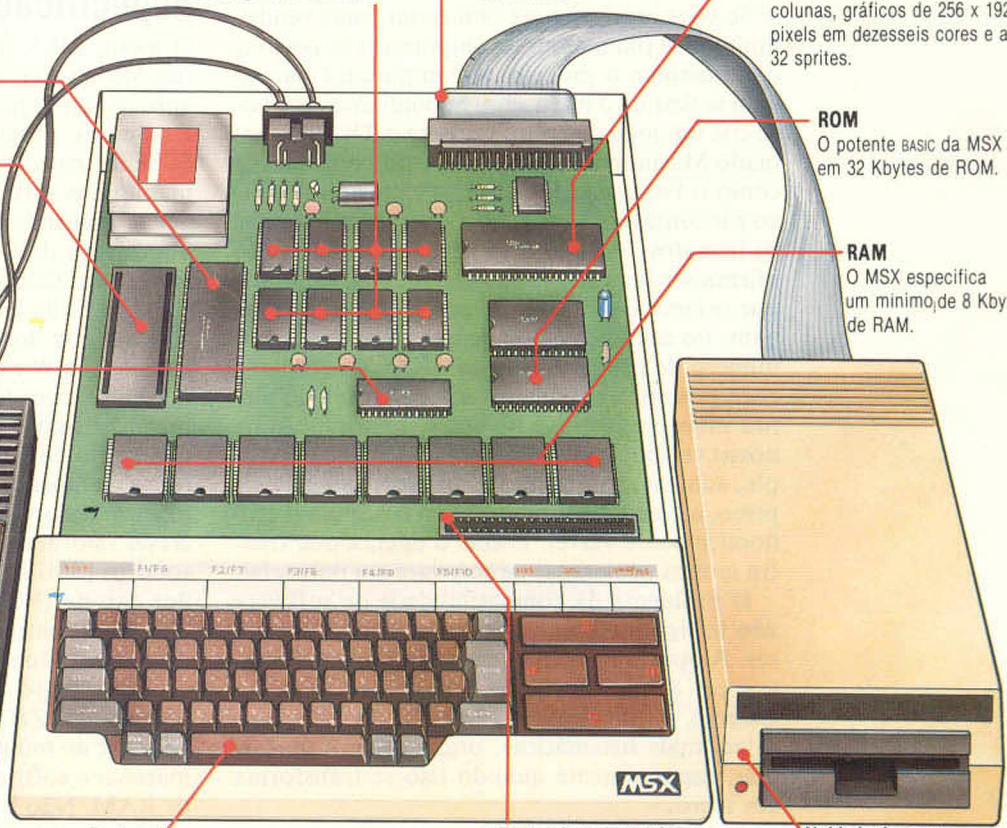
**RAM de vídeo**  
Separada em alguns sistemas, libera áreas de memória para os programas em BASIC.

**Conector de expansão**  
Todas as máquinas têm conector de expansão padronizado.

**Chip de vídeo**  
Um Texas 9918 ou equivalente proporciona a apresentação na tela, com 24 linhas de 32 colunas, gráficos de 256 x 192 pixels em dezesseis cores e até 32 sprites.

**ROM**  
O potente BASIC da MSX reside em 32 Kbytes de ROM.

**RAM**  
O MSX especifica um mínimo de 8 Kbytes de RAM.



Isso tudo significa que programas escritos para uma máquina MSX e armazenados em qualquer disco certamente rodarão sem problemas em qualquer outro computador da linha MSX, com plena utilização de seus recursos gráficos e sonoros. As vantagens são óbvias, tanto para o fabricante como para o consumidor.

O padrão MSX corre apenas alguns riscos. O primeiro é que qualquer "padrão", uma vez adotado, não conseguirá incorporar as inovações da área. Se, por exemplo, surgir no mercado um novo chip controlador de vídeo com recursos muito superiores, os programas MSX não poderão aproveitá-lo e deixarão o campo aberto aos concorrentes capazes de utilizá-lo.

O segundo risco está nos microprocessadores de 8 bits, dos quais o mais popular é o Z80, pois por princípio são incapazes de endereçar diretamente mais de 64 Kbytes de memória principal; também não podem processar de uma só vez dados de valor superior a 256.

À época do lançamento do MSX, parecia provável que nos próximos dez anos o mercado de micros passasse a ser dominado pelos microprocessadores de 16 bits. Se o padrão MSX conseguisse o sucesso esperado, sua consequência

mais positiva seria a de alertar os fabricantes que a padronização deve vir cedo — e não tarde — no projeto de qualquer inovação.

O MSX poderia facilitar a venda imediata dos produtos com base no Z80, mas dificilmente isso seria vantajoso a longo prazo. Seu real impacto estava em convencer o resto do mundo de que a padronização é importante.

No campo dos micros de 16 bits, a IBM provou que "contra a força não há argumentos", ao impor, com seu computador pessoal, na prática, um padrão. O MSX seria capaz de fazer o mesmo para os micros de 8 bits? Até fins de 1985, ele foi apoiado por muitos fabricantes, como a Yamaha, JVC, Hitachi, Sony, Sanyo, National, Pioneer, Canon, Fujitsu e Mitsubishi do Japão, a americana Spectravideo, a coreana Daewoo e Gradiente e Sharp no Brasil. Contudo, nenhum fabricante europeu entrara para o clube. Somente o tempo e a reação do mercado diriam, nos próximos anos, se havia necessidade de mais padronização, ou do tipo de inovação individualista que se espera de empresários como os pioneiros da computação, que, com seu engenho, criaram em precárias condições os primeiros microcomputadores.

**Convenções do projeto**  
Para que os programas e dispositivos adicionais sejam compatíveis com todos os sistemas MSX, o hardware de um micro MSX segue regras estritas. Uma vez que a máquina se conforma à configuração básica mostrada aqui, os projetistas podem acrescentar dispositivos extras.



# PADRÃO MSX

**O MSX é o padrão para micros adotado por mais de doze fabricantes japoneses, alguns bastante conhecidos pelos seus produtos eletrônicos. Vamos examinar dois desses equipamentos: o Hit-Bit e o HX-10.**

O padrão MSX prescreve: o microprocessador a ser usado (Z80), a quantidade mínima de ROM (32 Kbytes) e de RAM (8 Kbytes), o tipo de chip de gráficos e som, as teclas que devem estar presentes (embora o layout do teclado possa variar), o número mínimo de interfaces (e seus modelos), as telas gráficas e de texto, e a linguagem BASIC contida na ROM.

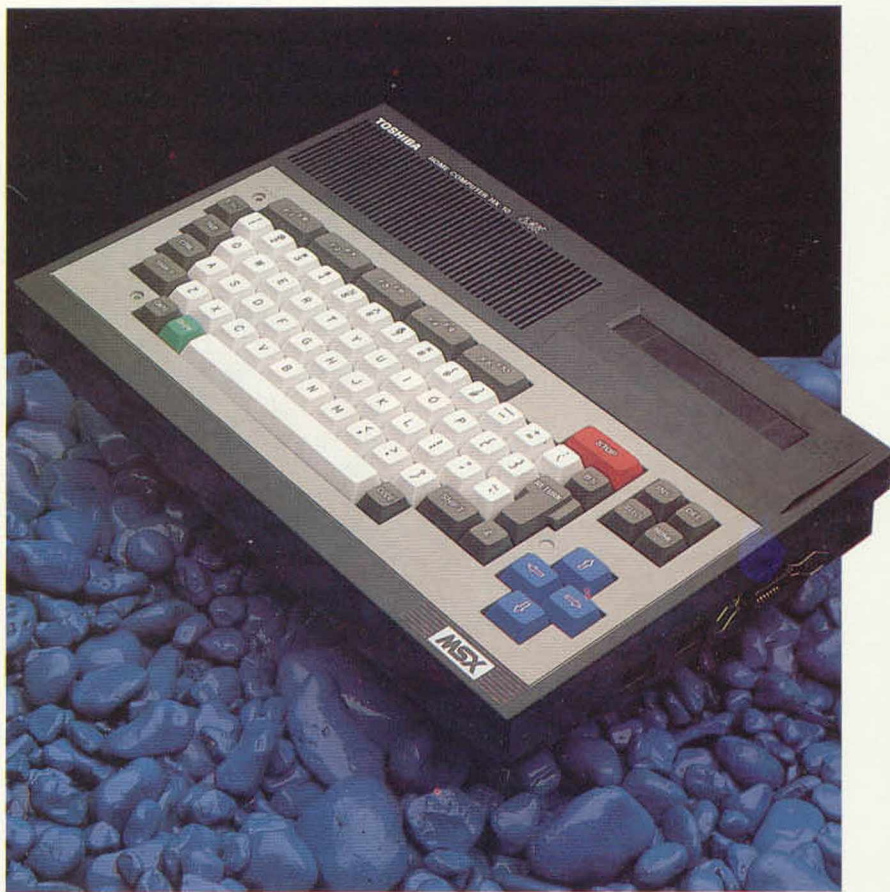
Sendo o MSX um projeto padronizado, é de se esperar que todos os micros MSX se assemelhem. Os diversos modelos, no entanto, admitem flexibilidade: a quantidade de memória que pode ir além daquele mínimo, o tipo de teclado e o número extra de interfaces. Assim, a maioria dos equipamentos MSX atingiu especificações acima dos requisitos mínimos.

Entre os modelos estrangeiros, o Hit-Bit da Sony e o HX-10 da Toshiba possuem um teclado de boa qualidade, mas há quem ache as teclas demasiadamente sensíveis ao toque. Os dois micros vêm com 64 Kbytes de memória principal e 16 Kbytes adicionais de RAM, exclusivos para o funcionamento da tela. Esse total de 80 Kbytes é superior ao fornecido pela maioria dos micros. Ambos os modelos possuem interfaces para impressora com saída paralela padrão Centronics e duas saídas para joysticks — itens frequentemente opcionais.

Esperava-se que os micros MSX se tornassem máquinas baratas, mas as flutuações no câmbio e o aumento dos custos de fabricação empurraram os preços para cima. Outra razão para o encarecimento foi a pressa com que se procurou colocar essas máquinas no mercado europeu.

Uma das primeiras coisas que nos chama a atenção quando ligamos um micro MSX é uma linha com palavras na parte inferior da tela. São comandos da linguagem BASIC, tais como RUN, CLOAD, LIST etc. Cinco teclas de função apresentam as palavras mais usadas, que, na tela, servem de rótulo para as teclas de função, de modo que o usuário não precisa memorizar a função de cada tecla.

Essas teclas são automaticamente definidas quando se liga a máquina, mas é fácil mudar a definição usando-se o comando KEY. Embora só existam cinco teclas, podem ser acessadas até dez funções. Para tanto, pressionam-se, ao mesmo



#### Sistema padronizado

O Toshiba HX-10 possui uma interface para impressora com saída paralela padrão Centronics e saídas para joystick (duas) e para cartucho de ROM. O BASIC do MSX lida com os joysticks da mesma maneira como comanda os movimentos do cursor, de modo que jogos desenvolvidos para um tipo de controle podem automaticamente utilizar outro. À esquerda, o Sony Hit-Bit e seus periféricos.

tempo, a função desejada e a tecla [Shift]. Esta, quando acionada, muda as palavras na parte inferior da tela, apresentando as novas funções designadas às teclas. Cada atribuição de função pode conter até quinze caracteres, mas apenas os sete primeiros aparecem na tela.

O teclado e o editor da tela funcionam concatenados para facilitar a edição. Quatro teclas movimentam o cursor pela tela, podendo-se fazer



Saída para tevê

Saída para cartucho de ROM  
Conforme o projeto padrão MSX

Saída para impressora

Chip adaptador de interface periférica

Este chip, chamado PIA (Peripheral Interface Adaptor), controla a entrada e a saída dos periféricos.

Saídas para joystick

Estas saídas usam o padrão Atari para joysticks.

Controlador de som AY-389 10

Este chip fornece som em três canais.

Chip de vídeo TI TMM 9929

O padrão MSX requer um TI 9918 ou equivalente. O chip 9929 é o equivalente para as telas do sistema PAL.

ROM do BASIC

O BASIC Extended, da Microsoft, está gravado neste chip de 32 Kbytes.

RAM de vídeo

Estes chips, com capacidade de 16 Kbytes, contêm os requisitos de memória para a informação de vídeo, liberando a memória principal para o usuário.

Modulador de PAL

RAM do usuário com 64 Kbytes

CPU Z80A

alterações em qualquer lugar da mesma, bastando escrever em cima dos caracteres existentes. A inserção e supressão de caracteres requerem um único toque. No HX-10, as teclas do cursor têm o mesmo tamanho das outras; já no Hit-Bit, esse conjunto é bem maior e bastante diferenciado. Como tais teclas são intensamente usadas, esse tipo de layout revela-se muito conveniente.

Assim como o hardware e o software do MSX estão cheios de recursos extras, seu BASIC inclui comandos como AUTO e RENUM — que facilitam a codificação — e contém vários comandos para manipulação de som, gráficos e interrupções. Três deles são básicos para a geração de gráficos. O comando LINE traça uma linha ligando dois pontos, mas pode também ser usado para desenhar um quadrado — basta acrescentar a letra B (box) logo após as coordenadas. Adicionando-se as letras BF (box fill), obtém-se

um quadrado preenchido com uma cor. O comando CIRCLE é usado para desenhar elipses e arcos, assim como círculos básicos. E PAINT preencherá totalmente qualquer contorno, mesmo os mais complicados, com uma cor.

O BASIC do MSX inclui muitas outras características úteis, mas o conjunto mais sofisticado de comandos, para “manipulação de interrupções”, pode não ser apreciado de imediato. São comandos muito úteis para programar gráficos em alta velocidade. Há várias situações em que um programa deve realizar uma tarefa, ao mesmo tempo que verifica se outra coisa está acontecendo.

Exemplo típico disso são jogos do tipo Invasores do Espaço. O programa deve manter os alienígenas em constante movimento pela tela e, simultaneamente, verificar se o botão de “fogo” foi acionado. Precisa, portanto, fazer as duas



## Desfraldando a bandeira

### Padrão MSX

<b>CPU</b>	Z80A, de 3,58 MHz
<b>RAM</b>	8 Kbytes, no mínimo
<b>ROM</b>	32 Kbytes, incluindo o BASIC
<b>TELA</b>	Dezesseis cores, gráficos de 256 x 132 pixels, 32 sprites, texto de 24 linhas x 40 colunas (ou 24 x 32); chip de vídeo TI 9918 ou equivalente
<b>SOM</b>	Três canais, podendo ser acessado diretamente pelos comandos do BASIC; chip controlador de som AY 38910
<b>INTERFACES</b>	Para cartucho MSX, monitor de tevê, impressora e para cassete
<b>TECLADO</b>	Tipo QWERTY, com teclas especiais de função, quatro de movimentação do cursor e dez de funções programáveis

### Variações MSX

<b>SONY HIT-BIT</b>	Software de banco de dados gravado em ROM, saída RGB, expansões opcionais de 4 Kbytes de RAM
<b>TOSHIBA HX-10</b>	Bus de expansão, duas saídas para joystick
<b>YAMAHA CX-5</b>	Miniteclado musical e software para a interface MIDI
<b>PIONEER</b>	Interface controladora de videodisco a laser
<b>SANYO MPC100</b>	Caneta óptica opcional e software
<b>JVC HC7GB</b>	Saída RGB
<b>SVI 728</b>	Teclado numérico completo
<b>SPECTRAVIDEO</b>	

Apesar de o padrão MSX requerer um mínimo de 8 Kbytes de memória, todos os fabricantes acima fornecem 64 Kbytes de RAM para o usuário, mais 16 Kbytes para a RAM de tela.

coisas ao mesmo tempo, trocando rápido de tarefa.

A solução do MSX é designar certas coisas como "eventos". São fornecidas instruções ao computador para que esteja atento a um evento. Se algum ocorre, o computador chama automaticamente uma sub-rotina para ocupar-se dele.

O modo gráfico do MSX pode mostrar dezesseis cores com uma resolução de 256 x 192 pixels. Definem-se até 32 sprites de 8 x 8 pixels (ou 16 sprites de 16 x 16 pixels; ou 8 sprites de 32 x 32 pixels). Para o aproveitamento máximo dos sprites, o BASIC do MSX inclui um conjunto completo de comandos dedicados a eles; por exemplo, SPRITE para definir um sprite, e PUT SPRITE para posicioná-lo em qualquer lugar da tela.

Já se encontra disponível um grande número de software em cartuchos. E a promessa de compatibilidade parece estar sendo cumprida — os aplicativos para o HX-10 rodam perfeitamente no Hit-Bit e vice-versa. Isso vale tanto para aplicativos em cartucho como para programas em cassete. Depois de anos de sistemas não compatíveis, parece quase mágico poder tirar um cartucho de um micro e usá-lo em outro. Os fabricantes do padrão MSX, confiantes no sucesso do equipamento graças a essa característica, procuram colocar bem rápido no mercado uma ampla gama de aplicativos.

No Brasil, foi com o padrão MSX que a Gradient e a Sharp ingressaram, em 1985, na área da informática. Seus micros, o Expert e o Hit-Bit, respectivamente, visam sobretudo o mercado educacional. É a contribuição brasileira a esse padrão de computadores bem equipados, divertidos de usar e de preço bem razoável. E que vêm cumprindo à altura as promessas dos fabricantes.

## MSX TOSHIBA HX-10

### MICROPROCESSADOR

Z80A.

### CLOCK

3,58 MHz.

### MEMÓRIA

64 Kbytes de RAM (28 Kbytes disponíveis para o BASIC), 16 Kbytes de RAM de tela, 32 Kbytes de ROM, incluindo o BASIC.

### VIDEO

Modo texto: 4 linhas x 40 colunas; modo gráfico: 256 x 192 pixels, dezesseis cores e até 32 sprites.

### TECLADO

Estilo máquina de escrever, com 68 teclas, sendo quatro para o cursor e cinco de funções programáveis.

### LINGUAGEM

BASIC expandido da Microsoft.

### INTERFACES

Saídas para impressora paralela (padrão Centronics), monitor, tevê, áudio, cassete, joystick, duas aberturas para cartucho de ROM, bus de expansão.

### DOCUMENTAÇÃO

Guia para instalação e guia de referência de programação em BASIC. Ambos são bem feitos, mas não suficientemente completos.





# FORÇA BRUTA



## O favorito

O Cray-1 (foto acima) oferece uma enorme capacidade de processamento, sob o controle de um computador frontal como um IBM, DEC, ou similar. A arquitetura do Cray-1 (diagrama abaixo) apresenta 32 Mbytes de memória principal e 4.888 bytes de espaço de registradores. Todo esse sistema aloja-se em nada menos que 3.400 painéis de circuitos impressos, com mais de 90 km de fios para conectá-los.

Muitas áreas da ciência exigem processamento rápido de imensas quantidades de dados. Para esse fim, são usados "supercomputadores", como o Cray-1, que estudamos neste artigo.

A potência de um computador, em termos simples, é função de seu tamanho de palavra (quantos bits são processados por vez), velocidade de transferência de dados, tamanho da memória principal e velocidade de ciclos da unidade central de processamento. Nos micros, as funções principais da CPU residem num único chip microprocessador. Este pode ser o conhecido Z80, os Intel 8088 e 8086, e o Motorola M68000. Todos utilizam a tecnologia MOS (Metal Oxide Semiconductor, "semicondutor de óxido metálico") para os circuitos lógicos e a memória do chip. O processamento e transferência de dados se dá em paralelo, a 8 ou 16 bits por vez. As frequências de clock dos microprocessadores variam de 1 a 12 MHz.

Todos esses recursos, mesmo parecendo avançados, não são suficientes para que os micros deem conta do colossal volume de processamento

de dados necessário para aplicações como animação de imagens, dinâmica dos fluidos e previsão de tempo.

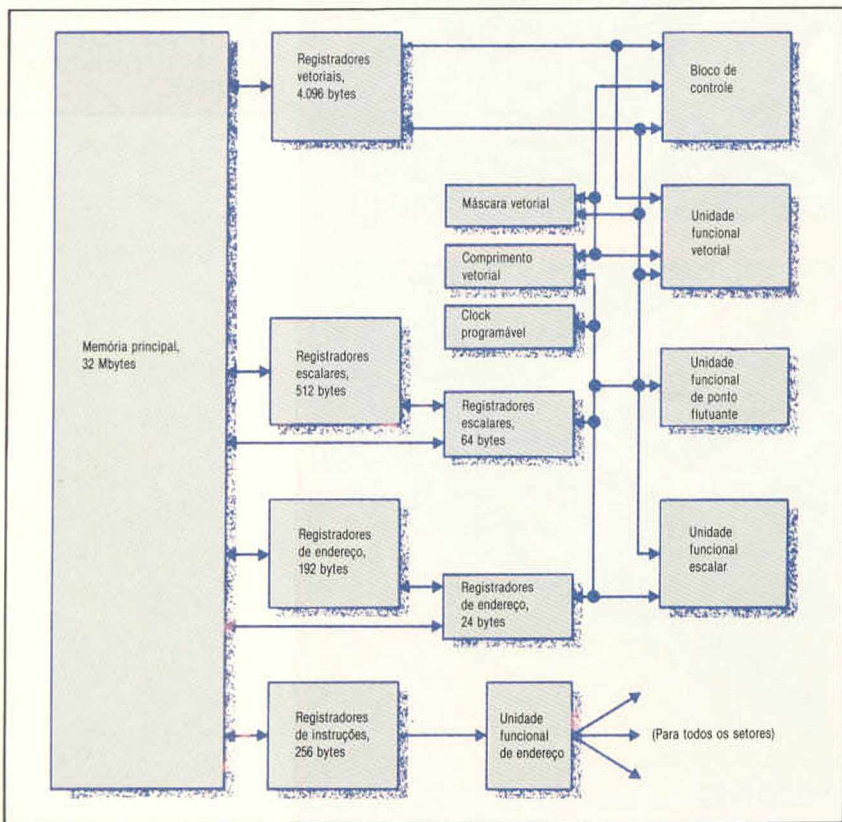
Vejamos um exemplo concreto. Imagine os requisitos necessários para fazer um filme com gráficos animados gerados por computador, com resolução de 6.000 x 6.000 pixels e 24 fotogramas, ou quadros, por segundo. Como os gráficos são animados, os pontos na imagem deslocam-se de quadro em quadro, de modo que a posição de cada ponto terá de ser calculada a cada novo fotograma. Isso significa 864 milhões de cálculos por segundo, cada um bastante complexo e envolvendo dezenas ou centenas de instruções em código de máquina. Isso resulta em bilhões de instruções por segundo.

Os supercomputadores, nome dado às máquinas de maior porte, funcionam de modo muito semelhante aos micros. As instruções e os dados vêm da memória; o processador manipula os dados de acordo com as instruções e armazena os resultados na memória. A diferença principal está na escala e na velocidade de execução dessas operações, bem como na arquitetura dos componentes do hardware.

Vejamos um exemplo típico de supercomputador: o Cray-1 S/4400, fabricado pela Cray Research. O primeiro Cray-1 foi instalado em 1976 e vem se firmando como o mais famoso e o mais popular supercomputador de grande porte. A CPU do Cray-1 ocupa um gabinete semicircular de quase 2 m de altura, parecido com um sofá curvo (o sistema de resfriamento e a fonte de alimentação ficam sob os "assentos"). Sua velocidade resulta da lógica de semicondutores bipolares, que são transistores "comuns", ao contrário do MOS, CMOS, NMOS, FET e tantos outros tipos. A lógica bipolar e a memória provêm de mais de 200.000 circuitos integrados, dispostos em 3.400 painéis; usam-se mais de 90 km de fios para interconectá-los.

O tamanho de palavra usado nas computações do Cray-1 é de 64 bits, ou seja, oito vezes mais que os 8 bits processados por um Z80. O clock do sistema tem uma velocidade de 80 MHz (80 milhões de ciclos por segundo); como resultado, uma adição em 64 bits leva apenas 37,5 nanossegundos, ao passo que uma adição em 8 bits, num Z80 operando a 4 MHz, leva 1,75 microssegundo.

A memória principal no Cray é parte integrante da CPU. Ela contém 32 Mbytes, dispostos sob a forma de 4.194.304 palavras, e transfere até 2,56 bilhões de bytes por segundo. Como se pode ver, o Cray possui um poderoso conjunto de registradores na CPU. Há 72 registradores de en-





dereço (cada um com 24 bits de comprimento), 72 registradores escalares (cada um com 64 bits de comprimento) e oito registradores vetoriais (cada qual com 64 palavras de comprimento). O espaço total dedicado aos registradores na CPU do Cray é, portanto, de 4.888 bytes; em comparação, o microprocessador Z80 possui apenas 26.

## O terminal frontal

Ao contrário da maioria dos computadores, que constituem unidades completas e independentes, os computadores Cray foram projetados para ser extensões de um computador de grande porte ou um minicomputador já existente. Este, chamado computador frontal, funciona como uma interface, aceitando entradas de terminais ou de leitoras de cartão e enviando as saídas para periféricos como impressoras ou fitas magnéticas.

Entre a CPU do Cray e o computador frontal fica o subsistema de entrada e saída do Cray, projetado para agilizar o fluxo de dados para a CPU. Há também uma interface, que adapta o sistema Cray às características específicas do IBM, DEC, Data General ou outro computador usado como terminal frontal. O subsistema de entrada e saída do Cray consiste em dois a quatro processadores de entrada e saída, sendo cada um deles, sozinho, um potente minicomputador.

Veja no boxe o diagrama de blocos da CPU do Cray-1, para uma idéia mais precisa de sua sofisticação. Poderíamos ainda citar o conjunto de instruções, as treze unidades funcionais que operam em paralelo e o processamento vetorial, que permite trabalhar com até 64 pares de operandos com uma única instrução. Em resumo, não exageramos ao dizer que o Cray é extremamente poderoso.

## Aplicações dos supercomputadores

### Meteorologia

Previsões de tempo resultam da coleta de dados em âmbito mundial, comunicações via satélite e modelagem computacional. Um "modelo" elaborado por computador requer centenas de milhões de cálculos para se obterem previsões em questão de horas. E apenas computadores extremamente potentes como o Cray processam tamanha quantidade de dados com a rapidez necessária.

### Dinâmica dos fluidos

Ciência fundamental no projeto de automóveis econômicos, sistemas de refrigeração de usinas nucleares, engenharia aeronáutica e muitas outras aplicações. Prever o movimento de fluidos exige a análise de imensas quantidades de dados, e os problemas se avolumam quando se necessita de resultados em tempo real. Cada partícula num fluido — e há incontáveis trilhões delas — atua sobre todas as outras do conjunto em movimento. Assim, o cálculo do comportamento de um sistema de fluidos exige uma capacidade maciça de processamento de dados.

### Previsões econômicas

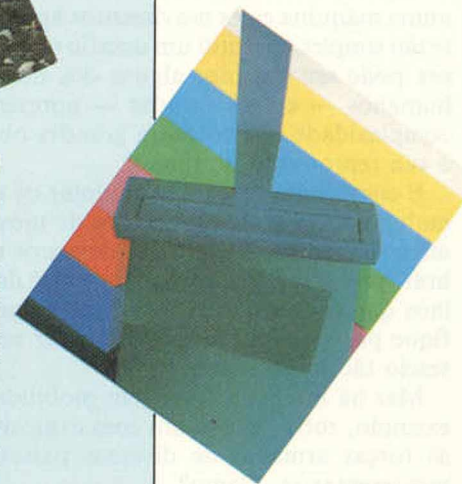
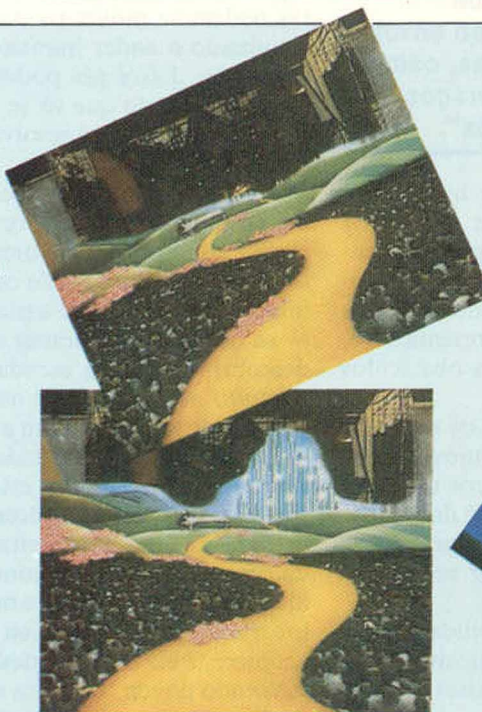
É difícil construir modelos econômicos, pois, como na dinâmica dos fluidos, pequenas mudanças num elemento repercutem sobre todo o sistema. Mesmo modelos simples são de enorme complexidade. Também aqui se recorre a computadores extremamente rápidos e potentes, para que os analistas não tenham de esperar meses pelos resultados.

## Alta velocidade

Para comparar a capacidade de processamento de um supercomputador Cray-1 à de um micro baseado no Z80, considere a produção de uma seqüência de 10 minutos de filme com gráficos animados em alta resolução. Vamos admitir uma resolução de 6.000x6.000 pixels, e 24 fotogramas por segundo; cada pixel precisa ser calculado para cada novo fotograma. Na versão para o Z80, necessita-se de cem instruções em código de máquina, com um tempo de execução médio de dezenove ciclos de clock (4,75 microssegundos). Para o Cray-1, com suas poderosas instruções de processamento vetorial, diremos que são necessárias 25 instruções em código de máquina (um cálculo pessimista), com um tempo médio de execução de quatro ciclos de clock (50 nanossegundos). Para a produção desse filme, um micro baseado no Z80 levaria

$$6.000^2 \times 24 \times 60 \times 10 \times 100 \times 4,75 \times 10^{-6} = 2,4624 \times 10^8 \text{ segundos, ou seja, } 7,8 \text{ anos.}$$

Um Cray-1 levaria  $6.000^2 \times 24 \times 60 \times 10 \times 25 \times 50 \times 10^{-9} = 6,48 \times 10^5 \text{ segundos, ou seja, apenas } 7,5 \text{ dias.}$



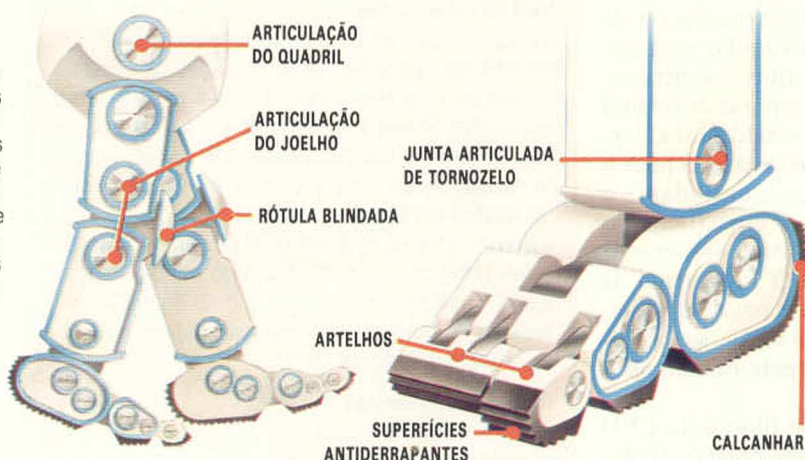
### Revolução visual

Os gráficos gerados por computador estão revolucionando a produção de vídeos e filmes. Criam-se imagens, como as mostradas, por meio da moderna tecnologia de micros e minicomputadores. Trabalhando em maior escala, usou-se o Cray-1 para gerar mais de 20 minutos de imagens no filme *The last starfighter (O último combatente do espaço)*, executando cálculos que ocuparam um micro de 8 bits por mais de quinze anos.

# O PASSO DO ROBÔ

## Caminhando

Andar é um movimento da maior complexidade. Para fazer isso, um robô deveria ter tantas articulações quanto o homem e estar programado para dar seus passos exatamente assim: deve mudar seu centro de gravidade, inclinando o corpo para a frente e levantando a perna que ficou atrás. O ombro gira e os braços balançam, equilibrando o corpo sobre o pé de apoio, com a ajuda dos artelhos, até que o outro pé vá à frente dar apoio.



**Para ser útil, um robô precisa de movimentos. Mas conseguir isso envolve a resolução de vários problemas, como a construção de pernas, pés, braços, mãos, articulações e “músculos”.**

Movimentar o corpo, para um ser humano, é mais do que trivial: é natural. Mas reproduzir numa máquina esses movimentos aparentemente tão simples constitui um desafio que nem sempre pode ser vencido: alguns dos movimentos humanos — como o andar — apresentam tal complexidade que colocam grandes obstáculos à sua reprodução mecânica.

E como fazer, então, para dotar os robôs de mobilidade, isto é, capacidade de mover-se de um lugar para outro? Bem, devemos nos lembrar, primeiro, de que a maior parte dos trabalhos que se pretende de um robô exige que ele fique parado. Assim, fazê-lo andar acaba não sendo tão importante.

Mas há tarefas que exigem mobilidade. Por exemplo, robôs que lidam com explosivos para as forças armadas de diversos países têm de movimentar-se. Como?

Há diversas maneiras de fazer um robô andar,

especialmente se ele tiver pernas. Nesse caso, estas podem se mover para a frente e para trás, simulando o andar humano, arrastando os pés pelo chão. Esses pés podem ter rodas com catracas, de modo que só se movam para diante. E aí o problema está resolvido — só que fica difícil manobrar o robô.

A solução seria fazê-lo erguer os pés do chão para caminhar. Mas aí surge outro problema: ele terá de equilibrar-se em uma das pernas enquanto anda. As soluções são complicadas, mas não impossíveis. Um robô assim até subiria escadas — só ficaria difícil montar o mecanismo para ele descobrir quando a escada acaba.

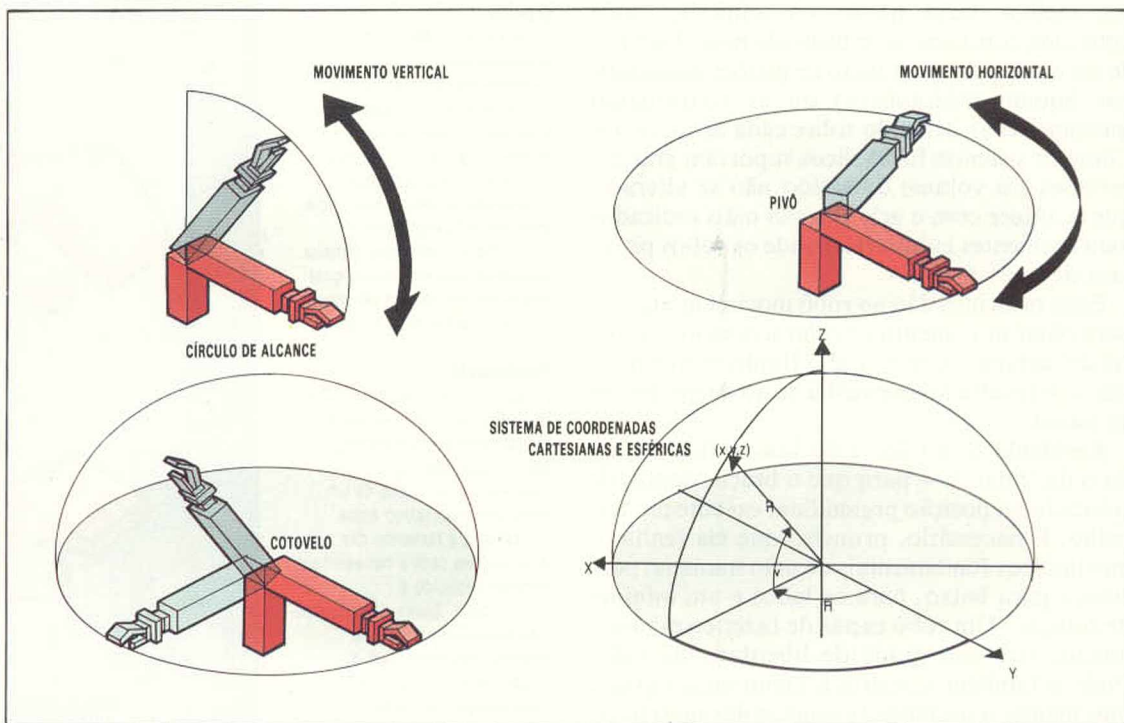
Muito mais simples é usar esteiras, como os tanques de guerra — com a desvantagem de que os robôs não são tão grandes, perdendo, portanto, para os tanques em estabilidade. Além disso, os movimentos carecem de precisão. Para manobrar, uma das esteiras tem de ficar parada, enquanto a outra continua a mover-se. O ideal seria que a esteira parada apenas girasse sobre o solo, em torno de seu centro. Mas isso não acontece, e ela acaba se deslocando, o que, mesmo sendo pouco, significa um problema. No caso do tanque, o deslocamento é corrigido pelo homem que o dirige.



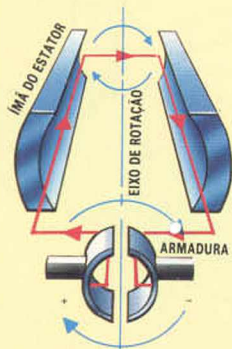


**rotações possíveis**

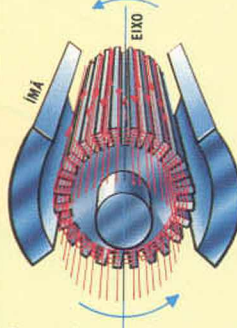
Mesmo o mais simples robô deve poder movimentar o braço como nestas ilustrações. Para isso, o cotovelo é articulado, permitindo um movimento vertical semicircular (de 90°) e horizontal circular (360°). Assim, o braço alcança qualquer ponto num hemisfério cujo raio tem seu tamanho. E a localização desse ponto é dada por coordenadas do seguinte modo (veja o desenho): o ângulo H corresponde ao do arco tangente x/y; o ângulo V é o do arco seno z/R (arc sen z/R). Assim, o programa que leva o braço a um determinado ponto deve conter as coordenadas x,y,z desse ponto. Esses valores são transformados em movimentos dos servomotores, que dirigem então o braço até o ponto exato de trabalho.



**Passo a passo**



Em qualquer motor elétrico, uma corrente na bobina do rotor cria um fluxo magnético contrário ao do estator. É justamente essa oposição de forças que produz movimento no rotor.



O rotor de um motor a passo pode ter centenas de bobinas. Passando-se a corrente de alimentação de uma para outra obtêm-se movimentos pequenos de grande precisão.

Por essas e outras razões, robôs usam rodas, que produzem movimento suave e são mais fáceis de controlar. Ficam faltando, então, um meio de o robô saber exatamente onde está e um mecanismo para movimentar as rodas com precisão.

Para o primeiro problema, a solução está no plano cartesiano, definido por dois eixos de coordenadas. Isso basta para localizar o robô e determinar os pontos para os quais ele tem de se deslocar.

Fazer isso com precisão, contudo, exige dispositivos especiais. Não podemos, nesse caso, usar motores elétricos comuns para movimentar as rodas, as quais, cortada a alimentação, ainda giram pelo menos meia volta, e isso constitui um grave inconveniente para aplicações que exigem precisão. Nesse caso, a solução está no uso de motores a passo, ou graduais, que produzem movimentos precisos, de qualquer amplitude. É esse o tipo de motor usado nas tartarugas, que têm rodas e um suporte para caneta, podendo, assim, executar desenhos.

Mas tudo se complica bem mais quando se trata de reproduzir os movimentos da mão e do braço do ser humano, necessários para que o robô trabalhe. Em primeiro lugar, deve-se estabelecer um sistema para determinar a posição do braço em qualquer instante; depois, esse braço precisa de uma estrutura e de "músculos".

A primeira questão tem solução também pelos eixos de coordenadas; mas nesse caso há um complicador, que é um terceiro eixo (já que o movimento do braço se faz em três dimensões). Com isso, pode-se descrever a posição do braço em qualquer ponto do espaço em que ele atua. É possível, então, contruir um robô cujo braço se desloque ao longo desses três eixos de coor-

denadas — para a frente e para trás, para cima e para baixo, para a direita e para a esquerda. Fica meio desajeitado, mas funciona bem para trabalhos numa área limitada.

Esse sistema funciona bem para robôs fixos a um suporte, mas não tem a mesma eficiência em outras situações. Para estas, há métodos de descrever o espaço e os pontos dentro dele. Um desses usa como modelo um cilindro e suas coordenadas. Imagine uma lata cilíndrica. Localiza-se qualquer ponto dentro da lata, associando: a distância a que está do centro do círculo que forma o fundo da lata (essa distância é um segmento de raio); o ângulo que ele forma em relação a determinado ponto de referência no círculo; e a altura em relação ao fundo da lata. Com essas três informações, há uma localização exata.

Um sistema parecido tem uma esfera como modelo de espaço. A diferença com o anterior está em que, ao invés de usar a altura em relação ao fundo, a informação necessária consiste no ângulo formado entre o ponto, o centro da esfera e seu diâmetro horizontal.

O processo mais comum para o posicionamento exato do braço de um robô, no entanto, usa somente ângulos: um para descrever a rotação na base, outro para a elevação do braço e um terceiro para uma segunda articulação.

**Força muscular**

O sistema de coordenadas adotado acabará de determinando o tipo de estrutura do braço, faltando, então, os "músculos" que vão produzir o movimento. Podem ser hidráulicos, elétricos ou pneumáticos.

Se a opção for pelos motores elétricos, estes deverão ser de passo, ligados direta ou indiretamente às partes que formam o braço. Um siste-



ma melhor seria imitar os animais, cujos músculos contraem-se e distendem-se. Isso pode ser conseguido por meio de pistões acionados por líquido (hidráulicos) ou ar comprimido (pneumáticos), atuando sobre cada articulação. Como os sistemas hidráulicos suportam grandes pressões e o volume do fluido não se altera (o que acontece com o ar), eles são mais indicados para ambientes industriais, onde os robôs precisam de mais força.

Esses músculos dão ao robô movimento linear; para obter movimento circular usa-se uma espécie de turbina sobre a qual o fluido exerce pressão — o resultado assemelha-se ao de um motor de passo.

Resolvido o problema do braço, fica faltando o da “mão” — para que o braço, depois de colocado na posição pretendida, execute um trabalho. É necessário, primeiro que ela tenha os movimentos fundamentais da mão humana: para cima e para baixo, para os lados e um mínimo de rotação. Um robô capaz de fazer esses movimentos tem seis graus de liberdade na mão. Pode-se também, construí-los com menos graus, mas manter a mobilidade implica dar mais liberdade às outras articulações do braço.

## A mão do robô

Tudo isso definido, fica faltando apenas estabelecer como será a mão do robô. O ideal, naturalmente, é fazê-la tão parecida com a do homem quanto possível (e até existem robôs assim). O formato mais comum consiste num “polegar” e dois “dedos”, habilitando o robô a apanhar objetos como o homem.

A energia para movimentar essa mão pode ser qualquer das três já citadas — elétrica, pneumática ou hidráulica —, dependendo da tarefa que o robô vai executar. Se ele tiver de trabalhar com objetos grandes e pesados, provavelmente precisará de musculatura hidráulica. Para outras aplicações, porém, motores elétricos ou a ar comprimido bastarão — especialmente se essas aplicações não exigirem precisão absoluta.

Haverá casos em que o robô não precisará de uma mão ou qualquer coisa parecida. Um robô de solda, por exemplo, só requer o equipamento de soldagem ligado diretamente ao punho. Existem robôs capazes até mesmo de escolher e trocar uma ferramenta do punho, de acordo com o tipo de trabalho a executar. Podem, digamos, descartar uma chave de fenda e inserir uma pistola de pintura — desde que o encaixe seja simples —, o que os torna extremamente versáteis.

A grande maioria, contudo, costuma trabalhar o tempo inteiro apenas com uma ferramenta — como os robôs de solda usados na indústria automobilística, que têm características totalmente distintas dos usados em pinturas. Seria inútil um robô de solda capaz de substituir ferramentas, já que seu trabalho é contínuo e exercido sempre no mesmo lugar. Mesmo porque, na fase de soldagem o carro ainda não está pronto para receber a pintura.

### O pulso

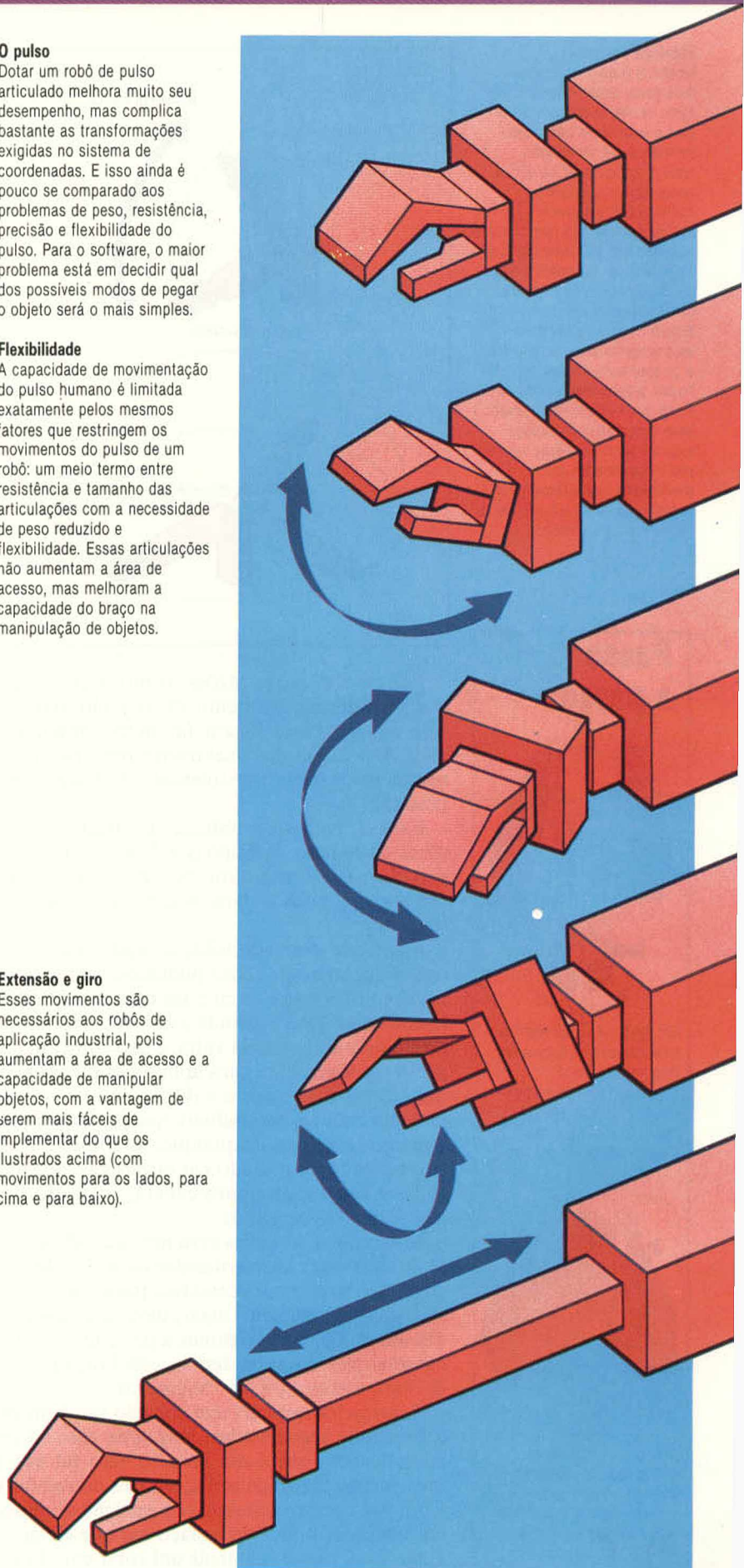
Dotar um robô de pulso articulado melhora muito seu desempenho, mas complica bastante as transformações exigidas no sistema de coordenadas. E isso ainda é pouco se comparado aos problemas de peso, resistência, precisão e flexibilidade do pulso. Para o software, o maior problema está em decidir qual dos possíveis modos de pegar o objeto será o mais simples.

### Flexibilidade

A capacidade de movimentação do pulso humano é limitada exatamente pelos mesmos fatores que restringem os movimentos do pulso de um robô: um meio termo entre resistência e tamanho das articulações com a necessidade de peso reduzido e flexibilidade. Essas articulações não aumentam a área de acesso, mas melhoram a capacidade do braço na manipulação de objetos.

### Extensão e giro

Esses movimentos são necessários aos robôs de aplicação industrial, pois aumentam a área de acesso e a capacidade de manipular objetos, com a vantagem de serem mais fáceis de implementar do que os ilustrados acima (com movimentos para os lados, para cima e para baixo).





# MUDANÇA DE ENDEREÇO

**A linguagem ASSEMBLY deve grande parte de sua versatilidade às diferentes maneiras de calcular endereços na memória. Examinamos aqui os endereçamentos direto, indireto e indexado, bem como suas combinações.**

Qualquer instrução em linguagem ASSEMBLY relaciona-se implícita ou explicitamente a um conteúdo da memória. Uma vez que um byte só se diferencia de outro por seu endereço, toda instrução em ASSEMBLY refere-se, portanto, a um ou mais endereços. Às vezes a referência é direta e óbvia, como em LDA \$E349, que significa “carregar o acumulador com o conteúdo do endereço \$E349”. Essa instrução menciona sem ambigüidade o endereço \$E349 e sua relação com o acumulador (cujo endereço é um nome, e não um número).

Outras instruções, contudo, referem-se a um endereço de maneira menos clara; por exemplo, RET, que significa “retornar de uma sub-rotina”. À primeira vista, aqui não há menção a nenhum endereço; mas RET, na realidade, significa “o endereço da próxima instrução a ser executada é o lugar de onde se chamou a última sub-rotina”. Aqui não se menciona o endereço cujo conteúdo será alterado (ou seja, o contador de programas — o registrador que contém o endereço da próxima instrução a ser executada). Também há referência ao endereço do novo conteúdo. Essas duas instruções representam modos de endereçamento totalmente diferentes.

Até agora, vimos instruções com dois diferentes modos de endereçamento: o imediato (como em LDA, \$45 ou ADC #31) e o direto absoluto (como em STA \$58A7 ou LD (\$696C),A). Esses modos, “naturais”, parecem abranger todos os casos possíveis, exceto os implícitos, como RST ou RET; há, porém, outras possibilidades, que examinaremos agora.

## Endereçamento da página zero

Usamos esse endereçamento, ou endereçamento curto, sempre que uma instrução se refere a um endereço entre \$0000 e \$00FF. Todos os endereços nessa faixa têm \$00 como byte mais alto e, portanto, ficam na página zero da memória. Tais instruções necessitam de apenas 2 bytes — um para o código de operação e outro para o byte mais baixo do endereço. Quando a CPU detecta um endereço de um só byte num ponto onde costuma haver dois, assume que o byte mais

alto é \$00 e, assim, refere-se diretamente à página zero. A vantagem desse processo está na velocidade de execução: o acesso aos dados da página zero é bem mais rápido que a qualquer outra página, exatamente porque requer um endereço de apenas 1 byte.

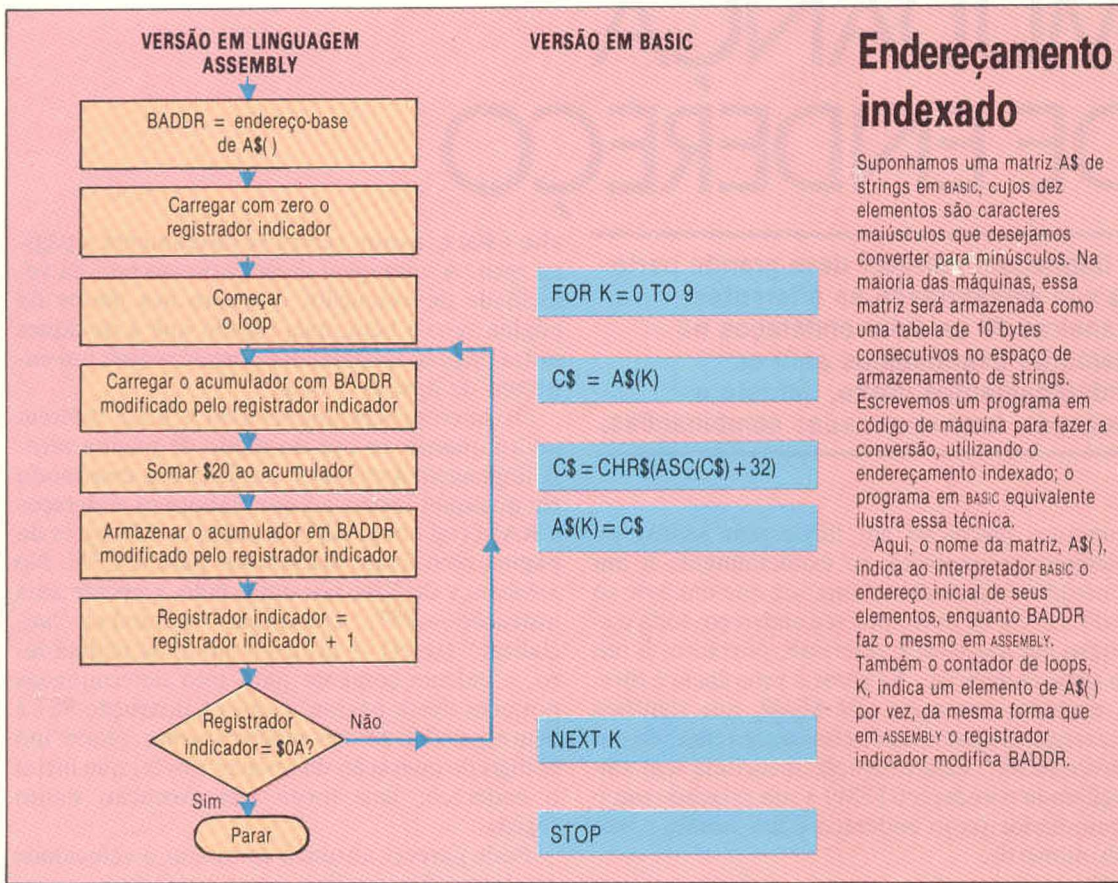
Os microprocessadores Z80 e 6502 diferem muito quanto ao uso do modo de página zero. A linguagem ASSEMBLY do 6502 usa esse modo em qualquer instrução que elabore os endereços à RAM (tal como LDA), e todos os 256 bytes da página zero ficam disponíveis para a CPU. No ASSEMBLY do Z80, por outro lado, somente uma instrução — RST, “recomeçar” (restart) ou “inicializar” (reset) — utiliza o modo de página zero, e calcula somente para oito determinadas posições dessa página. Como a instrução RST é bem específica em seu endereçamento, requer um código de operação de apenas 1 byte, que inclui o endereço; isso torna sua execução muito rápida.

Pode parecer absurdo comparar a velocidade das instruções em linguagem ASSEMBLY, quando sabemos que o tempo de execução da mais lenta é medido em microssegundos. Entretanto, às vezes a economia de 1 microssegundo por instrução representa a diferença entre o sucesso e o fracasso de um programa. Por exemplo, os jogos que mostram gráficos animados em alta resolução, coloridos e tridimensionais, envolvem milhões de instruções por operação em tela, e devem executar esses comandos o mais rápido possível para obter movimentos suaves e contínuos. Cortar 1 microssegundo de uma operação é muito importante quando ela faz parte de um loop a ser executado 64.000 vezes seguidas.

## Endereçamento indexado

O modo indexado é vital para a programação em ASSEMBLY, pois permite a elaboração de estruturas matriciais de dados. Sem tais estruturas, os programas se restringem ao endereçamento de posições de memória uma a uma; foi o que fizemos até agora em nossos programas. Por meio da indexação, entretanto, manipulam-se muito mais dados, e o programador conta com quantidade bem maior de recursos.

Os elementos essenciais do modo indexado são o endereço-base e o *índice*. Vamos supor que queremos montar uma tabela de dados — os códigos dos caracteres ASCII, por exemplo — em bytes consecutivos. O endereço-base da tabela é o do primeiro byte. A partir disso, podemos nos referir a qualquer byte subsequente da tabela por sua posição em relação ao endereço-base; o primeiro byte fica na posição zero, o segundo na



### Endereçamento indexado

Suponhamos uma matriz A\$ de strings em BASIC, cujos dez elementos são caracteres maiúsculos que desejamos converter para minúsculos. Na maioria das máquinas, essa matriz será armazenada como uma tabela de 10 bytes consecutivos no espaço de armazenamento de strings. Escrevemos um programa em código de máquina para fazer a conversão, utilizando o endereçamento indexado; o programa em BASIC equivalente ilustra essa técnica.

Aqui, o nome da matriz, A\$( ), indica ao interpretador BASIC o endereço inicial de seus elementos, enquanto BADDR faz o mesmo em ASSEMBLY. Também o contador de loops, K, indica um elemento de A\$( ) por vez, da mesma forma que em ASSEMBLY o registrador indicador modifica BADDR.

posição um, o terceiro na posição dois, e assim por diante. A posição de um byte com relação ao endereço-base da tabela chama-se índice, e calcula-se o endereço absoluto de qualquer byte da tabela pela soma do endereço-base com o índice do byte. Se construirmos um loop num programa em ASSEMBLY, e usarmos o contador do loop como índice, poderemos endereçar cada byte da tabela seqüencialmente, da mesma forma como acessamos os elementos de uma matriz em BASIC usando um loop FOR-NEXT.

Uma vez mais, as versões do ASSEMBLY do Z80 e do 6502 diferem quanto ao endereçamento indexado. O chip do 6502 contém dois registradores de um só byte, chamados X e Y; cada um dispõe de um índice que modifica um endereço-base. Isso limita o comprimento da tabela a 256 bytes, que é o maior número representado por um byte. Já o chip do Z80 contém dois registradores de 2 bytes, IX e IY, que contêm o próprio endereço-base e podem ser incrementados ou decrementados para indicar os sucessivos bytes da tabela. Sendo registradores de dois bytes, IX e IY podem conter o endereço de qualquer byte a que a própria CPU tem acesso. Seu conteúdo também se modifica por um índice de um só byte.

### Endereçamento indireto

Envolve o uso de endereços indicadores. Para compreender melhor esse conceito, imaginemos um grupo de pessoas que formou um cineclube

e se encontra todas as semanas para assistir a um filme. Este pode estar passando em um dos vários cinemas da cidade. Então, escolhido o filme da semana, o presidente escreve a hora e o lugar da projeção num cartaz e afixa-o na vitrine de uma loja no centro da cidade. Os membros do cineclube não sabem qual será o cinema escolhido, mas sabem onde fica a loja, e esta lhes "indica" o local correto. O endereço da loja é, indiretamente, o endereço do cinema.

No modo de endereçamento indireto escrevem-se instruções que contêm o endereço de um indicador e que atuam sobre o conteúdo da po-



**Indicador**  
Exemplo de endereçamento indireto encontrado no cotidiano. Na foto, o quadro indicador dos trens, com destinos e plataformas, contém dados necessários a quem viaja. A seta "Train indicator" (indicador de trens) mostra onde encontrar o quadro; assim, ela indiretamente endereça os dados. Em ASSEMBLY, o endereçamento indireto significa que o endereço no operando é o do byte que armazena os dados; o do operando, um indicador que aponta para o endereço real.



sição para onde o indicador aponta (e não sobre o conteúdo do próprio indicador). As vantagens desse modo são consideráveis, especialmente quando combinado com o tipo indexado. Suponhamos, por exemplo, que você escreva uma rotina em ASSEMBLY que pesquise, numa tabela de dados, um certo caractere, e retorne com sua posição indexada. Suponhamos ainda que você queira manter várias tabelas em diferentes locais da memória e usar a mesma rotina para pesquisar qualquer uma delas. Se sua rotina encontrar o endereço-base da tabela indiretamente, por meio de um indicador, poderá ser usada com qualquer tabela, desde que se modifique o conteúdo do indicador antes de se chamar a rotina.

Em geral, os programas requerem combinações desses modos de endereçamento. A instrução LDA do 6502, por exemplo, pode ser usada nos seguintes modos:

Código de operação	Operando	Modo
LDA	#\$34	Imediato
LDA	\$A2	Direto, página zero
LDA	\$967F	Direto, absoluto
LDA	\$A2,X	Página zero, indexado com X
LDA	\$967F,X	Absoluto, indexado com X
LDA	\$967F,Y	Absoluto, indexado com Y
LDA	(\$A2,X)	Indireto, pré-indexado com X
LDA	(\$A2),Y	Indireto, pós-indexado com Y

Exemplificamos com uma instrução do 6502 porque mostra com clareza as combinações de diversos modos de endereçamento. Essa tabela pode, de início, parecer um pouco confusa, mas, na verdade, o uso de vários modos torna claro seu significado; até agora usamos LDA e ADC em dois modos — imediato e absoluto — sem confusão.

A tabela responde a uma pergunta natural: como saber qual o modo de endereçamento de uma instrução quando o código mnemônico é o mesmo em todos os casos? Vemos que o formato do operando difere em cada modo, e a única ambigüidade possível é quanto a uma instrução como LDA SYMB1 — ela está no modo absoluto ou página zero? Um programa assembler resolve esse problema. Mas, se você quiser compilar o programa manualmente, terá de determinar se SYMB1 foi definida como um valor de 1 ou de 2 bytes.

De modo geral, uma vez que se comece a usar um programa assembler, pode-se esquecer dos códigos de operação e do número de bytes por instrução e concentrar-se em aprender as técnicas de programação do ASSEMBLY. É importante entender como funciona o código de máquina, mas a linguagem ASSEMBLY, usada em conjun-

to com um bom programa assembler, é um método de programação muito melhor, que combina a potência do código de máquina com muitos recursos das linguagens de alto nível.

Eis dois programinhas úteis em ASSEMBLY prontos para execução.

```
; CONVHA — CONVERTE BYTES HEXA EM CARACTERES ASCII
; SEQUENCIA DA CHAMADA

; LXI    D,<OP1>   CARREGA O ENDEREÇO DE DESTINO
; LXI    H,<OP2>   CARREGA O ENDEREÇO-FONTE
; MVI    C,<CONT>  CONTADOR DE BYTES HEXA (1-255)
; CALL   CONVHA

CONVHA:  MOV    A,M      ; CARREGA DOIS DIGITOS HEXA DE 4 BITS
         RAR          ; EXAMINA O PRIMEIRO DA ESQUERDA
         RAR          ;
         RAR          ;
         RAR          ;
         CALL  COHAD   ;
         MOV    A,M      ; CARREGA-OS NOVAMENTE E
         CALL  COHAD   ; EXAMINA O DIGITO DA DIREITA
         INX    H        ; INDICA O BYTE HEXA SEGUINTE
         DCR    C        ; CONT = CONT-1
         JNZ   CONVHA   ; CONTINUA O LOOP
         RET            ; RETORNA AO USUARIO

COHAD:   XCHG          ; TROCA O CONTEUDO DOS INDICADORES
         ANI    OFH     ; MASCARA ZERO NOS 4 BYTES DA ESQUERDA
         OPI    10      ; EXAMINA A FAIXA
         JNO   COHA1    ;
         ADI    D30H    ; FAIXA 0-9
         JMP   COHA2    ;
COHA1:   ADI    037H    ; NA FAIXA A-F
COHA2:   MOV    M,A     ; ARMAZENA O RESULTADO
         INX    H        ; INDICA O PROXIMO BYTE
         XCHG          ; DESTROCA OS INDICADORES
         RET            ; RETORNA

; TRANS: TRANSFERE DADOS DE UMA AREA (OP1) PARA OUTRA AREA (OP2)
; SEQUENCIA DA CHAMADA:

; LXI    D,<OP1>   ; CARREGA ENDEREÇO DE ORIGEM
; LXI    H,<OP2>   ; CARREGA ENDEREÇO DE DESTINO
; MVI    C,<CONT>  ; CARREGA TAMANHO DO CAMPO OP1
; CALL   TRANS     ; TRANSFERE OS DADOS DE OP1

TRANS:   PUSH   H       ; SALVA ENDEREÇO DE OP2
TRANS1:  PUSH   D       ; SALVA ENDEREÇO DE OP1
         LDAX  D       ; ACESSA UM DADO DE OP1
         MVI   D,0     ; ZERA O REGISTRADOR D
         MOV   E,A     ; CARREGA O ACUMULADOR NO REGISTRADOR
         E     ;
         DAD   D       ; ENDEREÇO DE ENTRADA DE OP2
         MOV   A,M     ; ACESSA UM DADO DE OP1
         POP   D       ; RESTAURA ENDEREÇO DE OP1
         POP   H       ; RESTAURA ENDEREÇO DE OP2
         PUSH  H       ;
         STAX  D       ;
         INX   D       ; INCREMENTA O INDICADOR DE OP1
         DCR   C       ; INCREMENTA A CONTAGEM
         JNZ   TRANS1  ; LOOP ATE O FINAL
         POP   H       ;
         RET            ; RETORNA
```



# LABO

**A Labo, indústria brasileira de minis, produz também micros, terminais bancários e sistemas aplicativos. Modulares e compatíveis entre si, seus equipamentos vêm conquistando crescente fatia do mercado.**

Criada em 1961, a Labo Eletrônica S.A. iniciou suas atividades fabricando instrumentos eletrônicos de alta precisão. Em 1979 entrou na área da informática, firmando um acordo de transferência integral de tecnologia com a empresa alemã Nixdorf Computer A.G. Esta, a maior fábrica européia de computadores e detentora de tecnologia avançada, já instalou mais de 110.000 sistemas em todo o mundo. Em 1983, a Labo participou com 27,7% das unidades comercializadas no Brasil, alcançando em 1984 a marca dos 1.500 equipamentos instalados.

A linha de produção da empresa inclui os minicomputadores da série Labo 8000, com vários modelos totalmente compatíveis em hardware e software; os superminis, de arquitetura modular e atualizada; e os microcomputadores, que constituem a série Labo 8200. Fornece, ainda, sistemas bancários on line e off line e terminais de venda para lojas e supermercados.

Modulares e compatíveis entre si, os equipamentos Labo expandem-se e integram-se facilmente. Seus micros e minis comunicam-se em rede. Mantendo os investimentos já realizados em hardware, software e treinamento, converte-se o míni 8034 num 8038, com um processador

central duas vezes mais rápido e um poderoso software básico.

O micro Labo 8221, concebido para operar em configuração multiterminal, dispõe de uma capacidade de memória superior a qualquer outro equipamento de seu porte. Nele, o processador central, a memória e o controlador da unidade de discos estão montados numa única placa, o que simplifica sua manutenção. Trabalhando com três modos de operação hierarquizados — usuário, sistema e núcleo (kernel) —, proporciona uma proteção aos dados só encontrada nos sistemas de maior porte. Seu software básico é o SOL (Sistema Operacional Labo), que permite, operando com diferentes terminais, executar dois programas simultaneamente.

Também o minicomputador Labo 8043, o maior fabricado pela empresa, oferece total proteção a programas e dados: qualquer acesso às informações é protegido por um conjunto de procedimentos conhecidos apenas pelo pessoal autorizado. Outro valioso recurso desse míni está no módulo ASA (Aceleração de Software Aplicativo). Ele reduz o número de acessos aos discos, mantendo na memória, de forma dinâmica, os blocos mais lidos e prevendo por estatística os próximos blocos a serem acessados. Totalmente desenvolvido no Brasil, o ASA agiliza o trabalho do míni, equiparando-se, nos recursos, aos equipamentos de grande porte.

Os componentes utilizados pela Labo são rigorosamente testados em seu laboratório de controle de qualidade, onde uma câmara de simulação de envelhecimento descarta todos os elementos que apresentarem falhas.

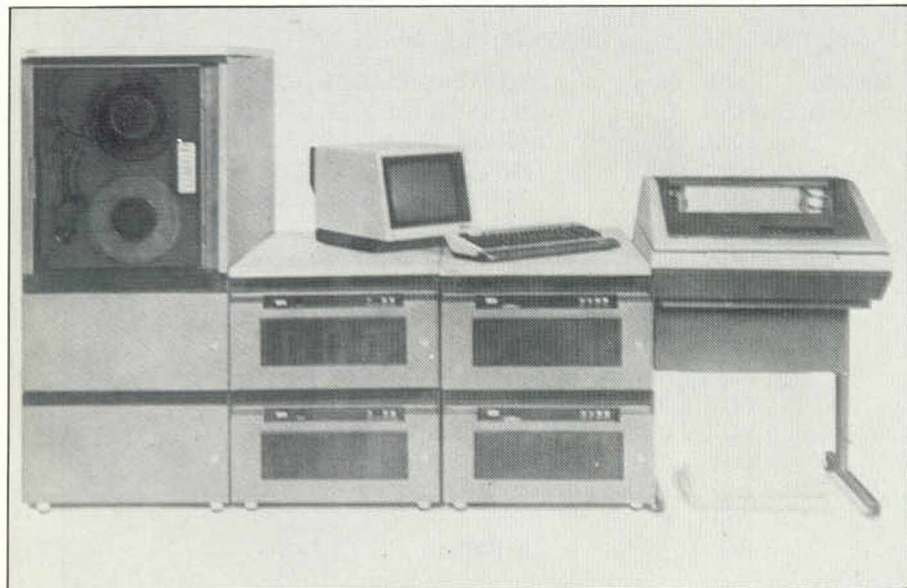
Visando oferecer soluções imediatas aos problemas de seu cliente, a Labo dispõe de uma série de programas prontos, destinados a diversos segmentos do mercado: consórcios, hospitais, transportadoras, frotistas, administradoras de imóveis, operadoras de open market, agroindústrias etc. Todos os programas são protegidos por códigos e procedimentos que impedem o acesso de pessoas não autorizadas às informações.

Com filiais nas principais cidades do Brasil, a Labo dedica especial atenção à manutenção de seus equipamentos, incluindo suporte remoto por telediagnóstico — serviço inédito que, através das linhas telefônicas, detecta irregularidades nos sistemas e programas.

Entre esses programas destacam-se o SACIL — Sistemas Aplicativos Comerciais Integrados Labo e o PLANCOI — Planejamento e Controle Industrial, destinado, principalmente, a empresas de médio e grande porte.

#### Programas protegidos

O 8043, o maior dos minis fabricados pela Labo, conta com recursos de proteção a programas e dados confidenciais. Dispõe também de um módulo ASA, que reduz os acessos aos discos, agilizando a execução.



# ANÁLISE DE JOGOS

Nas páginas a seguir, MICROCOMPUTADOR, CURSO PRÁTICO analisa os jogos eletrônicos de maior sucesso. Alguns proporcionaram o desenvolvimento de grande parte do software lúdico hoje existente no mercado internacional. Incluindo as três categorias — aventura, reflexão e animação —, esses jogos foram selecionados em vista não apenas do êxito comercial, mas também de sua riqueza estrutural.

## FORÇAS ESPACIAIS

Space Invaders (análise)

Jamais um jogo mexeu tanto com a imaginação do público como Space Invaders. Simples, mas eficiente, conseguiu tamanho sucesso nos fliperamas que acabou estimulando versões para os mais populares microcomputadores. Na verdade, para quase todos os micros existe uma versão do Space Invaders. O jogo tornou-se tão conhecido que seu nome acabou virando termo genérico, associando-se a todos os jogos de fliperama.

Quando foi lançado, em 1978, o jogo rapidamente produziu uma febre de proporções quase epidêmicas. Os pais se preocupavam, achando que seus filhos gastariam todo o seu tempo e dinheiro nos detestáveis fliperamas. Aqueles desgostosos pais não imaginavam, porém, que seus filhos, na verdade, estavam tomando contato com o futuro.

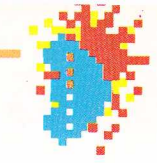
Pode-se dizer que o Space Invaders alterou a maneira como a sociedade via os computadores. Antes que o jogo tivesse sucesso na exploração das possibilidades gráficas do microprocessador, os computadores eram considerados pouco confiáveis — o exemplo clássico era o paranoico HAL do filme 2001: Uma odisseia no espaço.

O Space Invaders foi o pioneiro dos jogos para computadores do gênero extermínio. A partir dele surgiram centenas de jogos nos quais um único herói ou heroína tinha de enfrentar hordas de maldosos atacantes, contando apenas com sua agilidade no botão de disparo e três vidas.

É inegável que o Space Invaders está sujeito ao envelhecimento, principalmente porque se trata de um jogo muito simples, em que o jogador controla um canhão móvel de raios laser, usa-

```

" +N$(3) AND N$(3)>S$)+(S$+
S$+
" +N$(4) AND N$(4)>S$)+(S$+
" +N$(5) AND N$(5)>S
N$(6)<>S$ THEN LET H=12
IF N$(4)<>S$ THEN LET H=14
IF N$(5)<>S$ THEN LET H=16
IF H=0 THEN PRINT AT A-2,B-
PRINT AT A,B-1;" I$:"
FOR M=1 TO (40-I)/4
IF H=0 THEN GOTO 420
IF H=1 THEN GOSUB 1230
IF H=2 THEN GOSUB 1330
IF H=3 THEN GOSUB 1430
IF H=4 THEN GOSUB 1530
IF H=5 THEN GOSUB 1630
IF H=6 THEN GOSUB 1730
IF H=7 THEN GOSUB 1830
IF H=8 THEN GOSUB 1930
IF H=9 THEN GOSUB 2030
IF H=10 THEN GOSUB 2130
IF H=11 THEN GOSUB 2230
IF H=12 THEN GOSUB 2330
IF H=13 THEN GOSUB 2430
IF H=14 THEN GOSUB 2530
IF H=15 THEN GOSUB 2630
IF H=16 THEN GOSUB 2730
IF H=17 THEN GOSUB 2830
IF H=18 THEN GOSUB 2930
IF H=19 THEN GOSUB 3030
IF H=20 THEN GOSUB 3130
IF H=21 THEN GOSUB 3230
IF H=22 THEN GOSUB 3330
IF H=23 THEN GOSUB 3430
IF H=24 THEN GOSUB 3530
IF H=25 THEN GOSUB 3630
IF H=26 THEN GOSUB 3730
IF H=27 THEN GOSUB 3830
IF H=28 THEN GOSUB 3930
IF H=29 THEN GOSUB 4030
IF H=30 THEN GOSUB 4130
IF H=31 THEN GOSUB 4230
IF H=32 THEN GOSUB 4330
IF H=33 THEN GOSUB 4430
IF H=34 THEN GOSUB 4530
IF H=35 THEN GOSUB 4630
IF H=36 THEN GOSUB 4730
IF H=37 THEN GOSUB 4830
IF H=38 THEN GOSUB 4930
IF H=39 THEN GOSUB 5030
IF H=40 THEN GOSUB 5130
IF H=41 THEN GOSUB 5230
IF H=42 THEN GOSUB 5330
IF H=43 THEN GOSUB 5430
IF H=44 THEN GOSUB 5530
IF H=45 THEN GOSUB 5630
IF H=46 THEN GOSUB 5730
IF H=47 THEN GOSUB 5830
IF H=48 THEN GOSUB 5930
IF H=49 THEN GOSUB 6030
IF H=50 THEN GOSUB 6130
IF H=51 THEN GOSUB 6230
IF H=52 THEN GOSUB 6330
IF H=53 THEN GOSUB 6430
IF H=54 THEN GOSUB 6530
IF H=55 THEN GOSUB 6630
IF H=56 THEN GOSUB 6730
IF H=57 THEN GOSUB 6830
IF H=58 THEN GOSUB 6930
IF H=59 THEN GOSUB 7030
IF H=60 THEN GOSUB 7130
IF H=61 THEN GOSUB 7230
IF H=62 THEN GOSUB 7330
IF H=63 THEN GOSUB 7430
IF H=64 THEN GOSUB 7530
IF H=65 THEN GOSUB 7630
IF H=66 THEN GOSUB 7730
IF H=67 THEN GOSUB 7830
IF H=68 THEN GOSUB 7930
IF H=69 THEN GOSUB 8030
IF H=70 THEN GOSUB 8130
IF H=71 THEN GOSUB 8230
IF H=72 THEN GOSUB 8330
IF H=73 THEN GOSUB 8430
IF H=74 THEN GOSUB 8530
IF H=75 THEN GOSUB 8630
IF H=76 THEN GOSUB 8730
IF H=77 THEN GOSUB 8830
IF H=78 THEN GOSUB 8930
IF H=79 THEN GOSUB 9030
IF H=80 THEN GOSUB 9130
IF H=81 THEN GOSUB 9230
IF H=82 THEN GOSUB 9330
IF H=83 THEN GOSUB 9430
IF H=84 THEN GOSUB 9530
IF H=85 THEN GOSUB 9630
IF H=86 THEN GOSUB 9730
IF H=87 THEN GOSUB 9830
IF H=88 THEN GOSUB 9930
IF H=89 THEN GOSUB 10030
IF H=90 THEN GOSUB 10130
IF H=91 THEN GOSUB 10230
IF H=92 THEN GOSUB 10330
IF H=93 THEN GOSUB 10430
IF H=94 THEN GOSUB 10530
IF H=95 THEN GOSUB 10630
IF H=96 THEN GOSUB 10730
IF H=97 THEN GOSUB 10830
IF H=98 THEN GOSUB 10930
IF H=99 THEN GOSUB 11030
IF H=100 THEN GOSUB 11130
IF H=101 THEN GOSUB 11230
IF H=102 THEN GOSUB 11330
IF H=103 THEN GOSUB 11430
IF H=104 THEN GOSUB 11530
IF H=105 THEN GOSUB 11630
IF H=106 THEN GOSUB 11730
IF H=107 THEN GOSUB 11830
IF H=108 THEN GOSUB 11930
IF H=109 THEN GOSUB 12030
IF H=110 THEN GOSUB 12130
IF H=111 THEN GOSUB 12230
IF H=112 THEN GOSUB 12330
IF H=113 THEN GOSUB 12430
IF H=114 THEN GOSUB 12530
IF H=115 THEN GOSUB 12630
IF H=116 THEN GOSUB 12730
IF H=117 THEN GOSUB 12830
IF H=118 THEN GOSUB 12930
IF H=119 THEN GOSUB 13030
IF H=120 THEN GOSUB 13130
IF H=121 THEN GOSUB 13230
IF H=122 THEN GOSUB 13330
IF H=123 THEN GOSUB 13430
IF H=124 THEN GOSUB 13530
IF H=125 THEN GOSUB 13630
IF H=126 THEN GOSUB 13730
IF H=127 THEN GOSUB 13830
IF H=128 THEN GOSUB 13930
IF H=129 THEN GOSUB 14030
IF H=130 THEN GOSUB 14130
IF H=131 THEN GOSUB 14230
IF H=132 THEN GOSUB 14330
IF H=133 THEN GOSUB 14430
IF H=134 THEN GOSUB 14530
IF H=135 THEN GOSUB 14630
IF H=136 THEN GOSUB 14730
IF H=137 THEN GOSUB 14830
IF H=138 THEN GOSUB 14930
IF H=139 THEN GOSUB 15030
IF H=140 THEN GOSUB 15130
IF H=141 THEN GOSUB 15230
IF H=142 THEN GOSUB 15330
IF H=143 THEN GOSUB 15430
IF H=144 THEN GOSUB 15530
IF H=145 THEN GOSUB 15630
IF H=146 THEN GOSUB 15730
IF H=147 THEN GOSUB 15830
IF H=148 THEN GOSUB 15930
IF H=149 THEN GOSUB 16030
IF H=150 THEN GOSUB 16130
IF H=151 THEN GOSUB 16230
IF H=152 THEN GOSUB 16330
IF H=153 THEN GOSUB 16430
IF H=154 THEN GOSUB 16530
IF H=155 THEN GOSUB 16630
IF H=156 THEN GOSUB 16730
IF H=157 THEN GOSUB 16830
IF H=158 THEN GOSUB 16930
IF H=159 THEN GOSUB 17030
IF H=160 THEN GOSUB 17130
IF H=161 THEN GOSUB 17230
IF H=162 THEN GOSUB 17330
IF H=163 THEN GOSUB 17430
IF H=164 THEN GOSUB 17530
IF H=165 THEN GOSUB 17630
IF H=166 THEN GOSUB 17730
IF H=167 THEN GOSUB 17830
IF H=168 THEN GOSUB 17930
IF H=169 THEN GOSUB 18030
IF H=170 THEN GOSUB 18130
IF H=171 THEN GOSUB 18230
IF H=172 THEN GOSUB 18330
IF H=173 THEN GOSUB 18430
IF H=174 THEN GOSUB 18530
IF H=175 THEN GOSUB 18630
IF H=176 THEN GOSUB 18730
IF H=177 THEN GOSUB 18830
IF H=178 THEN GOSUB 18930
IF H=179 THEN GOSUB 19030
IF H=180 THEN GOSUB 19130
IF H=181 THEN GOSUB 19230
IF H=182 THEN GOSUB 19330
IF H=183 THEN GOSUB 19430
IF H=184 THEN GOSUB 19530
IF H=185 THEN GOSUB 19630
IF H=186 THEN GOSUB 19730
IF H=187 THEN GOSUB 19830
IF H=188 THEN GOSUB 19930
IF H=189 THEN GOSUB 20030
IF H=190 THEN GOSUB 20130
IF H=191 THEN GOSUB 20230
IF H=192 THEN GOSUB 20330
IF H=193 THEN GOSUB 20430
IF H=194 THEN GOSUB 20530
IF H=195 THEN GOSUB 20630
IF H=196 THEN GOSUB 20730
IF H=197 THEN GOSUB 20830
IF H=198 THEN GOSUB 20930
IF H=199 THEN GOSUB 21030
IF H=200 THEN GOSUB 21130
IF H=201 THEN GOSUB 21230
IF H=202 THEN GOSUB 21330
IF H=203 THEN GOSUB 21430
IF H=204 THEN GOSUB 21530
IF H=205 THEN GOSUB 21630
IF H=206 THEN GOSUB 21730
IF H=207 THEN GOSUB 21830
IF H=208 THEN GOSUB 21930
IF H=209 THEN GOSUB 22030
IF H=210 THEN GOSUB 22130
IF H=211 THEN GOSUB 22230
IF H=212 THEN GOSUB 22330
IF H=213 THEN GOSUB 22430
IF H=214 THEN GOSUB 22530
IF H=215 THEN GOSUB 22630
IF H=216 THEN GOSUB 22730
IF H=217 THEN GOSUB 22830
IF H=218 THEN GOSUB 22930
IF H=219 THEN GOSUB 23030
IF H=220 THEN GOSUB 23130
IF H=221 THEN GOSUB 23230
IF H=222 THEN GOSUB 23330
IF H=223 THEN GOSUB 23430
IF H=224 THEN GOSUB 23530
IF H=225 THEN GOSUB 23630
IF H=226 THEN GOSUB 23730
IF H=227 THEN GOSUB 23830
IF H=228 THEN GOSUB 23930
IF H=229 THEN GOSUB 24030
IF H=230 THEN GOSUB 24130
IF H=231 THEN GOSUB 24230
IF H=232 THEN GOSUB 24330
IF H=233 THEN GOSUB 24430
IF H=234 THEN GOSUB 24530
IF H=235 THEN GOSUB 24630
IF H=236 THEN GOSUB 24730
IF H=237 THEN GOSUB 24830
IF H=238 THEN GOSUB 24930
IF H=239 THEN GOSUB 25030
IF H=240 THEN GOSUB 25130
IF H=241 THEN GOSUB 25230
IF H=242 THEN GOSUB 25330
IF H=243 THEN GOSUB 25430
IF H=244 THEN GOSUB 25530
IF H=245 THEN GOSUB 25630
IF H=246 THEN GOSUB 25730
IF H=247 THEN GOSUB 25830
IF H=248 THEN GOSUB 25930
IF H=249 THEN GOSUB 26030
IF H=250 THEN GOSUB 26130
IF H=251 THEN GOSUB 26230
IF H=252 THEN GOSUB 26330
IF H=253 THEN GOSUB 26430
IF H=254 THEN GOSUB 26530
IF H=255 THEN GOSUB 26630
IF H=256 THEN GOSUB 26730
IF H=257 THEN GOSUB 26830
IF H=258 THEN GOSUB 26930
IF H=259 THEN GOSUB 27030
IF H=260 THEN GOSUB 27130
IF H=261 THEN GOSUB 27230
IF H=262 THEN GOSUB 27330
IF H=263 THEN GOSUB 27430
IF H=264 THEN GOSUB 27530
IF H=265 THEN GOSUB 27630
IF H=266 THEN GOSUB 27730
IF H=267 THEN GOSUB 27830
IF H=268 THEN GOSUB 27930
IF H=269 THEN GOSUB 28030
IF H=270 THEN GOSUB 28130
IF H=271 THEN GOSUB 28230
IF H=272 THEN GOSUB 28330
IF H=273 THEN GOSUB 28430
IF H=274 THEN GOSUB 28530
IF H=275 THEN GOSUB 28630
IF H=276 THEN GOSUB 28730
IF H=277 THEN GOSUB 28830
IF H=278 THEN GOSUB 28930
IF H=279 THEN GOSUB 29030
IF H=280 THEN GOSUB 29130
IF H=281 THEN GOSUB 29230
IF H=282 THEN GOSUB 29330
IF H=283 THEN GOSUB 29430
IF H=284 THEN GOSUB 29530
IF H=285 THEN GOSUB 29630
IF H=286 THEN GOSUB 29730
IF H=287 THEN GOSUB 29830
IF H=288 THEN GOSUB 29930
IF H=289 THEN GOSUB 30030
IF H=290 THEN GOSUB 30130
IF H=291 THEN GOSUB 30230
IF H=292 THEN GOSUB 30330
IF H=293 THEN GOSUB 30430
IF H=294 THEN GOSUB 30530
IF H=295 THEN GOSUB 30630
IF H=296 THEN GOSUB 30730
IF H=297 THEN GOSUB 30830
IF H=298 THEN GOSUB 30930
IF H=299 THEN GOSUB 31030
IF H=300 THEN GOSUB 31130
IF H=301 THEN GOSUB 31230
IF H=302 THEN GOSUB 31330
IF H=303 THEN GOSUB 31430
IF H=304 THEN GOSUB 31530
IF H=305 THEN GOSUB 31630
IF H=306 THEN GOSUB 31730
IF H=307 THEN GOSUB 31830
IF H=308 THEN GOSUB 31930
IF H=309 THEN GOSUB 32030
IF H=310 THEN GOSUB 32130
IF H=311 THEN GOSUB 32230
IF H=312 THEN GOSUB 32330
IF H=313 THEN GOSUB 32430
IF H=314 THEN GOSUB 32530
IF H=315 THEN GOSUB 32630
IF H=316 THEN GOSUB 32730
IF H=317 THEN GOSUB 32830
IF H=318 THEN GOSUB 32930
IF H=319 THEN GOSUB 33030
IF H=320 THEN GOSUB 33130
IF H=321 THEN GOSUB 33230
IF H=322 THEN GOSUB 33330
IF H=323 THEN GOSUB 33430
IF H=324 THEN GOSUB 33530
IF H=325 THEN GOSUB 33630
IF H=326 THEN GOSUB 33730
IF H=327 THEN GOSUB 33830
IF H=328 THEN GOSUB 33930
IF H=329 THEN GOSUB 34030
IF H=330 THEN GOSUB 34130
IF H=331 THEN GOSUB 34230
IF H=332 THEN GOSUB 34330
IF H=333 THEN GOSUB 34430
IF H=334 THEN GOSUB 34530
IF H=335 THEN GOSUB 34630
IF H=336 THEN GOSUB 34730
IF H=337 THEN GOSUB 34830
IF H=338 THEN GOSUB 34930
IF H=339 THEN GOSUB 35030
IF H=340 THEN GOSUB 35130
IF H=341 THEN GOSUB 35230
IF H=342 THEN GOSUB 35330
IF H=343 THEN GOSUB 35430
IF H=344 THEN GOSUB 35530
IF H=345 THEN GOSUB 35630
IF H=346 THEN GOSUB 35730
IF H=347 THEN GOSUB 35830
IF H=348 THEN GOSUB 35930
IF H=349 THEN GOSUB 36030
IF H=350 THEN GOSUB 36130
IF H=351 THEN GOSUB 36230
IF H=352 THEN GOSUB 36330
IF H=353 THEN GOSUB 36430
IF H=354 THEN GOSUB 36530
IF H=355 THEN GOSUB 36630
IF H=356 THEN GOSUB 36730
IF H=357 THEN GOSUB 36830
IF H=358 THEN GOSUB 36930
IF H=359 THEN GOSUB 37030
IF H=360 THEN GOSUB 37130
IF H=361 THEN GOSUB 37230
IF H=362 THEN GOSUB 37330
IF H=363 THEN GOSUB 37430
IF H=364 THEN GOSUB 37530
IF H=365 THEN GOSUB 37630
IF H=366 THEN GOSUB 37730
IF H=367 THEN GOSUB 37830
IF H=368 THEN GOSUB 37930
IF H=369 THEN GOSUB 38030
IF H=370 THEN GOSUB 38130
IF H=371 THEN GOSUB 38230
IF H=372 THEN GOSUB 38330
IF H=373 THEN GOSUB 38430
IF H=374 THEN GOSUB 38530
IF H=375 THEN GOSUB 38630
IF H=376 THEN GOSUB 38730
IF H=377 THEN GOSUB 38830
IF H=378 THEN GOSUB 38930
IF H=379 THEN GOSUB 39030
IF H=380 THEN GOSUB 39130
IF H=381 THEN GOSUB 39230
IF H=382 THEN GOSUB 39330
IF H=383 THEN GOSUB 39430
IF H=384 THEN GOSUB 39530
IF H=385 THEN GOSUB 39630
IF H=386 THEN GOSUB 39730
IF H=387 THEN GOSUB 39830
IF H=388 THEN GOSUB 39930
IF H=389 THEN GOSUB 40030
IF H=390 THEN GOSUB 40130
IF H=391 THEN GOSUB 40230
IF H=392 THEN GOSUB 40330
IF H=393 THEN GOSUB 40430
IF H=394 THEN GOSUB 40530
IF H=395 THEN GOSUB 40630
IF H=396 THEN GOSUB 40730
IF H=397 THEN GOSUB 40830
IF H=398 THEN GOSUB 40930
IF H=399 THEN GOSUB 41030
IF H=400 THEN GOSUB 41130
IF H=401 THEN GOSUB 41230
IF H=402 THEN GOSUB 41330
IF H=403 THEN GOSUB 41430
IF H=404 THEN GOSUB 41530
IF H=405 THEN GOSUB 41630
IF H=406 THEN GOSUB 41730
IF H=407 THEN GOSUB 41830
IF H=408 THEN GOSUB 41930
IF H=409 THEN GOSUB 42030
IF H=410 THEN GOSUB 42130
IF H=411 THEN GOSUB 42230
IF H=412 THEN GOSUB 42330
IF H=413 THEN GOSUB 42430
IF H=414 THEN GOSUB 42530
IF H=415 THEN GOSUB 42630
IF H=416 THEN GOSUB 42730
IF H=417 THEN GOSUB 42830
IF H=418 THEN GOSUB 42930
IF H=419 THEN GOSUB 43030
IF H=420 THEN GOSUB 43130
IF H=421 THEN GOSUB 43230
IF H=422 THEN GOSUB 43330
IF H=423 THEN GOSUB 43430
IF H=424 THEN GOSUB 43530
IF H=425 THEN GOSUB 43630
IF H=426 THEN GOSUB 43730
IF H=427 THEN GOSUB 43830
IF H=428 THEN GOSUB 43930
IF H=429 THEN GOSUB 44030
IF H=430 THEN GOSUB 44130
IF H=431 THEN GOSUB 44230
IF H=432 THEN GOSUB 44330
IF H=433 THEN GOSUB 44430
IF H=434 THEN GOSUB 44530
IF H=435 THEN GOSUB 44630
IF H=436 THEN GOSUB 44730
IF H=437 THEN GOSUB 44830
IF H=438 THEN GOSUB 44930
IF H=439 THEN GOSUB 45030
IF H=440 THEN GOSUB 45130
IF H=441 THEN GOSUB 45230
IF H=442 THEN GOSUB 45330
IF H=443 THEN GOSUB 45430
IF H=444 THEN GOSUB 45530
IF H=445 THEN GOSUB 45630
IF H=446 THEN GOSUB 45730
IF H=447 THEN GOSUB 45830
IF H=448 THEN GOSUB 45930
IF H=449 THEN GOSUB 46030
IF H=450 THEN GOSUB 46130
IF H=451 THEN GOSUB 46230
IF H=452 THEN GOSUB 46330
IF H=453 THEN GOSUB 46430
IF H=454 THEN GOSUB 46530
IF H=455 THEN GOSUB 46630
IF H=456 THEN GOSUB 46730
IF H=457 THEN GOSUB 46830
IF H=458 THEN GOSUB 46930
IF H=459 THEN GOSUB 47030
IF H=460 THEN GOSUB 47130
IF H=461 THEN GOSUB 47230
IF H=462 THEN GOSUB 47330
IF H=463 THEN GOSUB 47430
IF H=464 THEN GOSUB 47530
IF H=465 THEN GOSUB 47630
IF H=466 THEN GOSUB 47730
IF H=467 THEN GOSUB 47830
IF H=468 THEN GOSUB 47930
IF H=469 THEN GOSUB 48030
IF H=470 THEN GOSUB 48130
IF H=471 THEN GOSUB 48230
IF H=472 THEN GOSUB 48330
IF H=473 THEN GOSUB 48430
IF H=474 THEN GOSUB 48530
IF H=475 THEN GOSUB 48630
IF H=476 THEN GOSUB 48730
IF H=477 THEN GOSUB 48830
IF H=478 THEN GOSUB 48930
IF H=479 THEN GOSUB 49030
IF H=480 THEN GOSUB 49130
IF H=481 THEN GOSUB 49230
IF H=482 THEN GOSUB 49330
IF H=483 THEN GOSUB 49430
IF H=484 THEN GOSUB 49530
IF H=485 THEN GOSUB 49630
IF H=486 THEN GOSUB 49730
IF H=487 THEN GOSUB 49830
IF H=488 THEN GOSUB 49930
IF H=489 THEN GOSUB 50030
IF H=490 THEN GOSUB 50130
IF H=491 THEN GOSUB 50230
IF H=492 THEN GOSUB 50330
IF H=493 THEN GOSUB 50430
IF H=494 THEN GOSUB 50530
IF H=495 THEN GOSUB 50630
IF H=496 THEN GOSUB 50730
IF H=497 THEN GOSUB 50830
IF H=498 THEN GOSUB 50930
IF H=499 THEN GOSUB 51030
IF H=500 THEN GOSUB 51130
IF H=501 THEN GOSUB 51230
IF H=502 THEN GOSUB 51330
IF H=503 THEN GOSUB 51430
IF H=504 THEN GOSUB 51530
IF H=505 THEN GOSUB 51630
IF H=506 THEN GOSUB 51730
IF H=507 THEN GOSUB 51830
IF H=508 THEN GOSUB 51930
IF H=509 THEN GOSUB 52030
IF H=510 THEN GOSUB 52130
IF H=511 THEN GOSUB 52230
IF H=512 THEN GOSUB 52330
IF H=513 THEN GOSUB 52430
IF H=514 THEN GOSUB 52530
IF H=515 THEN GOSUB 52630
IF H=516 THEN GOSUB 52730
IF H=517 THEN GOSUB 52830
IF H=518 THEN GOSUB 52930
IF H=519 THEN GOSUB 53030
IF H=520 THEN GOSUB 53130
IF H=521 THEN GOSUB 53230
IF H=522 THEN GOSUB 53330
IF H=523 THEN GOSUB 53430
IF H=524 THEN GOSUB 53530
IF H=525 THEN GOSUB 53630
IF H=526 THEN GOSUB 53730
IF H=527 THEN GOSUB 53830
IF H=528 THEN GOSUB 53930
IF H=529 THEN GOSUB 54030
IF H=530 THEN GOSUB 54130
IF H=531 THEN GOSUB 54230
IF H=532 THEN GOSUB 54330
IF H=533 THEN GOSUB 54430
IF H=534 THEN GOSUB 54530
IF H=535 THEN GOSUB 54630
IF H=536 THEN GOSUB 54730
IF H=537 THEN GOSUB 54830
IF H=538 THEN GOSUB 54930
IF H=539 THEN GOSUB 55030
IF H=540 THEN GOSUB 55130
IF H=541 THEN GOSUB 55230
IF H=542 THEN GOSUB 55330
IF H=543 THEN GOSUB 55430
IF H=544 THEN GOSUB 55530
IF H=545 THEN GOSUB 55630
IF H=546 THEN GOSUB 55730
IF H=547 THEN GOSUB 55830
IF H=548 THEN GOSUB 55930
IF H=549 THEN GOSUB 56030
IF H=550 THEN GOSUB 56130
IF H=551 THEN GOSUB 56230
IF H=552 THEN GOSUB 56330
IF H=553 THEN GOSUB 56430
IF H=554 THEN GOSUB 56530
IF H=555 THEN GOSUB 56630
IF H=556 THEN GOSUB 56730
IF H=557 THEN GOSUB 56830
IF H=558 THEN GOSUB 56930
IF H=559 THEN GOSUB 57030
IF H=560 THEN GOSUB 57130
IF H=561 THEN GOSUB 57230
IF H=562 THEN GOSUB 57330
IF H=563 THEN GOSUB 57430
IF H=564 THEN GOSUB 57530
IF H=565 THEN GOSUB 57630
IF H=566 THEN GOSUB 57730
IF H=567 THEN GOSUB 57830
IF H=568 THEN GOSUB 57930
IF H=569 THEN GOSUB 58030
IF H=570 THEN GOSUB 58130
IF H=571 THEN GOSUB 58230
IF H=572 THEN GOSUB 58330
IF H=573 THEN GOSUB 58430
IF H=574 THEN GOSUB 58530
IF H=575 THEN GOSUB 58630
IF H=576 THEN GOSUB 58730
IF H=577 THEN GOSUB 58830
IF H=578 THEN GOSUB 58930
IF H=579 THEN GOSUB 59030
IF H=580 THEN GOSUB 59130
IF H=581 THEN GOSUB 59230
IF H=582 THEN GOSUB 59330
IF H=583 THEN GOSUB 59430
IF H=584 THEN GOSUB 59530
IF H=585 THEN GOSUB 59630
IF H=586 THEN GOSUB 59730
IF H=587 THEN GOSUB 59830
IF H=588 THEN GOSUB 59930
IF H=589 THEN GOSUB 60030
IF H=590 THEN GOSUB 60130
IF H=591 THEN GOSUB 60230
IF H=592 THEN GOSUB 60330
IF H=593 THEN GOSUB 60430
IF H=594 THEN GOSUB 60530
IF H=595 THEN GOSUB 60630
IF H=596 THEN GOSUB 60730
IF H=597 THEN GOSUB 60830
IF H=598 THEN GOSUB 60930
IF H=599 THEN GOSUB 61030
IF H=600 THEN GOSUB 61130
IF H=601 THEN GOSUB 61230
IF H=602 THEN GOSUB 61330
IF H=603 THEN GOSUB 61430
IF H=604 THEN GOSUB 61530
IF H=605 THEN GOSUB 61630
IF H=606 THEN GOSUB 61730
IF H=607 THEN GOSUB 61830
IF H=608 THEN GOSUB 61930
IF H=609 THEN GOSUB 62030
IF H=610 THEN GOSUB 62130
IF H=611 THEN GOSUB 62230
IF H=612 THEN GOSUB 62330
IF H=613 THEN GOSUB 62430
IF H=614 THEN GOSUB 62530
IF H=615 THEN GOSUB 62630
IF H=616 THEN GOSUB 62730
IF H=617 THEN GOSUB 62830
IF H=618 THEN GOSUB 62930
IF H=619 THEN GOSUB 63030
IF H=620 THEN GOSUB 63130
IF H=621 THEN GOSUB 63230
IF H=622 THEN GOSUB 63330
IF H=623 THEN GOSUB 63430
IF H=624 THEN GOSUB 63530
IF H=625 THEN GOSUB 63630
IF H=626 THEN GOSUB 63730
IF H=627 THEN GOSUB 63830
IF H=628 THEN GOSUB 63930
IF H=629 THEN GOSUB 64030
IF H=630 THEN GOSUB 64130
IF H=631 THEN GOSUB 64230
IF H=632 THEN GOSUB 64330
IF H=633 THEN GOSUB 64430
IF H=634 THEN GOSUB 64530
IF H=635 THEN GOSUB 64630
IF H=636 THEN GOSUB 64730
IF H=637 THEN GOSUB 64830
IF H=638 THEN GOSUB 64930
IF H=639 THEN GOSUB 65030
IF H=640 THEN GOSUB 65130
IF H=641 THEN GOSUB 65230
IF H=642 THEN GOSUB 65330
IF H=643 THEN GOSUB 65430
IF H=644 THEN GOSUB 65530
IF H=645 THEN GOSUB 65630
IF H=646 THEN GOSUB 65730
IF H=647 THEN GOSUB 65830
IF H=648 THEN GOSUB 65930
IF H=649 THEN GOSUB 66030
IF H=650 THEN GOSUB 66130
IF H=651 THEN GOSUB 66230
IF H=652 THEN GOSUB 66330
IF H=653 THEN GOSUB 66430
IF H=654 THEN GOSUB 66530
IF H=655 THEN GOSUB 66630
IF H=656 THEN GOSUB 66730
IF H=657 THEN GOSUB 66830
IF H=658 THEN GOSUB 66930
IF H=659 THEN GOSUB 67030
IF H=660 THEN GOSUB 67130
IF H=661 THEN GOSUB 67230
IF H=662 THEN GOSUB 67330
IF H=663 THEN GOSUB 67430
IF H=664 THEN GOSUB 67530
IF H=665 THEN GOSUB 67630
IF H=666 THEN GOSUB 67730
IF H=667 THEN GOSUB 67830
IF H=668 THEN GOSUB 67930
IF H=669 THEN GOSUB 68030
IF H=670 THEN GOSUB 68130
IF H=671 THEN GOSUB 68230
IF H=672 THEN GOSUB 68330
IF H=673 THEN GOSUB 68430
IF H=674 THEN GOSUB 68530
IF H=675 THEN GOSUB 68630
IF H=676 THEN GOSUB 68730
IF H=677 THEN GOSUB 68830
IF H=678 THEN GOSUB 68930
IF H=679 THEN GOSUB 69030
IF H=680 THEN GOSUB 69130
IF H=681 THEN GOSUB 69230
IF H=682 THEN GOSUB 69330
IF H=683 THEN GOSUB 69430
```

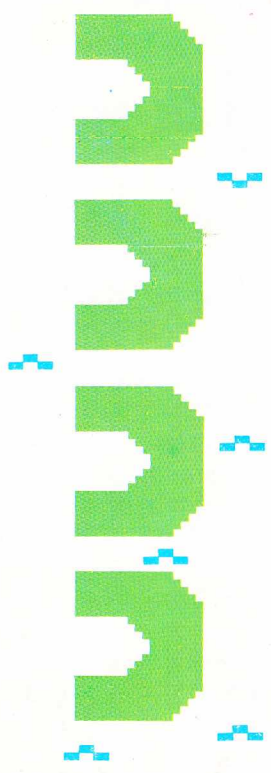


do para alvejar grupos de alienígenas invasores. Os inimigos se movem ameaçadoramente na direção da superfície da Terra. Perde-se uma "vida" quando o canhão de raios laser é atingido pelo fogo inimigo, ou quando os invasores conseguem chegar ao solo (alcançam a parte inferior da tela).

Há várias diferenças entre o fliperama original e as versões disponíveis para micros domésticos. Em vez de surgirem do nada, as hordas invasoras agora saem de um grande foguete, situado à esquerda da tela de tevê. Os próprios alienígenas são mais vivamente coloridos, formados por sprites mais complexos. As barreiras defensivas, atrás das quais o canhão de raios laser podia esconder-se na versão original, agora desaparecem, e os invasores percorrem uma distância menor antes de alcançar a parte inferior da tela. Mas um elemento permanece o mesmo: o som intimidante de batidas de coração, que acompanha a descida dos alienígenas, intensificando-se à medida que os monstros se aproximam. Isso acaba provocando uma estonteante descarga de adrenalina — provável razão do enorme sucesso do jogo.

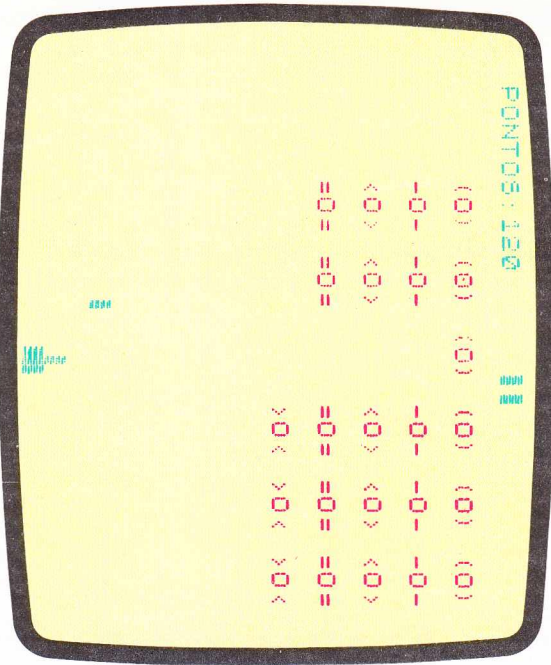
Outra característica tanto do jogo de fliperama como das versões para microcomputadores é o bônus misterioso, um prêmio oferecido ao jogador quando ele consegue atingir um dos discos voadores que ocasionalmente cruzam a tela da esquerda para a direita. Apesar do tempo e da disponibilidade de componentes de software mais sofisticados, o Space Invaders vem mantendo seu apelo, mostrando-se um jogo extremamente excitante e agradável — um verdadeiro "clássico do software".

**Space Invaders:** Para os computadores Apple II, Atari, Commodore 64, Sinclair Spectrum e TRS-80.  
**Editor:** Atari Corporation, Inglaterra.  
**Autor:** Atari.  
**Joystick:** Necessário.  
**Formato:** Cartucho.



# INVASORES PARA A LINHA SINCLAIR

Este é um jogo inspirado num clássico do fliperama, o Space Invaders. Você precisa defender a Terra dos invasores que se aproximam para destruí-la com suas bombas. No final, o planeta acabará sendo invadido, mas seu objetivo é conquistar o maior número de pontos. Quando atingir 1.500 pontos, você receberá um canhão extra e iniciará nova fase, mais difícil que a anterior. Seus controles são: [5] para mover para a esquerda, [8] para a direita e [0] para atirar. Após digitar o programa, inicie-o com o RUN.



**Prova de resistência**  
 Aproximando-se da Terra, os inimigos jogam suas bombas para eliminar toda e qualquer resistência. O jogo termina quando os alienígenas invadem o planeta ou destroem todos os canhões.

```

10 REM
   PARABENS SINCLAIR
   PONTOS: 1200
   (0) (8) (0) (0) (0)
   -0- -0- -0- -0-
   <0> <0> <0> <0>
   =0= =0= =0= =0=
   >0< >0< >0<
20 LET N=1
   GOTO 30
30 LET N=1
   GOTO 40
40 LET N=1
   GOTO 50
50 LET N=1
   GOTO 60
60 LET N=1
   GOTO 70
70 LET N=1
   GOTO 80
80 LET N=1
   GOTO 90
90 LET N=1
   GOTO 100
100 LET N=1
   GOTO 110
110 LET N=1
   GOTO 120
120 LET N=1
   GOTO 130
130 LET N=1
   GOTO 140
140 LET N=1
   GOTO 150
150 LET N=1
   GOTO 160
160 LET N=1
   GOTO 170
170 LET N=1
   GOTO 180
180 LET N=1
   GOTO 190
190 LET N=1
   GOTO 200
200 LET N=1
   GOTO 210
210 LET N=1
   GOTO 220
220 LET N=1
   GOTO 230
230 LET N=1
   GOTO 240
240 LET N=1
   GOTO 250
250 LET N=1
   GOTO 260
260 LET N=1
   GOTO 270
270 LET N=1
   GOTO 280
280 LET N=1
   GOTO 290
290 LET N=1
   GOTO 300
300 LET N=1
   GOTO 310
310 LET N=1
   GOTO 320
320 LET N=1
   GOTO 330
330 LET N=1
   GOTO 340
340 LET N=1
   GOTO 350
350 LET N=1
   GOTO 360
360 LET N=1
   GOTO 370
370 LET N=1
   GOTO 380
380 LET N=1
   GOTO 390
390 LET N=1
   GOTO 400
400 LET N=1
   GOTO 410
410 LET N=1
   GOTO 420
420 LET N=1
   GOTO 430
430 LET N=1
   GOTO 440
440 LET N=1
   GOTO 450
450 LET N=1
   GOTO 460
460 LET N=1
   GOTO 470
470 LET N=1
   GOTO 480
480 LET N=1
   GOTO 490
490 LET N=1
   GOTO 500
500 LET N=1
   GOTO 510
510 LET N=1
   GOTO 520
520 LET N=1
   GOTO 530
530 LET N=1
   GOTO 540
540 LET N=1
   GOTO 550
550 LET N=1
   GOTO 560
560 LET N=1
   GOTO 570
570 LET N=1
   GOTO 580
580 LET N=1
   GOTO 590
590 LET N=1
   GOTO 600
600 LET N=1
   GOTO 610
610 LET N=1
   GOTO 620
620 LET N=1
   GOTO 630
630 LET N=1
   GOTO 640
640 LET N=1
   GOTO 650
650 LET N=1
   GOTO 660
660 LET N=1
   GOTO 670
670 LET N=1
   GOTO 680
680 LET N=1
   GOTO 690
690 LET N=1
   GOTO 700
700 LET N=1
   GOTO 710
710 LET N=1
   GOTO 720
720 LET N=1
   GOTO 730
730 LET N=1
   GOTO 740
740 LET N=1
   GOTO 750
750 LET N=1
   GOTO 760
760 LET N=1
   GOTO 770
770 LET N=1
   GOTO 780
780 LET N=1
   GOTO 790
790 LET N=1
   GOTO 800
800 LET N=1
   GOTO 810
810 LET N=1
   GOTO 820
820 LET N=1
   GOTO 830
830 LET N=1
   GOTO 840
840 LET N=1
   GOTO 850
850 LET N=1
   GOTO 860
860 LET N=1
   GOTO 870
870 LET N=1
   GOTO 880
880 LET N=1
   GOTO 890
890 LET N=1
   GOTO 900
900 LET N=1
   GOTO 910
910 LET N=1
   GOTO 920
920 LET N=1
   GOTO 930
930 LET N=1
   GOTO 940
940 LET N=1
   GOTO 950
950 LET N=1
   GOTO 960
960 LET N=1
   GOTO 970
970 LET N=1
   GOTO 980
980 LET N=1
   GOTO 990
990 LET N=1
   GOTO 1000
1000 LET N=1
   GOTO 1010
1010 LET N=1
   GOTO 1020
1020 LET N=1
   GOTO 1030
1030 LET N=1
   GOTO 1040
1040 LET N=1
   GOTO 1050
1050 LET N=1
   GOTO 1060
1060 LET N=1
   GOTO 1070
1070 LET N=1
   GOTO 1080
1080 LET N=1
   GOTO 1090
1090 LET N=1
   GOTO 1100
1100 LET N=1
   GOTO 1110
1110 LET N=1
   GOTO 1120
1120 LET N=1
   GOTO 1130
1130 LET N=1
   GOTO 1140
1140 LET N=1
   GOTO 1150
1150 LET N=1
   GOTO 1160
1160 LET N=1
   GOTO 1170
1170 LET N=1
   GOTO 1180
1180 LET N=1
   GOTO 1190
1190 LET N=1
   GOTO 1200
1200 LET N=1
   GOTO 1210
1210 LET N=1
   GOTO 1220
1220 LET N=1
   GOTO 1230
1230 LET N=1
   GOTO 1240
1240 LET N=1
   GOTO 1250
1250 LET N=1
   GOTO 1260
1260 LET N=1
   GOTO 1270
1270 LET N=1
   GOTO 1280
1280 LET N=1
   GOTO 1290
1290 LET N=1
   GOTO 1300
1300 LET N=1
   GOTO 1310
1310 LET N=1
   GOTO 1320
1320 LET N=1
   GOTO 1330
1330 LET N=1
   GOTO 1340
1340 LET N=1
   GOTO 1350
1350 LET N=1
   GOTO 1360
1360 LET N=1
   GOTO 1370
1370 LET N=1
   GOTO 1380
1380 LET N=1
   GOTO 1390
1390 LET N=1
   GOTO 1400
1400 LET N=1
   GOTO 1410
1410 LET N=1
   GOTO 1420
1420 LET N=1
   GOTO 1430
1430 LET N=1
   GOTO 1440
1440 LET N=1
   GOTO 1450
1450 LET N=1
   GOTO 1460
1460 LET N=1
   GOTO 1470
1470 LET N=1
   GOTO 1480
1480 LET N=1
   GOTO 1490
1490 LET N=1
   GOTO 1500
1500 LET N=1
   GOTO 1510
1510 LET N=1
   GOTO 1520
1520 LET N=1
   GOTO 1530
1530 LET N=1
   GOTO 1540
1540 LET N=1
   GOTO 1550
1550 LET N=1
   GOTO 1560
1560 LET N=1
   GOTO 1570
1570 LET N=1
   GOTO 1580
1580 LET N=1
   GOTO 1590
1590 LET N=1
   GOTO 1600
1600 LET N=1
   GOTO 1610
1610 LET N=1
   GOTO 1620
1620 LET N=1
   GOTO 1630
1630 LET N=1
   GOTO 1640
1640 LET N=1
   GOTO 1650
1650 LET N=1
   GOTO 1660
1660 LET N=1
   GOTO 1670
1670 LET N=1
   GOTO 1680
1680 LET N=1
   GOTO 1690
1690 LET N=1
   GOTO 1700
1700 LET N=1
   GOTO 1710
1710 LET N=1
   GOTO 1720
1720 LET N=1
   GOTO 1730
1730 LET N=1
   GOTO 1740
1740 LET N=1
   GOTO 1750
1750 LET N=1
   GOTO 1760
1760 LET N=1
   GOTO 1770
1770 LET N=1
   GOTO 1780
1780 LET N=1
   GOTO 1790
1790 LET N=1
   GOTO 1800
1800 LET N=1
   GOTO 1810
1810 LET N=1
   GOTO 1820
1820 LET N=1
   GOTO 1830
1830 LET N=1
   GOTO 1840
1840 LET N=1
   GOTO 1850
1850 LET N=1
   GOTO 1860
1860 LET N=1
   GOTO 1870
1870 LET N=1
   GOTO 1880
1880 LET N=1
   GOTO 1890
1890 LET N=1
   GOTO 1900
1900 LET N=1
   GOTO 1910
1910 LET N=1
   GOTO 1920
1920 LET N=1
   GOTO 1930
1930 LET N=1
   GOTO 1940
1940 LET N=1
   GOTO 1950
1950 LET N=1
   GOTO 1960
1960 LET N=1
   GOTO 1970
1970 LET N=1
   GOTO 1980
1980 LET N=1
   GOTO 1990
1990 LET N=1
   GOTO 2000
2000 LET N=1
   GOTO 2010
2010 LET N=1
   GOTO 2020
2020 LET N=1
   GOTO 2030
2030 LET N=1
   GOTO 2040
2040 LET N=1
   GOTO 2050
2050 LET N=1
   GOTO 2060
2060 LET N=1
   GOTO 2070
2070 LET N=1
   GOTO 2080
2080 LET N=1
   GOTO 2090
2090 LET N=1
   GOTO 2100
2100 LET N=1
   GOTO 2110
2110 LET N=1
   GOTO 2120
2120 LET N=1
   GOTO 2130
2130 LET N=1
   GOTO 2140
2140 LET N=1
   GOTO 2150
2150 LET N=1
   GOTO 2160
2160 LET N=1
   GOTO 2170
2170 LET N=1
   GOTO 2180
2180 LET N=1
   GOTO 2190
2190 LET N=1
   GOTO 2200
2200 LET N=1
   GOTO 2210
2210 LET N=1
   GOTO 2220
2220 LET N=1
   GOTO 2230
2230 LET N=1
   GOTO 2240
2240 LET N=1
   GOTO 2250
2250 LET N=1
   GOTO 2260
2260 LET N=1
   GOTO 2270
2270 LET N=1
   GOTO 2280
2280 LET N=1
   GOTO 2290
2290 LET N=1
   GOTO 2300
2300 LET N=1
   GOTO 2310
2310 LET N=1
   GOTO 2320
2320 LET N=1
   GOTO 2330
2330 LET N=1
   GOTO 2340
2340 LET N=1
   GOTO 2350
2350 LET N=1
   GOTO 2360
2360 LET N=1
   GOTO 2370
2370 LET N=1
   GOTO 2380
2380 LET N=1
   GOTO 2390
2390 LET N=1
   GOTO 2400
2400 LET N=1
   GOTO 2410
2410 LET N=1
   GOTO 2420
2420 LET N=1
   GOTO 2430
2430 LET N=1
   GOTO 2440
2440 LET N=1
   GOTO 2450
2450 LET N=1
   GOTO 2460
2460 LET N=1
   GOTO 2470
2470 LET N=1
   GOTO 2480
2480 LET N=1
   GOTO 2490
2490 LET N=1
   GOTO 2500
2500 LET N=1
   GOTO 2510
2510 LET N=1
   GOTO 2520
2520 LET N=1
   GOTO 2530
2530 LET N=1
   GOTO 2540
2540 LET N=1
   GOTO 2550
2550 LET N=1
   GOTO 2560
2560 LET N=1
   GOTO 2570
2570 LET N=1
   GOTO 2580
2580 LET N=1
   GOTO 2590
2590 LET N=1
   GOTO 2600
2600 LET N=1
   GOTO 2610
2610 LET N=1
   GOTO 2620
2620 LET N=1
   GOTO 2630
2630 LET N=1
   GOTO 2640
2640 LET N=1
   GOTO 2650
2650 LET N=1
   GOTO 2660
2660 LET N=1
   GOTO 2670
2670 LET N=1
   GOTO 2680
2680 LET N=1
   GOTO 2690
2690 LET N=1
   GOTO 2700
2700 LET N=1
   GOTO 2710
2710 LET N=1
   GOTO 2720
2720 LET N=1
   GOTO 2730
2730 LET N=1
   GOTO 2740
2740 LET N=1
   GOTO 2750
2750 LET N=1
   GOTO 2760
2760 LET N=1
   GOTO 2770
2770 LET N=1
   GOTO 2780
2780 LET N=1
   GOTO 2790
2790 LET N=1
   GOTO 2800
2800 LET N=1
   GOTO 2810
2810 LET N=1
   GOTO 2820
2820 LET N=1
   GOTO 2830
2830 LET N=1
   GOTO 2840
2840 LET N=1
   GOTO 2850
2850 LET N=1
   GOTO 2860
2860 LET N=1
   GOTO 2870
2870 LET N=1
   GOTO 2880
2880 LET N=1
   GOTO 2890
2890 LET N=1
   GOTO 2900
2900 LET N=1
   GOTO 2910
2910 LET N=1
   GOTO 2920
2920 LET N=1
   GOTO 2930
2930 LET N=1
   GOTO 2940
2940 LET N=1
   GOTO 2950
2950 LET N=1
   GOTO 2960
2960 LET N=1
   GOTO 2970
2970 LET N=1
   GOTO 2980
2980 LET N=1
   GOTO 2990
2990 LET N=1
   GOTO 3000
3000 LET N=1
   GOTO 3010
3010 LET N=1
   GOTO 3020
3020 LET N=1
   GOTO 3030
3030 LET N=1
   GOTO 3040
3040 LET N=1
   GOTO 3050
3050 LET N=1
   GOTO 3060
3060 LET N=1
   GOTO 3070
3070 LET N=1
   GOTO 3080
3080 LET N=1
   GOTO 3090
3090 LET N=1
   GOTO 3100
3100 LET N=1
   GOTO 3110
3110 LET N=1
   GOTO 3120
3120 LET N=1
   GOTO 3130
3130 LET N=1
   GOTO 3140
3140 LET N=1
   GOTO 3150
3150 LET N=1
   GOTO 3160
3160 LET N=1
   GOTO 3170
3170 LET N=1
   GOTO 3180
3180 LET N=1
   GOTO 3190
3190 LET N=1
   GOTO 3200
3200 LET N=1
   GOTO 3210
3210 LET N=1
   GOTO 3220
3220 LET N=1
   GOTO 3230
3230 LET N=1
   GOTO 3240
3240 LET N=1
   GOTO 3250
3250 LET N=1
   GOTO 3260
3260 LET N=1
   GOTO 3270
3270 LET N=1
   GOTO 3280
3280 LET N=1
   GOTO 3290
3290 LET N=1
   GOTO 3300
3300 LET N=1
   GOTO 3310
3310 LET N=1
   GOTO 3320
3320 LET N=1
   GOTO 3330
3330 LET N=1
   GOTO 3340
3340 LET N=1
   GOTO 3350
3350 LET N=1
   GOTO 3360
3360 LET N=1
   GOTO 3370
3370 LET N=1
   GOTO 3380
3380 LET N=1
   GOTO 3390
3390 LET N=1
   GOTO 3400
3400 LET N=1
   GOTO 3410
3410 LET N=1
   GOTO 3420
3420 LET N=1
   GOTO 3430
3430 LET N=1
   GOTO 3440
3440 LET N=1
   GOTO 3450
3450 LET N=1
   GOTO 3460
3460 LET N=1
   GOTO 3470
3470 LET N=1
   GOTO 3480
3480 LET N=1
   GOTO 3490
3490 LET N=1
   GOTO 3500
3500 LET N=1
   GOTO 3510
3510 LET N=1
   GOTO 3520
3520 LET N=1
   GOTO 3530
3530 LET N=1
   GOTO 3540
3540 LET N=1
   GOTO 3550
3550 LET N=1
   GOTO 3560
3560 LET N=1
   GOTO 3570
3570 LET N=1
   GOTO 3580
3580 LET N=1
   GOTO 3590
3590 LET N=1
   GOTO 3600
3600 LET N=1
   GOTO 3610
3610 LET N=1
   GOTO 3620
3620 LET N=1
   GOTO 3630
3630 LET N=1
   GOTO 3640
3640 LET N=1
   GOTO 3650
3650 LET N=1
   GOTO 3660
3660 LET N=1
   GOTO 3670
3670 LET N=1
   GOTO 3680
3680 LET N=1
   GOTO 3690
3690 LET N=1
   GOTO 3700
3700 LET N=1
   GOTO 3710
3710 LET N=1
   GOTO 3720
3720 LET N=1
   GOTO 3730
3730 LET N=1
   GOTO 3740
3740 LET N=1
   GOTO 3750
3750 LET N=1
   GOTO 3760
3760 LET N=1
   GOTO 3770
3770 LET N=1
   GOTO 3780
3780 LET N=1
   GOTO 3790
3790 LET N=1
   GOTO 3800
3800 LET N=1
   GOTO 3810
3810 LET N=1
   GOTO 3820
3820 LET N=1
   GOTO 3830
3830 LET N=1
   GOTO 3840
3840 LET N=1
   GOTO 3850
3850 LET N=1
   GOTO 3860
3860 LET N=1
   GOTO 3870
3870 LET N=1
   GOTO 3880
3880 LET N=1
   GOTO 3890
3890 LET N=1
   GOTO 3900
3900 LET N=1
   GOTO 3910
3910 LET N=1
   GOTO 3920
3920 LET N=1
   GOTO 3930
3930 LET N=1
   GOTO 3940
3940 LET N=1
   GOTO 3950
3950 LET N=1
   GOTO 3960
3960 LET N=1
   GOTO 3970
3970 LET N=1
   GOTO 3980
3980 LET N=1
   GOTO 3990
3990 LET N=1
   GOTO 4000
4000 LET N=1
   GOTO 4010
4010 LET N=1
   GOTO 4020
4020 LET N=1
   GOTO 4030
4030 LET N=1
   GOTO 4040
4040 LET N=1
   GOTO 4050
4050 LET N=1
   GOTO 4060
4060 LET N=1
   GOTO 4070
4070 LET N=1
   GOTO 4080
4080 LET N=1
   GOTO 4090
4090 LET N=1
   GOTO 4100
4100 LET N=1
   GOTO 4110
4110 LET N=1
   GOTO 4120
4120 LET N=1
   GOTO 4130
4130 LET N=1
   GOTO 4140
4140 LET N=1
   GOTO 4150
4150 LET N=1
   GOTO 4160
4160 LET N=1
   GOTO 4170
4170 LET N=1
   GOTO 4180
4180 LET N=1
   GOTO 4190
4190 LET N=1
   GOTO 4200
4200 LET N=1
   GOTO 4210
4210 LET N=1
   GOTO 4220
4220 LET N=1
   GOTO 4230
4230 LET N=1
   GOTO 4240
4240 LET N=1
   GOTO 4250
4250 LET N=1
   GOTO 4260
4260 LET N=1
   GOTO 4270
4270 LET N=1
   GOTO 4280
4280 LET N=1
   GOTO 4290
4290 LET N=1
   GOTO 4300
4300 LET N=1
   GOTO 4310
4310 LET N=1
   GOTO 4320
4320 LET N=1
   GOTO 4330
4330 LET N=1
   GOTO 4340
4340 LET N=1
   GOTO 4350
4350 LET N=1
   GOTO 4360
4360 LET N=1
   GOTO 4370
4370 LET N=1
   GOTO 4380
4380 LET N=1
   GOTO 4390
4390 LET N=1
   GOTO 4400
4400 LET N=1
   GOTO 4410
4410 LET N=1
   GOTO 4420
4420 LET N=1
   GOTO 4430
4430 LET N=1
   GOTO 4440
4440 LET N=1
   GOTO 4450
4450 LET N=1
   GOTO 4460
4460 LET N=1
   GOTO 4470
4470 LET N=1
   GOTO 4480
4480 LET N=1
   GOTO 4490
4490 LET N=1
   GOTO 4500
4500 LET N=1
   GOTO 4510
4510 LET N=1
   GOTO 4520
4520 LET N=1
   GOTO 4530
4530 LET N=1
   GOTO 4540
4540 LET N=1
   GOTO 4550
4550 LET N=1
   GOTO 4560
4560 LET N=1
   GOTO 4570
4570 LET N=1
   GOTO 4580
4580 LET N=1
   GOTO 4590
4590 LET N=1
   GOTO 4600
4600 LET N=1
   GOTO 4610
4610 LET N=1
   GOTO 4620
4620 LET N=1
   GOTO 4630
4630 LET N=1
   GOTO 4640
4640 LET N=1
   GOTO 4650
4650 LET N=1
   GOTO 4660
4660 LET N=1
   GOTO 4670
4670 LET N=1
   GOTO 4680
4680 LET N=1
   GOTO 4690
4690 LET N=1
   GOTO 4700
4700 LET N=1
   GOTO 4710
4710 LET N=1
   GOTO 4720
4720 LET N=1
   GOTO 4730
4730 LET N=1
   GOTO 4740
4740 LET N=1
   GOTO 4750
4750 LET N=1
   GOTO 4760
4760 LET N=1
   GOTO 4770
4770 LET N=1
   GOTO 4780
4780 LET N=1
   GOTO 4790
4790 LET N=1
   GOTO 4800
4800 LET N=1
   GOTO 4810
4810 LET N=1
   GOTO 4820
4820 LET N=1
   GOTO 4830
4830 LET N=1
   GOTO 4840
4840 LET N=1
   GOTO 4850
4850 LET N=1
   GOTO 4860
4860 LET N=1
   GOTO 4870
4870 LET N=1
   GOTO 4880
4880 LET N=1
   GOTO 4890
4890 LET N=1
   GOTO 4900
4900 LET N=1
   GOTO 4910
4910 LET N=1
   GOTO 4920
4920 LET N=1
   GOTO 4930
4930 LET N=1
   GOTO 4940
4940 LET N=1
   GOTO 4950
4950 LET N=1
   GOTO 4960
4960 LET N=1
   GOTO 4970
4970 LET N=1
   GOTO 4980
4980 LET N=1
   GOTO 4990
4990 LET N=1
   GOTO 5000
5000 LET N=1
   GOTO 5010
5010 LET N=1
   GOTO 5020
5020 LET N=1
   GOTO 5030
5030 LET N=1
   GOTO 5040
5040 LET N=1
   GOTO 5050
5050 LET N=1
   GOTO 5060
5060 LET N=1
   GOTO 5070
5070 LET N=1
   GOTO 5080
5080 LET N=1
   GOTO 5090
5090 LET N=1
   GOTO 5100
5100 LET N=1
   GOTO 5110
5110 LET N=1
   GOTO 5120
5120 LET N=1
   GOTO 5130
5130 LET N=1
   GOTO 5140
5140 LET N=1
   GOTO 5150
5150 LET N=1
   GOTO 5160
5160 LET N=1
   GOTO 5170
5170 LET N=1
   GOTO 5180
5180 LET N=1
   GOTO 5190
5190 LET N=1
   GOTO 5200
5200 LET N=1
   GOTO 5210
5210 LET N=1
   GOTO 5220
5220 LET N=1
   GOTO 5230
5230 LET N=1
   GOTO 5240
5240 LET N=1
   GOTO 5250
5250 LET N=1
   GOTO 5260
5260 LET N=1
   GOTO 5270
5270 LET N=1
   GOTO 5280
5280 LET N=1
   GOTO 5290
5290 LET N=1
   GOTO 5300
5300 LET N=1
   GOTO 5310
5310 LET N=1
   GOTO 5320
5320 LET N=1
   GOTO 5330
5330 LET N=1
   GOTO 5340
5340 LET N=1
   GOTO 5350
5350 LET N=1
   GOTO 5360
5360 LET N=1
   GOTO 5370
5370 LET N=1
   GOTO 5380
5380 LET N=1
   GOTO 5390
5390 LET N=1
   GOTO 5400
5400 LET N=1
   GOTO 5410
5410 LET N=1
   GOTO 5420
5420 LET N=1
   GOTO 5430
5430 LET N=1
   GOTO 5440
5440 LET N=1
   GOTO 5450
5450 LET N=1
   GOTO 5460
5460 LET N=1
   GOTO 5470
5470 LET N=1
   GOTO 5480
5480 LET N=1
   GOTO 5490
5490 LET N=1
   GOTO 5500
5500 LET N=1
   GOTO 5510
5510 LET N=1
   GOTO 5520
5520 LET N=1
   GOTO 5530
5530 LET N=1
   GOTO 5540
5540 LET N=1
   GOTO 5550
5550 LET N=1
   GOTO 5560
5560 LET N=1
   GOTO 5570
5570 LET N=1
   GOTO 5580
5580 LET N=1
   GOTO 5590
5590 LET N=1
   GOTO 5600
5600 LET N=1
   GOTO 5610
5610 LET N=1
   GOTO 5620
5620 LET N=1
   GOTO 5630
5630 LET N=1
   GOTO 5640
5640 LET N=1
   GOTO 5650
5650 LET N=1
   GOTO 5660
5660 LET N=1
   GOTO 5670
5670 LET N=1
   GOTO 5680
5680 LET N=1
   GOTO 5690
5690 LET N=1
   GOTO 5700
5700 LET N=1
   GOTO 5710
5710 LET N=1
   GOTO 5720
5720 LET N=1
   GOTO 5730
5730 LET N=1
   GOTO 5740
5740 LET N=1
   GOTO 5750
5750 LET N=1
   GOTO 5760
5760 LET N=1
   GOTO 5770
5770 LET N=1
   GOTO 5780
5780 LET N=1
   GOTO 5790
5790 LET N=1
   GOTO 5800
5800 LET N=1
   GOTO 5810
5810 LET N=1
   GOTO 5820
5820 LET N=1
   GOTO 5830
5830 LET N=1
   GOTO 5840
5840 LET N=1
   GOTO 5850
5850 LET N=1
   GOTO 5860
5860 LET N=1
   GOTO 5870
5870 LET N=1
   GOTO 5880
5880 LET N=1
   GOTO 5890
5890 LET N=1
   GOTO 5900
5900 LET N=1
   GOTO 5910
5910 LET N=1
   GOTO 5920
5920 LET N=1
   GOTO 5930
5930 LET N=1
   GOTO 5940
5940 LET N=1
   GOTO 5950
5950 LET N=1
   GOTO 5960
5960 LET N=1
   GOTO 5970
5970 LET N=1
   GOTO 5980
5980 LET N=1
   GOTO 5990
5990 LET N=1
   GOTO 6000
6000 LET N=1
   GOTO 6010
6010 LET N=1
   GOTO 6020
6020 LET N=1
   GOTO 6030
6030 LET N=1
   GOTO 6040
6040 LET N=1
   GOTO 6050
6050 LET N=1
   GOTO 6060
6060 LET N=1
   GOTO 6070
6070 LET N=1
   GOTO 6080
6080 LET N=1
   GOTO 6090
6090 LET N=1
   GOTO 6100
6100 LET N=1
   GOTO 6110
6110 LET N=1
   GOTO 6120
6120 LET N=1
   GOTO 6130
6130 LET N=1
   GOTO 6140
6140 LET N=1
   GOTO 6150
6150 LET N=1
   GOTO 6160
6160 LET N=1
   GOTO 6170
6170 LET N=1
   GOTO 6180
6180 LET N=1
   GOTO 6190
6190 LET N=1
   GOTO 6200
6200 LET N=1
   GOTO 6210
6210 LET N=1
   GOTO 6220
6220 LET N=1
   GOTO 6230
6230 LET N=1
   GOTO 6240
6240 LET N=1
   GOTO 6250
6250 LET N=1
   GOTO 6260
6260 LET N=1
   GOTO 6270
6270 LET N=1
   GOTO 6280
6280 LET N=1
   GOTO 6290
6290 LET N=1
   GOTO 6300
6300 LET N=1
   GOTO 6310
6310 LET N=1
   GOTO 6320
6320 LET N=1
   GOTO 6330
6
```