

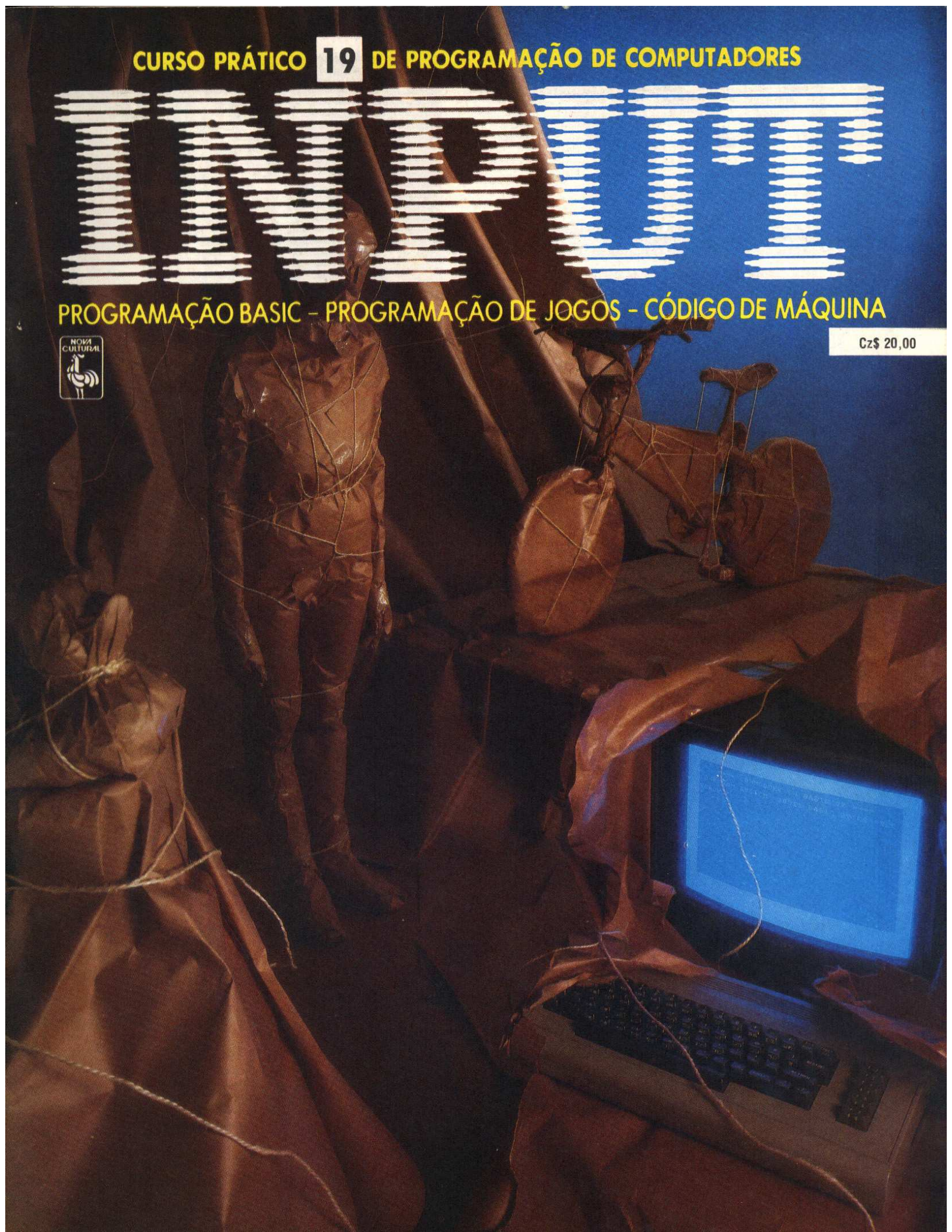
CURSO PRÁTICO **19** DE PROGRAMAÇÃO DE COMPUTADORES

INFORMÁTICA

PROGRAMAÇÃO BASIC – PROGRAMAÇÃO DE JOGOS – CÓDIGO DE MÁQUINA



Cz\$ 20,00



INPUT

Vol. 2

Nº 19

NESTE NÚMERO

PROGRAMAÇÃO BASIC

TRABALHE COM O CÓDIGO ASCII

Linguagem comum aos computadores, o código ASCII permite a comunicação entre essas máquinas. Use números em vez de caracteres. Um programa para codificar e decodificar as mais diversas mensagens361

CÓDIGO DE MÁQUINA

CÓDIGOS PARA O MSX

O que são os códigos de controle 367

PROGRAMAÇÃO DE JOGOS

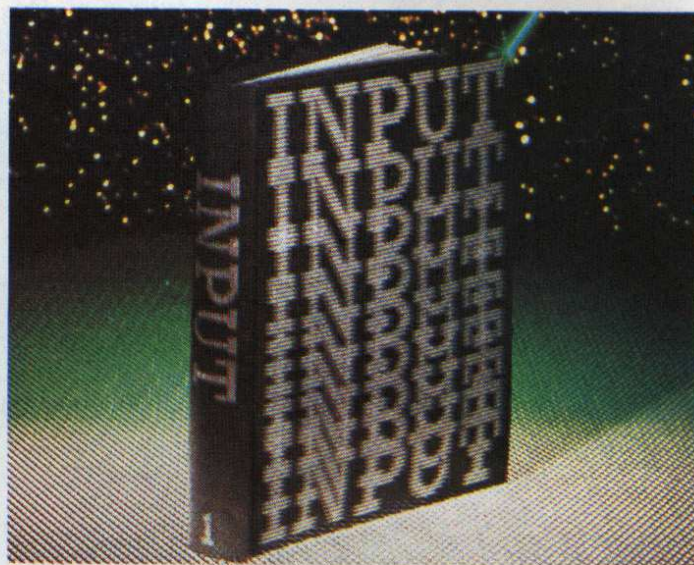
UM JOGO DE TIRO AO PATO

Até mesmo o mais ferrenho ecologista pode participar desse inocente jogo de tiro ao pato, que tem a vantagem de limitar a mortandade dos simpáticos animaizinhos à tela do computador. Veja também como utilizar a sub-rotina de controle do joystick368

APLICAÇÕES

CONVERSÕES NO COMPUTADOR

Ao contrário de outros povos, ingleses e norte-americanos empregam sistemas de pesos e medidas diferentes do sistema métrico decimal. Aprenda a fazer conversões deste para aqueles (e vice-versa), no computador374



PLANO DA OBRA

“INPUT” é uma obra editada em fascículos semanais, e cada conjunto de 15 fascículos compõe um volume. A capa para encadernação de cada volume estará à venda oportunamente.

COMPLETE SUA COLEÇÃO

Exemplares atrasados, até seis meses após o encerramento da coleção, poderão ser comprados, a preços atualizados, da seguinte forma: **1. Pessoalmente** — por meio de seu jornaleiro ou dirigindo-se ao distribuidor local, cujo endereço poderá ser facilmente conseguido junto a qualquer jornaleiro de sua cidade. Em São Paulo os endereços são: Rua Brigadeiro Tobias, 773 (Centro); Av. Industrial, 117 (Santo André); e, no Rio de Janeiro: Rua da Passagem, 93 (Botafogo). **2. Por carta** — Poderão ser solicitados exemplares atrasados também por carta, que deve ser enviada para DINAP — Distribuidor Nacional de Publicações — Números Atrasados — Estrada Velha de Osasco, 132 (Jardim Tereza) — CEP 06000 — Osasco — São Paulo. **3. Por telex** — Utilize o nº (011) 33670 ABSA. Em Portugal, os pedidos devem ser feitos à Distribuidora Jardim de Publicações Ltd. — Qta. Pau Varais, Azinhaga de Fetais — 2685, Camarate — Lisboa; Tel. 257-2542 — Apartado 57 — Telex 43 069 JARLIS P.

Não envie pagamento antecipado. O atendimento será feito pelo reembolso postal e o pagamento, incluindo as despesas postais, deverá ser efetuado ao se retirar a encomenda na Agência do Correio. **Atenção:** Após seis meses do encerramento da coleção, os pedidos serão atendidos, dependendo da disponibilidade de estoque. **Obs.:** Quando pedir livros, mencione sempre o título e/ou o autor da obra, além do número da edição.

COLABORE CONOSCO

Encaminhe seus comentários, críticas, sugestões ou reclamações ao Serviço de Atendimento ao Leitor — Caixa Postal 9442, São Paulo — SP.



Editor
VICTOR CIVITA

REDAÇÃO

Diretora Editorial: Iara Rodrigues

Editor chefe: Paulo de Almeida
Editor de texto: Cláudio A.V. Cavalcanti
Editor de Arte: Eduardo Barreto
Chefe de Arte: Carlos Luiz Batista
Assistentes de Arte: Ailton Oliveira Lopes, Dilvacy M. Santos, José Maria de Oliveira, Grace A. Arruda, Monica Lenardon Corradi
Secretária de Redação/Coordenadora: Stefania Crema
Secretários de Redação: Beatriz Hagström, José Benedito de Oliveira Damião, Maria de Lourdes Carvalho, Marisa Soares de Andrade, Mauro de Queiroz
Secretário Gráfico: Antonio José Filho

COLABORADORES

Consultor Editorial Responsável: Dr. Renato M.E. Sabbatini (Diretor do Núcleo de Informática Biomédica da Universidade Estadual de Campinas)
Execução Editorial: DATAQUEST Assessoria em Informática Ltda. Campinas, SP.
Tradução: Maria Fernanda Sabbatini
Adaptação, programação e redação: Abílio Pedro Neto, Aluísio J. Dornellas de Barros, Marcelo R. Pires Therezo, Raul Neder Porrelli
Coordenação geral: Rejane Felizatti Sabbatini
Editora de Texto: Ana Lúcia B. de Lucena
Assistente de Arte: Dagmar Bastos Sampaio
COMERCIAL
Diretor Comercial: Roberto Martins Silveira
Gerente Comercial: Flávio Ferruccio Maculan
Gerente de Circulação: Denise Maria Mozol

PRODUÇÃO

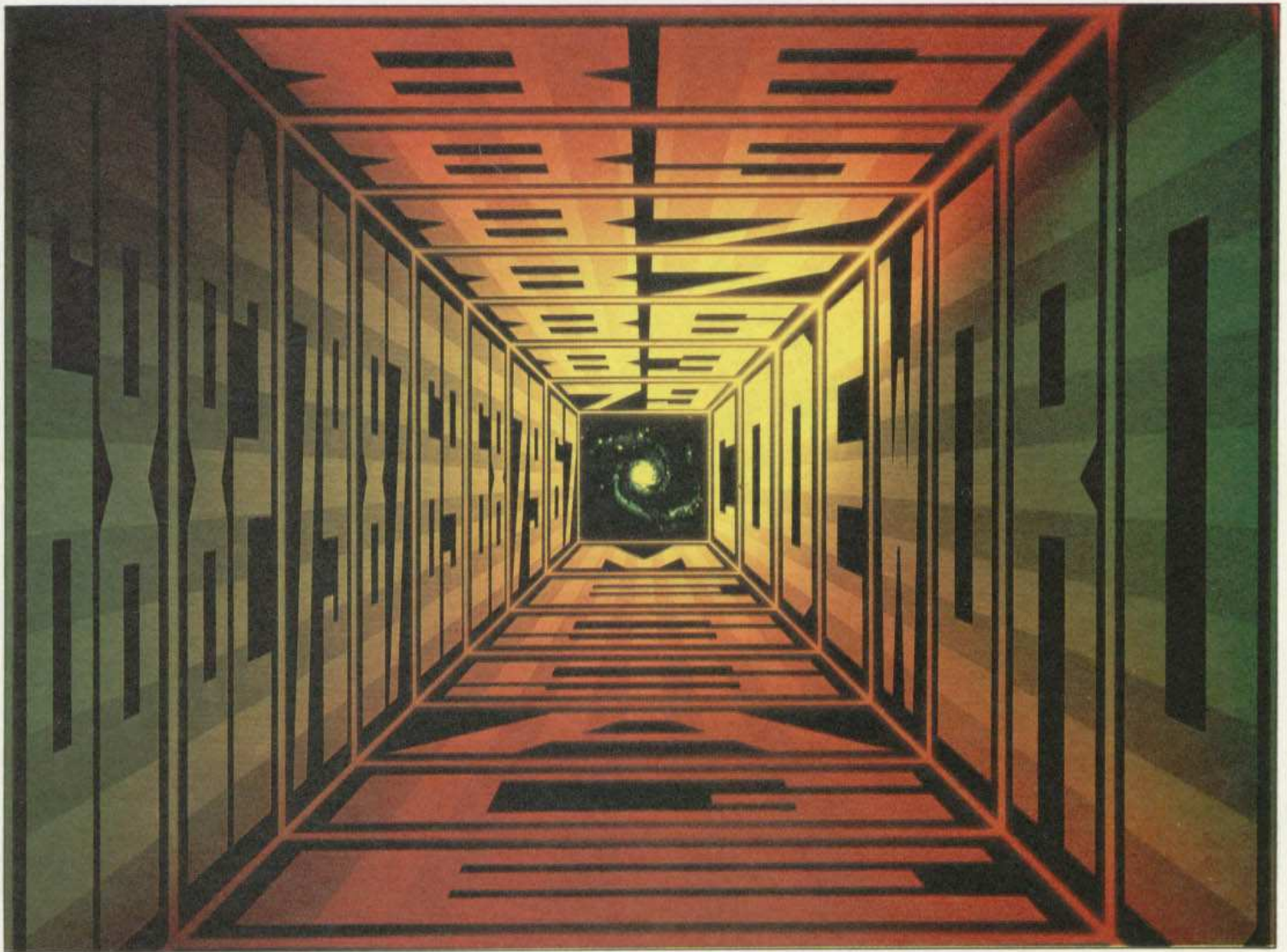
Gerente de Produção: João Stungis
Coordenador de Impressão: Atilio Roberto Bonon
Preparador de Texto/Coordenador: Eliel Silveira Cunha

Preparadores de Texto: Ana Maria Dilguerian, Antonio Francellino de Oliveira, Karina Ap. V. Grechi, Levon Yacubian, Maria Teresa Galluzzi, Paulo Felipe Mendrone
Revisor/Coordenador: José Maria de Assis
Revisoras: Conceição Aparecida Gabriel, Isabel Leite de Camargo, Lígia Aparecida Ricetto, Maria do Carmo Leme Monteiro, Maria Luiza Simões, Maria Teresa Martins Lopes.

© Marshall Cavendish Limited, 1984/85.
© Editora Nova Cultural Ltda., São Paulo, Brasil, 1986.
Edição organizada pela Editora Nova Cultural Ltda.
(Artigo 15 da Lei 5 988, de 14/12/1973).
Esta obra foi composta na AM Produções Gráficas Ltda. e impressa na Divisão Gráfica da Editora Abril S.A.

TRABALHE COM O CÓDIGO ASCII

- O QUE É O CÓDIGO ASCII?
- USE NÚMEROS EM VEZ DE CARACTERES
- UM PROGRAMA PARA CODIFICAR E DECODIFICAR MENSAGENS



Conjunto de códigos alfanuméricos padronizados, o código ASCII foi concebido com o objetivo de tornar possível a transferência de dados de um computador para outro.

Cada tecla ou combinação de teclas do seu micro corresponde a um código eletrônico dentro do computador. Em BASIC, códigos desse tipo são representados por valores decimais entre 0 e 255.

A letra A, por exemplo, tem o código decimal 65, B é 66, e assim por diante. Toda vez que uma letra ou palavra é digitada, o computador armazena os códigos correspondentes.

Os valores e caracteres correspondentes não são, porém, os mesmos em todos os computadores; mas felizmente existe alguma padronização no modo com que eles são usados.

Os valores de código são referidos pela sigla ASCII (que vem de *American Standard Code for Information Interchange*, ou seja, Código Padrão Americano para Troca de Informação). Es-

se código foi montado com o objetivo de possibilitar a transferência de dados de um computador para outro. Os micros TRS, MSX, Apple e Spectrum costumam empregá-lo, pelo menos em parte. Apenas o ZX-81 tem um código completamente diferente.

O CÓDIGO ASCII

O conjunto completo de caracteres ASCII não se apresenta do mesmo modo em todos os computadores. Mas há semelhanças entre os diversos conjun-



tos. A maior delas está na faixa de 33 a 90, que cobre os símbolos mais comuns — pontuação, números e letras maiúsculas. Existe um modo muito simples de descobrir o código ASCII de um caractere: basta digitar **PRINT ASC ("X")** (ou **PRINT CODE "X"** no Spectrum). Você terá, assim, o código de X ou de qualquer outro caractere que for colocado em seu lugar. O próximo programa facilita as coisas para você:



```
10 PRINT "DIGITE QUALQUER LETRA,
```

```
NUMERO OU CHARACTER"
20 INPUT AS
30 PRINT "O CODIGO ASCII PARA ";
AS;" E ";ASC(AS)
40 GOTO 20
```



No Spectrum, mude a linha 30 para:

```
30 PRINT "O CODIGO ASCII PARA
";AS;" E ";CODE AS
```

O oposto de **ASC** ou **CODE** é **CHR\$,** que converte o número de código para o caractere correspondente. O próximo

programa converte todos os códigos ASCII de 33 a 90 em caracteres:



```
10 PRINT "CODIGO ASCII", "CARACTE
R"
20 FOR N=33 TO 90
30 PRINT N,CHR$(N).
40 FOR D=1 TO 500:NEXT D
50 NEXT N
```

As letras minúsculas estão na faixa de 97 a 122 e podem ser vistas mudando-se o último número da linha 20 para 122. Ao substituírmos este último por 255,

teremos uma visão do conjunto completo de caracteres. A listagem dos caracteres e códigos correspondentes está no manual do seu micro.

No TRS-Color, os caracteres minúsculos aparecerão na tela iguais aos maiúsculos, só que invertidos. A impressora ligada ao micro provavelmente interpretará esses caracteres como minúsculos de verdade. O TK-2000 não possui letras minúsculas.

COMO USAR O CÓDIGO ASCII

A vantagem de se usar um número em lugar de uma letra é que se pode alterá-lo matematicamente de várias maneiras. Então, ao se imprimir o **CHR\$** do novo número, o que se obtém é uma letra diferente. Base para muitos programas codificadores, essa possibilidade de conversão é muito interessante, visto que não se pode manipular uma letra por meio de regras matemáticas.

Códigos simples apenas adicionam um valor constante a cada número, de modo a fazer com que a letra "viaje" pelo alfabeto. Por exemplo, a letra A vira G, o B torna-se H, etc. Um código desse tipo é tão fácil de montar quanto de decifrar. Assim, um programa bem simples pode tentar as 26 combinações possíveis; um rápido exame determinará qual dessas combinações é a correta.

Para ser útil, contudo, o programa deve fazer coisas mais complicadas. O programa abaixo, por exemplo, usa uma palavra de código que funciona como chave. Assim, cada letra da mensagem é modificada de uma forma diferente. Muito mais difícil de ser decifrado, esse código exige que se conheça a palavra-chave.

S

```
10 POKE 23658,8: LET CP=0:
LET PM=0: LET DS=""
20 CLS
30 INPUT "QUAL E A SENHA? ";
LINE CS
40 PRINT "INTRODUZA A MENSAGEM"
50 INPUT LINE MS
60 LET CP=CP+1: IF CP>LEN CS
THEN LET CP=1
70 LET PM=PM+1
80 IF PM>LEN MS THEN GOTO
200
90 LET FS=MS(PM)
100 IF FS<"A" OR FS>"Z" THEN
GOTO 150
110 LET F=CODE FS+CODE CS(CP)-
65
```

```
120 IF F>90 THEN LET F=F-26
130 LET DS=DS+CHR$ F
140 GOTO 60
150 IF FS<"0" OR FS>"9" THEN
LET DS=DS+FS: GOTO 70
160 LET F=CODE FS+CODE CS(CP)-
48
170 IF F>57 THEN LET F=F-10:
GOTO 170
180 LET DS=DS+CHR$ F
190 GOTO 60
200 PRINT "A MENSAGEM CODIFICADA E:"
210 PRINT 'DS
220 STOP
```



```
10 CLEAR 300
20 CLS
30 LINEINPUT"QUAL E A SENHA?";C
$
40 PRINT"INTRODUZA A MENSAGEM"
50 LINEINPUT MSS
60 CP=CP+1:IF CP>LEN(CS)THEN CP
=1
70 PM=PM+1
80 IF PM>LEN(MSS)THEN 200
90 FS=MID$(MSS,PM,1)
100 IF FS<"A" OR FS>"Z" THEN 15
0
110 F=ASC(FS)+ASC(MID$(CS,CP,1)
)-65
120 IF F>90 THEN F=F-26
130 CDS=CDS+CHR$(F)
140 GOTO 60
150 IF FS<"0" OR FS>"9" THEN CD
$=CDS+FS:GOTO 70
160 F=ASC(FS)+ASC(MID$(CS,CP,1)
)-48
170 IF F>57 THEN F=F-10:GOTO 17
0
180 CDS=CDS+CHR$(F)
190 GOTO 60
200 PRINT:PRINT"A MENSAGEM CODI
FICADA E:"
210 PRINT:PRINT CDS
220 END
```



```
10 HOME
30 INPUT "Palavra chave? ";CS
40 PRINT "Digite a mensagem:"
50 INPUT MSS
60 CP = CP + 1: IF CP > LEN (C
S) THEN CP = 1
70 PM = PM + 1
80 IF PM > LEN (MSS) THEN 200
90 FS = MID$(MSS,PM,1)
100 IF FS < "A" OR FS > "Z" TH
EN 150
110 F = ASC (FS) + ASC ( MID$(
CS,CP,1) ) - 65
120 IF F > 90 THEN F = F - 26
130 CDS = CDS + CHR$ (F)
140 GOTO 60
150 IF FS < "0" OR FS > "9" TH
EN CDS = CDS + FS: GOTO 70
160 F = ASC (FS) + ASC ( MID$(
CS,CP,1) ) - 48
```

```
170 IF F > 57 THEN F = F - 10:
GOTO 170
180 CDS = CDS + CHR$ (F)
190 GOTO 60
200 PRINT : PRINT "A mensagem
codificada e:"
210 PRINT : PRINT CDS
220 END
```



```
20 CLS
30 LINEINPUT"Palavra de código?"
";CS
40 PRINT"Digite a mensagem:"
50 LINEINPUT MSS
60 CP=CP+1:IFCP>LEN(CS)THENCPCP=1
70 PM=PM+1
80 IFPM>LEN(MSS)THEN200
90 FS=MID$(MSS,PM,1)
100 IFFS<"A"ORFS>"Z"THEN150
110 F=ASC(FS)+ASC(MID$(CS,CP,1)
)-65
120 IFF>90THENF=F-26
130 CDS=CDS+CHR$(F)
140 GOTO60
150 IFFS<"0"ORFS>"9"THENCDS=CDS
+FS:GOTO70
160 F=ASC(FS)+ASC(MID$(CS,CP,1)
)-48
170 IFF>57THENF=F-10:GOTO170
180 CDS=CDS+CHR$(F)
190 GOTO60
200 PRINT:PRINT"A mensagem codi
ficada e:"
210 PRINT:PRINTCDS
220 END
```

Depois de pedir que sejam digitadas a palavra-chave e a mensagem, o programa faz a codificação, mostrando-a na tela. A mensagem ficará então preservada (ou seja, somente alguém que conheça o código poderá decifrá-la).

O modo como isso funciona é bastante simples, assemelhando-se muito ao método descrito anteriormente. A diferença está no fato de que, em vez de adicionar um valor fixo a cada letra; o código ASCII da primeira letra da palavra-chave é acrescentado à primeira letra da mensagem; a segunda letra da palavra-chave é adicionada à segunda da mensagem, e assim por diante. Quando o fim da palavra-código é encontrado, o ciclo recomeça.

Se o resultado de algumas das adições for maior do que 90 (o que significa que a letra correspondente ficará situada além do Z), devemos subtrair 26, de modo a trazer o caractere para dentro do alfabeto. Os números são manipulados separadamente nas linhas 150 a 180 para manter o código entre 48 e 57, que corresponde a 0 e 9.

O programa decodificador é muito semelhante ao primeiro, diferindo apenas em alguns sinais + e - que foram mudados, além de certos valores. Os números das linhas não coincidem; assim,

pode-se usar o programa combinado com o anterior. Algumas linhas foram colocadas de modo que se possa escolher entre codificar e decodificar uma mensagem.

Se você está familiarizado com os comandos de remuneração e edição de linhas do seu computador, tente usá-los para transformar o programa anterior no programa que apresentamos agora. Depois, junte-os (MERGE).

S

```

12 INPUT "(C)ODIFICAR OU (D)E
CODIFICAR?"; LINE A$
14 IF A$="D" THEN GOTO 400
16 IF A$<>"C" THEN GOTO 12
400 CLS
410 INPUT "QUAL E A SENHA? ";
LINE C$
420 PRINT "INTRODUZA A MENSAGE
M CODIFICADA "
430 INPUT LINE M$
440 LET CP=CP+1: IF CP>LEN C$
THEN LET CP=1
450 LET PM=PM+1
460 IF PM>LEN M$ THEN GOTO
580
470 LET F$=M$(PM)
480 IF F$<"A" OR F$>"Z" THEN
GOTO 530
490 LET F=CODE F$-CODE C$(CP)+
65
500 IF F<65 THEN LET F=F+26
510 LET D$=D$+CHR$ F
520 GOTO 440
530 IF F$<"0" OR F$>"9" THEN
LET D$=D$+F$: GOTO 450
540 LET F=CODE F$-CODE C$(CP)+
48
550 IF F<48 THEN LET F=F+10:
GOTO 550
560 LET D$=D$+CHR$ F
570 GOTO 440
580 PRINT "'A MENSAGEM DECODIF
ICADA E "
590 PRINT 'D$
600 STOP

```

T T

```

12 INPUT (C)ODIFICAR OU (D)ECOD
IFICARPRINT;A$
14 IF A$="D" THEN GOTO 400
16 IF A$<>"C" THEN GOTO 12
400 CLS
410 LINEINPUT"QUAL E A SENHA?";
C$
420 PRINT"INTRODUZA A MENSAGEM
CODIFICADA"
430 LINEINPUT MSS
440 CP=CP+1:IF CP>LEN(C$) THEN
CP=1
450 PM=PM+1
460 IF PM>LEN(MS$) THEN 580
470 F$=MID$(MSS,PM,1)
480 IF F$<"A" OR F$>"Z" THEN 53
0
490 F=ASC(F$)-ASC(MID$(C$,CP,1)
)+65
500 IF F<65 THEN F=F+26

```



```

510 CDS=CDS+CHR$(F)
520 GOTO 440
530 IF F$<"0" OR F$>"9" THEN CD
S=CDS+F$:GOTO 450
540 F=ASC(F$)-ASC(MID$(C$,CP,1)
)+48
550 IF F<48 THEN F=F+10:GOTO 55
0
560 CDS=CDS+CHR$(F)
570 GOTO 440
580 PRINT"A MENSAGEM DECODIFICA
DA E: "
590 PRINT:PRINT CDS
600 END

```



```

12 INPUT "[C]odificar ou [D]ec
odificar ";AS
14 IF AS = "D" THEN 410
16 IF AS < > "C" THEN 12
400 HOME
410 INPUT "Palavra chave? ";CS
420 PRINT "Digite a mensagem:"
430 INPUT MSS
440 CP = CP + 1: IF CP > LEN (
CS) THEN CP = 1
450 PM = PM + 1
460 IF PM > LEN (MSS) THEN 58
0
470 F$ = MID$(MSS,PM,1)
480 IF F$ < "A" OR F$ > "Z" TH
EN 530
490 F = ASC (F$) - ASC ( MID$
(C$,CP,1)) + 65
500 IF F < 65 THEN F = F + 26
510 CDS = CDS + CHR$( F)
520 GOTO 440
530 IF F$ < "0" OR F$ > "9" TH
EN CDS = CDS + F$: GOTO 450
540 F = ASC (F$) - ASC ( MID$
(C$,CP,1)) + 48
550 IF F < 48 THEN F = F + 10:
GOTO 550
560 CDS = CDS + CHR$( F)
570 GOTO 440
580 PRINT : PRINT "A mensagem
decodificada e: "
590 PRINT : PRINT CDS
600 END

```



```

12 INPUT"(C)odificar ou (D)ecod
ificar? ";AS
14 IF AS="D"THEN400
16 IF AS<"C"THEN12
400 CLS
410 LINEINPUT"Palavra de código
? ";CS
420 PRINT"Digite a mensagem:"
430 LINEINPUT MSS
440 CP=CP+1:IFCP>LEN(C$)THENC
P=
1
450 PM=PM+1
460 IFPM>LEN(MSS)THEN580
470 F$=MID$(MSS,PM,1)
480 IFF$<"A"ORF$>"Z"THEN530
490 F=ASC(F$)-ASC(MID$(C$,CP,1)

```



```

)+65
500 IFF<65THENF=F+26
510 CDS=CDS+CHRS(F)
520 GOTO440
530 IFFS<"0"ORFS>"9"THENCDS=CDS
+FS:GOTO450
540 F=ASC(FS)-ASC(MIDS(C$,CP,1)
)+48
550 IFF<48THENF=F+10:GOTO550
560 CDS=CDS+CHRS(F)
570 GOTO440
580 PRINT:PRINT"A mensagem deco
dificada é:"
590 PRINT:PRINTCDS
600 END

```

Se você quer que o seu código seja realmente indecifrável, codifique a mensagem usando uma segunda palavra-chave. Desse modo, ele se tornará à prova de espíões. Mas tenha o cuidado de não trocar a ordem das duas palavras-chave na hora de decodificar. Do contrário, você poderá causar uma confusão monumental.

COMPARAÇÕES

Uma das funções mais importantes do código ASCII consiste em facilitar a comparação de palavras (veja página 241 e seguintes).

Essa operação é feita letra por letra pelo computador.

Suponhamos, por exemplo, que se queira comparar as palavras COSMONAUTA e COSMOPOLITA. O computador começará confrontando os dois C; em seguida, ele passará sucessivamente para os dois O, os dois S, os dois M, os dois O, até chegar ao N e ao P.

Vale lembrar que o que está sendo comparado não são posições alfabéticas arbitrárias, mas os códigos ASCII de cada caractere.

Como P tem um valor ASCII maior que N, nesta comparação, a palavra COSMOPOLITA tem um valor maior do que a expressão COSMONAUTA. Note que nem a soma dos códigos nem o número de letras são levados em conta.

Cada letra é comparada individualmente com a sua correspondente (para um aprofundamento do tema, veja outros exemplos de comparações de palavras e cordões no artigo *Cadeia de Caracteres*, página 241).

A seguir, apresentamos alguns exemplos de comparações de vários valores possíveis para A\$ e B\$, juntamente com os códigos ASCII envolvidos. É importante comparar os valores do código de cada caractere aos pares: o primeiro caractere de A\$ com o primeiro de B\$ e assim por diante, até o fim da palavra mais curta.

A\$ ASCII B\$ ASCII RELAÇÃO

```

ABC 65,66,67 ABC 65,66,67 A$=B$
ABD 65,66,68 ABCD 65,66,67 A$>B$
ABD 65,66,68 ABCD 65,66,67
A$<>B$
ABC 65,66,67 Abc 65,97,98 A$<B$
COSMI 67,79,83 COSMO 67,79,83
A$<B$
77,73 77,79
$1 36,49 $1.0 36,49,46 A$<B$

```

Note que no segundo e terceiro exemplos é possível escrever a relação de várias maneiras. No último exemplo, como todos os caracteres são iguais, a palavra mais longa é a de maior valor.

VERIFIQUE AS ENTRADAS

As funções **ASC** e **CODE** funcionam apenas no primeiro caractere de uma variável alfanumérica. Assim, **PRINT ASC ("BRASIL")** — ou **PRINT CODE "BRASIL"**, no Spectrum — mostrará 65, o código de B. Isto é muito útil para se checar comandos **INPUT** do programa. Veja, por exemplo, o programa anterior. A linha 12 pede que você digite um "C" para codificar, ou um "D" para decodificar. As duas linhas seguintes direcionam o programa da forma adequada. Mas, para evitar que o computador não aceite as entradas **CODIFICAR** ou **DECODIFICAR**, escritas por extenso, pode-se reescrever as linhas 14 e 16 como se segue:



```

14 IF ASC(A$)-68 THEN GOTO 400
16 IF ASC(A$)<>67 THEN GOTO 12

```



```

14 IF CODE A$=68 THEN GOTO
400
16 IF CODE A$<>67 THEN GOTO
12

```

CÓDIGOS DE CONTROLE

Alguns códigos ASCII não têm caracteres associados. Quando você tecla **<ENTER>** ou **<RETURN>** o micro identifica o valor 13; mas, em vez de imprimir um caractere na tela, o cursor é colocado no início da linha seguinte; ou, como acontece no Spectrum, o programa é listado. O código 13 é chamado de código de retorno do carro (*Carriage Return*; daí o **<CR>** dos modelos Apple). Experimente colocar no computador **PRINT "A"; CHRS (13); "B"**. O "B" será impresso na linha abaixo do "A".

Este programa permite ver os códigos de controle:



```

10 PRINT CODE INKEYS
20 GOTO 10

```



```

5 A$=INKEYS:IF A$="" THEN 5
10 PRINT ASC(A$)
20 GOTO 10

```



```

5 GET A$
10 PRINT ASC (A$)
20 GOTO 5

```

Tente teclar **<ENTER>** ou **<ESC>** ou pressionar a tecla de retrocesso, ou qualquer outra do teclado, sozinha ou simultaneamente com **<SHIFT>** ou **<CTRL>**, e vá descobrindo os códigos que seu micro utiliza.

Alguns computadores usam mais esses códigos do que outros. Afinal, existem muitos números disponíveis entre 0 e 31 e alguns acima de 90. Tais códigos foram originalmente definidos quando os computadores trabalhavam apenas com as antigas impressoras. Atualmente, o computador é ligado a um televisor ou monitor de vídeo e muitos desses códigos tornaram-se obsoletos.



O Spectrum teve muitos de seus antigos códigos redefinidos, de forma a trabalhar com a tela de TV. A instrução **PAPER** tem código 17 e **INK**, 16. Assim, para escrever a palavra "TÍTULO" em vermelho sobre fundo verde, é preciso digitar o seguinte:

```

10 LET A$=CHRS 17+CHRS 4+CHRS
16+CHRS 2+"TITULO"
20 PRINT A$

```

Usado muitas vezes em um programa, esse cabeçalho precisa ser definido apenas uma vez. Sempre que se quiser usá-lo basta digitar **PRINT A\$**.



O Apple também usa caracteres de controle, tanto a nível operacional como a nível de programas. Assim, você deve conhecer muito bem o **CTRL-S**, o **CTRL-C** ou o **CTRL-D**. Nesse computador, os códigos dos caracteres de controle de **CTRL-A** até **CTRL-Z** ocupam os números de 1 a 26. Desta forma, **CHRS (1)** equivale a **CTRL-A**; **CHRS (2)**, a **CTRL-B**, etc.

CÓDIGOS PARA O MSX

Muito úteis para uma série de truques de programação, os códigos de controle ocupam o espaço que vai de 0 a 31 no código ASCII. Aprenda a usá-los nos micros MSX.

Os códigos de controle correspondem, na maioria dos microcomputadores, aos números entre 0 e 31 no sistema ASCII, e servem para realizar uma série de funções relacionadas principalmente com o controle do vídeo. Em artigos anteriores desta série, mostramos como os diferentes fabricantes de micros aproveitam a faixa de códigos de 0 a 31 para implementar essas funções de controle. Ao contrário do espaço de código ASCII que vai de 32 em diante, não há nessa faixa qualquer padronização entre os diversos modelos de computadores. Vejamos agora como os usuários de micros da linha MSX podem explorar os seus códigos de controle.

A FUNÇÃO CHR\$

Embora o MSX disponha de uma tecla <CONTROL>, que serve para gerar os códigos de 0 a 31 (por exemplo, pressionando-se as teclas <CONTROL> e <A> simultaneamente obtém-se o código 1), um programa BASIC só pode acionar os códigos de controle por intermédio da função CHR\$ (abreviatura de *character*). Essa função permite converter um código numérico inteiro entre 0 e 255 em seu caractere correspondente. Por exemplo, PRINT CHR\$ (65) escreverá na tela a letra A.

Usando o PRINT CHR\$ dessa forma, podemos controlar diversas funções do vídeo, por meio dos códigos de 0 a 31. A seguir, apresentamos os códigos que proporcionam os efeitos mais interessantes no MSX:

- 1 - determina caractere gráfico
- 7 - aciona o alarma sonoro
- 8 - apaga o caractere antes do cursor
- 9 - tabula
- 10 - pula uma linha (LINEFEED)
- 11 - coloca o cursor na posição 1,1 (HOME)

- 12 - equivale ao CLS
- 13 - retorno de carro (RETURN).

Ao contrário dos códigos ASCII de 32 em diante, a função CHR\$ associada aos códigos de 0 a 31 não imprime qualquer caractere visível na tela, causando apenas um efeito imediato.

O programa a seguir mostra uma aplicação interessante dos códigos de controle. Ele está estruturado de tal modo que, toda vez que uma mensagem de erro for mostrada na tela, o computador acionará o alarma sonoro interno (bipe) e pulará uma linha em branco antes da mensagem. Em vez de repetir a mesma seqüência de comandos para cada mensagem de erro, definimos uma variável CS\$, que contém os códigos 7 e 10. Depois, basta usar o PRINT da mensagem, precedendo-a com essa variável.

```
10 CS$=CHR$(10)+CHR$(7)+"*** ERR
O: "
20 CLS:PRINT "Cálculo da raiz q
uadrada"
30 PRINT
35 X$=""
40 LINE INPUT "ENTRE UM NUMERO:
";X$
45 IF X$="" THEN END
50 X=VAL(X$)
60 IF X=0 THEN PRINT CS;"Entrad
a deve ser um número":GOTO 40
70 IF X<0 THEN PRINT CS;"Numero
deve ser maior do que zero":GO
TO 40
80 PRINT
90 PRINT "A raiz quadrada de";X
;"é igual a";SQR(X)
100 PRINT
110 GOTO 40
```

COMO PASSEAR PELO VÍDEO

Outra aplicação interessante dos códigos de controle é a manipulação do cursor de texto. Assim, os códigos 28 a 31 movimentam o cursor da seguinte maneira:

- 28 - move o cursor para a direita
- 29 - move o cursor para a esquerda
- 30 - move o cursor para cima
- 31 - move o cursor para baixo.

Um pequeno programa, que ilustra bem o uso destes códigos é mostrado a seguir:

■	O QUE SÃO OS CÓDIGOS DE CONTROLE
■	COMO UTILIZÁ-LOS EM PROGRAMAS BASIC
■	MENSAGENS COM SONS

```
5 CLS
10 LOCATE 10,15
20 FOR I=1 TO 1000
30 R=INT(RND(1)*4)
35 PRINT CHR$(28+R);".";CHR$(29
);
40 NEXT I
```

Este é um curtíssimo exemplo do chamado "passeio aleatório em duas dimensões". Com a execução do programa, o cursor passará a se deslocar pela tela nas quatro direções possíveis.

As linhas 5 e 10 limpam a tela e colocam o cursor em seu centro. O laço que vai das linhas 20 a 40 coloca 1 000 pontos aleatoriamente na tela. A linha 30 sorteia um número entre 0 e 3. A linha 35 movimenta o cursor com a ajuda da função CHR\$(28+R) (que gera um número entre 28 e 31), imprime um ponto e recua o cursor de novo com a função CHR\$(29).

A TECLA DE CONTROLE

Como foi explicado, a tecla <CONTROL> serve apenas para acionar os códigos de controle diretamente a partir do teclado (ou seja, não é possível "enxertar" códigos de controle dentro de mensagens entre aspas, como é feito com os micros da linha Apple). Por outro lado, diversas teclas já existentes no MSX para limpeza de tela, controle do cursor, etc. são duplicadas por uma combinação de <CONTROL> com outra tecla. Por exemplo, <CONTROL> <I> avança uma posição de tabulação na tela, e equivale à tecla <TAB>.

Entretanto, algumas funções — como o apagamento total ou parcial de uma linha, o avanço do cursor até a palavra seguinte, etc. — não têm teclas correspondentes, e podem ser usadas com vantagem na edição de programas.

Para gerar um código de controle por meio da tecla <CONTROL>, basta lembrar-se da seguinte relação: o código 1 corresponde à letra A, o código 2 corresponde à letra B, e assim por diante. Por exemplo, para fazer o cursor descer uma linha (caractere de controle chamado LINEFEED, LF, ou "alimentação de linha"), basta pressionar conjuntamente as teclas <CONTROL> e J.

UM JOGO DE TIRO AO PATO

Pratique tiro ao alvo, caçando patos ... no computador. Mas, na vida real, respeite o direito dos bichinhos à existência, deixando-os saborear os doces frutos da liberdade.

Se você leu o artigo anterior de *Programação de Jogos*, já deve ter gravado em fita o programa que movimentava uma alça de mira pela tela com auxílio do joystick. No entanto, embora seja interessante ver como se faz isso em BASIC, o programa não terá muito sentido se não houver um alvo.

Agora, mostraremos como incluir a sub-rotina de controle do joystick em programas de jogos. Isso é feito acrescentando-se ao programa original os subprogramas apresentados a seguir. Dessa forma, obteremos um jogo de tiro ao pato.

Um de cada vez, dez patos aparecerão por um breve momento na tela. O objetivo do jogo é alvejá-los antes que eles desapareçam. A contagem de pontos depende da velocidade e da pontaria do jogador; a cada tiro perdido são descontados alguns pontos. A esse desconto daremos o nome de multa.

Carregue ou digite o programa anterior no computador e acrescente as linhas a seguir (evidentemente, estas variarão de acordo com o tipo de computador ao qual estão destinadas).

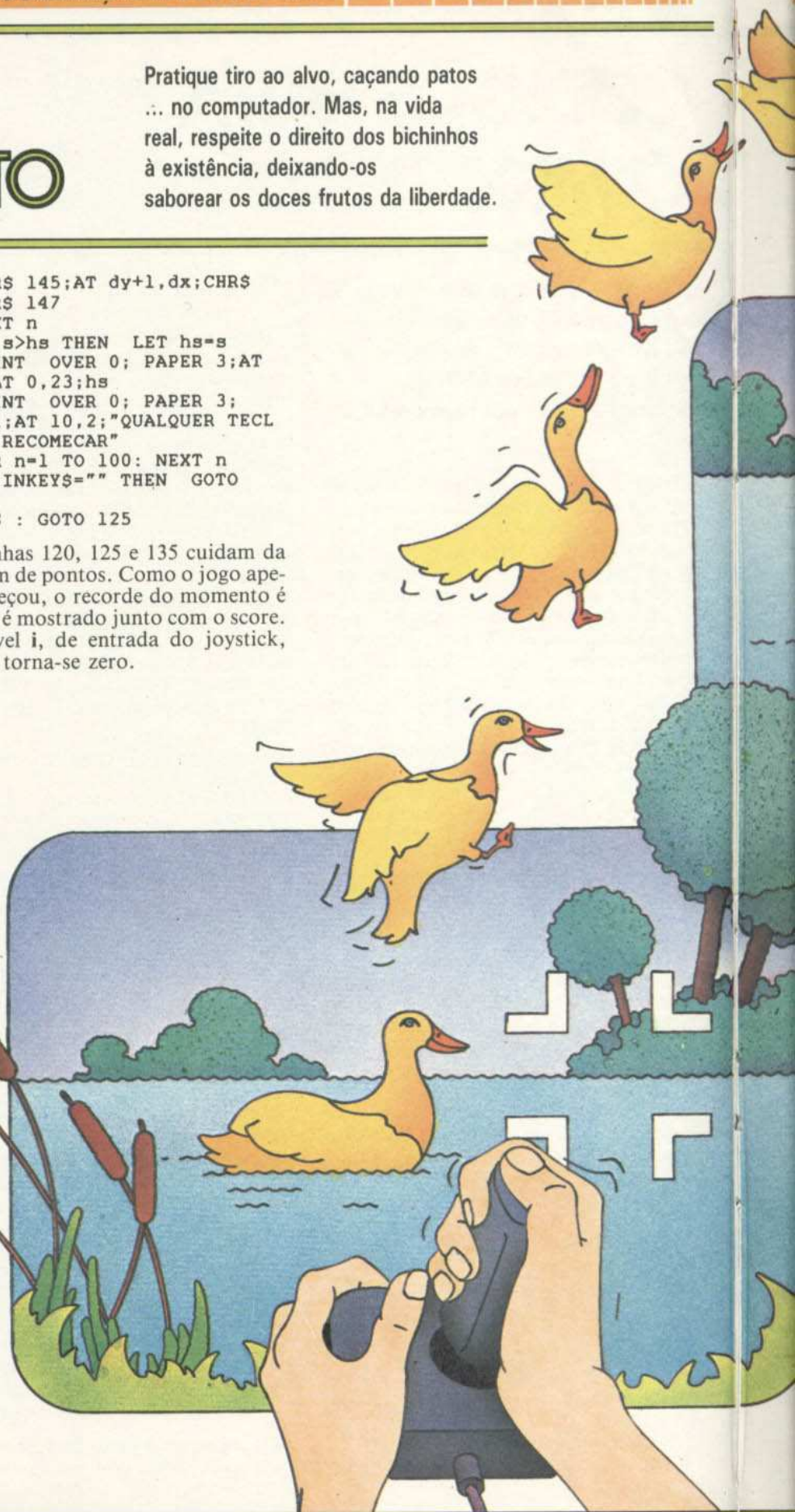
S

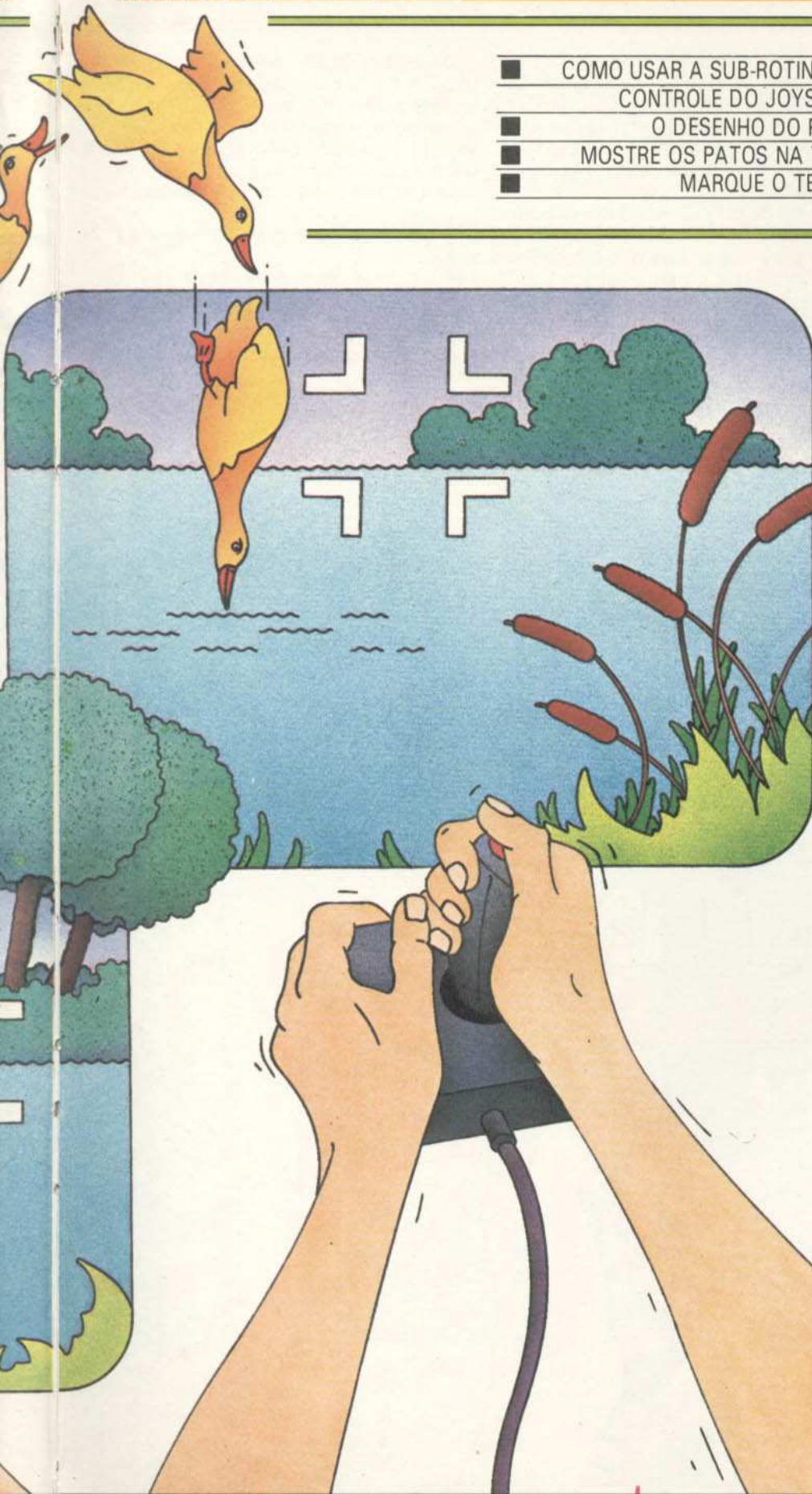
Acrescente as próximas linhas ao programa que movimentava o joystick.

```
120 LET hs=0
125 LET i=0: PRINT PAPER 2;
INK 7;" SCORE";TAB 14;"RECORDE
";TAB 31;" "
135 PRINT PAPER 3; OVER 0;AT
0,23;hs
145 LET s=0
150 FOR n=1 TO 10
160 LET dx=INT (RND*31)
170 LET dy=INT (RND*20)+1
180 PRINT INK 6;AT dy,dx;CHRS
144;CHRS 145;AT dy+1,dx;CHRS
146;CHRS 147
190 POKE 23672,0: POKE 23673,0
205 PRINT OVER 0; PAPER 3;AT
0,8;s;" "
210 IF i<>48 THEN GOTO 400
220 IF ATTR (y,x)=14 AND ATTR
(y+1,x+1)=14 THEN LET s=s+260
-(PEEK 23672): SOUND .02,30:
GOTO 410
230 LET s=s-20: LET i=0
400 IF PEEK 23572+256*PEEK
23673<240 THEN GOTO 200
410 PRINT INK 7;AT dy,dx;CHRS
```

```
144;CHRS 145;AT dy+1,dx;CHRS
146;CHRS 147
420 NEXT n
430 IF s>hs THEN LET hs=s
440 PRINT OVER 0; PAPER 3;AT
0,8;s;AT 0,23;hs
450 PRINT OVER 0; PAPER 3;
FLASH 1;AT 10,2;"QUALQUER TECL
A PARA RECOMEÇAR"
460 FOR n=1 TO 100: NEXT n
470 IF INKEYS="" THEN GOTO
470
480 CLS : GOTO 125
```

As linhas 120, 125 e 135 cuidam da contagem de pontos. Como o jogo apenas começou, o recorde do momento é zero; ele é mostrado junto com o score. A variável *i*, de entrada do joystick, também torna-se zero.





- COMO USAR A SUB-ROTINA DE CONTROLE DO JOYSTICK
- O DESENHO DO PATO
- MOSTRE OS PATOS NA TELA
- MARQUE O TEMPO

- COMO SABER SE O BOTÃO DO JOYSTICK FOI PRESSIONADO
- COMO SABER SE O TIRO ATINGIU O ALVO
- CONTE PONTOS E RECORDES

O laço **FOR...NEXT** entre as linhas 150 e 420 coloca na tela uma série de dez patos que o jogador tentará acertar. A cada volta do laço, as linhas 160 e 170 escolhem uma nova posição para o pato. A linha 180 coloca a ave na tela, imprimindo os quatro caracteres (UDG) que formam o desenho.

A linha 190 coloca o valor zero no contador de tempo. Este será usado para medir os segundos que o pato permanecerá na tela. Quando esse tempo se esgotar, o pato será desenhado em outra posição.

Depois que o programa retornar da sub-rotina de controle do joystick, a linha 205 apagará o primeiro caractere do score. Esta é uma precaução para quando o score estiver diminuindo; assim, nenhuma parte do score anterior permanecerá na tela.

O joystick do Atari tem um botão para atirar; quando acionado, esse botão envia ao computador o caractere de código 48. A linha 210 detecta o tiro, mas só com a mira parada. É possível verificar se foi dado um tiro com a mira em movimento, mas isto ocuparia mais quatro comandos **IF** — um para cada direção. Não é aconselhável fazer isso em BASIC porque diminuiríamos a velocidade do programa.

Caso um tiro tenha sido disparado, o programa verificará se o pato foi atingido, usando **ATTR** na linha 220. **ATTR** verifica a cor de determinada porção da tela. Se a cor for a do pato, o tiro será considerado certo, e o jogador ganhará pontos. O cálculo desses pontos é baseado no tempo gasto pelo jogador para acertar o pato. Cada tiro bem-sucedido é anunciado por um bip do computador.

Se **ATTR** não fornecer a cor do pato, o programa continuará na linha 230, onde uma multa de vinte pontos será subtraída do total, e o valor de **i**, igualado a zero para que a multa não seja cobrada novamente (o que só acontecerá quando outro tiro for desperdiçado). A linha 400 verifica se o limite de tempo foi ultrapassado, saltando para a linha 200, se isto ainda não tiver acontecido. Caso mais de cinco segundos tenham se passado, a linha 410 fará o pato desaparecer.

Depois que os dez patos forem colocados na tela, o programa alcançará a

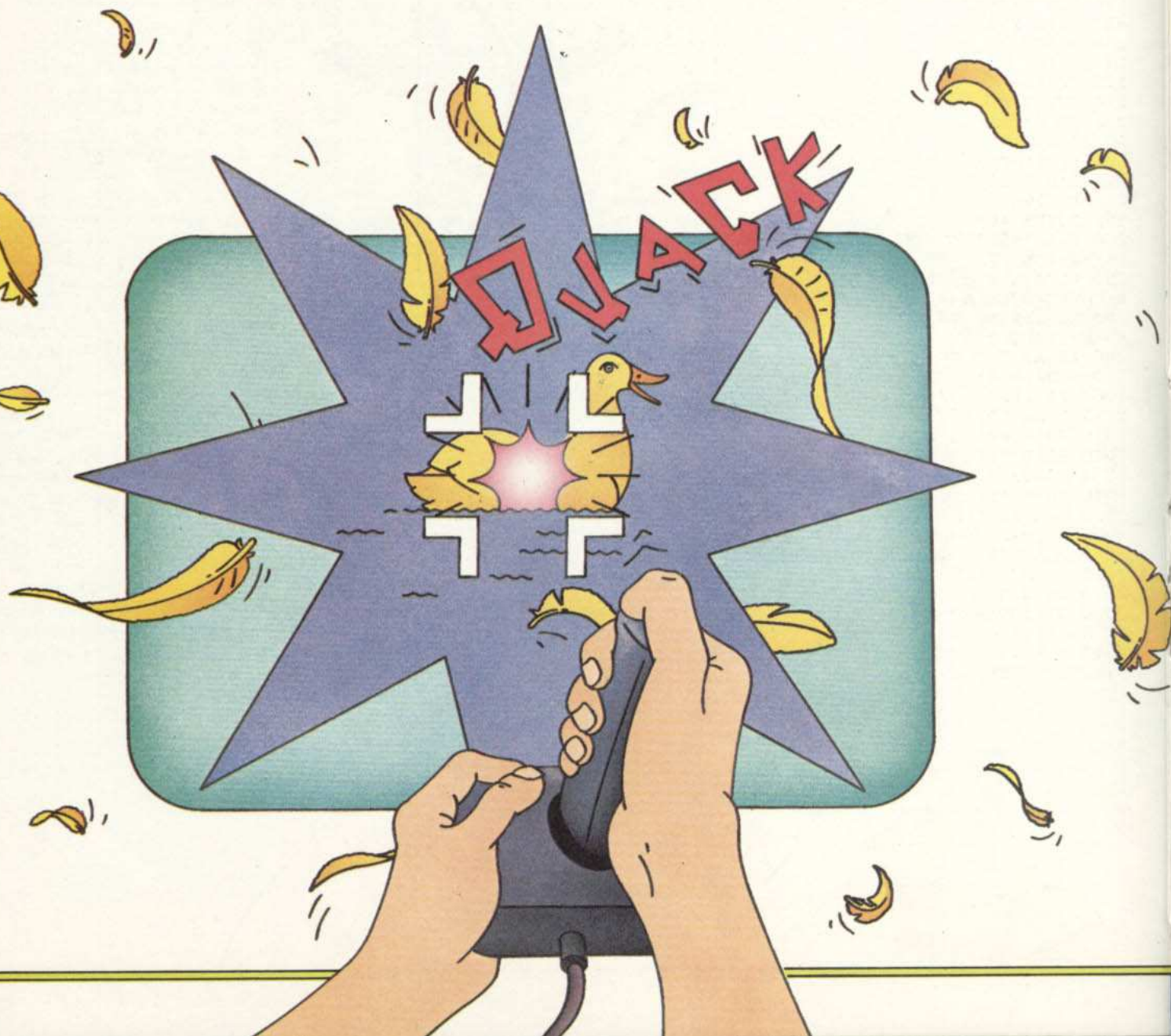
linha 430, que fará a comparação entre o score do jogador e o recorde atual. Se for o caso, o recorde será atualizado. O recorde e o score são mostrados pela linha 440. O programa termina com a frase usual: "Qualquer tecla para recomeçar".

O nível de dificuldade do jogo pode ser alterado modificando-se, na linha 400, o limite de tempo que o jogador tem para acertar o pato. Valores compatíveis para a conquista de pontos e para a multa devem ser colocados nas linhas 220 e 230.

S

Não há programa de tiro ao pato para o ZX-81, mas as modificações que se seguem permitirão que você movimente o monstro do programa da página 316 usando o joystick. O botão do joystick provoca a inversão da tela (você deve usar a versão completa do programa).

```
115 LET A=CODE AS
120 IF A=33 AND X>0 THEN LET X=X-1
130 IF A=36 AND X<28 THEN LET X=X+1
140 IF A=35 AND Y>0 THEN LET Y=Y-1
150 IF A=34 AND Y<20 THEN LET Y=Y+1
155 IF A=28 THEN RAND USR 16606
```



28 é o código enviado ao computador pelo botão do joystick.



O programa do MSX é completado pelas seguintes linhas e modificações.

```
10 SCREEN 1,2:KEY OFF:COLOR 15,
1,10:R=RND(-TIME)
20 FOR I=1 TO 64
55 SPRITES(1)=RIGHTS(AS,32,
70 FOR I=4 TO 10
80 VPOKE BASE(6)+I,15*16+1
90 NEXT
100 VPOKE BASE(6)+30,1*16+1
105 VPOKE BASE(6)+31,6*16+6
110 CLS:STRIG(1) ON:SC=0
120 GOSUB 4000
130 FOR I=0 TO 32:VPOKE BASE(5)
+I,255:NEXT
140 FOR I=1 TO 22:VPOKE BASE(5)
+31+I*32,255:VPOKE BASE(5)+32+I
*32,255:NEXT
150 FOR I=0 TO 31:VPOKE BASE(5)
+23*32+I,255:NEXT
160 GOSUB 3000
180 D=10
190 TIME=0
210 PUT SPRITE 1,(DX,DY),10
220 ON STRIG GOSUB 2000,2000,20
00,2000,2000:IF D<1 THEN 250
230 IF TIME<500 THEN 200
240 D=D-1:IF D>0 THEN GOSUB 300
0:GOTO 190
250 GOSUB 4000
260 FOR I=257 TO 286:VPOKE BASE
(5)+I,245:NEXT
280 LOCATE 1,2:PRINT "QUER JOGA
R NOVAMENTE (S/N) ?"
290 AS=INKEYS:IF AS<>"S" AND AS
<>"N" THEN 290
300 IF AS="S" THEN 110
310 SCREEN 0:COLOR 15,4,4:END
2000 IF ABS(X-DX)>1 AND ABS(Y-D
Y)>1 THEN 2070
2010 PUT SPRITE 1,(DX,209)
2020 PLAY "T25506CL32EG"
2030 GOSUB 3000
2040 SC=SC+550-TIME:GOSUB 4000
2050 D=D-1:TIME=0
2060 RETURN
2070 PLAY "T25501DDE"
2080 SC=SC-25:GOSUB 4000
2090 RETURN
3000 DX=INT(RND(1)*220+10):DY=I
NT(RND(1)*145+25):RETURN
4000 FOR I=33 TO 62:VPOKE BASE(
5)+I,245:NEXT
```

```
4010 LOCATE 0,1:PRINT "SCORE:";
:LOCATE 7,1:PRINTSC;
4020 IF SC>HI THEN HI=SC
4030 LOCATE 14,1:PRINT "RECORDE
:";:LOCATE 24,1:PRINT HI;
4040 RETURN
5000 DATA 3,12,16,32,3
2,64,64,127,64,64,3
2,32,16,12,3,0,224
,152,132,130,130,129,
129,255,129,129,130,
130,132,152,224,0
5010 DATA 14,27,127,31,15,
7,15,31,31,29,30,15,3,
1,1,3,0,0,0,0,0,192,11
2,188,206,30,124,248,224,
64,64,224
```

A linha 10 foi modificada para mudar as cores da tela e iniciar o gerador de números aleatórios.

A linha 20 foi alterada para poder ler os dados adicionais da linha 5010. O sprite do pato é criado pela linha 55.

As linhas 70 a 105 mudam a tabela de cores — **BASE(6)** — com o objetivo de alterar as cores dos caracteres. Isto não é essencial ao nosso programa, mas permitirá que você modifique os valores **P** e **T** do comando **VPOKE BASE(6) + I, T * 16, P** (**P** é a cor de fundo e **T**, a cor de frente).

A linha 110 apaga a tela e liga a função **STRIG(1)** que permite a detecção de um tiro (ela também reduz o score a zero). Quando o botão de tiro do joystick 1 é pressionado, a função **STRIG(N) ON** desvia o curso do programa para o endereço do comando **ON STRIG GOSUB**. **N** determina o joystick: 1 e 3, joystick da direita; 2 e 4, joystick da esquerda. O número 0 permite à tecla de espaços disparar tiros.

A linha 120 chama a sub-rotina 4000, que coloca o score e o recorde na tela.

As linhas 130 a 150 desenharam uma moldura vermelha, colocando na tela uma série de caracteres com **VPOKE**. A cor da moldura é determinada pela linha 105.

A posição em que o pato vai aparecer é escolhida ao acaso pela linha 160. As linhas 180 e 190 estabelecem as condições iniciais: dez patos na variável **D**, score zero e contador de tempo — **TIME**, uma variável interna do MSX —

também igual a zero.

A linha 210 desenha o pato. A linha 220 diz ao programa para onde ir, caso o botão de tiro seja pressionado. São necessários cinco endereços após **ON STRIG** para cobrir os valores de 0 a 4. Se, ao retornar da sub-rotina 2000, onde é verificado se o tiro atingiu o alvo, não houver mais patos — ou seja, **D<1** — o programa irá para a linha 250.

A linha 230 verifica se o tempo que cada pato pode ficar na tela foi ultrapassado. Se isto ainda não aconteceu, o programa voltará à linha 200.

Caso o pato não seja atingido dentro do prazo, a linha 240 diminuirá o número de aves. Se ainda houver patos, a nova posição será calculada pela sub-rotina 3000 e o programa voltará para a linha 190.

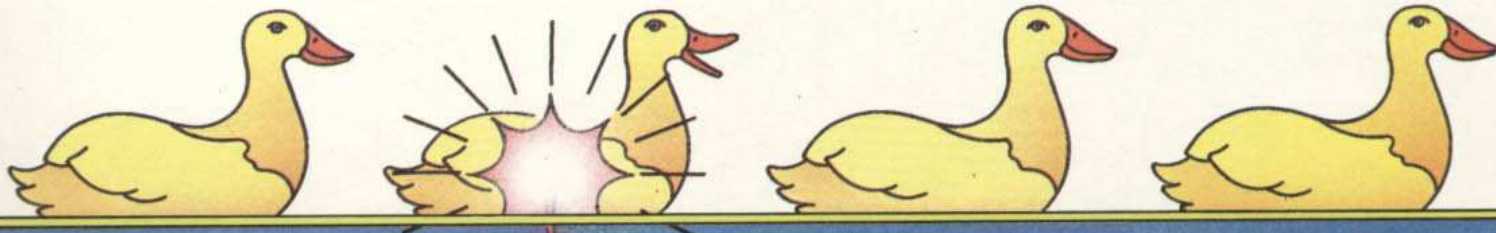
Quando não houver mais patos, a linha 250 chamará a sub-rotina 4000, que mostrará o placar.

A linha 280 faz então a velha pergunta: "Quer jogar novamente?" (a linha 260 apaga a região onde é escrita a mensagem). Caso a resposta seja afirmativa, a linha 300 recomeçará o jogo. Caso seja negativa, a linha 310 terminará com ele. A linha 290 cuida das outras respostas.

A sub-rotina 2000 verifica se o tiro dado acertou o pato. Na linha 2000, as condições **ABS(X-DX)>1** e **ABS(Y-DY)>1** dão uma certa margem de erro à posição da mira. Se exigirmos que as coordenadas do pato — **DX, DY** — sejam exatamente iguais às coordenadas da mira — **X, Y** —, o jogo ficará muito difícil.

A linha 2010 faz com que o sprite do pato desapareça da tela, pois torna sua coordenada vertical igual a 209. Uma pequena melodia celebra o sucesso do jogador, na linha 2020. A linha 2030 chama a sub-rotina 3000 que calcula uma nova posição para o pato.

Os pontos a que o jogador fez jus são calculados pela linha 2040 com base no tempo que ele levou para acertar o pato. A sub-rotina 4000 mostra o novo placar. A linha 2050 reduz em 1 o número de patos e faz o contador de tempo começar novamente do 0.



Caso o jogador erre o tiro, a linha 2070 produzirá algumas notas dissonantes. Uma multa de 25 pontos será então cobrada e o placar, modificado e mostrado ao jogador.

A linha 3000 chamada como sub-rotina calcula ao acaso a nova posição do pato.

A sub-rotina que começa na linha 4000 mostra na tela o score, atualizando e exibindo o recorde.

Os valores da linha 5010 contêm o padrão do sprite do pato.

O nível de dificuldade do jogo pode ser modificado alterando-se o prazo que o jogador tem para acertar o pato. Isso é feito na linha 230. Acompanhando essa alteração, o cálculo dos pontos (na linha 2040) e da multa (na linha 2080) deve sofrer modificações proporcionais.



Complete o programa anterior com as linhas:

```
60 HOME : HGR : SCALE= 1: ROT=
  0:X = 130:Y = 90
100 DX = INT ( RND ( 1) * 256) :
DY = INT ( RND ( 1) * 138)
110 D = 10:SC = 0
120 CV = 0
210 HCOLOR= 3: GOSUB 700
220 IF PEEK ( - 16287) > 127
OR PEEK (16286) > 127 THEN GO
SUB 1000: IF D < 1 THEN 250
230 IF CV < 40 THEN 200
240 D = D - 1: IF D > 0 THEN G
OSUB 1500: GOTO 120
250 : HOME = HTAB 5: VTAB 23: P
RINT "SCORE = ";SC;
260 IF SC > HI THEN HI = SC
270 PRINT TAB( 20);"RECORDE =
";HI
280 PRINT TAB( 5);"QUER JOGAR
NOVAMENTE (S/N) ? "
290 GET AS: IF AS < > "S" AND
AS < > "N" THEN 290
300 IF AS = "S" THEN 60
310 HOME : TEXT : END
500 LX = X:LY = Y:CV = CV + 1
700 DRAW 1 AT DX,DY
710 DRAW 2 AT DX,DY + 8
720 DRAW 3 AT DX + 8,DY
730 DRAW 4 AT DX + 8,DY + 8
740 RETURN
```

```
1000 IF ABS ( X - DX - 4) > 8
OR ABS ( Y - DY - 4) > 8 THEN 1
070
1010 FOR I = 1 TO 20
1020 P = PEEK ( - 16336) : NEXT
```

```
1030 GOSUB 1500
1040 SC = SC + 50 - CV
1050 D = D - 1:CV = 0
1060 RETURN
1070 FOR I = 1 TO 3
1080 P = PEEK ( - 16336) : NEXT
```

```
1090 SC = SC - 5
1100 RETURN
1500 HCOLOR= 0: GOSUB 700
1510 HCOLOR= 3: DRAW 5 AT X,Y
1520 DX = INT ( RND ( 1) * 256)
: DY = INT ( RND ( 1) * 138)
1530 RETURN
```

O HOME da linha 60 foi acrescentado para que o placar pudesse ser mostrado. A linha 110 calcula ao acaso uma posição para o pato. As linhas 120 e 130 estabelecem condições iniciais para o número de patos (D), para o score (S) e para o contador de tempo (CV).

A linha 210 desenha o pato. A linha 220 verifica se o botão do joystick foi pressionado — neste caso, o valor de um dos endereços dos comandos PEEK se torna maior que 127. Caso o programa não funcione, o joystick pode vir a utilizar o endereço -16285.

A linha 230 verifica se o prazo que o jogador tem para acertar o pato se esgotou, voltando para a linha 200, se isto ainda não aconteceu.

Caso um dos patos não tenha sido abatido dentro do limite de tempo, seu número diminuirá em um. Se ainda sobrarem patos, a sub-rotina 1500 desenhará um deles na tela e o jogo continuará na linha 190.

Se os dez patos já foram desenhados, o jogo terminará e o score será exibido na porção inferior do vídeo. A linha 260 atualiza o recorde, que é impresso pela linha 270. Uma opção de jogar novamente é então oferecida ao jogador.

A linha 500 foi modificada para incluir um contador de tempo. A sub-rotina 700 desenha ou apaga o pato da tela, conforme a definição de cor atual.

As linhas de 1000 a 1100 verificam se

o tiro foi certo. As condições do IF da linha 1000 permitem uma certa margem de erro ao tiro. Se exigirmos que as coordenadas do pato coincidam com as da mira, o jogo se tornará muito difícil (lembre-se de que não conseguimos controlar totalmente o cursor em BASIC; além disso, a mira se move de oito em oito pontos).

As linhas 1020 e 1030 produzem uma série de vinte "cliques" no alto-falante, informando o sucesso do tiro. Um novo pato é gerado pela linha 1030. Os pontos ganhos são então calculados, com base no tempo gasto pelo jogador para abater a ave. O número de patos é diminuído e o contador de tempo volta a zero.

Se o tiro não atingir o pato, as linhas 1070 e 1080 emitirão três "cliques" e uma multa de cinco pontos será cobrada.

A sub-rotina 1500 apaga o pato de sua antiga posição — linha 1500 — e desenha a mira nesse lugar, calculando ao acaso uma nova posição.

O nível de dificuldade do jogo pode ser alterado, modificando-se o limite de tempo na linha 230. A contagem de pontos e a multa devem ser alteradas proporcionalmente, nas linhas 1040 e 1090.



Para o TK-2000, acrescente as linhas válidas para o Apple, com as seguintes modificações:

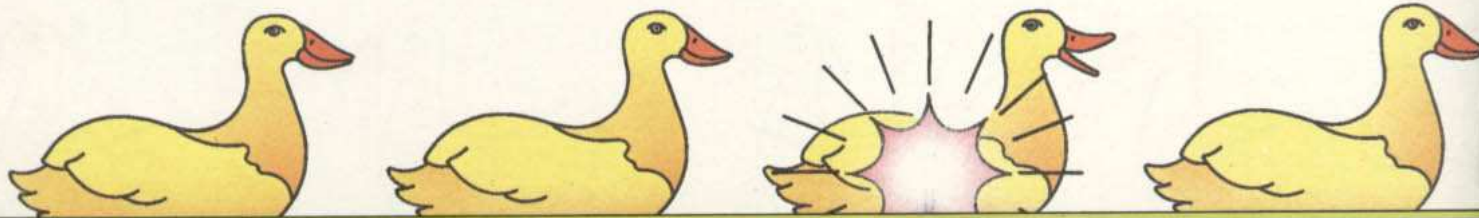
```
220 IF A = 46 OR A = 76 THEN
GOSUB 1000: IF D < 1 THEN 250
570 IF A = 46 OR A = 76 THEN
RETURN
```

Essas linhas verificam se um dos botões de tiro do joystick da microdigital foi pressionado. 46 e 76 são os códigos que os botões de tiro enviam ao micro.



Ao adicionar as próximas linhas, você terá o jogo completo.

```
80 FOR K=1536 TO 2272 STEP 32
90 READ A,B:POKE K,A:POKE K+1,B
100 NEXT
```




```

110 GET(0,0)-(13,11),D,G
120 GET(0,12)-(13,23),H,G
150 DX=RND(239)+1:DY=RND(178)+1
180 D=10:SC=0
190 TIMER=0
210 PUT(DX,DY)-(DX+13,DY+11),D,
OR
220 IF(PEEK(65280)AND1)=0 GOSU
B 2000:IF D<1 THEN 250
230 IF TIMER<200 THEN 200
240 D=D-1:IF D>0 GOSUB 3000:GOT
O 190
250 CLS:PRINT @139,"SCORE=";SC
260 IF SC>HI THEN HI=SC
270 PRINT @234,"RECORDE=";HI
280 PRINT @389,"QUER RECOMECAR
(S/N)?"
290 AS=INKEYS:IF AS<>"S" AND AS
<>"N" THEN 290
300 IF AS="S" THEN 130
310 END
2000 IF PPOINT(X,Y)=1 THEN 2070
2010 PUT(X-6,Y-5)-(X+7,Y+5),H,P
SET
2020 PLAY "T5004CEEEG"
2030 GOSUB 3000
2040 SC=SC+250-TIMER
2050 D=D-1:TIMER=0
2060 RETURN
2070 PLAY "T25001DDE"
2080 SC=SC-10
2090 RETURN
3000 PUT(DX,DY)-(DX+13,DY+11),B
,PSET
3010 PUT(X-8,Y-5)-(X+9,Y+5),S,P
SET
3020 DX=RND(239)+1:DY=RND(178)+
1:RETURN
4020 DATA 4,0,25,0,149,0,21,0,5
,0,5,64,5,80,21,148,22,148,22,8
4,21,84,5,80
4030 DATA 48,48,0,0,195,12,51,4
8,15,192,51,48,204,204,15,192,5
1,48,192,12,0,0,51,48

```

Dois desenhos estão contidos nas linhas **DATA** 4020 e 4030: o pato e o que restou dele após ser atingido pelo tiro. Os números são lidos e colocados na tela pelas linhas 80 a 100 com auxílio de comandos **POKE**. Dois comandos **GET** são usados para colocar o pato na matriz **D** e o efeito do tiro em **H**.

A linha 150 seleciona ao acaso uma posição para o pato, enquanto a linha 210 utiliza os valores de **DX** e **DY** para desenhá-lo na tela.

Os valores iniciais do número de pa-

tos, do score e do contador de tempo são estabelecidos pelas linhas 180 e 190.

Os tiros são detectados pela linha 220. Se o botão da direita for pressionado, o bit 0 do endereço 65280 da memória mudará de 1 para 0; se o botão pressionado for o da esquerda, o bit mudará de 0 para 1. A linha 220 mostra como verificar essas mudanças em 65280. Se utilizarmos os comandos **PEEK** e **AND1** e pressionarmos o botão da direita, teremos o valor zero. **PEEK** e **AND2** verificam o botão da esquerda. Tal como está, a linha 220 se refere ao botão da direita. Se ele for pressionado, o programa será desviado para a sub-rotina 2000, que verificará se o tiro acertou o alvo.

A parte final da linha 220 averigua se os dez patos já foram desenhados; se a resposta for positiva, o programa irá para a linha 250.

Uma vez pressionado o botão do joystick, a linha 2000 verifica a cor do ponto que está no centro da mira. Se a função **PPOINT** sugerir que a cor é verde, isso significa que o jogador errou o alvo, e o programa irá para a linha 2070, que produzirá um som de erro. Uma multa de dez pontos será cobrada e a sub-rotina terminará na linha 2090.

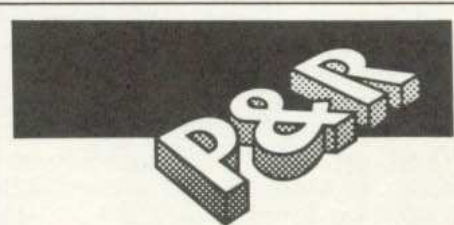
Se a cor no centro da mira não for verde, isso quer dizer que o tiro atingiu o alvo. Um desenho será colocado sobre o pato, e se ouvirá um som de sucesso. A seguir, é chamada a sub-rotina 3000. Sua função é apagar o que restou do pato e desenhar a mira no mesmo local. A linha 3020 calcula ao acaso a nova posição do pato.

A linha 2040 calcula o score. Este depende do tempo levado para acertar o pato. O número de aves é diminuído de um a cada exemplar abatido e o contador de tempo recomeça de zero.

Após completar a sub-rotina, o programa retorna à linha 230, onde é verificado o prazo para o pato ser atingido. Se a variável **TIMER** for menor que 200, o programa voltará à linha 200. Caso contrário, a linha 240 diminuirá o número de patos, verificando se sobrou algum. Em caso positivo, a sub-rotina 3000 apagará a ave restante e a linha 3020 desenhará um novo pato.

Se os patos tiverem acabado, o score e o recorde serão mostrados pelas linhas 250 a 270. Estas também atualizam o recorde.

As linhas 280 a 310 oferecem ao jogador a possibilidade de jogar novamente.



O que fazer para transformar o desenho do pato em um alvo diferente?

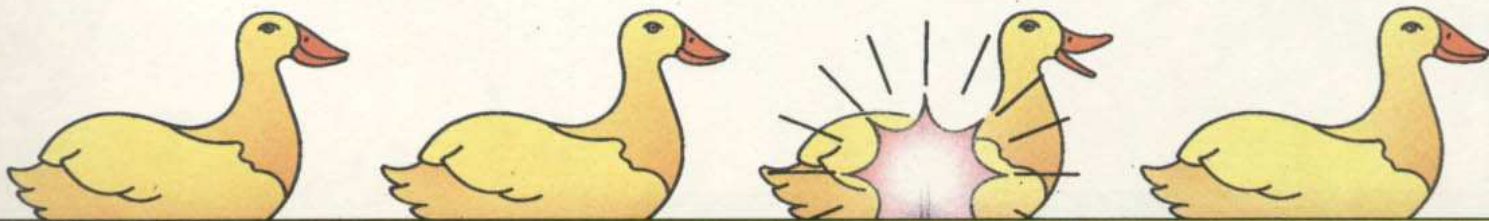
A resposta depende do computador que estamos usando. Basicamente, devemos alterar as linhas **DATA** e, talvez, as linhas que colocam caracteres (**UDG**) na tela.

O programa do Spectrum usa um conjunto de quatro **UDG** definidos nas linhas 1000 a 1030. Para conjuntos maiores, a linha 180, que coloca o desenho na tela, usando **PRINT**, deve ser modificada.

Usuários do MSX também precisarão mudar as linhas **DATA**. Se usarem mais de um sprite, o tamanho do **FOR** da linha 20 deverá ser alterado e novos **PUT SPRITE**, acrescentados. É possível também que sejam necessárias algumas modificações nos comandos que contêm coordenadas. O programa editor de sprites que fornecemos pode ser útil como ponto de referência.

A chave da modificação das linhas **DATA** do Apple e do TK-2000 está no programa editor de tabelas de figuras. A sub-rotina 700 deve ser alterada em caso de modificação do tamanho da figura. Neste caso, todos os comandos que lidam com coordenadas devem ser igualmente alterados.

Já no TRS-Color é muito simples criar caracteres **UDG** de qualquer tamanho. Mude as linhas **DATA** e os **FOR...NEXT** que as lêem. Talvez seja necessário mudar também as coordenadas dos comandos **PUT** e **GET**.



CONVERSÕES NO COMPUTADOR

Adotado pela maioria dos países do mundo ocidental, o sistema métrico decimal de pesos e medidas tem contribuído, desde o seu surgimento no século XVIII, para facilitar as relações comerciais e o intercâmbio cultural entre os povos. Os Estados Unidos e a Grã-Bretanha, porém, permanecem apegados a ele, preferindo empregar seus próprios sistemas. Essa divergência de critérios tem criado não poucos problemas técnicos, pois grande parte das máquinas e ferramentas em circulação no mercado mundial é produzida nesses dois países, exigindo constantes conversões de medidas, como, por exemplo, de polegadas para centímetros ou, então, de li-

bras para quilogramas.

O programa a seguir faz conversões desse tipo, incluindo até mesmo medidas menos freqüentes, como a de pressão em milímetros de mercúrio (abreviada para mmHG).

Inicialmente, digite o programa e execute-o. Um menu de opções aparecerá imediatamente. A primeira delas (SAIR) permite que você retorne ao BASIC, enquanto as outras se referem aos tipos de unidades que podem ser convertidas. As escolhas possíveis são: comprimento, área, volume, massa (ou peso), pressão e temperatura.

As opções são precedidas por números situados à sua esquerda. Para selecionar uma delas, pressione a tecla com o número correspondente. Para converter, por exemplo, unidades de comprimen-

Você tem dificuldade para descobrir quantas polegadas há em um metro ou quantos litros contém um galão? O programa conversor de INPUT resolve esse problema.

to, pressione a tecla "1". Não importa, por enquanto, se a conversão será do sistema métrico decimal para o sistema britânico ou vice-versa.

QUAIS UNIDADES

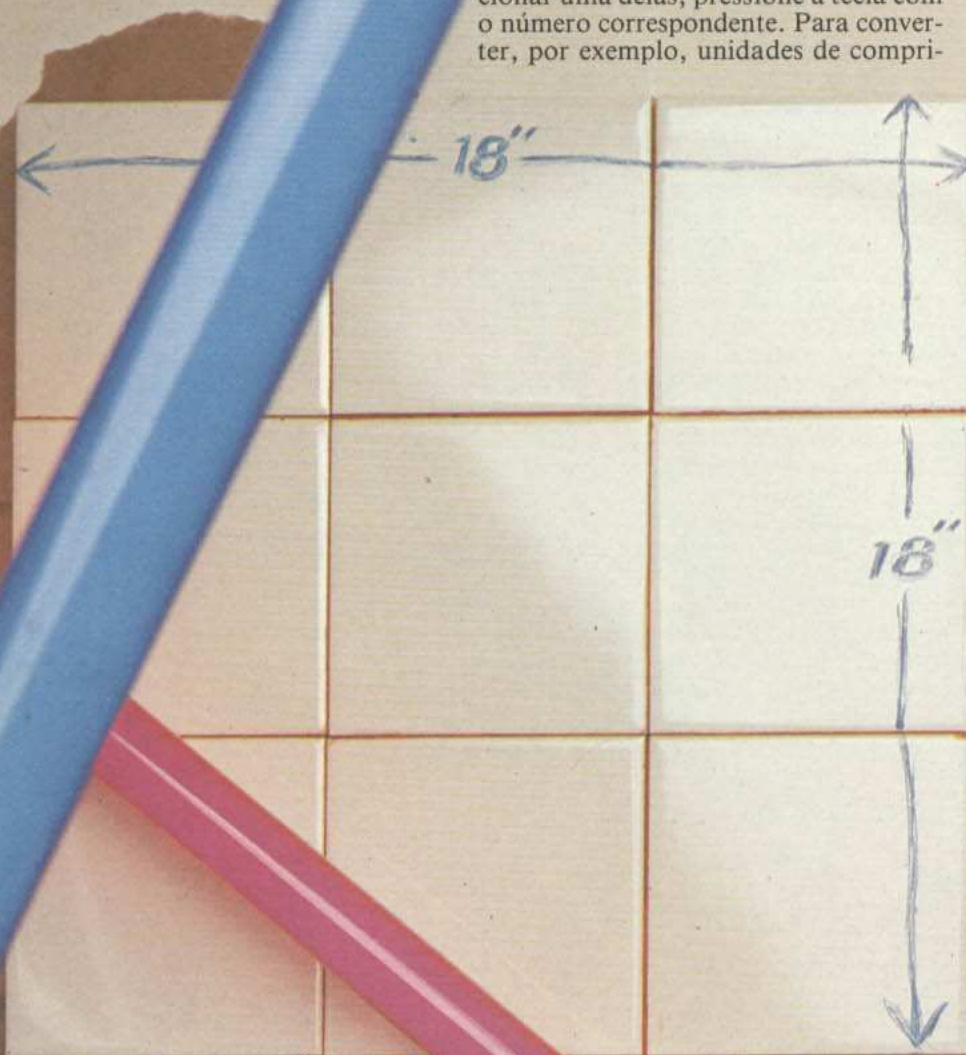
Depois de fazer sua escolha a respeito do gênero de conversão, pressione a tecla com o número correspondente. O computador mostrará então um segundo menu, que lhe permitirá definir a unidade a ser convertida.

Assim, se você pressionou a tecla "1" para converter unidades de comprimento, tem agora uma lista de todas as unidades possíveis (polegadas, pés, milímetros, centímetros, metros, etc.). Se quiser converter polegadas em unidades decimais, pressione "1" novamente e digite o número de polegadas. Não se esqueça de teclar <ENTER> ou <RETURN> depois disso.

ENCONTRE AS RESPOSTAS

O computador faz as conversões e a seguir mostra os resultados em todas as unidades de um dos sistemas de medidas (métrico decimal ou britânico). Desta forma, se você digitar um valor em polegadas, terá a resposta em milímetros, centímetros, metros e quilômetros.

Depois de mostrar todos os valores, o computador esperará até que uma tecla seja pressionada para prosseguir. Se ela for <ENTER> ou <RETURN>, o programa voltará a mostrar o menu principal. Se você quiser continuar a converter o mesmo tipo de unidade, qualquer outra tecla o levará ao menu anterior.



- CONVERSÃO DO SISTEMA MÉTRICO PARA O SISTEMA BRITÂNICO E VICE-VERSA
- COMO USAR O PROGRAMA
- COMO CONVERTER VALORES DE

- ÁREA, PESO, COMPRIMENTO, VOLUME, PRESSÃO E TEMPERATURA
- CONVERSÃO DE UNIDADES MISTAS E FRAÇÕES

Para interromper o programa, retorne ao menu principal e tecla "O" para a opção SAIR.

UNIDADES MISTAS

É possível ainda que você precise converter o valor de uma medida britânica em que haja unidades e subunidades. Tomemos, por exemplo, dois pés e seis polegadas. Há duas maneiras de resolver o problema.

O programa aceita números que não sejam inteiros; assim, se você souber quantas polegadas há em um pé, pode calcular a fração decimal correspondente e digitá-la no computador. Neste caso não há dificuldade, visto que seis polegadas são exatamente meio pé. Assim, você digitaria 2.5.

O segundo método (mais empregado no caso de frações de cálculo difícil) consiste em converter o valor em duas etapas. Dessa forma, passe para o sistema decimal, em primeiro lugar, o valor expresso em pés; converta depois o valor em polegadas. Em seguida, tudo o que você tem a fazer é somar os dois resultados (lembre-se de somar sempre valores da mesma unidade decimal).

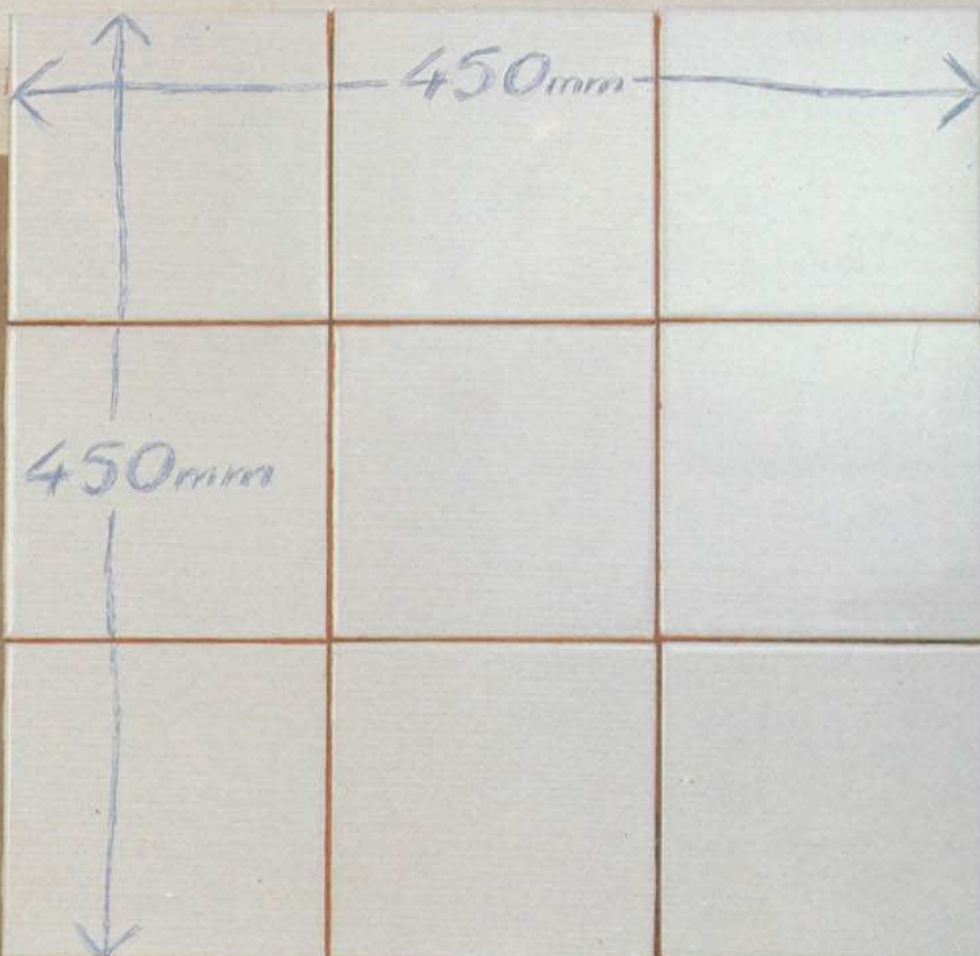
Se ocorrer o contrário, ou seja, se o número não inteiro for do sistema decimal, calcule a fração conveniente e coloque-a no computador.



```
10 DIM L(7), LS(7), A(6), AS(6), V(6), VS(6), M(5), MS(5), P(4), PS(4)
20 FOR K=0 TO 7:READ L(K),LS(K):NEXT
30 DATA 1,POLEGADAS,12,PES,36,JARDDAS,63360,MILHAS,.03937,MILIMETROS,.3937,CENTIMETROS,39.37,M
```

```
ETROS,39370,KILOMETROS
40 FOR K=0 TO 6:READ A(K),AS(K):NEXT
50 DATA 1,POLEGADAS QUADRADAS,144,PES QUADRADOS,6272640,ACRES,4.0145E9,MILHAS QUADRADAS,.155,CENTIMETROS QUADRADOS,1550,METROS QUADRADOS,1.55E7,HECTARES
60 FOR K=0 TO 6:READ V(K),VS(K):NEXT
70 DATA 1,POLEGADAS CUBICAS,1728,PES CUBICOS,34.67,PINTAS,277.36,GALOES,.06102,CENTIMETROS CUBICOS,61.024,LITROS,61024,METROS CUBICOS
80 FOR K=0 TO 5:READ M(K),MS(K):NEXT
90 DATA 1,ONCAS,16,LIBRAS,35840,TON. INGLESAS,.03527,GRAMAS,35.27,KILOGRAMAS,35270,TON. METRICAS
100 FOR K=0 TO 4:READ P(K),PS(K):NEXT
```

```
110 DATA 1,PSI,51.73,MM HG,6895,N/METRO QUADRADO,.0681,ATMOSFERAS,68.95,MILIBARES
120 CLS:PRINT"QUAL CATEGORIA (0-6)?"
130 PRINT @72,"0- SAIDA":PRINT @136,"1- COMPRIMENTO":PRINT @200,"2- AREA":PRINT @264,"3- VOLUME":PRINT @328,"4- PESO":PRINT @392,"5- PRESSAO":PRINT @456,"6- TEMPERATURA"
140 AS=INKEYS:IF AS<"0" OR AS>"6" THEN 140
150 IF AS="0" THEN CLS:END
160 CLS:ON VAL(AS) GOSUB 1000,1500,2000,2500,3000,3500
170 GOTO 120
1000 PRINT @9,"comprimento":PRINT @34,"SELECIONE UNIDADE ORIGINAL":FOR K=0 TO 7:PRINT @136+32*K,K+1;"- ";LS(K):NEXT
1010 BS=INKEYS:IF BS<"1" OR BS>"8" THEN 1010
```




```

1020 B=VAL(B$)-1:CLS:PRINT" INT
RODUZA O NUMERO DE ";L$(B)
1030 INPUT VL
1040 CLS:PRINT VL;L$(B);" EQUIV
ALE A "
1050 IF B>3 THEN 1080
1060 FOR K=0 TO 3:PRINT @96+K*6
4,VL*L(B)/L(K+4),L$(K+4):NEXT
1070 GOTO 1090
1080 FOR K=0 TO 3:PRINT @96+K*6
4,VL*L(B)/L(K),L$(K):NEXT
1090 AS=INKEY$:IF AS="" THEN 10
90
1100 IF AS=CHR$(13) THEN RETURN
ELSE CLS:GOTO 1000
1500 PRINT @13,"area":PRINT @35
,"SELECIONE UNIDADE ORIGINAL":F
OR K=0 TO 6:PRINT @131+32*K,K+1
;"- ";AS(K):NEXT
1510 BS=INKEY$:IF BS<"1" OR BS>
"7" THEN 1510
1520 B=VAL(B$)-1:CLS:PRINT" INT
RODUZA O NUMERO DE ";AS(B)
1530 INPUT VL
1540 CLS:PRINT VL;AS(B);" EQUIV
ALE A "
1550 IF B>3 THEN 1580
1560 FOR K=0 TO 2:PRINT @96+K*6
4,VL*A(B)/A(K+4),AS(K+4):NEXT
1570 GOTO 1590
1580 FOR K=0 TO 3:PRINT @96+K*6
4,VL*A(B)/A(K),AS(K):NEXT
1590 AS=INKEY$:IF AS="" THEN 15
90
1600 IF AS=CHR$(13) THEN RETURN
ELSE CLS:GOTO 1500
2000 PRINT @12,"volume":PRINT @
35,"SELECIONE UNIDADE ORIGINAL"
:FOR K=0 TO 6:PRINT @133+32*K,K
+1;"- ";VS(K):NEXT
2010 BS=INKEY$:IF BS<"1" OR BS>
"7" THEN 2010
2020 B=VAL(B$)-1:CLS:PRINT" INT
RODUZA O NUMERO DE ";VS(B)
2030 INPUT VL
2040 CLS:PRINT VL;VS(B);" EQUIV
ALE A "
2050 IF B>3 THEN 2080
2060 FOR K=0 TO 2:PRINT @96+K*6
4,VL*V(B)/V(K+4),VS(K+4):NEXT
2070 GOTO 2090
2080 FOR K=0 TO 3:PRINT @96+K*6
4,VL*V(B)/V(K),VS(K):NEXT
2090 AS=INKEY$:IF AS="" THEN 20
90
2100 IF AS=CHR$(13) THEN RETURN
ELSE CLS:GOTO 2000
2500 PRINT @13,"peso":PRINT @35
,"SELECIONE UNIDADE ORIGINAL":F
OR K=0 TO 5:PRINT @136+32*K,K+1
;"- ";MS(K):NEXT
2510 BS=INKEY$:IF BS<"1" OR BS>
"6" THEN 2510
2520 B=VAL(B$)-1:CLS:PRINT" INT
RODUZA O NUMERO DE ";MS(B)
2530 INPUT VL
2540 CLS:PRINT VL;MS(B);" EQUIV
ALE A "
2550 IF B>2 THEN 2580
2560 FOR K=0 TO 2:PRINT @96+K*6
4,VL*M(B)/M(K+3),MS(K+3):NEXT
2570 GOTO 2590
2580 FOR K=0 TO 2:PRINT @96+K*6
4,VL*M(B)/M(K),MS(K):NEXT
2590 AS=INKEY$:IF AS="" THEN 25
90
2600 IF AS=CHR$(13) THEN RETURN
ELSE CLS:GOTO 2500
3000 PRINT @11,"pressao":PRINT
@36,"SELECIONE UNIDADE ORIGINAL"
:FOR K=0 TO 4:PRINT @134+32*K,
K+1;"- ";PS(K):NEXT
3010 BS=INKEY$:IF BS<"1" OR BS>
"5" THEN 3010
3020 B=VAL(B$)-1:CLS:PRINT" INT
RODUZA O NUMERO DE ";PS(B)
3030 INPUT VL
3040 CLS:PRINT VL;PS(B);" EQUIV
ALE A "
3050 T=0:FOR K=0 TO 4:IF K=B TH
EN 3070
3060 PRINT @96+T*64,VL*P(K)/P(B
),PS(K):T=T+1
3070 NEXT
3080 AS=INKEY$:IF AS="" THEN 30
80
3090 IF AS=CHR$(13) THEN RETURN
ELSE CLS:GOTO 3000
3500 CLS:PRINT @10,"temperatura"
:PRINT @37,"SELECIONE CONVERSA
O : ":PRINT" CENTIGRADOS PARA F
AHRENHEIT (C) FAHRENHEIT PARA C
ENTIGRADOS (F)"
3510 BS=INKEY$:IF BS<>"C" AND B
S<>"F" THEN 3510
3520 IF BS="C" THEN 3560
3530 PRINT" INTRODUZA GRAUS FAH
RENHEIT":INPUT VL
3540 CLS:PRINT @33,VL;"GRAUS FA
HRENHEIT EQUIVALEM A"
3550 PRINT @97,(VL-32)*5/9;"GRA
US CENTIGRADOS":GOTO 3590
3560 PRINT" INTRODUZA GRAUS CEN
TIGRADOS":INPUT VL
3570 CLS:PRINT @33,VL;"GRAUS CE
NTIGRADOS EQUIVALEM A"
3580 PRINT @97,32+VL*9/5;"GRAUS
FAHRENHEIT"
3590 AS=INKEY$:IF AS="" THEN 35
90
3600 IF AS=CHR$(13) THEN RETURN
ELSE CLS:GOTO 3500
5 POKE 23658,8
10 DIM L(8): DIM L$(8,12):
DIM A(7): DIM AS(7,16): DIM V
(7): DIM VS(7,16): DIM M(6):
DIM V(7): DIM VS(7,12): DIM M
(6): DIM MS(6,12): DIM P(5):
DIM PS(5,17)
20 FOR K=1 TO 8: READ L(K),L$(
K): NEXT K
30 DATA 1,"POLEGADAS",12,"PES
",36,"JARDAS",63360,"MILHAS",
.03937,"MILIMETROS",.3937,"CE
NTIMETROS",39.37,"METROS",
39370,"QUILOMETROS"
40 FOR K=1 TO 7: READ A(K),AS
(K): NEXT K
50 DATA 1,"POL.QUADRADAS",144
,"PES QUADRADOS",6272540,"ACR
ES",4.0145E9,"MILHAS QUADRADA
S",.155,"CENT.QUADRADOS",
1550,"METROS QUADRADOS",
1.55E7,"HECTARES"
60 FOR K=1 TO 7: READ V(K),VS
(K): NEXT K
70 DATA 1,"POL.CUBICAS",1728,
"PES CUBICOS",34.67,"PINTS",
277.36,"GALOES",.06102,"CENT.
CUBICOS",61.024,"LITROS",
61024,"METROS CUBICOS"
80 FOR K=1 TO 6: READ M(K),MS
(K): NEXT K
90 DATA 1,"ONCAS",16,"LIBRAS"
,35840,"TON.INGLESAS",.03527,
"GRAMAS",35.27,"QUILOGRAMAS",
35270,"TON.METRICAS"
100 FOR K=1 TO 5: READ P(K),PS
(K): NEXT K
110 DATA 1,"LIBRA/POL.QUADR.",
51.73,"mmHG",6895,"N/METRO QUA
DR.",.0681,"ATMOSFERAS",68.95,
"MILIBARES"
120 CLS : PRINT INVERSE 1;"
TAB 5;"QUE CATEGORIA (0 A 6)?"
;TAB 31;" "
130 PRINT AT 6,8;"0- SAIDA";AT
8,8;"1- COMPRIMENTO";AT 10,8;"
2- AREA";AT 12,8;"3- VOLUME";
AT 14,8;"4- PESO";AT 16,8;"5-
PRESSAO";AT 18,8;"6- TEMPERATU
RA"
140 LET Z$=INKEY$: IF Z$<"0"
OR Z$>"6" THEN GOTO 140
150 IF Z$="0" THEN CLS :
STOP
160 CLS : GOSUB 1000+(VAL Z$-1
)*500
170 GOTO 120
1000 PRINT INVERSE 1;AT 0,12;"
COMPRIMENTO ": PRINT AT 2,6;"S
ELECIONE UNIDADE ORIGINAL": PRI
NT : FOR K=1 TO 8: PRINT 'TAB 1
0;K;"- ";L$(K): NEXT K
1010 LET BS=INKEY$: IF BS<"1" O
R BS>"8" THEN GOTO 1010
1020 LET B=VAL BS: INPUT "INTRO
DUZA NUMERO DE ";(L$(B)),VL
1040 CLS : PRINT AT 2,4;VL;" ";
L$(B);" EQUIVALE A "
1050 IF B>4 THEN GOTO 1080
1060 FOR K=1 TO 4: PRINT AT K*2
+4,3;VL*L(B)/L(K+4);TAB 18;L$(K
+4): NEXT K
1070 GOTO 1090
1080 FOR K=1 TO 4: PRINT AT K*2
+4,3;VL*L(B)/L(K);TAB 18;L$(K):
NEXT K
1090 LET Z$=INKEY$: IF Z$="" TH
EN GOTO 1090
1100 IF Z$=CHR$ 13 THEN RETURN
1110 CLS : GOTO 1000
1500 PRINT INVERSE 1;AT 0,13;"
AREA ": PRINT AT 2,6;"SELECION
E UNIDADE ORIGINAL": PRINT : FO
R K=1 TO 7: PRINT 'TAB 10;K;"-
";AS(K): NEXT K
1510 LET BS=INKEY$: IF BS<"1" O
R BS>"7" THEN GOTO 1510
1520 LET B=VAL BS: INPUT "INTRO
DUZA O NUMERO DE ";(AS(B)),VL
1540 CLS : PRINT AT 2,4;VL;" ";
AS(B);" EQUIVALE A "
1550 IF B>4 THEN GOTO 1580
1560 FOR K=1 TO 3: PRINT AT K*2

```




1pt/0.56 litre

```
+4,3;VL*A(B)/A(K+4);TAB 18;A$(K
+4): NEXT K
1570 GOTO 1590
1580 FOR K=1 TO 4: PRINT AT K*2
+4,3;VL*A(B);TAB 18;A$(K): NEXT
K
1590 LET Z$=INKEY$: IF Z$="" TH
EN GOTO 1590
1600 IF Z$=CHR$ 13 THEN RETURN
1610 CLS : GOTO 1500
2000 PRINT INVERSE 1;AT 0,12;"
VOLUME ": PRINT AT 2,6;"SELECI
ONE UNIDADE ORIGINAL": PRINT :
FOR K=1 TO 7: PRINT 'TAB 10;K;"
- ";V$(K): NEXT K
2010 LET B$=INKEY$: IF B$<"1" O
R B$>"7" THEN GOTO 2010
2020 LET B=VAL B$: INPUT "INTRO
DUZA O NUMERO DE ";(V$(B)),VL
2040 CLS : PRINT AT 2,4;VL;" ";
V$(B);" EQUIVALE A "
2050 IF B>4 THEN GOTO 2080
2060 FOR K=1 TO 3: PRINT AT K*2
+4,3;VL*V(B)/V(K+4);TAB 18;V$(K
+4): NEXT K
2070 GOTO 2090
2080 FOR K=1 TO 4: PRINT AT K*2
+4,3;VL*V(B)/V(K);TAB 18;V$(K):
NEXT K
2090 LET Z$=INKEY$: IF Z$="" TH
EN GOTO 2090
```

```
2100 IF Z$=CHR$ 13 THEN RETURN
2110 CLS : GOTO 2000
2500 PRINT INVERSE 1;AT 0,13;"
PESO ": PRINT AT 2,6;"SELECI
E UNIDADE ORIGINAL": PRINT : FO
R K=1 TO 6: PRINT 'TAB 10;K;"-
";M$(K): NEXT K
2510 LET B$=INKEY$: IF B$<"1" O
R B$>"6" THEN GOTO 2510
2520 LET B=VAL B$: INPUT "INTRO
DUZA O NUMERO DE ";(M$(B)),VL
2540 CLS : PRINT AT 2,4;VL;" ";
M$(B);" EQUIVALE A "
2550 IF B>3 THEN GOTO 2580
2560 FOR K=1 TO 3: PRINT AT K*2
+4,3;VL*M(B)/M(K+3);TAB 18;M$(K
+3): NEXT K
2570 GOTO 2590
2580 FOR K=1 TO 3: PRINT AT K*2
+4,3;VL*M(B)/M(K);TAB 18;M$(K):
NEXT K
2590 LET Z$=INKEY$: IF Z$="" TH
EN GOTO 2590
2600 IF Z$=CHR$ 13 THEN RETURN
2610 CLS : GOTO 2500
3000 PRINT INVERSE 1;AT 0,11;"
PRESSAO ": PRINT AT 2,6;"SELEC
IONE UNIDADE ORIGINAL": PRINT :
FOR K=1 TO 5: PRINT 'TAB 10;K;
- -";P$(K): NEXT K
```

```
3010 LET B$=INKEY$: IF B$<"1" O
R B$>"5" THEN GOTO 3010
3020 LET B=VAL B$: INPUT "INTRO
DUZA O NUMERO DE ";(P$(B)),VL
3040 CLS : PRINT AT 2,4;VL;" ";
P$(B);" EQUIVALE A "
3050 LET T=0: FOR K=1 TO 5: IF
K=B THEN GOTO 3070
3060 PRINT AT K*2+4,3;VL*P(K)/P
(B);TAB 18;P$(K): LET T=T+1
3070 NEXT K
3080 LET Z$=INKEY$: IF Z$="" TH
EN GOTO 3080
3090 IF Z$=CHR$ 13 THEN RETURN
3100 CLS : GOTO 3000
3500 PRINT INVERSE 1;AT 0,9;"
TEMPERATURA ": PRINT AT 3,11;"C
ONVERSAO:" : PRINT ' " CENTIGRADO
PARA FAHRENHEIT (C) FAHRENHEI
T PARA CENTIGRADO (F)"
3510 LET B$=INKEY$: IF B$<>"C"
AND B$<>"F" THEN GOTO 3510
3520 IF B$="C" THEN GOTO 3560
3530 INPUT "INTRODUZA GRAUS FAH
RENHEIT",VL
3540 CLS : PRINT AT 1,2;VL;" GR
AUS FAHRENHEIT EQUIVALEM A "
3550 PRINT 'TAB 2;(VL-32)*5/9;"
GRAUS CENTIGRADOS ": GOTO 3590
3560 INPUT "INTRODUZA GRAUS CEN
```



```

TIGRADOS",VL
3570 CLS : PRINT AT 1,2;VL;" GR
AUS CENTIGRADOS EQUIVALEM A "
3580 PRINT 'TAB 2;32+VL*9/5;" G
RAUS FAHRENHEIT"
3590 PAUSE 0: LET Z$=INKEY$: IF
Z$="" THEN GOTO 3590
3600 IF Z$=CHR$ 13 THEN RETURN
3610 CLS : GOTO 3500

```



```

10 DIM L(7),LS(7),A(6),AS(6),V
(6),VS(6),M(5),MS(5),P(4),PS(4)

```

```

20 FOR K = 0 TO 7: READ L(K),L
S(K): NEXT

```

```

30 DATA 1,POLEGADAS,12,PES,36
,JARDAS,63360,MILHAS,.03937,MIL
IMETROS,.3937,CENTIMETROS,39.37
,METROS,39370,KILOMETROS

```

```

40 FOR K = 0 TO 6: READ A(K),A
S(K): NEXT

```

```

50 DATA 1,POLEGADAS QUADRADAS
,144,PES QUADRADOS,6272640,ACRE
S,4.0145E9,MILHAS QUADRADAS,.15
5

```

```

55 DATA CENTIMETROS QUADRADO
S,1550,METROS QUADRADOS,1.55E7,
HECTARES

```

```

60 FOR K = 0 TO 6: READ V(K),V
S(K): NEXT

```

```

70 DATA 1,POLEGADAS CUBICAS,1
728,PES CUBICOS,34.67,PINTAS,27
7.36,GALOES

```

```

75 DATA .06102,CENTIMETROS CU
BICOS,61.024,LITROS,61024,METRO
S CUBICOS

```

```

80 FOR K = 0 TO 5: READ M(K),M
S(K): NEXT

```

```

90 DATA 1,ONCAS,16,LIBRAS,358
40,TONS,.03527,GRAMAS,35.27,KIL
OGRAMAS,35270,TONELADAS

```

```

100 FOR K = 0 TO 4: READ P(K),
PS(K): NEXT

```

```

110 DATA 1,LIBRAS/POL2,51.73,
MMHG,6895,NEWTONS/M2,.0681,ATMO
SFERAS,68.95,MILIBARES

```

```

120 HOME : HTAB 10: VTAB 5: PR
INT "QUAL CATEGORIA? (0-6)"

```

```

130 VTAB 10: HTAB 10: PRINT "0
- SAIR": HTAB 10: PRINT "1 - C
OMPRIMENTO": HTAB 10: PRINT "2
- AREA"

```

```

140 HTAB 10: PRINT "3 - VOLUME
": HTAB 10: PRINT "4 - PESO": H
TAB 10: PRINT "5 - PRESSAO"

```

```

150 HTAB 10: PRINT "6 - TEMPER
ATURA"

```

```

160 GET AS: IF AS < "0" OR AS
> "6" THEN 160

```

```

170 HOME : IF AS = "0" THEN E
ND

```

```

180 ON VAL (AS) GOSUB 1000,15
00,2000,2500,3000,3500

```

```

190 GOTO 120

```

```

1000 INVERSE : PRINT SPC( 39)
:: HTAB 14: PRINT "COMPRIMENTO"
: NORMAL

```

```

1010 PRINT : HTAB 5: PRINT "ES
COLHA A UNIDADE": PRINT

```

```

1020 FOR K = 0 TO 7: PRINT : H

```



```

TAB 5: PRINT K + 1;"- ";LS(K):
NEXT

```

```

1030 GET BS: IF BS < "1" OR BS
> "8" THEN 1030

```

```

1040 B = VAL (BS) - 1: HOME :
PRINT "NUMERO DE ";LS(B);"?"

```

```

1050 INPUT VL

```

```

1060 HOME : PRINT VL; CHR$ (32)
);LS(B);" EQUIVALEM A:"

```

```

1070 IF B > 3 THEN 1100

```

```

1080 VTAB 10: FOR K = 0 TO 3:
PRINT : PRINT VL * L(B) / L(K) +
4);: HTAB 20: PRINT LS(K + 4):
NEXT

```

```

1090 GOTO 1110

```

```

1100 VTAB 10: FOR K = 0 TO 3:
PRINT : PRINT VL * L(B) / L(K);
: HTAB 20: PRINT LS(K): NEXT

```

```

1110 GET AS: IF AS < > CHR$
(13) THEN HOME : GOTO 1000

```

```

1120 RETURN

```

```

1500 INVERSE : PRINT SPC( 39)
:: HTAB 18: PRINT "AREA": NORMA
L

```

```

1510 PRINT : HTAB 5: PRINT "ES

```

```

COLHA A UNIDADE": PRINT

```

```

1520 FOR K = 0 TO 6: PRINT : H
TAB 5: PRINT K + 1;"- ";AS(K):
NEXT

```

```

1530 GET BS: IF BS < "1" OR BS
> "7" THEN 1530

```

```

1540 B = VAL (BS) - 1: HOME :
PRINT "NUMERO DE ";AS(B);"?"

```

```

1550 INPUT VL

```

```

1560 HOME : PRINT VL; CHR$ (32)
);AS(B);" EQUIVALEM A:"

```

```

1570 IF B > 3 THEN 1600

```

```

1580 VTAB 10: FOR K = 0 TO 2:
PRINT : PRINT VL * A(B) / A(K) +
4);: HTAB 19: PRINT AS(K + 4):
NEXT

```

```

1590 GOTO 1610

```

```

1600 VTAB 10: FOR K = 0 TO 3:
PRINT : PRINT VL * A(B) / A(K);
: HTAB 19: PRINT AS(K): NEXT

```

```

1610 GET AS: IF AS < > CHR$
(13) THEN HOME : GOTO 1500

```

```

1620 RETURN

```

```

2000 INVERSE : PRINT SPC( 39)
:: HTAB 17: PRINT "VOLUME": NOR

```


19lb/8.61 Kg

```

MAL
2010 PRINT : HTAB 5: PRINT "ES
COLHA A UNIDADE": PRINT
2020 FOR K = 0 TO 6: PRINT : H
TAB 5: PRINT K + 1; "- "; VS(K):
NEXT
2030 GET BS: IF BS < "1" OR BS
> "7" THEN 2030
2040 B = VAL (BS) - 1: HOME :
PRINT "NUMERO DE "; VS(B); "?"
2050 INPUT VL
2060 HOME : PRINT VL; CHRS (32
); VS(B); " EQUIVALEM A:"
2070 IF B > 3 THEN 2100
2080 VTAB 10: FOR K = 0 TO 2:
PRINT : PRINT VL * V(B) / V(K +
4);: HTAB 19: PRINT VS(K + 4):
NEXT
2090 GOTO 2110
2100 VTAB 10: FOR K = 0 TO 3:
PRINT : PRINT VL * V(B) / V(K):
: HTAB 19: PRINT VS(K): NEXT
2110 GET AS: IF AS < > CHRS
(13) THEN HOME : GOTO 2000
2120 RETURN
2500 INVERSE : PRINT SPC( 39)
;: HTAB 18: PRINT "PESO": NORMA
L
2510 PRINT : HTAB 5: PRINT "ES
COLHA A UNIDADE": PRINT
2520 FOR K = 0 TO 5: PRINT : H
TAB 5: PRINT K + 1; "- "; MS(K):
NEXT
2530 GET BS: IF BS < "1" OR BS
> "6" THEN 2530
2540 B = VAL (BS) - 1: HOME :
PRINT "NUMERO DE "; MS(B); "?"
2550 INPUT VL
2560 HOME : PRINT VL; CHRS (32
); MS(B); " EQUIVALEM A:"
2570 IF B > 2 THEN 2600
2580 VTAB 10: FOR K = 0 TO 2:
PRINT : PRINT VL * M(B) / M(K +
3);: HTAB 19: PRINT MS(K + 3):
NEXT
2590 GOTO 2610
2600 VTAB 10: FOR K = 0 TO 3:
PRINT : PRINT VL * M(B) / M(K):
: HTAB 19: PRINT MS(K): NEXT
2610 GET AS: IF AS < > CHRS
(13) THEN HOME : GOTO 2500
2620 RETURN
3000 INVERSE : PRINT SPC( 39)
;: HTAB 16: PRINT "PRESSAO": NO
RMAL
3010 PRINT : HTAB 5: PRINT "ES
COLHA A UNIDADE": PRINT
3020 FOR K = 0 TO 4: PRINT : H
TAB 5: PRINT K + 1; "- "; PS(K):
NEXT
3030 GET BS: IF BS < "1" OR BS
> "5" THEN 3030
3040 B = VAL (BS) - 1: HOME :
PRINT "NUMERO DE "; PS(B); "?"
3050 INPUT VL
3060 HOME : PRINT VL; CHRS (32
); PS(B); " EQUIVALEM A:"
3070 T = 0: VTAB 10: FOR K = 0
TO 4: IF K = B THEN 3100
3090 PRINT : PRINT VL * P(K) /
P(B);: HTAB 19: PRINT PS(K): T
= T + 1
3100 NEXT

```



```

3110 GET A$: IF A$ < > CHR$(13) THEN HOME : GOTO 3000
3120 RETURN
3500 INVERSE : PRINT SPC(39)
;: HTAB 14: PRINT "TEMPERATURA"
: NORMAL
3510 PRINT : HTAB 5: PRINT "CONVERTER:" : PRINT
3520 PRINT : HTAB 5: PRINT "CENTIGRADO PARA FARENHEIT (C)"
3530 PRINT : HTAB 5: PRINT "FARENHEIT PARA CENTIGRADO (F)"
3540 GET B$: IF B$ < > "C" AND B$ < > "F" THEN 3540
3550 IF B$ = "C" THEN 3590
3560 HOME : INPUT "GRAUS FARENHEIT? "; VL
3570 PRINT : PRINT VL; " GRAUS FARENHEIT EQUIVALEM A "; (VL - 32) * 5 / 9; " GRAUS CENTIGRADOS." : GOTO 3620
3590 HOME : INPUT "GRAUS CENTIGRADOS? "; VL
3600 PRINT : PRINT VL; " GRAUS CENTIGRADOS EQUIVALEM A "; 32 + VL * 9 / 5; " GRAUS FARENHEIT."
3620 GET A$: IF A$ < > CHR$(13) THEN HOME : GOTO 3500
3630 RETURN

```



```

5 DEFSNG A
10 DIM L(7), L$(7), A(6), A$(6), V(6), V$(6), M(5), M$(5), P(4), P$(4)
20 FORK=0TO7: READ L(K), L$(K): NEXT
30 DATA 1, polegadas, 12, pés, 36, jardas, 63360, milhas, .03937, milímetros, .3937, centímetros, 39.37, metros, 39370, quilômetros
40 FORK=0TO6: READ A(K), A$(K): NEXT
50 DATA 1, polegadas quadradas, 144, pés quadrados, 6272640, acres, 4.0145e9, milhas quadradas, .155, centímetros quadrados, 1550, metros quadrados, 1.55e7, hectares
60 FORK=0TO6: READ V(K), V$(K): NEXT
70 DATA 1, polegadas cúbicas, 1728, pés cúbicos, 34.67, pintas, 277.36, galões, .06102, centímetros cúbicos, 61.024, litros, 61024, metros cúbicos
80 FORK=0TO5: READ M(K), M$(K): NEXT
90 DATA 1, onças, 16, libras, 35840, tons, .03527, gramas, 35.27, quilogramas, 35270, toneladas
100 FORK=0TO4: READ P(K), P$(K): NEXT
110 DATA 1, libras/pol, .51.73, mm Hg, 6895, newtons/m, .0681, atmosferas, 68.95, milibares
120 CLS: COLOR 15, 4, 4: PRINT "Qual categoria? (0-6)"
130 LOCATE 5, 5: PRINT "0 - Sair": LOCATE 5, 7: PRINT "1 - Comprimento": LOCATE 5, 9: PRINT "2 - Área"
135 LOCATE 5, 11: PRINT "3 - Volume": LOCATE 5, 13: PRINT "4 - Peso": LOCATE 5, 15: PRINT "5 - Pressão":

```

```

LOCATE 5, 17: PRINT "6 - Temperatura"
140 A$=INKEY$: IF A$ < "0" OR A$ > "6" THEN 140
150 IF A$ = "0" THEN CLS: END
160 CLS: ON VAL(A$) GOSUB 1000, 1500, 2000, 2500, 3000, 3500
170 GOTO 120
1000 COLOR 15, 6, 6: PRINT "Comprimeto": PRINTSTRING$(39, 195): LOCATE 5, 3: PRINT "Selecione a unidade:"
1010 FORK=0TO7: LOCATE 5, 6+2*K: PRINTK+1; "- "; L$(K): NEXT
1020 B$=INKEY$: IF B$ < "1" OR B$ > "8" THEN 1020
1030 B=VAL(B$)-1: CLS: PRINT "Digite o número de "; L$(B)
1040 INPUT VL
1050 CLS: PRINTVL; CHR$(32); L$(B); " equivalem a:"
1060 IF B > 3 THEN 1090
1070 FORK=0TO3: LOCATE 1, 6+2*K: A=VL*L(B)/L(K+4): PRINTA; TAB(14); L$(K+4): NEXT
1080 GOTO 1100
1090 FORK=0TO3: LOCATE 1, 6+2*K: A=VL*L(B)/L(K): PRINTA; TAB(14); L$(K): NEXT
1100 A$=INKEY$: IF A$ = "" THEN 1100
1110 IF A$ = CHR$(13) THEN RETURN ELSE CLS: GOTO 1000
1500 COLOR 15, 10, 10: PRINT "Área": PRINTSTRING$(39, 195): LOCATE 5, 3: PRINT "Selecione a unidade:"
1510 FORK=0TO6: LOCATE 5, 6+2*K: PRINTK+1; "- "; A$(K): NEXT
1520 B$=INKEY$: IF B$ < "1" OR B$ > "7" THEN 1520
1530 B=VAL(B$)-1: CLS: PRINT "Digite o número de "; A$(B)
1540 INPUT VL
1550 CLS: PRINTVL; CHR$(32); A$(B); " equivalem a:"
1560 IF B > 3 THEN 1590
1570 FORK=0TO2: LOCATE 1, 6+2*K: A=VL*A(B)/A(K+4): PRINTA; TAB(14); A$(K+4): NEXT
1580 GOTO 1600
1590 FORK=0TO3: LOCATE 1, 6+2*K: A=VL*A(B)/A(K): PRINTA; TAB(14); A$(K): NEXT
1600 A$=INKEY$: IF A$ = "" THEN 1600
1610 IF A$ = CHR$(13) THEN RETURN ELSE CLS: GOTO 1500
2000 COLOR 15, 12, 12: PRINT "Volume": PRINTSTRING$(39, 195): LOCATE 5, 3: PRINT "Selecione a unidade:"
2010 FORK=0TO6: LOCATE 5, 6+2*K: PRINTK+1; "- "; V$(K): NEXT
2020 B$=INKEY$: IF B$ < "1" OR B$ > "7" THEN 2020
2030 B=VAL(B$)-1: CLS: PRINT "Digite o número de "; V$(B)
2040 INPUT VL
2050 CLS: PRINTVL; CHR$(32); V$(B); " equivalem a:"
2060 IF B > 3 THEN 2090
2070 FORK=0TO2: LOCATE 1, 6+2*K: A=VL*V(B)/V(K+4): PRINTA; TAB(14); V$(K+4): NEXT
2080 GOTO 2100
2090 FORK=0TO3: LOCATE 1, 6+2*K: A

```

```

=VL*V(B)/V(K): PRINTA; TAB(14); V$(K): NEXT
2100 A$=INKEY$: IF A$ = "" THEN 2100
2110 IF A$ = CHR$(13) THEN RETURN ELSE CLS: GOTO 2000
2500 COLOR 15, 13, 13: PRINT "Peso": PRINTSTRING$(39, 195): LOCATE 5, 3: PRINT "Selecione a unidade:"
2510 FORK=0TO5: LOCATE 5, 6+2*K: PRINTK+1; "- "; M$(K): NEXT
2520 B$=INKEY$: IF B$ < "1" OR B$ > "6" THEN 2520
2530 B=VAL(B$)-1: CLS: PRINT "Digite o número de "; M$(B)
2540 INPUT VL
2550 CLS: PRINTVL; CHR$(32); M$(B); " equivalem a:"
2560 IF B > 2 THEN 2590
2570 FORK=0TO2: LOCATE 1, 6+2*K: A=VL*M(B)/M(K+3): PRINTA; TAB(14); M$(K+3): NEXT
2580 GOTO 2600
2590 FORK=0TO2: LOCATE 1, 6+2*K: A=VL*M(B)/M(K): PRINTA; TAB(14); M$(K): NEXT
2600 A$=INKEY$: IF A$ = "" THEN 2600
2610 IF A$ = CHR$(13) THEN RETURN ELSE CLS: GOTO 2500
3000 COLOR 15, 14, 14: PRINT "Pressão": PRINTSTRING$(39, 195): LOCATE 5, 3: PRINT "Selecione a unidade:"
3010 FORK=0TO4: LOCATE 5, 6+2*K: PRINTK+1; "- "; P$(K): NEXT
3020 B$=INKEY$: IF B$ < "1" OR B$ > "5" THEN 3020
3030 B=VAL(B$)-1: CLS: PRINT "Digite o número de "; P$(B)
3040 INPUT VL
3050 CLS: PRINTVL; CHR$(32); P$(B); " equivalem a:"
3060 T=0: FORK=0TO4: IFK=B THEN 3080
3070 LOCATE 1, 6+2*T: A=VL*P(K)/P(B): PRINTA; TAB(14); P$(K): T=T+1
3080 NEXT
3090 A$=INKEY$: IF A$ = "" THEN 3090
3100 IF A$ = CHR$(13) THEN RETURN ELSE CLS: GOTO 3000
3500 COLOR 15, 6, 6: PRINT "Temperatura": PRINTSTRING$(39, 195): LOCATE 5, 3: PRINT "Converter:"
3510 LOCATE 5, 6: PRINT "Centígrados para Farenheit (C)": LOCATE 5, 8: PRINT "Farenheit para Centígrados (F)"
3520 B$=INKEY$: IF B$ < > "C" AND B$ < > "F" THEN 3520
3530 IF B$ = "C" THEN 3560
3540 CLS: INPUT "Graus Farenheit": VL
3550 LOCATE 3, 7: PRINTVL; " graus Farenheit equivalem a ": PRINT(VL-32)*5/9; " graus Centígrados." : GOTO 3580
3560 CLS: INPUT "Graus Centígrados": VL
3570 LOCATE 3, 7: PRINTVL; " graus Centígrados equivalem a ": PRINT(32+VL*9/5); " graus Farenheit."
3580 A$=INKEY$: IF A$ = "" THEN 3580
3590 IF A$ = CHR$(13) THEN RETURN ELSE CLS: GOTO 3500

```


Pág. 26 - legenda à esquerda

Onde se lê:

No TK-80X o Caleidoscópio...

Leia-se:

No TK-90X o Caleidoscópio...

Pág. 34 - 1ª col. - 4º parágrafo

Onde se lê:

5 + 10 + 200 + 3000

Leia-se:

5 + 70 + 200 + 3000

Pág. 53 - 1ª col. - 2º parágrafo

Onde se lê:

...conector DID...

Leia-se:

...conector DIN...

Pág. 56 - TÍTULO DA MATÉRIA

Onde se lê:

ARITIMÉTICA HEXADECIMAL

Leia-se:

ARITMÉTICA HEXADECIMAL

Pág. 60 - 3ª col. - quadro

Onde se lê:

O octal tem 8 dígitos: 0, 1, 2, 3, 4, 5, 6, 7 e 8.

Leia-se:

O octal tem 8 dígitos: 0, 1, 2, 3, 4, 5, 6 e 7.

Pág. 245 - 1ª col. - 1º e 4º parágrafos

Onde se lê:

VAL(=A\$) ou VAL(=B\$)

Leia-se:

VAL(A\$) ou VAL(B\$)

Pág. 245 - 1ª col. - 4º parágrafo

Onde se lê:

B\$“45 MARCOS”

Leia-se:

B\$ = “45 MARCOS”

Pág. 260 - 1ª col. - 5º, 6º e 7º parágrafos

e Pág. 260 - 2ª col. - 2º parágrafo

Onde se lê:

CHR\$

Leia-se:

CHR\$

Pág. 290 - 1ª col. - último parágrafo

Pág. 291 - 2ª col. - último parágrafo

Onde se lê:

...os comandos GET\$ e

Leia-se:

...os comandos GET e

Pág. 294 - 1ª col. - 2º parágrafo

Onde se lê:

...poderão ser chocados...

Leia-se:

...poderão ser checados...

S

Pág. 7 - 2ª col. - 2º parágrafo

Onde se lê:

(só vale para o TK-90X)

Leia-se:

(vale para todos os micros da linha Sinclair)

Pág. 30 - 1ª col. - programa

Substitua a flecha para cima (↑) na linha 60 pela letra I

Pág. 32 - 3ª col. - programa

Substitua a flecha para cima (↑) na linha 150 pela letra I

Pág. 39 - 1ª col. - programa

Para funcionar no ZX-81, coloque tudo em maiúsculas e substitua a linha 130 por:

```
130 PRINT , , , "OUTRO NUMERO (S/ N)?"
```

Pág. 41 - 1ª col. - programa

Para funcionar no ZX-81, mude tudo para maiúsculas e faça as seguintes correções no programa:

```
25 LET T=0
26 LET C=0
40 IF N=-99 THEN GOTO 80
80 PRINT "MEDIA=";T/C
90 STOP
```

Pág. 42 - 2ª col. - programa

Para funcionar no ZX-81, mude tudo para maiúsculas e faça as seguintes correções no programa:

```
50 IF G=N THEN GOTO 90
90 PRINT "MUITO BEM"
92 PAUSE 100
94 GOTO 10
```

Pág. 43 - 1ª col. - 4º parágrafo

Adicione:

No programa para o ZX-81, retire a linha 94.

Pág. 43 - 1ª col. - 2º programa

Para funcionar no ZX-81, mude tudo para maiúsculas e separe as instruções da linha 110 em duas linhas.

Pág. 77 - 1ª col. - 2º programa

Para funcionar no ZX-81, divida a linha 15 em:

```
15 INPUT A
16 INPUT B
```

Pág. 78 - 2ª col. - programa

Para funcionar no ZX-81, divida em várias linhas os comandos separados por dois pontos, nas linhas 130, 140, 150 e 160.

Pág. 94 - 2ª col. - 4º parágrafo

Onde se lê:

No ZX-81 a tecla RAMD fornece RAND na tela)

Leia-se:

No ZX-81, a tecla RAND pode ser usada para esta finalidade)

Pág. 244 - 2ª col. - 1º programa

O programa também funciona para o Sinclair ZX-81. Apenas coloque o comando GOTO 10 da linha 30 em uma linha adicional.

S

Pág. 30 - 1ª col. - programa

Substitua a flecha para cima (↑) na linha 60 pela letra I

Pág. 32 - 3ª col. - programa

Substitua a flecha para cima (↑) na linha 150 pela letra I

Pág. 46 - 1ª col. - programa

Adicione a linha:

90 DIM a\$(32)

Pág. 94 - 2ª col. - 3º programa

Pág. 100 - 1ª col. - 2º programa

O logotipo não se aplica a este programa

Pág. 136 - 3ª col. - programa

Corrija a linha:

200 CLS:PRINT BRIGHT v; PAPER 2; INK 6; AT 2; 4; "MENU PRINCIPAL"

Pág. 168 - 3ª col. - 2º parágrafo

Adicione:

No micro nacional TK-90X, este comando recebeu o nome de SOUND. Portanto, se você tem um TK-90X, substitua todas as ocorrências do comando BEEP pelo comando SOUND, nas linhas 8010 (programas da página 169, 1ª coluna e 3ª coluna), 8010 e 8015 (programa da página 170, 1ª coluna) e 400 (página 170, 2ª coluna).

Pág. 213 - 3ª col. - 1º parágrafo

Onde se lê:

Os comandos ld de, 16384 e ld hl, 1,16385...

Leia-se:

Os comandos ld de, 16384 e ld hl, 16385...

Pág. 244 - 2ª col. - 1º programa

O programa também funciona para o Sinclair Spectrum

NO PRÓXIMO NÚMERO

PROGRAMAÇÃO BASIC

Conheça melhor os comandos gráficos de seu computador e saiba explorar plenamente os recursos disponíveis.

CÓDIGO DE MÁQUINA

O que faz um programa rastreador. Como detectar e corrigir os erros de um programa longo.

PROGRAMAÇÃO BASIC

Como tirar o máximo do comando **EDIT** e facilitar a edição de programas no TRS-80 e no TRS-Color.

PROGRAMAÇÃO DE JOGOS

Invente aventuras originais usando como base os programas de **INPUT**.

CURSO PRÁTICO **20** DE PROGRAMAÇÃO DE COMPUTADORES

PROGRAMAÇÃO BASIC - PROGRAMAÇÃO DE JOGOS - CÓDIGO DE MÁQUINA

Cz\$ 20,00

