

# MICROCOMPUTADOR CURSO PRÁTICO

TIRE O MÁXIMO DO MICRO EM 30 SEMANAS

PUBLICAÇÃO SEMANAL ILUSTRADA Cr\$ 6.500

4



**Impressoras matriciais • Loops**  
**A espionagem eletrônica**

*rioGráfica*

## MICROCOMPUTADOR CURSO PRÁTICO



### APLICAÇÕES

Computador espião 77



### LINGUAGENS

Divida e governe 79

Perdida no espaço 82



### PROJETOS DE PROGRAMAS

Ordene os dígitos 85



### RAIO X

Macintosh 86



### PERIFÉRICOS

Impressoras matriciais 88



### SOFTWARE

Arquivos seqüenciais 90



### CIÊNCIA DA COMPUTAÇÃO

Refinando o processo 92



### TÉCNICAS DE PROGRAMAÇÃO

Linhas de loop 95



### PERFIL

Investindo no futuro 96

# TESTE

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

- 1) Como se movimenta o cursor no computador Macintosh?
- 2) Qual a diferença entre um loop do tipo WHILE e outro do tipo REPEAT?
- 3) Como são formados os caracteres numa impressora matricial?
- 4) O que faz um "despistador" telefônico?
- 5) O que são diagramas de Venn?

### RESPOSTAS DO TESTE ANTERIOR

- 1) A impressora que necessita de papel especial metalizado é a eletrostática.
- 2) A vantagem de um algoritmo genérico é que ele pode ser reutilizado em outros programas.
- 3) Arquivos gravados em fita só podem ser acessados seqüencialmente, isto é, na ordem em que foram gravados.
- 4) O Sharp PC-1251 armazena dados em microfita.
- 5) Para impedir que a tartaruga do LOGO saia fora dos limites da tela, usa-se o comando FECHADO.

### PLANO DA OBRA

MICROCOMPUTADOR - CURSO PRÁTICO é uma coleção de 31 fascículos de periodicidade semanal. Os trinta primeiros terão, cada um, vinte páginas internas (miolo) e quatro capas, além de uma introdução de dezesseis páginas acompanhando o fascículo 1. O fascículo 31 terá doze páginas e quatro capas, e 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 25 (**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 31); miolos dos fascículos de 16 a 31; e guardas.

**Jogos** - Guardas; frontispício e sumário (que constituem as 3.<sup>as</sup> e 4.<sup>as</sup> capas do fascículo 31); 3.<sup>as</sup> e 4.<sup>as</sup> capas dos fascículos de 1 a 30; e guardas.

**Nota:** Muitos dos equipamentos que aparecem na seção RAI X não são normalmente comercializados no Brasil. Sua divulgação é feita a título de informação ao leitor.

**Errata:** A RAM básica do IBM PC tem 64 K, e não 60 K, como indicamos na página 12.

**rioGráfica**

**Diretoria Executiva:** Oscar Neves; Filipe Zander; Nilo Sérgio de Almeida; Danilo Esteves Costa; **GRUPO EDUCAÇÃO E CULTURA - Planejamento e Comercialização - Diretor:** Roberto Combochi; **Gerente de Marketing:** Jaime Rodrigues; **Gerente de Produto:** José Mauro Porto; **Gerente do P.C.P.:** Aylton Menezes; **Gerente de Vendas:** Rubens Barbosa; **Coordenadores de Promoção:** Paulo Cesar M. Seixas; Edgar Mello M. Neto; **Gerente de Circulação:** Norberto Martin; **Supervisor de Operações:** Abel Pereira Pinto - **Editorial - Diretor:** João J. Noro; **Editor-chefe:** Maurício Rittner;

**Administração Editorial:** Heitor de Souza Paixão; **Editor:** Jesse Navarro; **Revisores:** Isis Augusta Loyolla; Caolida Guerra; Luiz Vicente Vieira Filho; Marcos Domingos Agathão; Maria da Graça Mendonça Couto; **Diretor de Arte:** Aníbal Monteiro; **Chefe de Arte:** Bonifácio Duardes Miranda; **Chefe de Estúdio:** José Yuji Kuribayashi; **Assistente de Arte:** Miguel Luis Escámez Simón - **Colaboradores - Editores-assistentes:** Isa Mara Lando; Luiz Carlos Pizarro Marin; Cláudio Marcondes; **Texto:** Ana Luíza Mahlmeister; **Preparação de Texto:** Luiz Carlos Cardoso; **Tradução:** Maria Ester Menezes Marinho; Maria Clara Cescato; Newton Ruberval Eichemberg; Patrícia de Paiva e Castro; **Pesquisa Iconográfica:** Stella Maria Quente; **Arte:** Toshio Morobosi; **Assessoria Técnica:** Derval Junqueira de Aquino; Lauro de Lauro; Luiz Henrique Alayon. As fotos não creditadas pertencem à obra original. Siguem-se os créditos das demais: Itautec; Photoservice; Fernando Scavone; Salvador de Rosa. © APSIF Copenhagen, 1984; © Orbis Publishing Ltd., 1984; © Editora Rio Gráfica Ltda., 1985, para a língua portuguesa. **Composição:** ERG Ltda. e AM Produções Gráficas Ltda. **Impressão:** JIBG

**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: Electroliber Lda. - Rua Prof. Reinaldo dos Santos, 1468, Lisboa.



# COMPUTADOR ESPIÃO



**Em quase todo o mundo moderno — inclusive no Brasil —, órgãos responsáveis pelo cumprimento da lei envolvem-se em atividades ilegais ao lançar mão dos mais recentes avanços da espionagem eletrônica.**

O uso oficial da espionagem computadorizada vem ganhando terreno e atinge cada vez mais nosso cotidiano.

Em certos casos, o controle tornou-se rotina de vital importância para os órgãos de segurança. É o caso do registro de certificados de propriedade de automóvel, carteiras de motorista e carteiras de identidade. A possibilidade de integração dos vários sistemas é crescente, devido à correlação entre os arquivos e, por exemplo, os computadores da polícia. Isso dá às autoridades uma capacidade extra de controle e vigilância das atividades de toda a população.

Em muitos países já existe preocupação com essa possibilidade e iniciativas na elaboração de leis que protejam o cidadão do abuso de poder.

No entanto, alguns acreditam que tais leis serão constantemente atropeladas pelos avanços da tecnologia.

Muitos dos processos utilizados em espionagem computadorizada e em sistemas de segurança baseiam-se no reconhecimento de padrões. A técnica permite ao computador comparar aquilo que ele “vê” com modelos já armazenados em sua memória. A desvantagem desse tipo de sistema é que ocupa muito espaço de memória e muita capacidade de processamento, além de um dispêndio de tempo relativamente grande. Mas a disponibilidade e o barateamento do espaço de memória e da capacidade de processamento possibilitaram importantes avanços nessa área.

## Policiais auxiliares

Um exemplo desse avanço é o sistema de impressões digitais instalado na década de 80 em alguns países europeus. Ele tem capacidade de armazenar 650.000 impressões digitais e 100.000 “marcas”, isto é, impressões parciais encontradas em locais onde ocorreram crimes. O sistema compara as marcas com as impressões armazenadas e verifica se correspondem a alguma delas.

## “Grampeando” o escritório

Há inúmeros lugares onde os aparelhos de escuta podem ser escondidos, variando do mais óbvio aos de acesso mais difícil. Na foto, os percevejos poderiam estar ocultos nos seguintes locais:

- 1 O telefone. O percevejo pode estar no bocal, no corpo do telefone ou na luminária ao lado.
- 2 O vaso de planta. Na terra, embaixo do vaso ou camuflado como um percevejo real.
- 3 A parede. Percevejos já foram encontrados enfiados na parede como tachas ou atrás de revestimentos de cortiça.
- 4 O quadro. Disfarçado como um gancho de suporte ou escondido atrás da moldura.
- 5 A escrivaninha. Embaixo do tampo ou em qualquer das gavetas.



## As ferramentas do ofício

### Proteção computadorizada

Pessoas que lidam com informações sigilosas protegem-se da espionagem eletrônica usando dispositivos como este despistador telefônico computadorizado. Em vez de falar ao telefone, o usuário do aparelho digita sua mensagem no teclado. O dispositivo a envia de forma silenciosa pelas linhas telefônicas usuais. A mensagem é então recebida na tela de um sistema complementar. Um sintetizador de voz também pode converter os impulsos que chegam em mensagem audível.



### Mensagem em código

Similar ao despistador, esse sistema manda uma mensagem codificada a partir de um manuscrito. A mensagem codificada conserva-se protegida de bisbilhoteiros e "grampeadores", pois é transmitida silenciosamente. Mesmo assinaturas podem ser enviadas dessa forma.



### Analisador de tensão

Mede a incidência de tensão na voz. O aparelho demonstra os resultados imediatamente, sob forma numérica simples. Trata-se de sofisticado detector de mentiras e indica também se a pessoa está tensa ou ansiosa.



Essa aplicação é conseguida por minicomputadores capazes de processar matrizes altamente elaboradas, monitores e câmaras de televisão de excelente desempenho. Mesmo assim, o equipamento pode checar apenas duzentas ou trezentas marcas por dia.

Um sistema similar de reconhecimento e comparação de padrões foi montado numa ponte de grande movimento, na Inglaterra. Câmaras dirigidas às pistas de trânsito captam imagens das placas dos carros que se aproximam. O computador analisa as imagens e compara os números das placas com os de um arquivo de carros procurados. Quando é o caso, transmite-se por rádio, diretamente às viaturas da polícia rodoviária, a informação de que um desses carros foi visto.

Uma das áreas em que o desenvolvimento de hardware para microcomputadores teve efeitos mais significativos foi na produção de dispositivos de espionagem. Por suas dimensões cada vez menores, foram apelidados de "percevejos".

A tecnologia do chip possibilitou a produção de radiotransmissores do tamanho de um grão de arroz com sofisticados controles eletrônicos incorporados. Um dispositivo típico dessa categoria extrai sua energia da fonte que alimenta o telefone em que está instalado.

Há também percevejos com fonte de energia própria. Se deixados num canto da sala, captarão toda a conversa daquele ambiente e a transmitirão a um receptor distante. Como o outro, ele só é acionado quando alguém estiver conversando.

Mais ao estilo James Bond, existe ainda o percevejo remoto. O dispositivo lança um raio laser a uma janela e capta as vibrações provocadas no vidro por uma conversa. A conversa é restaurada, por computador, a partir das interferências que aparecem no raio refletido.

Em operações militares, o computador tem ainda outra função. Com o objetivo de evitar a espionagem do inimigo, os transmissores e receptores de rádio, no campo de batalha, utilizam um sistema controlado por microprocessadores. A mensagem sofre modificação de frequência e "pula" para a outra segundo um código preestabelecido. Utiliza-se esse procedimento para evitar que outro receptor capte a transmissão.

## Terminais do futuro

Os computadores usados para espionagem e segurança são equipamentos de grande porte, do mesmo tipo e capacidade das máquinas usadas na decifração de códigos secretos em órgãos como a Agência Central de Informações (CIA), nos Estados Unidos. Mas os avanços da tecnologia de hardware caminham para a miniaturização. Logo a identificação de impressões digitais, vozes e rostos será feita em minutos e a baixo custo.

Num futuro próximo, carros de polícia deverão estar equipados com computadores capazes de acessar instantaneamente dados de um registro escolar, criminal, médico, de seguro social ou qualquer outro arquivo oficial, pela simples introdução de um cartão magnético de identificação numa das entradas do terminal. A máquina analisará impressões digitais ou fotografias comparando-as com as dos arquivos centrais para certificar-se da identidade do suspeito.



# DIVIDA E GOVERNE

Valendo-se de procedimentos recursivos (que chamam a si próprios), o LOGO produz imagens gráficas de características incomuns inventadas por matemáticos para ilustrar certos paradoxos.

Ilustra a recursion no LOGO um programa projetado para traçar figuras de “árvores”. No princípio, desenhamos apenas o tronco com dois galhos, um à direita e outro à esquerda. A ramagem é produzida do mesmo modo (embora seus galhos sejam cada vez menores), com um caule central e os galhos direito e esquerdo.

O procedimento para o traçado de uma “árvore binária” desse tipo requer a entrada de dois dados: o comprimento do caule e o grau do “nível”. O comprimento dos ramos divide-se ao meio a cada nível que se afasta do tronco.

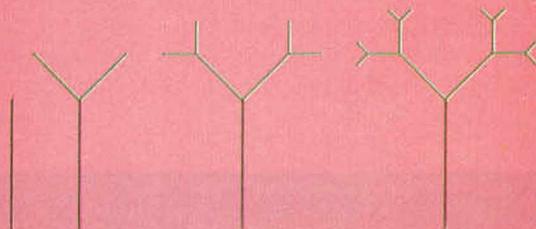
```
AP RAMO :COMP :NIVEL
SE :NIVEL = 0 ENTÃO PARE
FR :COMP
ES 45
RAMO (:COMP/2) (:NIVEL - 1)
DI 90
RAMO (:COMP/2) (:NIVEL - 1)
ES 45
VO :COMP
FIM
```

Observe que o procedimento não modifica a posição e a direção — ou seja, o “estado” — da tartaruga. Isso é importante, pois, de outro modo, o estado dela mudaria todas as vezes que o procedimento chamasse a si mesmo, tornando impossível a realização do desenho.

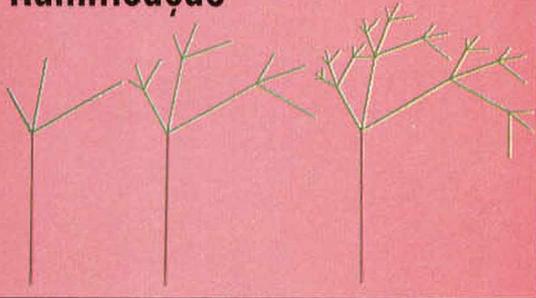
A árvore traçada por esse procedimento não é das mais realistas. Para torná-la mais parecida com as árvores verdadeiras, pode-se adaptar de várias maneiras o procedimento. A versão que apresentamos em seguida traça, a cada nível, três ramos de comprimento diferente:

```
AP RAMO1 :COMP :NIVEL
SE :NIVEL = 0 ENTÃO PARE
FR :COMP
ES 30
RAMO1 (:COMP/3) (:NIVEL - 1)
DI 40
RAMO1 (:COMP/2) (:NIVEL - 1)
DI 50
RAMO1 (:COMP/1.5) (:NIVEL - 1)
ES 60
VO :COMP
FIM
```

## Árvore binária



## Ramificação



## Polígonos quadriculados

```
AP TABULEIRO :COMP :NIVEL
SE :NIVEL = 0 ENTÃO REPITA 4 [FR :COMP DI 90]
PARE
TABULEIRO (:COMP/2) (:NIVEL - 1)
FR (:COMP/2)
TABULEIRO (:COMP/2) (:NIVEL - 1)
DI 90
FR (:COMP/2)
ES 90
TABULEIRO (:COMP/2) (:NIVEL - 1)
VO (:COMP/2)
TABULEIRO (:COMP/2) (:NIVEL - 1)
ES 90
FR (:COMP/2)
DI 90
FIM
```

Elabore um procedimento análogo para dividir um triângulo em quatro triângulos menores. Depois divida cada um desses em quatro, e assim por diante. Isso vai constituir o **exercício 5** desta seção.

## Flocos de neve

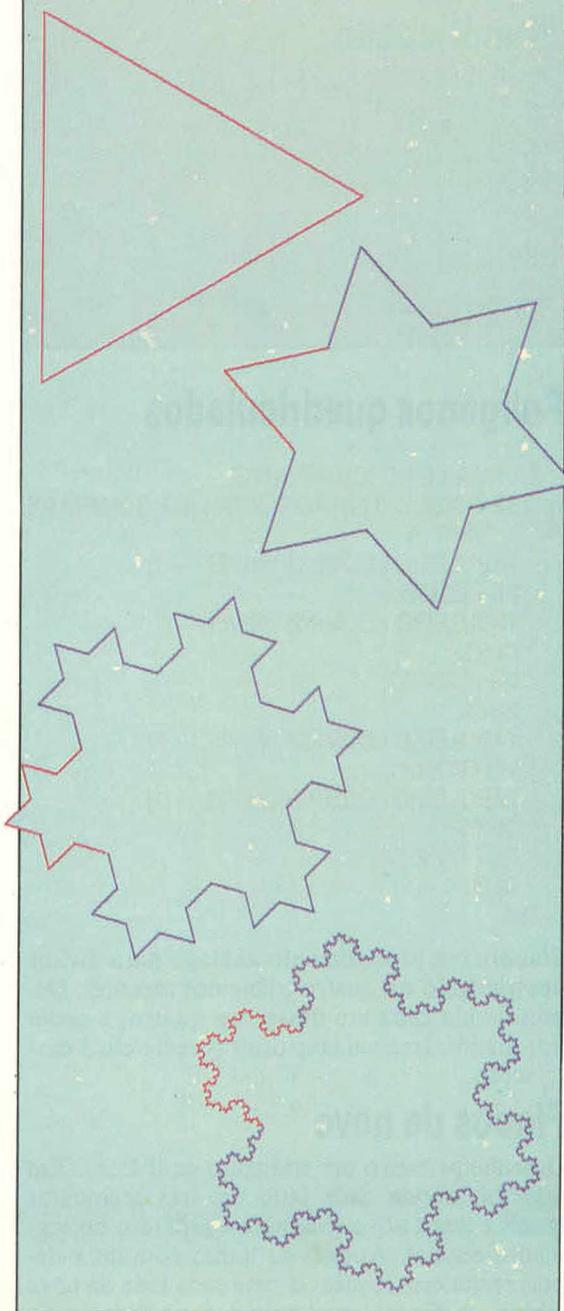
Desenhe primeiro um triângulo equilátero. Em seguida, divida cada lado em três segmentos iguais e trace novo triângulo equilátero no segmento central. Apague as linhas comuns e depois repita essa seqüência para cada lado da nova figura. A figura resultante é conhecida como



“curva dos flocos de neve”, por causa de sua aparência.

```
AP NEVE :TAM :NIVEL
  REPITA 3 (LADO :TAM :NIVEL DI 120)
FIM
AP LADO :TAM :NIVEL
  SE :NIVEL = 0 ENTAO FR :TAM PARE
  LADO (:TAM/3) (:NIVEL - 1)
  ES 60
  LADO (:TAM/3) (:NIVEL - 1)
  DI 120
  LADO (:TAM/3) (:NIVEL - 1)
  ES 60
  LADO (:TAM/3) (:NIVEL - 1)
FIM
```

### Curva dos flocos de neve



Note que LADO não é transparente ao estado, mas deixa a tartaruga no ponto certo para o traçado da figura seguinte.

Se esse processo de divisão se repetir indefinidamente (até o limite, na expressão dos matemáticos), o resultado será uma curva de comprimento infinito que, no entanto, circunda uma área finita. Pode-se provar que essa curva não é unidimensional nem bidimensional, mas possui características intermediárias.

Linha semelhante é montada a partir de um quadrado, dividindo-se cada um de seus lados em três partes iguais para construir quadrados nos segmentos médios, e assim por diante. Tente elaborar um procedimento que faça isso. (Exercício 6.)

As curvas mostradas a seguir foram inventadas pelo matemático polonês Waclaw Sierpinski (1882-1969). Se o processo é levado até o limite, obtém-se uma curva (uma linha unidimensional) que passa por todos os pontos do quadrado circundante (uma figura bidimensional). Várias outras curvas desse tipo apresentam comprimento pouco usual.

O procedimento empregado para o traçado delas é complexo. No nível 1, por exemplo, a curva compõe-se de quatro lados (mostrados em azul), interligados por quatro diagonais.

Assim, o procedimento principal, SIERP, apenas divide o processo em quatro seções, para que o procedimento UM.LADO opere com um deles por vez.

Considere agora apenas um dos lados, que é formado de três linhas — uma diagonal, uma vertical ou horizontal e outra diagonal. No nível 2, um conjunto menor de três linhas substitui cada uma das diagonais. A linha vertical ou horizontal é substituída por dois conjuntos semelhantes de três linhas unidas por outra.

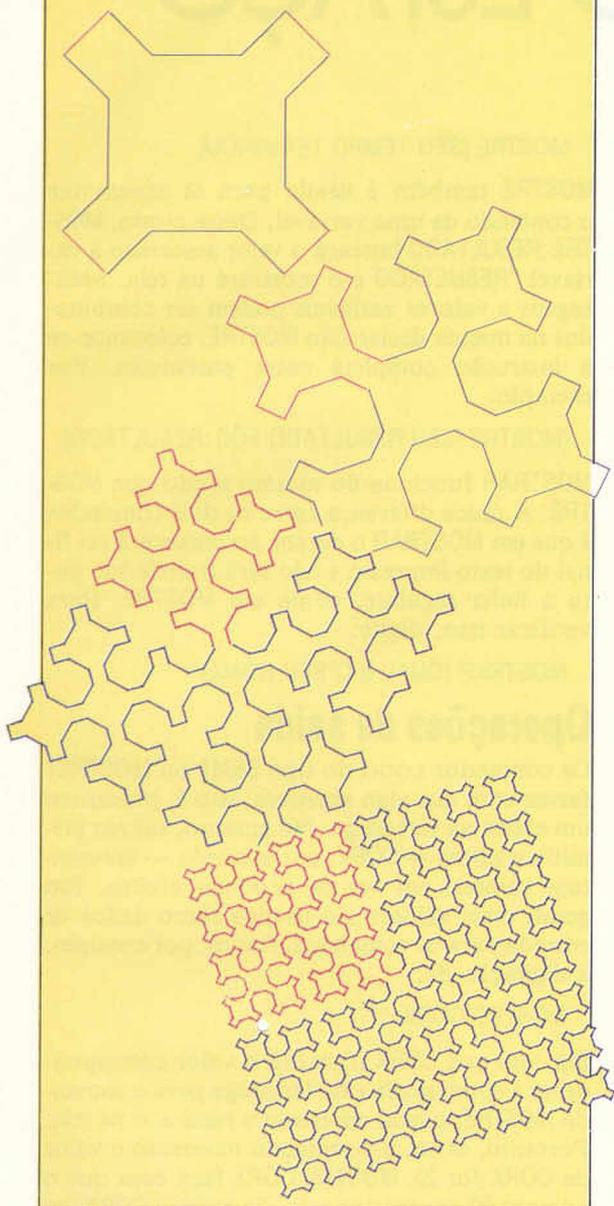
A seguir apresentamos os procedimentos para o traçado das curvas, com o comando FACA usado para inicializar DIAG:

```
AP SIERP :LADO :NIVEL
  FACA "DIAG :LADO / RQD2
  REPITA 4 [UM.LADO :NIVEL DI 45 FR :DIAG DI 45]
FIM
```

```
AP UM.LADO :NIVEL
  SE :NIVEL = 0 PARE
  UM.LADO (:NIVEL - 1)
  DI 45
  FR :DIAG
  DI 45
  UM.LADO (:NIVEL - 1)
  ES 90
  FR :LADO
  ES 90
  UM.LADO (:NIVEL - 1)
  DI 45
  FR :DIAG
  DI 45
  UM.LADO (:NIVEL - 1)
FIM
```



## Curva de Sierpinski



## Simplificação

Abreviações dos comandos apresentados nesta matéria:

FRENTE	FR
VOLTE	VO
DIREITA	DI
ESQUERDA	ES
LISTAR	LI

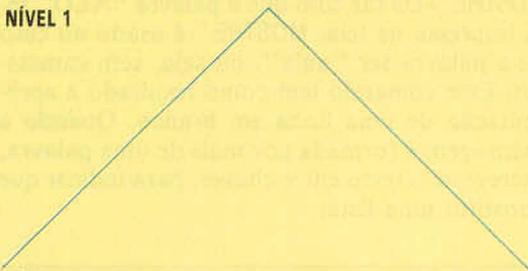
Outros comandos empregados:

SE  
ENTAO  
REPITA  
FACA  
FIM

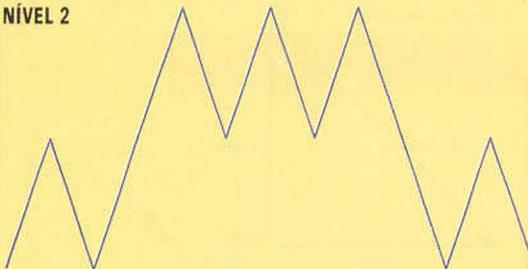
## Exercício 7

As ilustrações abaixo mostram figuras que, no limite, definem curvas sem gradiente. O primeiro nível consiste em duas linhas — uma que sobe e outra que desce. Para passar ao nível seguinte, substitua a linha ascendente por uma linha interrompida formada de seis partes. Esta se eleva até a metade da altura da linha original e, em seguida, desce até embaixo. Depois, sobe de novo até a metade da altura, continua até a altura máxima, cai até a metade e, por fim, sobe de novo até a altura máxima. A linha descendente divide-se em seis seções pelo mesmo processo.

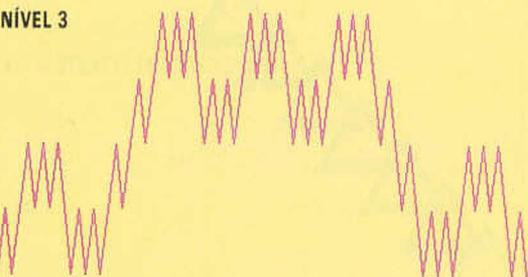
NÍVEL 1



NÍVEL 2



NÍVEL 3



Tente escrever um conjunto de procedimentos que trace essa série de curvas. Use o comando DEFXY em vez de FR e DI. Seu procedimento mais abrangente deve dividir a tarefa em duas partes — uma para a subida e outra para a descida. A seguir, serão necessários dois procedimentos separados para cada parte do procedimento de nível mais alto. (Tais procedimentos podem chamar-se uns aos outros, assim como a si mesmos.)

## Respostas do exercício 4

Um procedimento recorrente para o traçado da torre de quadrados:

```
AP TORRE :TAM
SE :TAM < 5 ENTAO PARE
QUAD :TAM
MOVA :TAM
TORRE (:TAM/2)
FIM
```

```
AP QUAD :TAM
REPITA 4 [FR :TAM DI 90]
FIM
```

```
AP MOVA :TAM
FR :TAM
DI 90
FR (:TAM/4)
ES 90
FIM
```







do no momento e passa o controle ao procedimento que o chamou. Isso pode ser verificado no procedimento MAX, que apresenta como dado de saída o maior dentre dois números:

```
AP MAX :X :Y
SE :X > :Y ENTAO SAIDA :X
SAIDA :Y
FIM
```

MOSTRE MAX 6 2 apresentará como resultado o número 6. Tente criar um procedimento que dê o valor absoluto de um número, de modo que tanto MOSTRE ABS 4 como MOSTRE ABS (- 4) terão como resultado o valor 4.

O jogo Tartaruga no Espaço pedirá que você digite seu nome e pressione RETURN. Isso é feito pelo procedimento que mostramos a seguir:

```
AP DAR.NOME
MEIATELA
MOSTRAR [QUAL E O SEU NOME?]
FACA "NOME PRIMEIRO ENTRE
(MOSTRAR "ALO :NOME)
FIM
```

ENTRE espera que uma linha seja digitada e finalizada com um RETURN. Ele então apresenta a linha sob a forma de lista. PRIMEIRO produz o primeiro elemento de uma lista. Experimente o procedimento DAR.NOME e digite o nome "PAULO". Em seguida, veja o que acontece ao digitar "PAULO MARCELO".

Os movimentos da tartaruga na tela serão controlados pelo acionamento das teclas D, E e C. D fará com que a tartaruga gire 30° em sentido horário (para a direita); E fará com que ela gire 30° em sentido anti-horário (para a esquerda). Por fim, C é usado para "chutar" a tartaruga — aumentando a velocidade dela qualquer que seja sua direção em determinado momento. A tartaruga estará em movimento pela tela e reagirá de imediato se as teclas forem pressionadas. Seria muito útil se dispuséssemos de um primitivo LOGO — LEIATECLA, talvez — que mostrasse a última tecla acionada. Se ele existisse, poderíamos escrever:

```
AP COMANDO
FACA "COM LEIATECLA
SE :COM = "D ENTAO DI 30
SE :COM = "E ENTAO ES 30
SE :COM = "C ENTAO CHUTE
FIM
```

Não existe um primitivo desse tipo. Contudo, podemos escrevê-lo sob a forma de procedimento:

```
AP LEIATECLA
SE CP? ENTAO SAIDA PEGUE
SAIDA"
FIM
```

Quando se pressiona uma tecla, ela é gravada na memória temporária (buffer) do teclado. PEGUE apenas lê o último caractere introduzido no buffer. Caso este esteja vazio, PEGUE esperará que uma tecla seja pressionada para ler o caractere relevante. CP? será verdadeiro se o buffer con-

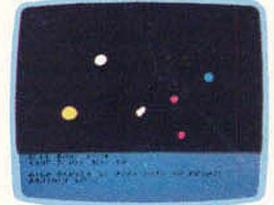
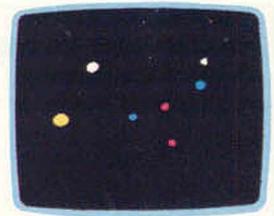
tiver algum caractere, e falso se estiver vazio. Desse modo, LEIATECLA terá como resultado o último caractere introduzido no buffer, ou então uma palavra nula no caso de este estar vazio.

## A tartaruga dinâmica

Na realidade, essa tartaruga viajando pelo espaço é uma "tartaruga dinâmica": tem velocidade, assim como posição e direção, como qualquer tartaruga viva.

Por estar no espaço, não haverá atrito ou gravidade. Ela obedecerá, portanto, às leis do movimento formuladas por Newton. Nossa ilustração tornará isso mais claro, mas, como exemplo, vamos supor que a tartaruga esteja se movendo da esquerda para a direita da tela com velocidade de 1 unidade. Se a tecla E for pressionada, a tartaruga mudará de direção, passando a mover-se para a parte de cima da tela. No entanto, seu impulso fará com que continue se movimentando na horizontal. Quando a tecla C é acionada, a tartaruga leva um "chute" na direção em que foi posicionada. Se for impulsiona para cima com a velocidade 1, o movimento resultante terá a velocidade de 1,4 na diagonal. Portanto, a tartaruga dinâmica permite a realização de experiências com um corpo sujeito às leis de Newton. Ela foi especialmente projetada para que você desenvolva uma compreensão intuitiva dessas leis, sem que seja necessário entender todas as equações envolvidas.

No programa, a velocidade da tartaruga dinâmica é considerada em termos de duas componentes, relativas aos eixos  $x$  e  $y$ . Essas componentes são determinadas por meio das funções SEN e COS. Os únicos controles são os três já mencionados. Para começar o jogo, digite INICIO. Você dispõe de tempo limitado para alcançar seu objetivo, e o programa mantém um registro do melhor resultado obtido.



### Tartaruga extraterrestre

O programa que vimos desenha apenas a tartaruga e seu alvo. As estrelas e os planetas destas figuras foram criados por simples procedimentos para o traçado de círculos.

## Projeto de programa

Elabore um programa para o jogo Alunissagem. Nele, você está pilotando uma nave a certa distância acima da superfície lunar. Resta pouco combustível e, devido à atração constante da gravidade, a velocidade de descida também aumenta de forma constante a cada segundo. Acionando a tecla F, você liga os motores da nave, reduzindo a velocidade de descida. Isso provoca, no entanto, maior consumo de combustível. O objetivo do jogo é fazer com que a descida seja lenta o suficiente para permitir uma alunissagem segura.

## Novos comandos

```
CORX
CORY
DEFXY
MEIATELA
TODATELA
```





## Respostas do exercício 5

### A) Triângulos aninhados

```
AP TRI :TAM :NIVEL
SE :NIVEL = 0 ENTAO REPITA
3 [FR :TAM DI 120]
PARE
TRI (:TAM/2) (:NIVEL - 1)
FR (:TAM/2)
TRI (:TAM/2) (:NIVEL - )
DI 60
TRI (:TAM/2) (:NIVEL - 1)
FR (:TAM/2)
DI 60
TRI (:TAM/2) (:NIVEL - 1)
ES 60
VO (:TAM/2)
ES 60
VO (:TAM/2)
FIM
```

### B) Floco de neve

```
AP NEVE1 :TAM :NIVEL
REPITA 4 [LADO1 :TAM :NIVEL DI 90]
FIM
AP LADO1 :TAM :NIVEL
SE :NIVEL = 0 ENTAO FR :TAM PARE
LADO1 (:TAM/3) (:NIVEL - 1)
ES 90
LADO1 (:TAM/3) (:NIVEL - 1)
DI 90
LADO1 (:TAM/3) (:NIVEL - 1)
DI 90
LADO1 (:TAM/3) (:NIVEL - 1)
ES 90
LADO1 (:TAM/3) (:NIVEL - 1)
FIM
```

### C) Curva sem gradiente

```
AP CURVA :FASEX :FASEY :NIVEL
SOBE :FASEX :FASEY :NIVEL
DESCE :FASEX :FASEY :NIVEL
FIM
AP SOBE :FASEX :FASEY :NIVEL
SE :NIVEL = 0 ENTAO DEFXY (CORY + :FASEX)
(CORY + :FASEY) PARE
SOBE (:FASEX/6) (:FASEY/2) (:NIVEL - 1)
DESCE (:FASEX/6) (:FASEY/2) (:NIVEL - 1)
SOBE (:FASEX/6) (:FASEY/2) (:NIVEL - 1)
SOBE (:FASEX/6) (:FASEY/2) (:NIVEL - 1)
DESCE (:FASEX/6) (:FASEY/2) (:NIVEL - 1)
DESCE (:FASEX/6) (:FASEY/2) (:NIVEL - 1)
SOBE (:FASEX/6) (:FASEY/2) (:NIVEL - 1)
DESCE (:FASEX/6) (:FASEY/2) (:NIVEL - 1)
FIM
AP DESCE :FASEX :FASEY :NIVEL
SE :NIVEL = 0 ENTAO DEFXY (CORY + :FASEX)
(CORY - :FASEY) PARE
DESCE (:FASEX/6) (:FASEY/2) (:NIVEL - 1)
SOBE (:FASEX/6) (:FASEY/2) (:NIVEL - 1)
DESCE (:FASEX/6) (:FASEY/2) (:NIVEL - 1)
DESCE (:FASEX/6) (:FASEY/2) (:NIVEL - 1)
SOBE (:FASEX/6) (:FASEY/2) (:NIVEL - 1)
DESCE (:FASEX/6) (:FASEY/2) (:NIVEL - 1)
FIM
```

## Tartaruga no espaço

```
AP INICIO
FAÇA "MAX 0
DESENHE
ST
ALVO
JOGAR
FIM

AP ALVO
SC DEFXY 0 5 CL
DI 90
REPITA 36 [FR 31.4/36
DI 10]
SC
FIM

AP JOGAR
DAR.NOME
INICIALIZAR
IMPULSO
FIM

AP DAR.NOME
MEIATELA
MOSTRE [EQUAL E'O SEU NOME?]
FAÇA "NOME PRIMEIRO ENTRE
FIM

AP INICIALIZAR
DEFXY 100 100
DEFD 270
FAÇA "VELX 0
FAÇA "VELY 0
TODATELA
MT
FIM

AP IMPULSO
COMANDO
DINA.MOVE
SE DISTANCIA < 10 ENTAO
FINAL PARE
FAÇA RESULTADO :RESULTADO-1
SE RESULTADO = 0 ENTAO
TEMPO.ENCERRADO
PARE
IMPULSO
FIM

AP COMANDO
FAÇA "COM PEGUE
SE :COM = "D ENTAO DI 30
SE :COM = "E ENTAO ES 30
SE :COM = "C ENTAO CHUTE
FIM

AP CHUTE
FAÇA "VELX :VELX + 3 * SEN DIRECAO
FAÇA "VELY :VELY + 3 * COS DIRECAO
FIM

AP DINA.MOVE
DEFXY ( CORY + :VELX )
(CORY + :VELY )
FIM

AP DISTANCIA
SAIDA ROD
(CORY * CORY + CORY * CORY )
FIM

AP FINAL
MOSTRAR"
MEIATELA
(MOSTRE [MUITO BEM] :NOME )
(MOSTRE [SEU RESULTADO FOI]
:RESULTADO
REPORTAR
DE NOVO
FIM

AP REPORTAR
SE:RESULTADO > :MAX ENTAO FAÇA
"MAX :RESULTADO FAÇA "MELHOR :NOME
MOSTRAR"
(MOSTRE [MELHOR RESULTADO E' ]
:MELHOR [COM] : MAX [PONTOS] )
FIM

AP DE.NOVO
MOSTRE [OUTRA PARTIDA?]
FAÇA RES PRIMEIRO ENTRE
SE:RES ="SIM ENTAO NOVO JOGO
PARE
SE :RES ="NAO ENTAO PARE
MOSTRAR [DECIDA-SE, SIM OU NAO?]
DE.NOVO
FIM

AP TEMPO.ENCERRADO
MOSTRAR"
MEIATELA
MOSTRE [SEU TEMPO TERMINOU]
DE.NOVO
FIM

AP NOVO.JOGO
SEMT
DAR.NOME
INICIALIZAR
IMPULSO
FIM
```





# ORDENE OS DÍGITOS

No jogo conhecido como Inverso, o objetivo é colocar uma lista de números em ordem crescente, com um mínimo de tentativas. Parece simples, mas os melhores quebra-cabeças derivam de conceitos básicos.

O programa gera, aleatoriamente, uma lista de números para a classificação, e você só muda a ordem dos números invertendo grupos específicos de números dentro da lista. Por exemplo, ao pedido de uma lista de nove números, o computador gera esta lista:

2 8 4 7 1 5 6 9 3

O jogador especifica "5" em resposta à mensagem "Inverter?". A ordem dos cinco primeiros números se altera e a lista fica assim:

1 7 4 8 2 5 6 9 3

Aparentemente, uma charada como essa não exige tempo ou esforço para ser resolvida: basta recorrer a um algoritmo preestabelecido. Na prática, porém, é difícil encontrar um algoritmo satisfatório. Suponhamos que haja  $n$  números na lista. O algoritmo mais óbvio seria este:

- Achar o maior número da lista e inverter todos os números até sua posição. (O número mais alto está agora na extremidade esquerda da lista.)

- Inverter todos os números, de forma que o maior fique na posição desejada (na extremidade direita da lista).

- Encontrar o segundo maior número, e repetir o procedimento. Para levar esse número até a posição desejada, é preciso executar o movimento "Inverter  $n-1$ ".

- Repetir o procedimento até colocar todos os números na ordem.

Esse algoritmo sempre resolve o problema em  $2n-3$  movimentos. Mas é possível obter uma solução com menos jogadas.

Para demonstrar como uma estratégia de pensamento avançado pode reduzir o número de jogadas, considere o exemplo do quadro.

Pelo procedimento inicial, gastaríamos sete ( $2 \times 5 - 3$ ) jogadas, mas um jogador habilidoso resolve o problema em apenas quatro.

Esse programa é um exemplo simples de toda uma série de jogos de inversão. Você pode desenvolver jogos com inversões a partir de qualquer extremidade da lista, ou substituir os números por quadrados de diversas cores.

## Quantos números?

```

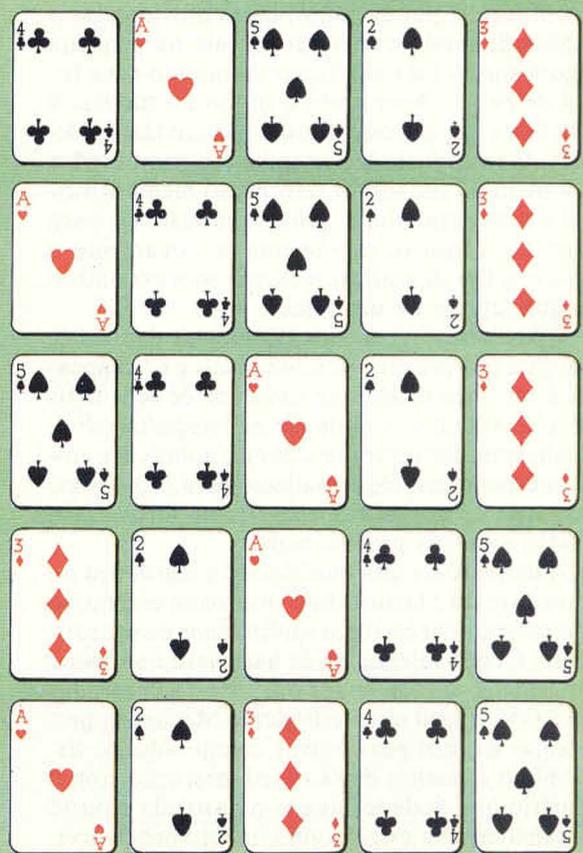
5 REM *** PARA A LINHA SINCLAIR ***
10 REM ***** INVERSO *****
20 DIM A(20)
30 CLS
35 PRINT "INVERSO"
40 INPUT "QUANTOS NUMEROS ?":N
50 IF N < 0 OR N > 20 OR N < > INT (N) THEN GOTO 30
60 REM *** ORDENANDO A LISTA ***
70 FOR I = 1 TO N
72 LET A(I) = I
74 NEXT I
80 REM *** ESCOLHENDO AO ACASO ***
90 FOR I = 1 TO N
100 LET R = INT ( RND * N + 1)
110 LET X = A(R)
112 LET A(R) = A(I)
114 LET A(I) = X
120 NEXT I
130 T = 1
135 REM *** IMPRIMINDO O QUADRO ***
140 CLS
142 PRINT "INVERSO NUMERO "I": " A LISTA E "
144 PRINT
150 FOR I = 1 TO N
151 PRINT A (I) ; " "
152 NEXT I
153 REM *** CHECA SE BANDOU ***
154 LET I = 1
156 IF A(I) = I THEN LET I = I + 1
157 IF I <= N THEN GOTO 156
158 IF I > N THEN GOTO 230
159 REM *** LE A TENTATIVA ***
160 PRINT
162 PRINT
164 INPUT "INVERSO ? "R
170 IF R < > INT (R) OR R < 0 OR R > N THEN
GOTO 140
180 LET T = T + 1
190 FOR I = 1 TO INT (R / 2)
200 LET X = A(I)
202 LET A(I) = A(R - I + 1)
204 LET A(R - I + 1) = X
210 NEXT I
220 GOTO 140
230 REM *** O VENCEDOR ***
240 PRINT
242 PRINT
244 PRINT "VOCE CONSEGUIU EM "T": "TENTATIVAS "
250 PRINT
252 INPUT "QUER JOGAR DE NOVO (S/N)? "A$
260 IF A$ = "S" THEN RUN
270 CLS
275 END
    
```

## Variações

O programa aqui apresentado para a linha Sinclair pode ser adaptado com facilidade para micros de outras linhas. Os compatíveis com Apple exigem uma única modificação importante, que se refere à sintaxe do comando RND. Assim, na linha 100, escreva:  $R = \text{INT}(\text{RND}(3) * N) + 1$ . Substitua também CLS por HOME nas linhas 30, 140 e 270.

## 2-3-5-3

O objetivo de Inverso é classificar uma lista reordenando repetidamente setores dela. O setor a ser invertido deve começar sempre pelo número mais à esquerda, e é descrito pela quantidade de elementos incluídos. Aqui, a sequência mais satisfatória de números é 2-3-5-3, isto é: inverte as duas primeiras cartas da esquerda; depois, as três primeiras cartas da esquerda; e assim por diante.





# MACINTOSH

**Combinando a utilização do mouse com menus iconográficos, o Macintosh, da Apple, facilita a interação do usuário com a máquina e garante a seu fabricante a reputação de inovador no mercado de micros.**

O Macintosh, nome emprestado de uma seleta variedade de maçã, não se parece com nenhum outro micro do mercado. A começar por seu estojo com alça, bastante pequeno para um equipamento com sua potência de processamento, o que permite classificá-lo como micro portátil. Incluindo o teclado, o mouse e o estojo — com compartimentos para todos os componentes —, o conjunto pesa 11,6 kg. O teclado, semelhante aos das máquinas de escrever, é dotado de processador para manipular funções especiais e conjuntos internacionais de caracteres.

O mais revolucionário no Mac, como costuma ser chamado, é a combinação do mouse que equipa a máquina com os menus iconográficos oferecidos tanto nos programas aplicativos como nos utilitários. Em outras palavras, é o sistema de mais fácil interação com o usuário, uma vez que não exige conhecimento de comandos de operação específicos. Para abrir um arquivo de documentos, por exemplo, basta movimentar o mouse de modo que o cursor caia na pequena figura simbólica (ícone) representando uma folha de papel. Apertando o botão do mouse, a tela fica a sua disposição para essa atividade. Depois de introduzir seu arquivo no computador por meio do teclado, o retorno ao menu principal é realizado com o próprio mouse; aí, para gravar o arquivo, movimentam-se novamente o mouse a fim de colocar o cursor sobre o símbolo que representa um disco.

O Macintosh vem com 128 Kbytes de memória. Sua tela possui 512 x 342 pixels e é “mapeada a bit”, de modo que cada um de seus mais de 175.000 pontos pode ser endereçado individualmente. Isso permite algumas aplicações gráficas espantosas, muito valiosas para projetistas, arquitetos, consultores, relações-públicas, fotógrafos e outros profissionais.

A despeito da alta qualidade da segurança do hardware do Macintosh, o que torna este micro tão excepcional é a força suplementar de seu software. Com a integração de hardware e software e os intensivos comandos operacionais baseados na ROM, é fácil transferir para o Macintosh programas escritos para outros computadores. Este micro é, assim, de tão fácil interação com o usuário que pode ser ligado na tomada e posto a trabalhar sem exigir conhecimento prévio acerca da operação do computador.

## Placa analógica

Controla o monitor de vídeo e a alimentação do micro. O Macintosh não precisa de ventilador. O excesso de calor é canalizado, através de placas de metal, para as fendas de ventilação no gabinete.



Alto-falante embutido

Cabeça de leitura/gravação da unidade de disco

Controle de contraste na tela

Unidade de disco, Sony 3 1/2

Fabricada especialmente para a Apple, esta unidade de disco suporta 400 K num só lado. A capacidade pode dobrar se for utilizada dupla face.

RAM de vídeo

São exigidos 22 K para o funcionamento do vídeo, acessados por DMA (Direct Memory Access).

Teclado

O teclado destacável do Macintosh tem seu próprio processador, que pode manipular todos os caracteres de línguas estrangeiras e funções especiais. O mouse torna desnecessárias as teclas que comandam o cursor.

Conector do teclado

Mouse

Controla o movimento do cursor e é utilizado para emitir comandos, representados de forma pictórica em pequenos menus exibidos na borda da tela.



**Conexão analógica digital**

Podem ser conectados dispositivos analógicos, digitais.

**Bus serial**

Também conhecido por "slots virtuais", o bus serial permite adicionar toda uma variedade de dispositivos periféricos.

Saída de áudio

Saída para impressora

Conector para unidade de disco

Conector do mouse

**MACINTOSH**

**DIMENSÕES**

343 x 254 x 254 mm (video/CPU /unidade de disco).

**CPU**

Motorola 68000, 7,83 MHz

**MEMÓRIA**

RAM de 128 K, ROM de 64 K

**TELA**

Monitor monocromático embutido, com 512 x 342 pixels, janelas.

**INTERFACES**

Mouse, impressora, unidade de disco externa, amplificador hi-fi, bus serial e analógico/digital.

**LINGUAGENS DISPONÍVEIS**

BASIC, COBOL, PASCAL

**TECLADO**

Tipo máquina de escrever, 59 teclas, teclado numérico opcional.

**DOCUMENTAÇÃO**

Existe um manual de operação com um audiocassete e uma instrução dirigida em disquete. Os manuais — MacPaint e MacWrite — são incluídos num só pacote, que combina disquete e cassete.

**Adaptador de I/O 6522**

O 6522, também utilizado para controle de I/O no Apple IIe, manipula o teclado, o mouse e circuitos de "tempo real".

Controlador de comunicação serial

Controlador da unidade de disco

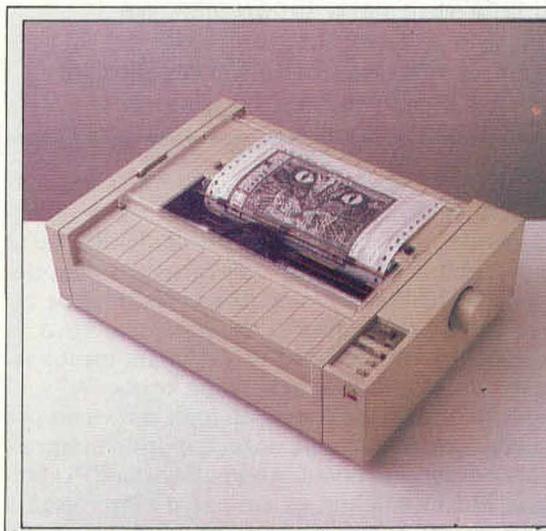
Este chip controla a unidade de disco embutida, assim como uma unidade adicional externa.

**CPU**

É o Motorola 68000. Este chip executa operações internas 32 bits por vez, porém envia e recebe dados a 16 bits; Isso representa maior velocidade de processamento e extensão de memória endereçável.

**RAM disponível para o usuário, de 128 K**

Estes dezesseis chips podem ser substituídos por chips de RAM de 256 K, o que dá ao Macintosh um total de 512 K de memória disponível. No entanto, para as aplicações existentes, bastam 128 K.



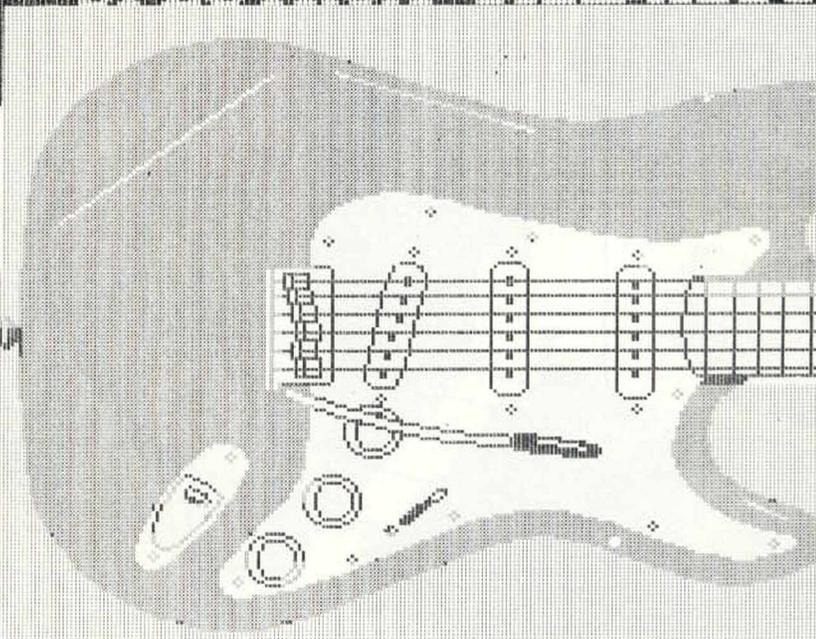
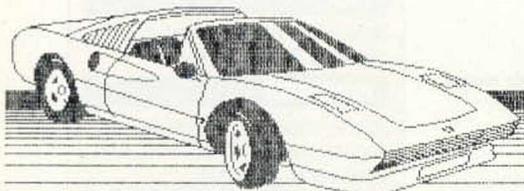
**Melhores imagens**

A ImageWriter é uma impressora gráfica serial capaz de imprimir até 120 caracteres por segundo. Sua velocidade é ainda mais evidente no modo gráfico: as figuras são traçadas por meio de um processo de "mapeamento a bit", seguindo o formato da tela.



# IMPRESSORAS MATRICIAIS

## Ferrari



### Arte impressa

Estes grafismos mostram o efeito que se consegue com algumas impressoras matriciais. Cada agulha da cabeça de impressão é controlada individualmente, permitindo padrões complexos e satisfatórios. As imagens acima foram criadas pelo software PaintBox.

**Quase todos os usuários de micros logo sentem falta de uma impressora. Usada para listar programas, ela permite a depuração de erros. Além disso, mostra-se imprescindível para o processamento de textos.**

Se você pretende adquirir uma impressora matricial, alguns pontos a considerar são a velocidade de impressão e a qualidade do texto produzido. Os modelos mais caros possuem elementos adicionais, como diferentes tipos de caracteres e espaçamento proporcional (isto é, caracteres estreitos como o *i* ocupam menos espaço do que os maiores, como o *m*).

A velocidade de impressão (cps, caracteres por segundo) é importante porque o uso da impressora consome tempo do computador: a CPU precisa dividir-se entre as rotinas de impressão e outras tarefas. Um modelo caro, que opere a 200

cps, pode levar um minuto para imprimir a listagem de um programa extenso; modelos mais baratos, com velocidade de 30 cps, levariam mais de seis minutos para imprimir a mesma listagem. Em geral, durante todo esse tempo, o computador não pode ser usado para outra tarefa. Supera-se esse problema pelo uso de um buffer — uma memória auxiliar. Conectado entre o computador e a impressora, o buffer armazena os dados enquanto a impressora trabalha com eles, liberando o computador para outras operações.

As impressoras mais caras têm grandes buffers incorporados. Algumas armazenam só um caractere, ou só uma linha. A Alice, por exemplo, armazena 8 K de dados.

A velocidade de impressão apresentada pelo fabricante deve ser tomada com certa reserva. Como acontece com os números relativos ao consumo de combustível em automóveis, ela é dada para condições ideais, e com frequência tem pouca semelhança com o desempenho real.



A velocidade é calculada para a impressão de uma única linha, composta de um mesmo caractere. Um texto normal, com seus caracteres variados, espaços, linhas e retornos do carro, diminui a velocidade final de impressão. Assim, quando estivesse imprimindo a listagem de um programa, a impressora com velocidade calculada em 160 cps faria uma média de 100 cps.

A qualidade dos caracteres produzidos no papel varia bastante de uma impressora para outra. Depende principalmente do número de agulhas da matriz de impressão — o mecanismo que forma os caracteres no papel, projetando algumas dessas agulhas.

A maioria dos modelos tem matriz de 9 x 7 agulhas. Já a Cannon PW 1080 utiliza matriz de 16 x 23 para produzir seus caracteres. Em consequência, os pontos não se distinguem e os caracteres têm aparência compacta e bem definida.

Para listagens de programas, a qualidade da impressão não é tão importante, mas para o processamento de texto e gráficos ela se revela fundamental.

A impressora matricial é um microcomputador dedicado à impressão. Utiliza chips de memória ROM e RAM e possui um microprocessador; pode, portanto, ser programada para executar outras tarefas além da impressão de textos, bastando para isso que se enviem códigos de controle especiais do micro para a impressora, ou que se instalem pequenas chaves, conhecidas como DIP (Dual In-line Packages), no interior da impressora. Os caracteres padrão ASCII, armazenados na memória da impressora, podem ser alterados para que se ajustem a alfabetos diferentes. No Brasil, produzem cedilha, til e acentos. Outros efeitos especiais incluem caracteres de largura dupla, texto em negrito (mais escuro e compacto) e diferentes espaçamentos de linha.

A Grafix MX 100 é uma das impressoras matriciais mais versáteis e tem inúmeros efeitos especiais de impressão. Imprime em caracteres itálicos e sublinha o texto automaticamente, além de permitir o espaçamento proporcional.

A linha de impressoras japonesas Epson tornou-se uma espécie de padrão industrial. Isso significa que a maioria dos softwares que necessitam de impressora — pacotes de processamento de texto, programas de faturamento etc. — implicam o uso de uma Epson. Em alguns casos, é necessário adaptar o software para uma impressora não compatível com a Epson.

Uma impressora barata pode ser ótima para a produção ocasional de listagens, mas é improvável que resista ao uso contínuo e diário num escritório.

O fator ruído raramente é levado em conta, mas, se você é do tipo que gosta de queimar as pestanas trabalhando à noite, saiba que algumas impressoras podem tornar-se irritantes no silêncio da madrugada.

As impressoras matriciais vêm equipadas com um mecanismo de tração que trabalha apenas

com formulários contínuos — do tipo que tem furos dos lados. Para imprimir em folhas soltas é necessário um mecanismo de fricção semelhante ao da máquina de escrever comum.

A maioria das matriciais vem com interface paralela ou com interface serial RS232 C. Caso a impressora não possua a interface adequada a seu micro, procure saber junto ao revendedor se é possível acoplar a ela uma interface adicional. Mesmo com a interface adequada, é necessário o cabo correto para ligar a impressora ao computador.



**Cedilha, til, acento**

A Emília, da Elebra (acima) e a Grafix (ao lado) produzem cedilha, til e acentos, sinais imprescindíveis para a escrita em português. Além disso, conseguem muitos efeitos especiais. (Abaixo, teste de impressão da Grafix.)

MICROCOMPUTADOR	MICROCOMPUTADOR	MICROCOMPUTADOR

# ARQUIVOS SEQÜENCIAIS

**Este capítulo é dedicado à discussão dos métodos de utilização dos arquivos seqüenciais em seus programas e mostra como superar algumas das limitações próprias do sistema.**

O arquivo seqüencial consiste num grupo de registros gravados em bloco no disquete ou na fita cassete. Esse tipo de armazenamento apresenta limitações no que diz respeito a acesso (leitura) e atualização de dados.

O acesso a qualquer registro exige a leitura de todos os dados precedentes. E a atualização, em geral, obriga o usuário a munir-se de uma cópia do arquivo até o ponto em que as mudanças são necessárias; feito isso, ele pode acrescentar as alterações e prosseguir copiando o arquivo original.

A organização das informações no arquivo depende do programador e da aplicação que terá. Um arquivo que contenha apenas textos resume-se a uma seqüência de códigos ASCII, seguida por um marcador de fim de arquivo. No entanto, se o arquivo contiver um banco de dados, como, por exemplo, um catálogo de livros, as informações devem ser organizadas de outra forma. A mais comum é dividir o arquivo em *registros* e *campos*. Cada livro tem seu registro no arquivo e dentro de cada registro há vários campos, tais como o título do livro, o autor, a editora e assim por diante. Num arquivo seqüencial, essas divisões devem ser marcadas por meio de caracteres especiais colocados entre os itens de dados.

Para tanto, costuma-se usar o caractere “retorno do carro” (código ASCII número 13), que atua como marcador entre os campos e os registros. Uma vez que o arquivo terá o mesmo número de campos para cada registro, é fácil para o programa localizar onde termina um registro e começa outro. Depois de criado, o arquivo seqüencial deve dar condições de acesso e atualização. As operações básicas na manipulação de arquivos são: acessar (ler), acrescentar, eliminar e modificar (editar) registros. Os diagramas mostram as várias maneiras de realizar tais operações com os arquivos seqüenciais: à medida que faz a leitura (obrigatoriamente na ordem), o programa cria nova cópia do arquivo. Nessa cópia gravam-se as alterações, introduzidas quando chega sua vez. Por fim, a cópia substitui o arquivo antigo, que é descartado, ou mantido como cópia de reserva (back-up).

Essas técnicas simples são a base de todas as rotinas de arquivamento seqüencial. Partem do princípio de que o sistema operacional pode ter dois arquivos diferentes abertos ao mesmo tempo — para ler de um e gravar no outro simultaneamente. Isso, no entanto, só é possível em alguns sistemas a disco e nos micros baseados em fitas cassete com dois gravadores conectados. Alguns modelos de micro dispõem de interfaces para dois gravadores cassete. As máquinas com saída para um único cassete só comportam arquivos pequenos — o suficiente para que sejam lidos na íntegra na memória e aí processados.

Esses métodos de organização dos arquivos são vantajosos também porque oferecem duas cópias do arquivo: uma do antigo, como era antes da atualização, e outra do novo, com as alterações. A guarda de ambos é precaução necessária porque, se algo acontecer ao arquivo, pode-se recuperá-lo pela simples atualização (de uma geração) da cópia. Na maioria das empresas costuma-se guardar três gerações de cada arquivo, que no jargão técnico recebem os nomes de filho, pai e avô.

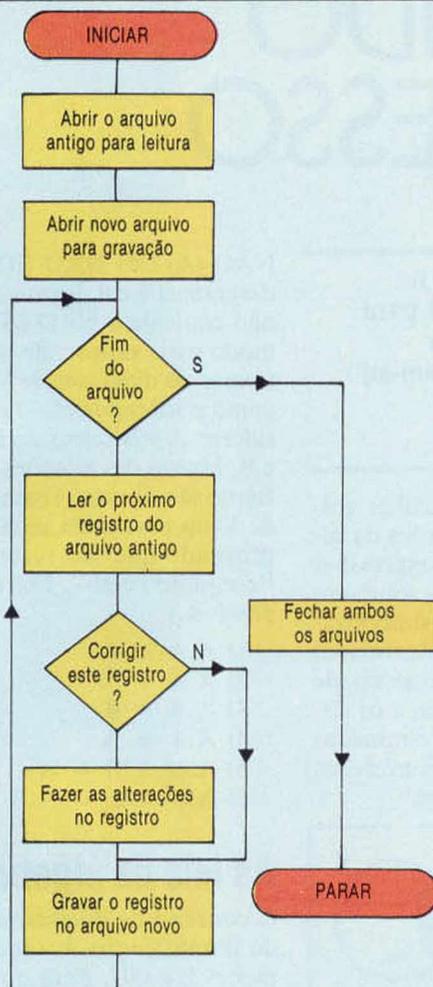
Essas técnicas são ideais para arquivos grandes, que não cabem na memória do computador — em geral, acessa-se, a cada vez, apenas uma parte do arquivo (alguns registros). No caso de arquivos pequenos, obtém-se melhor desempenho lendo todo o arquivo e colocando-o numa área da memória para aí ser processado. Todas as operações com arquivo se desenvolvem em alta velocidade na memória; em seguida, o novo arquivo é gravado na íntegra no disquete ou na fita cassete.

Com esse método corre-se um grande risco — as mudanças nos arquivos se tornam permanentes apenas quando as informações são gravadas de volta na fita ou no disquete; assim, podem-se perder os dados se o programa “travar” ou se o computador sofrer algum acidente ou for desligado durante a operação. Se você estiver usando programas que funcionem dessa maneira, certifique-se de que vêm sendo feitas cópias do arquivo para armazená-lo e verifique se foi produzida a cópia atualizada antes de terminar o programa.

Um pouco de experiência em lidar com arquivos seqüenciais vai mostrar-lhe que as técnicas envolvidas, apesar de trabalhosas, são úteis. Em muitos sistemas pequenos, o arquivo seqüencial é a única estrutura de arquivos disponível. Outras técnicas que complementam os arquivos seqüenciais, com acesso e atualização simples e rápidos, encontram-se no estudo dos arquivos de acesso aleatório.

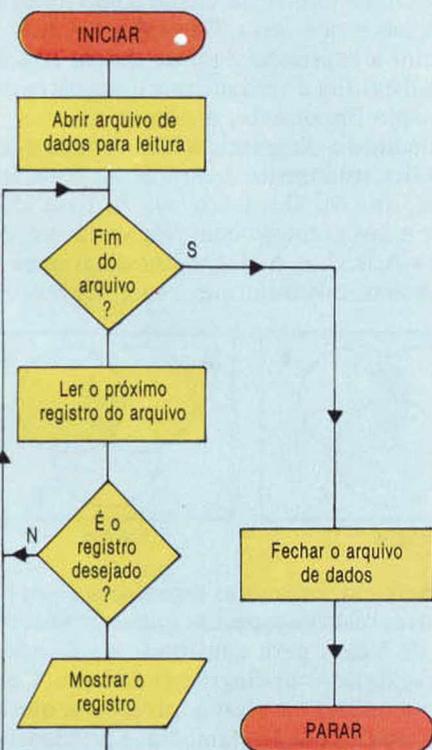
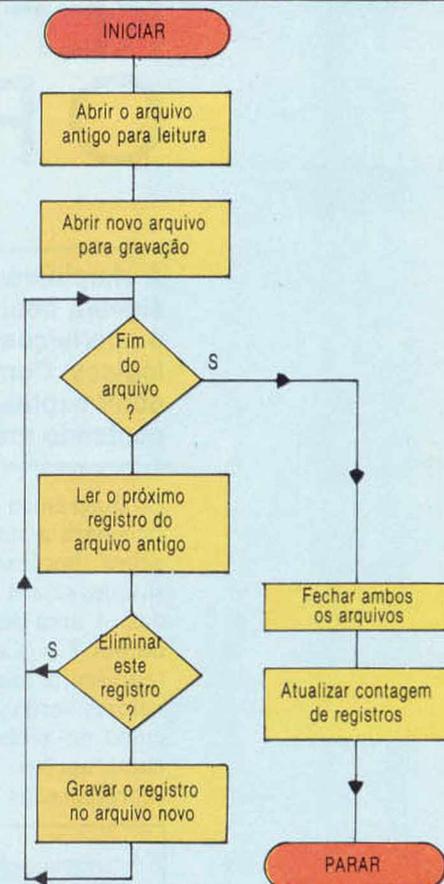
## Correção de registros

Para corrigir um registro (alterar as informações nele contidas), começa-se copiando todos os registros precedentes em novo arquivo. Quando se chegar ao registro a ser alterado, o programa faz a atualização. Depois de atualizado, o registro é gravado no arquivo novo e, em seguida, gravam-se todos os registros subsequentes do arquivo antigo. Qualquer número de registros pode ser atualizado numa só operação de busca por meio do arquivo.



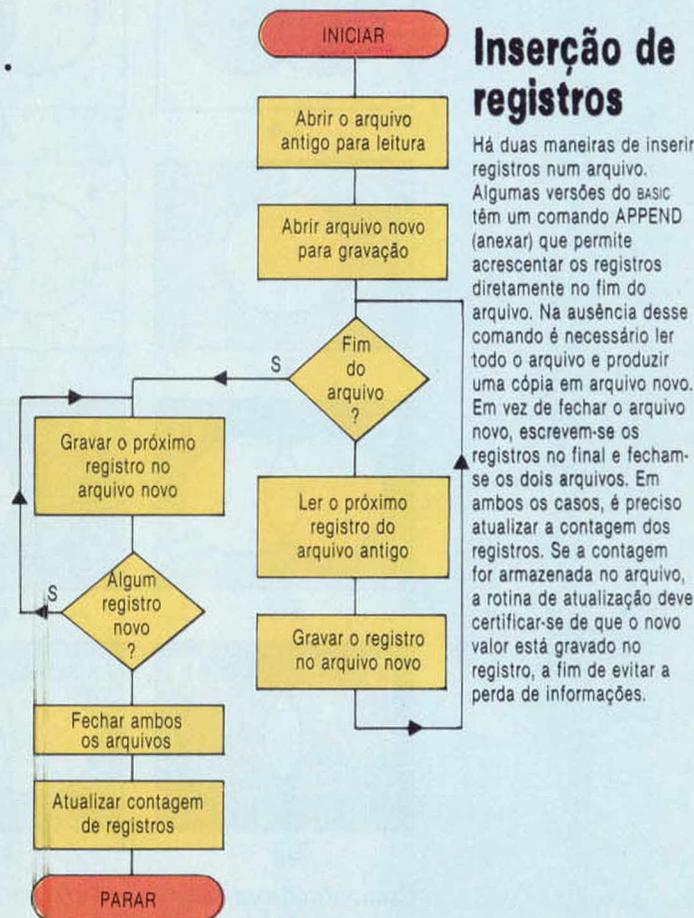
## Eliminação de registros

Para eliminar um registro, o programa faz a leitura e a cópia do arquivo, até chegar ao registro desejado. Este é lido, mas não copiado no arquivo novo. Em seguida, os registros remanescentes são lidos e copiados no novo arquivo. Vários registros podem ser eliminados numa única pesquisa. Da mesma forma que na inserção de registros, aqui também é essencial que se atualize sua contagem imediatamente.



## Acesso aos registros

Os arquivos sequenciais são os menos indicados para as aplicações que exigem a leitura de alguns registros apenas. Isso porque todo o arquivo tem de ser lido de novo, o que consome muito tempo. Se você estiver procurando determinado registro numa lista de nomes, o programa percorrerá um loop, examinando cada registro e prosseguindo no loop sempre que não preencher a condição. Se forem vários registros, eles podem ser lidos um após o outro, mas sempre na ordem em que aparecem no arquivo. Daí a importância de organizá-los em alguma ordem (por exemplo, alfabética) antes de armazená-los.



## Inserção de registros

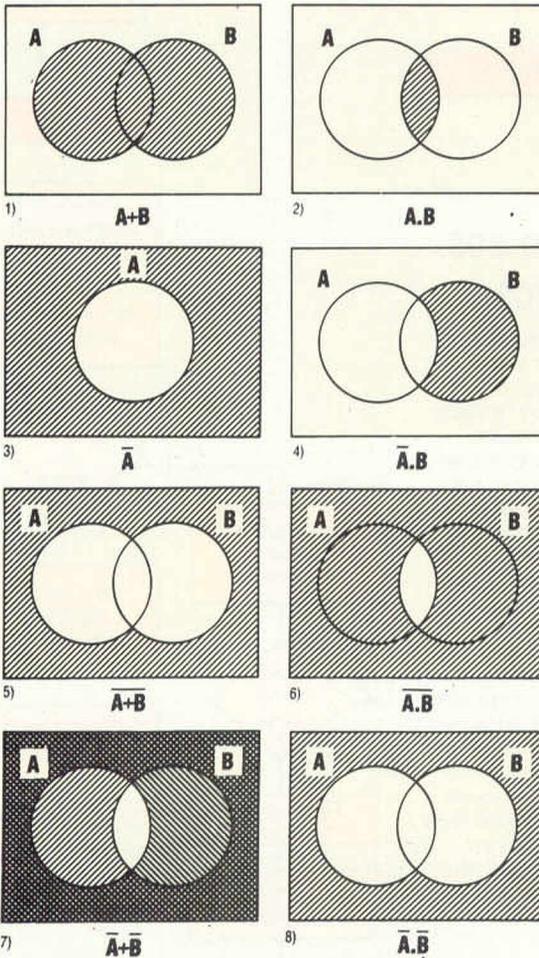
Há duas maneiras de inserir registros num arquivo. Algumas versões do BASIC têm um comando APPEND (anexar) que permite acrescentar os registros diretamente no fim do arquivo. Na ausência desse comando é necessário ler todo o arquivo e produzir uma cópia em arquivo novo. Em vez de fechar o arquivo novo, escrevem-se os registros no final e fecham-se os dois arquivos. Em ambos os casos, é preciso atualizar a contagem dos registros. Se a contagem for armazenada no arquivo, a rotina de atualização deve certificar-se de que o novo valor está gravado no registro, a fim de evitar a perda de informações.



# REFINANDO O PROCESSO

**A simplificação das expressões na álgebra booleana é fundamental para o aperfeiçoamento dos circuitos lógicos. Com esse recurso, podem-se obter expressões equivalentes contendo menos operadores.**

Os diagramas de Venn são um útil auxiliar gráfico para a simplificação das expressões da álgebra booleana, permitindo que expressões simples sejam desenhadas como áreas sombreadas. A área dentro de um retângulo (simbolizada por 1 = Conjunto Universo ou Identidade) representa todas as combinações possíveis de valores-verdade dos dados de entrada, e os círculos no retângulo representam determinadas combinações. Aqui mostramos algumas delas, representadas por diagramas de Venn:



Comparando os diagramas 5 e 7, nota-se de imediato que  $\overline{A+B}$  não é o mesmo que

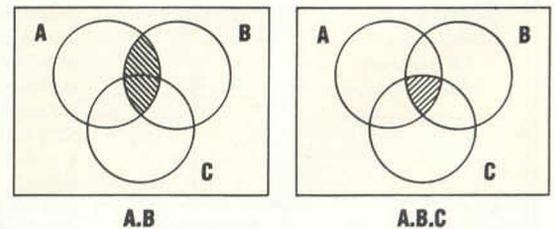
$\bar{A} \text{ OU } \bar{B}$ . Do mesmo modo, os diagramas 6 e 8 demonstram que  $\bar{A} \text{ E } B$  não equivale a  $\bar{A} \text{ E } \bar{B}$ . Talvez o modo mais simples de imaginar o E e o OU em termos do diagrama de Venn seja pensar em A.B como a sobreposição da área A à área B, e considerar A+B como as áreas combinadas de A e B. Muitas das relações existentes na álgebra de Boole são óbvias. Tente construir um diagrama de Venn para cada uma das relações seguintes, provando que são verdadeiras. (0 representa o "conjunto vazio", isto é, uma combinação impossível.)

- 1)  $A.A = A$
- 2)  $A.\bar{A} = 0$
- 3)  $A.0 = 0$
- 4)  $A.1 = A$
- 5)  $A.(A+B) = A$
- 6)  $A.(\bar{A}+B) = A.B$

## As leis da álgebra booleana

O conceito de *dualidade* é um eficiente recurso de simplificação, baseado na simetria das operações E e OU. Para obter a expressão dual de qualquer relação verdadeira, transforme todos os Es em OUs e vice-versa, assim como todos os zeros em uns e vice-versa. Exemplo:  $A + A.B = A$  constitui a expressão dual da quinta relação de nossa lista. Ela é verdadeira e demonstra outro princípio importante, a *absorção*.

Examinando o diagrama, é fácil ver que o termo A.B fica totalmente dentro de A; constata-se, assim, que foi absorvido por A. Essa idéia aplica-se a um exemplo com três variáveis, como  $A.B + A.B.C = A.B$ . Os dois diagramas de Venn, abaixo, mostram que isso é verdadeiro.



Como exercício, escreva as expressões duais das cinco outras relações especiais e desenhe seus diagramas de Venn, para confirmar a validade.

Observe de novo os diagramas de Venn. Comparando o quinto e o oitavo, percebe-se que esta importante relação também é verdadeira:  $\overline{A+B} = \bar{A}.\bar{B}$ . Já os diagramas 6 e 7 mostram que  $\bar{A}.B = \bar{A} + \bar{B}$ . Essas duas relações consti-



item as leis de De Morgan e incidem também sobre expressões mais complexas, como as de três variáveis ( $A+B+C = \overline{\overline{A}.\overline{B}.\overline{C}}$  e  $\overline{A.B.C} = \overline{A} + \overline{B} + \overline{C}$ ). As leis de De Morgan podem ser aplicadas em estágios:

$$\begin{aligned} & \overline{(A+B)}.C = \\ & = \overline{A.B.C} \quad (\text{usando-se a lei de De Morgan nos parênteses}); \text{ ou } = \\ & = \overline{A+B+C} \quad (\text{recombinando com o uso da lei de De Morgan}). \end{aligned}$$

Três outras regras aplicam-se, como na álgebra normal, à álgebra booleana. A lei associativa permite o deslocamento de parênteses:

$$\begin{aligned} (A.B).C &= A.(B.C) = A.B.C \\ (A+B)+C &= A+(B+C) = A+B+C \end{aligned}$$

A ordem na qual as letras são escritas pode ser mudada, de acordo com a lei comutativa:

$$\begin{aligned} A.B &= B.A \\ A+B &= B+A \end{aligned}$$

A lei distributiva permite a distribuição do multiplicador pelos parênteses:

$$A.(B+C) = A.B+A.C$$

## Exemplos de simplificação

- 1)  $\overline{(A+B)} + \overline{A.B}.B =$   
 $= (\overline{A.B} + \overline{A.B}).B$  (de De Morgan);  
 $= \overline{A.B}.B + \overline{A.B}.B$  (lei distributiva);  
 $= 0 + \overline{A.B}$  ( $\overline{B}.B = 0$ ,  $B.B = B$ );  
 $= \overline{A.B}$ ;
- 2)  $\overline{A.B} + \overline{A}.B + A.B =$   
 $= \overline{A}.(\overline{B} + B) + A.B$  (lei distributiva);  
 $= \overline{A} + A.B$  ( $\overline{B} + B = 1$ );  
 $= \overline{A} + B$  (dual da relação 6);
- 3)  $\overline{A+B} + \overline{A+B} + \overline{A.B} =$   
 $= \overline{A.B} + \overline{A.B} + \overline{A.B}$  (de De Morgan);  
 $= A.\overline{B} + A.B + \overline{A.B}$  ( $\overline{A} = A$ );  
 $= A.(\overline{B} + B) + \overline{A.B}$  (lei distributiva);  
 $= A + \overline{A.B}$  ( $\overline{B} + B = 1$ );  
 $= A + B$  (dual da relação 6).

## Porta OU exclusivo simplificada

Na matéria anterior desta seção examinamos um circuito não simplificado para porta OU exclusivo. Voltamos agora a abordar a questão, desta vez com o recurso de simplificar a expressão booleana e, com isso, também o circuito. A tabela de validação para a porta OU exclusivo é:

ENTRADA		SAÍDA
A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

A partir da tabela de validação, determinamos que  $C = \overline{A}.B + A.\overline{B}$ . É pouca a simplificação que

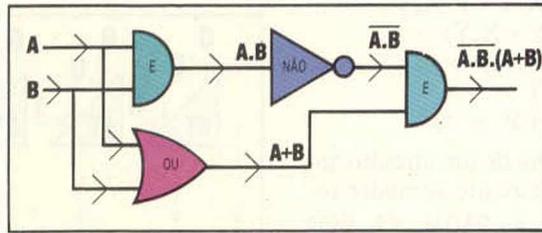
se pode fazer e seria necessário um circuito de cinco portas para executar a expressão. Contudo, há um modo alternativo de abordar o problema. Segundo a tabela de validação, C é 1 se A e B não forem ambos 1 ou ambos 0. Em termos booleanos, pode-se escrever uma expressão alternativa para C:

$$C = \overline{A.B} + \overline{A.B}$$

Usando as leis de De Morgan sucessivamente, simplifica-se o circuito e obtém-se:

$$C = (\overline{A.B}).(\overline{A.B}) = \overline{A.B}.(A+B)$$

Essa expressão exige apenas quatro portas:



## Circuito somador total

Já examinamos o processo de adição binária e projetamos um circuito simples (o somador parcial) para somar dois bits e produzir duas saídas para a soma e os dígitos de transporte da resposta. Se denominarmos X a primeira entrada e Y a segunda, podemos verificar pela tabela de validação para um somador parcial que a saída (S) da soma — ou seja, a resposta — pode ser representada pela expressão:  $S = \overline{X}.Y + X.\overline{Y}$ . Usando a lei de De Morgan, essa expressão pode ser simplificada para  $S = \overline{X.Y}.(X+Y)$ . A saída (C) de transporte é  $C = X.Y$ .

Na aritmética binária, três dígitos precisam ser acrescentados em qualquer das colunas do resultado da adição. Além dos dois dígitos, há um transporte das colunas anteriores a ser incluído. Para reproduzir o processo de adição binária, devemos desenhar um circuito com três entradas e duas saídas. Se chamarmos P o transporte da coluna anterior, a tabela de validação para o somador total (ST) será:

ENTRADAS			SAÍDAS	
P	X	Y	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



Tomando os exemplos em que  $S = 1$ , pode-se formar uma expressão para  $S$  a partir da tabela de validação:

$$S = \bar{P} \cdot \bar{X} \cdot Y + \bar{P} \cdot X \cdot \bar{Y} + P \cdot \bar{X} \cdot \bar{Y} + P \cdot X \cdot Y$$

Utilizando as regras vistas acima, podemos simplificar esta expressão:

$$S = \bar{P} \cdot (\bar{X} \cdot Y + X \cdot \bar{Y}) + P \cdot (\bar{X} \cdot \bar{Y} + X \cdot Y)$$

(lei distributiva);

$$S = \bar{P} \cdot (\bar{X} \cdot Y + X \cdot \bar{Y}) + P \cdot (\bar{X} \cdot Y \cdot X \cdot \bar{Y})$$

(de De Morgan)

De modo análogo, podemos formar uma expressão para  $C$  a partir da tabela de validação:

$$C = \bar{P} \cdot X \cdot Y + P \cdot \bar{X} \cdot Y + P \cdot X \cdot \bar{Y} + P \cdot X \cdot Y$$

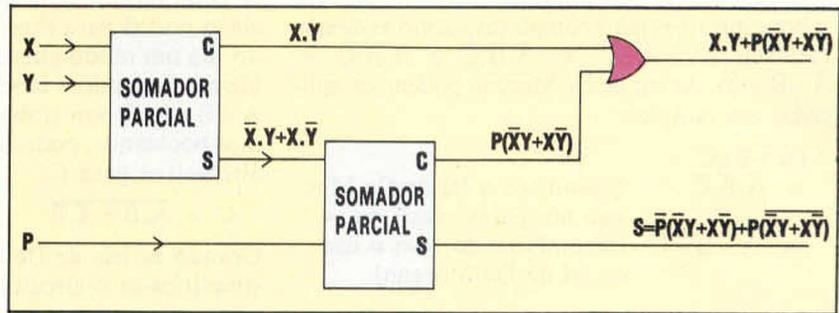
$$C = X \cdot Y \cdot (\bar{P} + P) + P \cdot (\bar{X} \cdot Y + X \cdot \bar{Y})$$

(lei distributiva)

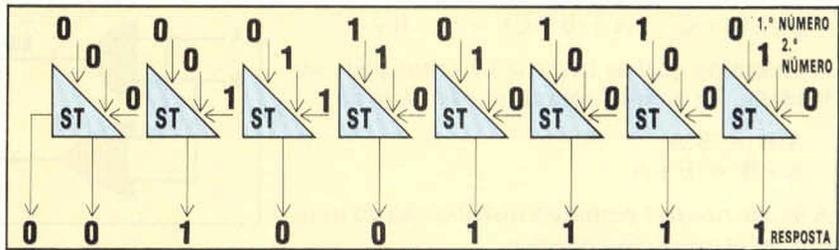
$$C = X \cdot Y + P \cdot (\bar{X} \cdot Y + X \cdot \bar{Y})$$

( $\bar{P} + P = 1$ )

$\bar{X} \cdot Y + X \cdot \bar{Y}$  é a saída da soma de um circuito somador parcial. Assim, um circuito somador total pode desenvolver-se a partir de dois somadores parciais.



Uma série de oito somadores totais se combina numa unidade lógica aritmética para executar uma adição binária de dois números de 8 bits:



### Exercício 3

1) Simplifique estas expressões:

- a)  $A \cdot (\bar{A} + \bar{B})$
- b)  $X + Y \cdot (X + Y) + X \cdot (\bar{X} + Y)$
- c)  $\bar{P} \cdot Q + \bar{P} \cdot Q + \bar{P} \cdot Q$
- d)  $X + Y \cdot Z + Z \cdot Y$

2) O alarme de um carro possui interruptores para ligar/desligar nas duas portas da frente e um interruptor geral. O alarme soará se qualquer uma ou ambas as portas forem abertas quando o interruptor geral estiver ligado. Faça uma tabela de validação mostrando as três entradas (porta A, porta B e o interruptor geral), e a saída de alarme. Use a tabela de validação para escrever uma expressão booleana para o alarme soando. Desenhe um circuito lógico para o sistema de alarme.

3) A lâmpada de uma saleta é operada por meio de um interruptor na porta, um interruptor na base da escada e outro no topo da escada. Desenhe um circuito lógico adequado.

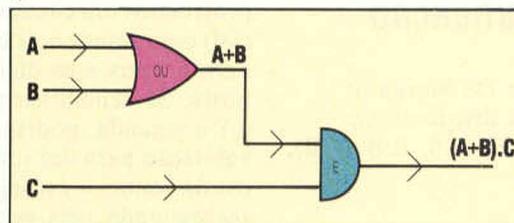
4) Você está preso numa ilha deserta com duas outras pessoas. Uma delas sempre fala a verdade, a outra sempre mente. Para sua própria sobrevivência, é importante descobrir qual delas fala a verdade. Há uma série de perguntas que você pode fazer a cada uma delas para determinar sua identidade. Desenhe as tabelas de validação para investigar as respostas possíveis. Eis um exemplo para você começar:

“Você sempre diz a verdade?”

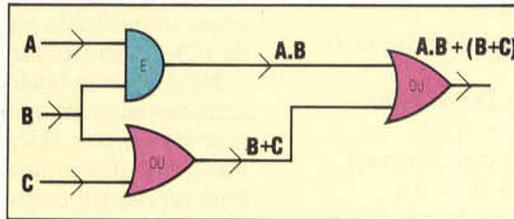
		RESPOSTAS POSSÍVEIS	
		SIM	NÃO
POSSÍVEL IDENTIDADE DE QUEM RESPONDE	MENTIROSO	1	0
	SINCERO	1	0

### Respostas do exercício 2

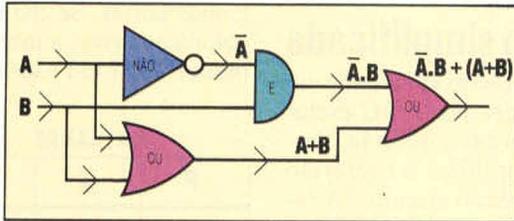
1)



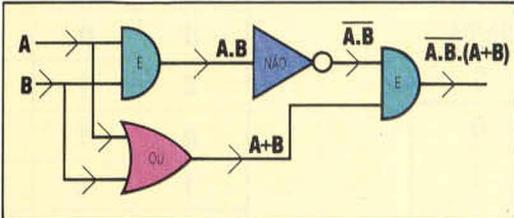
a)



b)



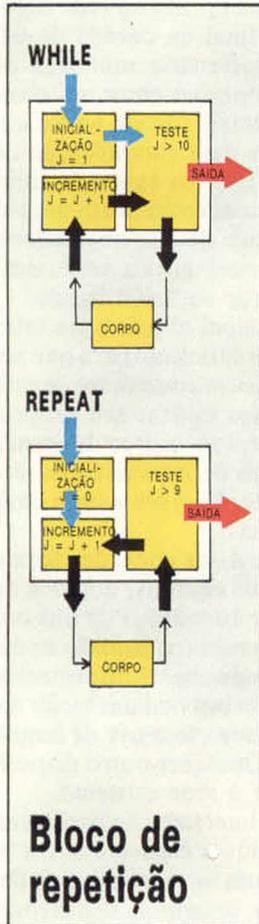
c)



d)

- 2a)  $C = A \cdot B$
- b)  $S = \bar{A} \cdot \bar{B} (A + B)$
- 3a)  $X = (A + B) \cdot (B \cdot C + C)$
- b)  $X = \bar{A} \cdot \bar{B} + \bar{B}$

# LINHAS DE LOOP



**Os fluxogramas convencionais nem sempre representam com clareza as estruturas de loop. Esta matéria estuda o funcionamento dos loops e apresenta novo símbolo para o fluxograma — os blocos de repetição.**

A repetição, ou loop, é uma das estruturas essenciais em qualquer linguagem de programação. Usa-se o loop quando uma decisão desvia o fluxo do programa para um trajeto que retorna à decisão inicial. O loop é, assim, uma estrutura responsável pela execução repetida de um corpo de instruções.

Abrangendo cerca de 50% de toda a atividade de processamento de dados, os loops merecem um estudo detalhado, sobretudo no que se refere a seu efeito sobre a estrutura geral do programa-algoritmo e às várias formas de sua construção e classificação.

Os loops costumam ser divididos em dois tipos, dependendo de sua semelhança com as duas estruturas de loop existentes nas linguagens de mais alto nível: REPEAT — UNTIL (repita-até) e WHILE — ENDWHILE (enquanto-fim).

Ambos os tipos são usados em PASCAL e o loop REPEAT existe em algumas versões do BASIC. No BASIC comum, o loop é feito da forma: IF-THEN-GOTO, que implica montar uma rotina de teste com quatro instruções no mínimo. Nas funções REPEAT e WHILE, esta rotina já está incorporada. Os dois tipos diferem quanto ao posicionamento dos testes de saída: no REPEAT, o teste vem no final do corpo do loop, enquanto no WHILE se encontra no início. Assim, o corpo de um loop REPEAT será sempre executado pelo menos uma vez; não ocorre o mesmo com o loop WHILE. O fluxograma de blocos permite visualizar essa diferença.

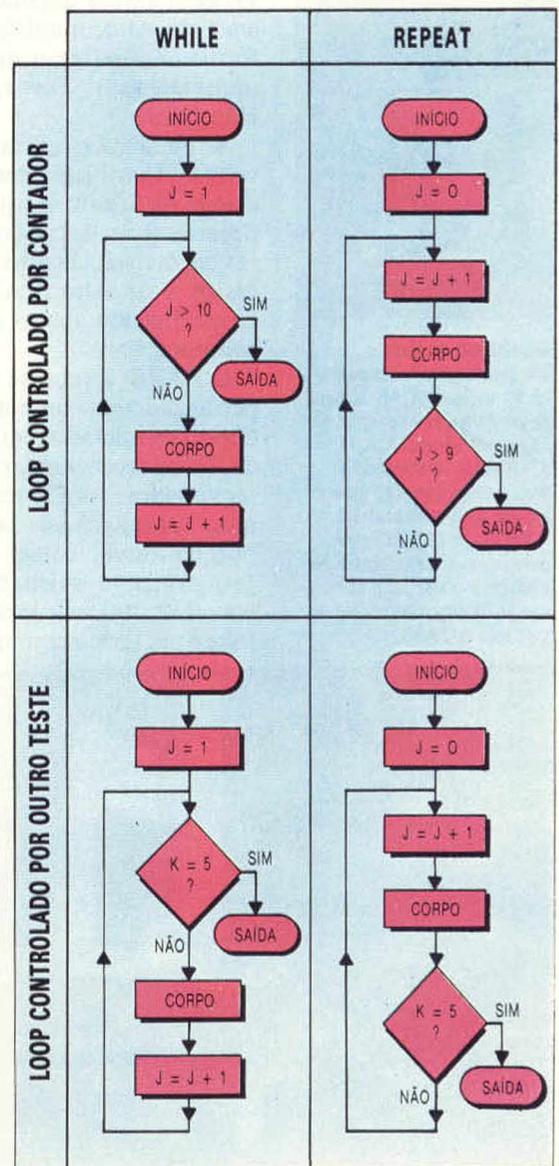
Outra forma de classificar os loops se refere ao controle de saída: pode ser realizado pelo próprio contador do loop ou por outra condição qualquer. Essa diferença evidencia-se menos num fluxograma de tipo linear convencional. Em fluxogramas desse tipo, a descrição dos loops é pouco útil: assemelha-se a uma simples ramificação e com freqüência exige o exame detalhado do algoritmo para distinguir um do outro.

Existe uma notação mais clara, chamada “bloco de repetição”, que marca o início de um loop e elimina a confusão entre loop e ramificação. Essa notação consiste em três blocos interligados: o primeiro mostra o início da contagem; o segundo, o incremento da contagem; e o terceiro contém o teste de saída do loop.

Tanto os loops REPEAT como os WHILE podem ser escritos dessa forma, mas diferem quanto ao fluxo do programa pelos blocos; o bloco de teste indica se o loop é controlado pelo contador ou não. O diagrama evidencia isso.

Num loop do tipo REPEAT, o fluxo das instruções é: início-corpo-teste-corpo-teste. Um loop do tipo WHILE segue o esquema início-teste-corpo-teste-corpo. As linhas que representam o fluxo saem e retornam ao diagrama de repetição, com o corpo do loop “suspensa” por tais linhas de controle.

## Classificação dos loops



# INVESTINDO NO FUTURO

**Quando a Xerox — a maior vendedora mundial de copiadoras — resolveu lançar-se no campo da automação de escritórios, sua reputação era tal que as pessoas já usavam “xerocar” como sinônimo de “fotocopiar”.**

No início dos anos 70, a Xerox planejou um intenso programa de pesquisas para que um sonho se tornasse realidade: dispor de informações em quantidade tão abundante no escritório que seu fluxo semelhasse uma corrente elétrica. Ela reuniu nos Estados Unidos uma equipe de pesquisas com ampla liberdade de ação, trabalhando em Palo Alto, município de Santa Clara, Califórnia, no extremo oposto das instalações da multinacional Xerox, em Rochester, New Hampshire.

A mudança para Santa Clara foi muito proveitosa. Localizado perto do campus da Universidade de Stanford, que possui um importante departamento de computação dedicado a pesquisas de inteligência artificial, o Centro de Pesquisas de Palo Alto (PARC, Palo Alto Research Center) atraiu alguns dos maiores cérebros em computação.

O PARC tornou-se um centro de cultura especializado em computação, criando um jargão que só os iniciados entendem. Muitos produtos da Xerox receberam apelidos enquanto eram desenvolvidos. As 820 séries de micros, por exemplo, foram chamadas de WORM (verme), porque iriam “comer a Apple” (maçã).

O principal objetivo da nova equipe era desenvolver uma rede local (LAN, Local Area Net). Esse é um termo comum hoje em dia, mas quan-

do a Xerox estabeleceu sua primeira rede experimental no Havaí, no final da década de 60, representava uma revolucionária mudança de conceito. Para que as conexões entre um computador central e um terminal fossem feitas, era necessário instalar cabos para comunicação de alta velocidade. Além disso, os cabos de comprimento superior a 20 m apresentavam muitos problemas. A rede pública de telefone poderia ser utilizada, mas isso restringiria a comunicação de dados à velocidade de 9.600 bauds.

Em Palo Alto, o principal objetivo era criar uma rede com velocidade suficiente para que pequenos computadores fossem conectados de modo que o usuário pudesse utilizar seu próprio aparelho, além de ter acesso a outros de grande porte, a seus discos e mais periféricos caros, como plotters e impressoras. Essa era a idéia básica da rede local Ethernet.

No sistema da Ethernet, as conexões foram feitas com cabos co-axiais comuns, dotados da capacidade de transmitir 10 milhões de bits por segundo. Esse sistema permite transmissão de dados digitalizados, incluindo sons e informações gráficas. Além disso, os cabos podiam medir até 500 m sem que fosse necessário o uso de amplificadores de repetição. Qualquer outro dispositivo poderia se conectar à rede existente.

A rede física (cabos e interfaces de hardware) é passiva: dados de qualquer espécie são transportados através dela e um interpretador de dados, em cada dispositivo, determina se a mensagem se destina àquele dispositivo. Se for assim, o interpretador decodifica a mensagem, apresentando-a de tal forma que aquele dispositivo — seja ele um microcomputador, uma impressora, um plotter ou qualquer outro — possa utilizá-la.

Por volta da metade dos anos 70, a Ethernet já estava funcionando. A Xerox percebeu que, se obtivesse ajuda de outras empresas, poderia fazer desse sistema um modelo universal de comunicação entre computadores. Ela levou seus projetos para a IBM, que não quis participar. A Digital Equipment Corporation, contudo, aderiu.

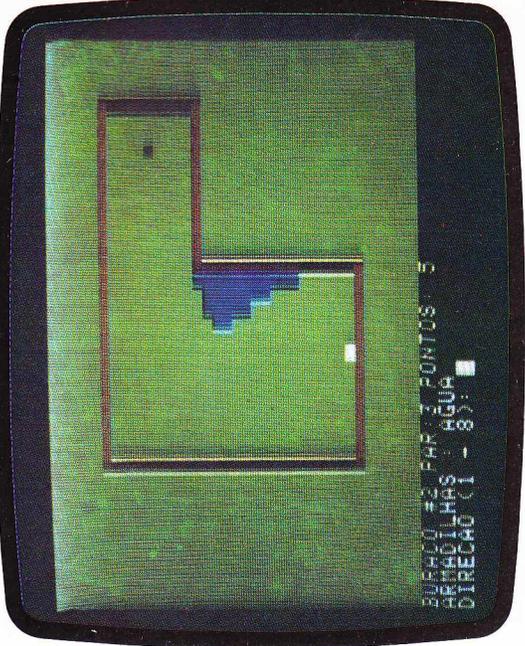
Em 1975, a Xerox também conseguiu a cooperação da Intel, fabricante de chips, que passou a produzir um chip dedicado à interpretação de dados. A rede Ethernet foi testada num escritório experimental e num complexo empresarial na Suécia. Depois dos testes, considerados satisfatórios, outros fabricantes adotaram o sistema. Tornou-se um modelo internacional admitido por empresas como a Hewlett-Packard e a ICL, na Grã-Bretanha, a Siemens, na Alemanha, e a Olivetti, na Itália.



## Nascimento do Lisa

Um dos maiores feitos da PARC foi a criação do STAR, sistema de programação que utiliza a linguagem SMALLTALK. O STAR opera combinando programas e dados no mesmo arquivo para processamento. A tecnologia Lisa da Apple deve muito a esse sistema. Na verdade, a maior parte da equipe de desenvolvimento do Lisa veio do PARC.





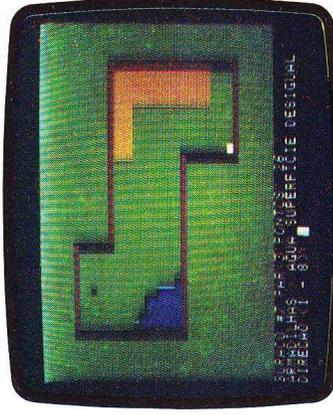
### As dificuldades

Existem quatro tipos de obstáculo: blocos, a serem contornados; areia, que interrompe a jogada; água, que também exige mais tacadas; e superfícies irregulares, que desviam a bola.

```

1050 PRINT :PRINT "2 1 8":PRINT "
3 * 7":PRINT "
4 5 6"
1065 VTAB 23: INPUT "TECLE RETURN PARA
CONTINUAR: ";ANS#
1070 HOME: VTAB 1:HTAB 9:PRINT "***
GOLFE MINIATURA ***"
1075 VTAB 5
1080 PRINT "AGORA VOCE PRECISA
DETERMINAR QUAL A VELOCIDADE DA
BOLA A VELOCIDADE DEVE SER DADA
POR UM NUMERO ENTRE 0.00 E 5.00.
POR EXEMELO VOCE PODE ATIRAR UMA
BOLA COM UMA VELOCIDADE RELATIVA
DE 1.3"
1085 PRINT "TENTE ALGUMAS VEZES ANTES
PARA PERCEBER A DIFICULDADE DE
PASSAR POR UM BURACO."
1090 PRINT "EXISTEM 4 TIPOS DE
OBSTACULOS. FIGUE ATENTO PARA
PERCEBERE-LOS E GUAL A RELACAO DELES
COM A BOLA."
1100 PRINT "EXISTEM 4 TIPOS DE
OBSTACULOS. FIGUE ATENTO PARA
PERCEBERE-LOS E GUAL A RELACAO DELES
COM A BOLA."
1110 VTAB 23: INPUT "TECLE RETURN PARA
CONTINUAR: ";ANS#
1120 HOME: VTAB 1:HTAB 9:PRINT "***GOLFE
MINIATURA ***"
1125 VTAB 5
1130 PRINT "BLOCOS : SAO COMO FAREDES,
VOCE PRECISA CONTORNA-LOS"
1135 PRINT "TELAS AMARELAS REPRESENTAM
AREIA. A SUA BOLA NAO PODE PENETRAR
NOS BANCOS DE AREIA. A PENALIDADE
PARA ISTO E' UM STROKE"
1145 PRINT
1150 PRINT"AGUA: ASSIM COMO A AREIA,ESTA
TELA AZUL VAI ATRASAR E INTERROMPER
A JOGADA, ACORRETANDO MAIS UM
STROKE"
1155 PRINT "SUPERFICIES IRREGULARES : OS
BLOCOS LARANJA DESVIAM A DIRECAO
DA BOLA. ALEATORIAMENTE, ESTA FALTA
NAO SOFRE PENALIDADE."
1170 VTAB 23: INPUT "TECLE RETURN PARA
CONTINUAR: ";ANS#

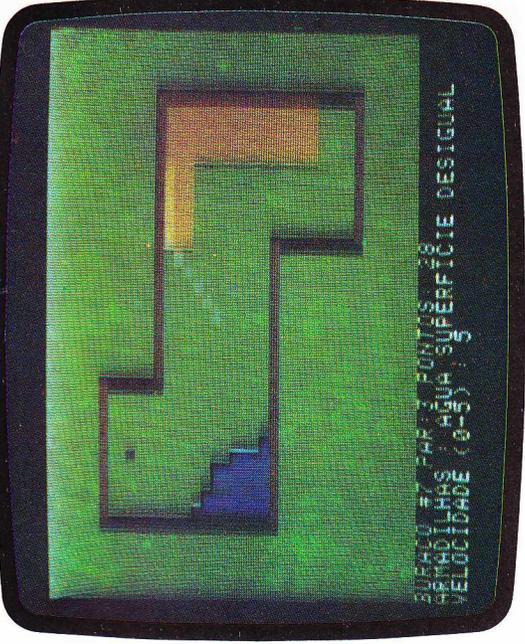
```



```

1180 HOME:VTAB 1:HTAB 9:PRINT "***
GOLFE MINIATURA***"
1185 VTAB 5
1190 RETURN
2000 REM *** INICIO ***
2005 BEL=CHR$(7)
2010 DIM HA(9,5):FOR I = 1 TO 9:FOR J =
1 TO 5:READ HA(I,J):NEXT J,I
2011 DATA 0,0,0,0,2
2012 DATA 1,0,0,0,3
2013 DATA 1,0,0,0,3
2014 DATA 1,1,0,0,3
2015 DATA 0,0,0,1,3
2016 DATA 0,0,1,1,3
2017 DATA 1,0,1,0,3
2018 DATA 0,1,1,0,4
2019 DATA 0,0,1,1,3
2020 DIM DI(8,2):FOR I = 1 TO 8:READ
DI(I,1),DI(I,2):NEXT I
2045 BR:HOME:RETURN
2050 COLOR=12
2052 FOR I = 0 TO 19
2054 HLINE 0,39 AT I:HLINE 0,39 AT 39-I
2056 VLINE 0,39 AT 1:VLINE 0,39 AT 39-I

```



### As penalidades

Depois de algumas partidas, você estará mais familiarizado com o modo de evitar as penalidades — ou seja, o aumento do número de tacadas (*strokes*). Mesmo assim, o jogo continuará emocionante.

```

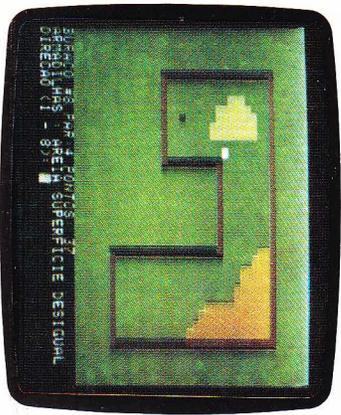
2058 NEXT :RETURN
2100 REM *** BURACO 1 ***
2105 GOSUB 2050
2106 COLOR = 1
2107 HLINE 10,24 AT 6: HLINE 10,24 AT 33
2108 VLINE 6,33 AT 10: VLINE 6,33 AT 24
2110 COLOR = 8: PLOT 17,9
2115 BY=32: BX=INT (RND (1) * 11) + 12:
COLOR = 15: PLOT BX,BY:RETURN
2200 REM *** BURACO 2 ***
2205 GOSUB 2050:COLOR = 1: HLINE 10,24 AT
33: VLINE 6,33 AT 10: VLINE 14,33 AT
24:HLINE 10,35 AT 6:HLINE 24,35 AT 16
:VLINE 6,16 AT 35
2207 COLOR = 2: VLINE 18,19 AT 20: VLINE
17,21 AT 21: VLINE 17,23 AT 22:VLINE
16,26 AT 23
2210 COLOR = 8:PLOT 32,11
2215 BY = 32: BX = INT ( RND (1) * 11 )
+ 12:COLOR = 15: PLOT BX,BY: RETURN
2300 REM *** BURACO 3 ***
2305 GOSUB 2050: COLOR = 1:HLINE 24,35 AT
6: HLINE 10,24 AT 15: HLINE 24,35 AT
24:HLINE 10,24 AT 33: VLINE 6,15 AT
24: VLINE 19,33 AT 10: VLINE 24,33 AT
28: VLINE 4,24 AT 35
2307 COLOR = 13: PLOT 30,23: VLINE 22,23
AT 31: VLINE 20,23 AT 32: VLINE 19,23
AT 33: VLINE 17,23 AT 34
2310 COLOR = 8:PLOT 32,9
2315 BY = 32: BX = INT (RND (1) * 11)
+ 12: COLOR = 15: PLOT BX,BY:RETURN
2400 REM *** BURACO 4 ***
2405 GOSUB 2050: COLOR = 1: HLINE 5,35 AT
6: HLINE 15,25 AT 15: HLINE 25,35 AT
24: HLINE 5,15 AT 15: HLINE 6,33 AT 5
: VLINE 15,33 AT 15: VLINE 15,24 AT
25: VLINE 5,24 AT 35
2407 COLOR = 13: PLOT 15,7:VLINE 7,8 AT
15:VLINE 7,9 AT 17: PLOT 20,7
2408 COLOR = 2:PLOT 15,14: VLINE 13,14 AT
16: VLINE 13,14 AT 17:VLINE 13,14 AT
18:VLINE 13,14 AT 19: VLINE 13,14 AT
20: VLINE 12,14 AT 21: VLINE 11,14 AT
22
2410 COLOR = 8:PLOT 32,21
2415 BY = 32: BX = INT (RND (1) * 9) + 6
: COLOR = 15: PLOT BX,BY: RETURN
2500 REM *** BURACO 5 ***
2505 GOSUB 2050: COLOR = 1:HLINE 10,24 AT
6:HLINE 10,24 AT 33: VLINE 6,33 AT 10:
VLINE 6,33 AT 24: HLINE 12,17 AT 17:
2510 COLOR = 8: PLOT 17,9
2515 BY = 32: BX = INT (RND (1) * 11) +
: COLOR = 15: PLOT BX,BY: RETURN
2600 REM *** BURACO 6 ***
2605 GOSUB 2050: COLOR = 1:HLINE 10,24 AT
33: VLINE 6,33 AT 10: HLINE 16,33 AT
24: HLINE 10,35 AT 6: HLINE 24,35 AT
16: HLINE 6,16 AT 35
2607 HLINE 26,29 AT 9: HLINE 26,29 AT 13:
VLINE 9,13 AT 26
2608 COLOR = 9: VLINE 13,18 AT 21: VLINE
13,18 AT 22: VLINE 13,18 AT 23: VLINE
13,15 AT 24: VLINE 14,15 AT 25: PLOT
26,15
2610 COLOR = 8: PLOT 32,11
2615 BY = 32: BX = INT (RND (1) * 11) +
12:COLOR = 15: PLOT BX,BY: RETURN
2700 REM *** BURACO 7 ***
2705 GOSUB 2050: COLOR = 1: HLINE 5,15 AT
6: HLINE 15,35 AT 12: HLINE 5,25 AT
24: HLINE 25,35 AT 33: VLINE 6,24 AT
5: VLINE 6,12 AT 15: VLINE 24,33 AT
25: VLINE 12,33 AT 35
2707 COLOR = 2: VLINE 16,23 AT 5: VLINE
17,23 AT 7: VLINE 18,23 AT 9: VLINE
20,23 AT 10: PLOT 11,23
2708 COLOR = 9:VLINE 13,28 AT 33: VLINE 13
,28 AT 32: VLINE 13,28 AT 33: VLINE 13
,28 AT 34: HLINE 25,30 AT 13: HLINE
25,30 AT 14: HLINE 25,30 AT 15
2710 COLOR = 8: PLOT 10,9
2715 BY = 32: BX = INT (RND (1) * 7)+26:
COLOR = 15: PLOT BX,BY: RETURN
2800 REM *** BURACO 8
2805 GOSUB 2050: COLOR = 1: HLINE 5,35 AT
6: HLINE 15,25 AT 15: HLINE 5,15 AT
24: HLINE 25,35 AT 33: VLINE 6,24 AT
5: VLINE 15,24 AT 15: VLINE 15,33 AT
25: VLINE 6,33 AT 35
2807 COLOR = 9: VLINE 7,8 AT 25: VLINE 7,9
AT 26: VLINE 7,11 AT 27: VLINE 7,11
AT 28: VLINE 7,12 AT 29: VLINE 7,12

```

```

AT 300: VLIN 7.17 AT 314: VLIN 7.18 AT
32: VLIN 7.19 AT 33: VLIN 7.20 AT
34
2808 COLOR = 13: VLIN 11.13 AT 8: VLIN
10.15 AT 9: VLIN 9.15 AT 10: VLIN 9
.16 AT 11: VLIN 9.16 AT 12
2810 COLOR = 8: PLOT 10,21
2815 BY = 32: BX = INT (RND (1) * 7) +
28: COLOR = 15: PLOT BX, BY: RETURN
2900 REM *** BURACO 9 ***
2905 GOSUB 2050: COLOR = 1: HLIN 5,35 AT
5: HLIN 5,25 AT 18: HLIN 25,35 AT
33: VLIN 6,18 AT 5: VLIN 18,33 AT
25: VLIN 6,33 AT 35
2907 HLIN 8,11 AT 9: VLIN 9,15 AT 11
2908 COLOR = 9: HLIN 25,34 AT 7: HLIN 27,
34 AT 8: HLIN 29,34 AT 9: HLIN 30,34
AT 10: HLIN 31,34 AT 11
2910 COLOR = 8: PLOT 8,12
2915 BY = 32: BX = INT (RND (1) * 7) + 25:
COLOR = 15: PLOT BX, BY: RETURN
3000 REM *** J O G O ***
3010 FOR HD = 1 TO 9: BC = 12
3020 DN HD GOSUB 2100,2200,2300,2400,25
00,2600,2700,2800,2900

```



```

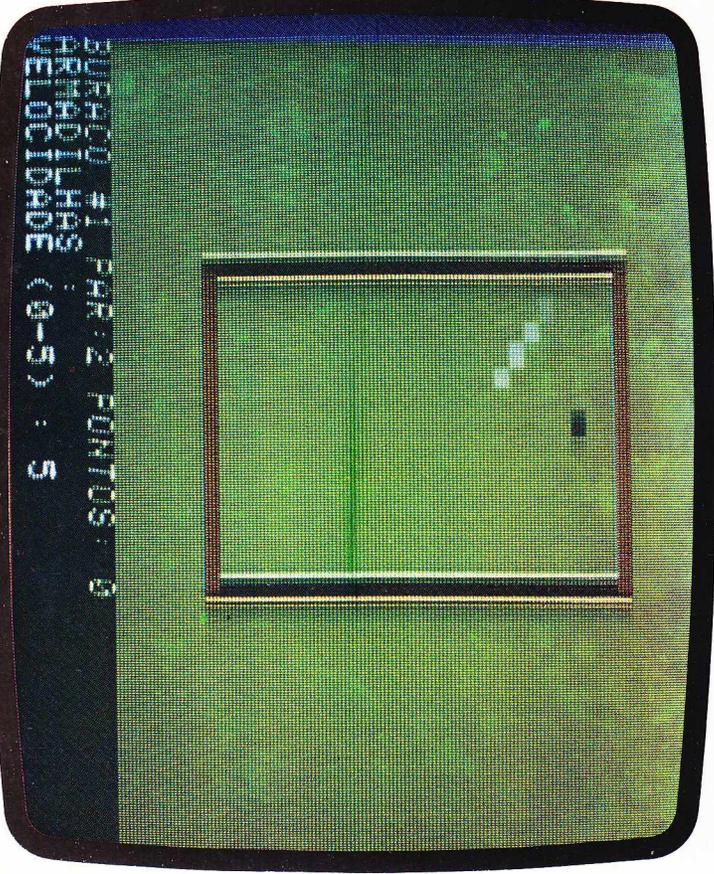
3030 HOME:PRINT "BURACO NUMERO:"HD" PAR:
"HA(HD,5)" PLACAR:ASC
3031 PRINT "TERRENO:":IF HA(HD,1) THEN
PRINT "AGUA ":
3032 IF HA(HD,2) THEN PRINT "AREIA ":
3033 IF HA(HD,3) THEN PRINT "TREGALAR":
3034 IF HA(HD,4) THEN PRINT "BLOCOS ":
3035 VTAB 23: HTAB 1: CALL -958: INPUT
"DIRCAD (1-8): "DIR
3040 IF DIR < 1 OR DIR > 8 OR DIR <> INT
(DIR) THEN VTAB 23: CALL -958:
PRINT "A DIRECAO E' ENTRE 1 E 8..."
:FOR PA = 1 TO 1500: NEXT PA: GOTO
3035
3045 VTAB 23: CALL -958: INPUT
"VELOCIDADE (0-5): "VSP
3050 IF SP < 0 OR SP > 5 THEN VTAB 23:
CALL -958:PRINT "A VELOCIDADE E'
ENTRE 0 E 5..." :FOR PA = 1 TO 1500
:NEXT PA: GOTO 3045
3055 UF = 0
3100 X2 = BX + DI(DI,1) :Y2 = BY +
DI(DI,2)
3110 IF SCRNX2,Y2 = 12 THEN COLOR = BC
: PLOT BX, BY: COLOR = 15: PLOT X2,Y
2: BX = X2: BY = Y2: BC = 12: GOTO
3900
3120 IF SCRNX2,Y2 = 01 THEN 3200
3121 IF SCRNX2,Y2 = 02 THEN 3500
3122 IF SCRNX2,Y2 = 13 THEN 3400
3123 IF SCRNX2,Y2 = 09 THEN 3500
3124 IF SCRNX2,Y2 = 08 THEN 3600

```

# GOLFE MINIATURA

## PARA LINHA APPLE

Neste programa, você vai dar três voltas pelo campo de golfe (sem carrinho) e tentar acertar a bola nos buracos. São nove telas com um buraco em cada uma — em três voltas você precisa acertar 27 buracos. Seus tacos serão as tacadas de 1 a 8, que lhe permitem direcionar as tacadas. Acerte a direção e a velocidade com o cuidado que exige um jogo de golfe e preste atenção para evitar os obstáculos. Há bancos de areia, pequenos lagos, superfícies irregulares e blocos dos quais você precisa desviar. Depois de embocar a bola, passe para outra tela. Quanto menos tacadas der, melhor.



```

3200 IF DI = 1 OR DI = 3 THEN DI = DI+4:
GOTO 3100
3201 IF DI = 7 OR DI = 5 THEN DI = DI-4:
GOTO 3100
3210 DN DI/2 GOTO 3220,3240,3260,3280
3220 IF SCRNX2+1,Y2=1 AND SCRNX2,Y2
+1=1 THEN DI=6:GOTO 3100
3225 IF SCRNX2+1,Y2=1 THEN DI=4: GOTO
3100
3235 DI=6:GOTO 3100
3240 IF SCRNX2+1,Y2=1 AND SCRNX2,Y2-
1=1 THEN DI=8: GOTO 3100
3245 IF SCRNX2+1,Y2=1 THEN DI=2: GOTO
3100
3250 IF SCRNX2+1,Y2=1 THEN DI=6: GOTO
3100
3255 DI=8:GOTO 3100
3260 IF SCRNX2+1,Y2=1 AND SCRNX2,Y2-
1=1 THEN DI=2: GOTO 3100
3265 IF SCRNX2+1,Y2=1 THEN DI=8: GOTO
3100
3270 IF SCRNX2,Y2-1=1 THEN DI=4: GOTO
3100
3275 DI=2: GOTO 3100
3280 IF SCRNX2-1,Y2=1 AND SCRNX2,Y2
+1=1 THEN DI=4: GOTO 3100
3285 IF SCRNX2-1,Y2=1 THEN DI=6: GOTO
3100
3290 IF SCRNX2,Y2+1=1 THEN DI=2: GOTO
3100
3295 DI=4: GOTO 3100
3300 IF TF THEN 3310
3305 TF=3:SC=SC+1
3310 COLOR=BC:PLOT BX, BY: COLOR=15: PLOT
X2,Y2:BX=X2:BY=Y2:BC=2
3320 GOTO 3900
3400 IF TF THEN 3410
3405 TF=3:SC=SC+1
3410 COLOR=BC: PLOT BX, BY: COLOR= 15: P
LOT X2,Y2:BX=X2:BY=Y2:BC= 13
3420 GOTO 3900
3500 IF UF THEN 3520
3505 UF=1
3510 DI=DI + INT (RND(1) *2) * 2 -1
3515 IF DI = 0 THEN DI = 8
3516 IF DI = 9 THEN DI = 1
3520 COLOR=BC: PLOT BX, BY: COLOR= 15 :
PLOT X2,Y2:BX=X2:BY=Y2:BC= 9: GOTO
3900
3600 COLOR=12: PLOT BX, BY: COLOR = 15 :
PLOT X2,Y2: COLOR = 8: PLOT X2,Y2
3602 DI = DI + INT (RND (1) *2)*2 -1
3604 IF DI = 0 THEN DI = 8
3605 SP = SP - .4:IF SP > 0 THEN X2=X2+
DI(DI,1):Y2=Y2+DI(DI,2): GOTO 3110
3606 IF DI = 9 THEN DI = 1
3610 PRINT BELL#BELL#BELL#SC = SC + 1:
GOTO 3990
3900 XX=PEEK (-16336)-PEEK (-16336)
3902 IF TF > 0 THEN TF = TF - 1:IF TF =0
THEN 3910
3905 SP = SP - .2: IF SP > 0 THEN 3100
3910 SC = SC + 1: GOTO 3030
3990 NEXT HO: RETURN
4000 REM *** FIM ***
4010 TEXT: HOME: VTAB 3: HTAB 9: PRINT
"*** GOLFE MINIATURA ***":VTAB 7
4020 PRINT BELL#BELL#BELL#FIM DE
JOGO!!!!
4022 PRINT
4025 PRINT "NO PAR 27 VOZE CONSEGUIU":
PRINT "CHEGAR COM "SC" TACADAS.
SUA MEDIA FOI DE "SC/9" TACADAS
POR BURACO"
4030 VTAB 22: PRINT "ESPERO QUE VOZE
TENHA GOSTADO DO JOGO!!"
4990 RETURN

```