GO TO Informática

IAN STEWART e ROBIN JONES

JOGOS DE PACIÊNCIAS E PUZZLES PARA O SPECTRUM



e ZX 81



EDIÇÕES DETOP

COLECÇÃO GO TO INFORMÁTICA

JOGOS DE PACIÊNCIAS E «PUZZLES»
— PARA O SPECTRUM E ZX81 —

Volumes publicados nesta colecção:

1 — Introdução à Programação em Pascal — Exercícios
2 — 13 Jogos para o Spectrum 16K ou 48K
3 — Jogos de Paciências e Puzzles para o Spectrum e ZX81

IAN STEWART e ROBIN JONES

JOGOS DE PACIÊNCIAS E «PUZZLES» PARA O SPECTRUM E ZX81

EDIÇÕES CETOP

APARTADO 33 ▪ 2726 MEM MARTINS CODEX
SEDE: ESTRADA LISBOA-SINTRA, km 14 ▪ TELEF. 926 32 22 ▪ TELEGS. EDICETOP

Este livro foi traduzido da edição original com o título:
Computer Puzzles: for Spectrum and ZX81

Capa: estúdios P. E. A.

Tradução e Revisão Técnica: Dr. Norberto Fernando G. Fernandes

© Ian Stewart e Robin Jones, 1982

Direitos em língua portuguesa reservados por
CETOP — Centro de Ensino Técnico e Profissional à Distância, Lda.

Nenhuma parte desta publicação pode ser reproduzida ou transmitida por qualquer forma ou por qualquer processo, electrónico, mecânico ou fotográfico, incluindo fotocópia, xerocópia ou gravação, sem autorização prévia e escrita do editor. Exceptua-se naturalmente a transcrição de pequenos textos ou passagens para apresentação ou crítica do livro. Esta excepção não deve de modo nenhum ser interpretada como sendo extensiva à transcrição de textos em recolhas antológicas ou similares donde resulte prejuízo para o interesse pela obra. Os transgressores são passíveis de procedimento judicial

Edição n.º 06 19 003/060

Editor: Tito Lyon de Castro

Execução técnica:
Gráfica Europam, Lda.
Mem Martins

ÍNDICE

		Pág.
	Introdução	7
	Leia isto primeiro	9
1	O lobo, a cabra e a couve — *The wolf, the goat and the cabbage*	11
2	Desordem alfabética — *Alphabetical disorder*	21
3	Troca-gatos — *Cat-swap*	27
4	O cento — *Stringing the ton*	31
5	Justas repartições — *Fair shares*	35
6	A floresta mágica — *The magic forest*	41
7	Rainhas dominadoras — *Queens dominant*	47
8	Bispos ruidosos — *Bishops blatant*	53
9	Torres dominantes — *Rooks rampant*	57
10	Senhoras tímidas — *Shy ladies*	61
11	«Puzzle» do macaco — *Monkey puzzle*	63
12	Encadeamento ecológico — *Ecological chain*	69
13	Torre de Hanói com bolos — *Tower of Hanoi with pancakes*	75
14	Cavalgando — *Several times knightly*	81
15	Animais em degraus — *Beasts on the steps*	89
16	Solitário — *Solitaire*	97
17	Anagramas contra-relógio — *Anagrams against the clock*	103
18	Palavras cruzadas — *Hidden words*	109
19	Criptogramas — *Cryptograms*	117

NOTA DO EDITOR PORTUGUÊS

Todos os programas contidos neste livro são apresentados de duas formas.

Na primeira reproduziu-se rigorosamente a versão original, em letra tipográfica e conforme o autor a concebeu.

Na segunda, adaptaram-se para português nomes de ficheiros, comentários, etc., de modo a tornar os jogos mais compreensíveis para quem não esteja familiarizado com a língua inglesa. A impressão dos programas, neste caso, foi efectuada numa impressora T.S. 2040.

INTRODUÇÃO

O grande criador de «puzzles» Henry Ernest Dudeney escreveu uma vez: «A história dos 'puzzles' perpetua, nem mais nem menos, a história actual do início e desenvolvimento do pensamento humano exacto. Se não tivessem existido 'puzzles' para solucionar, não teria havido perguntas para pôr; e se não tivesse havido perguntas a pôr que monótono mundo seria o nosso!»

Na actualidade, Dudeney faz a sua vida sem «puzzles» e, portanto, podemos esquecer, por enquanto, parte dos seus exageros; mas é provavelmente verdade que os «puzzles» tendem a encorajar algum pensamento lógico. As «palavras cruzadas» são uma das mais importantes artes da nossa época.

Mas esquecendo tudo o resto, os «puzzles» são sobretudo *divertidos*.

Há uma longa tradição de «puzzles» que vêm desde a antiga Babilónia. Muitos dos «puzzles» são tão fascinantes agora como o eram há setecentos anos atrás. E para provar isso mesmo, os «puzzles» entram agora na Idade do Computador. Este é um livro de «puzzles» para computador, que servem tanto no ZX Spectrum como no ZX81. Os computadores são excelentes para *fazer experiências* com «puzzles»: pode tentar-se uma resposta e verificar o que resulta.

Alguns «puzzles», de facto, *só* poderiam ser resolvidos com a ajuda de um computador. Por exemplo, você poderia «computar» um cubo de Rubik a quatro dimensões, mas nunca poderia *fazer* um. (Infelizmente também não conseguiria *resolvê-lo*, e daí nós não termos prosseguido com esta ideia neste livro.) Num computador a cores você poderá mudar as cores do mesmo modo que será possível fazê-lo com os tradicionais utensílios para o efeito: o papel e os lápis de cor.

Outros «puzzles» adaptam-se bem. Conhece aquele do lobo, da cabra e da couve tentando atravessar um rio? Num computador pode desenhar o rio, pôr o barco a navegar e ver quem é comido...

Finalmente, este livro tem três objectivos: colocar os «puzzles» na Idade do Computador; ensinar a pensar claro; mas, sobretudo, para simples divertimento. Foi divertido escrevê-lo: espero que se divirtam, tanto como eu, a lê-lo e fazendo os programas.

LEIA ISTO PRIMEIRO...

...ou mete-se numa grande confusão.

1 — A maioria dos programas deste livro vem em duas versões: a primeira para o ZX81 ou TIMEX 1000 (16 K) e a segunda para o ZX Spectrum. Os quatro últimos programas são apenas para os *entusiastas* do ZX Spectrum.

2 — A versão para ZX81 funcionará no Spectrum, mas a do Spectrum usa facilidades só nele disponíveis e por conseguinte não funcionará no ZX81.

3 — Para poupar espaço e papel e, portanto, o seu dinheiro, a listagem do Spectrum é feita modificando a do ZX81. A secção do Spectrum mostra como isso se poderá fazer em pormenor. Qualquer linha de programa do ZX81 que será modificada na versão do Spectrum está marcada com uma seta → de modo a poupar tempo aos possuidores do Spectrum na introdução dos programas.

4 — Outro aspecto: as varáveis no ZX81 são todas as letras maiúsculas (A, B, C, etc.); mas no Spectrum as variáveis são normalmente letras minúsculas (a, b, c, etc.). Para também poupar papel a versão do Spectrum não foi reescrita de modo a pôr letras minúsculas em todas as variáveis; assume-se portanto que quando você estiver a introduzir o programa, usará letras minúsculas. Portanto, PRINT X, Y na listagem será modificado pelos utilizadores do Spectrum para PRINT x, y ... etc.

5 — Uma caixa □ é usada para representar um espaço quando não é óbvio que isto não é mesmo necessário, sendo ■ um espaço em inverso vídeo, e a caixa envolvendo qualquer caracter quer dizer também inverso vídeo.

6 — No ZX81 faz-se "GOTO" e "GOSUB" e no Spectrum "GO TO" e "GO SUB". Nada se indica acerca disto nas listagens.

7 — No que se refere aos caracteres gráficos convencionou-se o seguinte:

ZX81: O g pequeno indica «carácter gráfico»: é seguido pela letra ou número a que refere o carácter. "gT" é .

Spectrum: Simultaneamente só para os caracteres que são acedidos com CAPS SHIFT é adicionada a letra "c". "g6" é ; e "g6c" é .

8 — No Spectrum deverá dar as cores directamente do teclado: BORDER PAPER e INK antes de inicializar qualquer progama. Para o fazer ponha em comando qualquer coisa como isto:

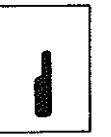
BORDER 1: PAPER 3: INK 7

para fundo magenta, bordo azul e letras a preto. (Tecle ENTER duas vezes para que o fundo apareça). Qualquer combinação é em princípio permitida mas não ficará muito bonita. Experimente; pode mudar sempre que quiser as cores antes de teclas RUN. Eu não incluí estes comandos iniciais nos programas do Spectrum porque são muito rotineiros: muitas das versões dos programas do Spectrum incluem de qualquer modo outras cores.

9 — Os programas estão prontos a funcionar por si próprios e podem ser introduzidos independentemente da ordem em que aparecem no livro: introduza o que mais lhe apetecer e siga as instruções.

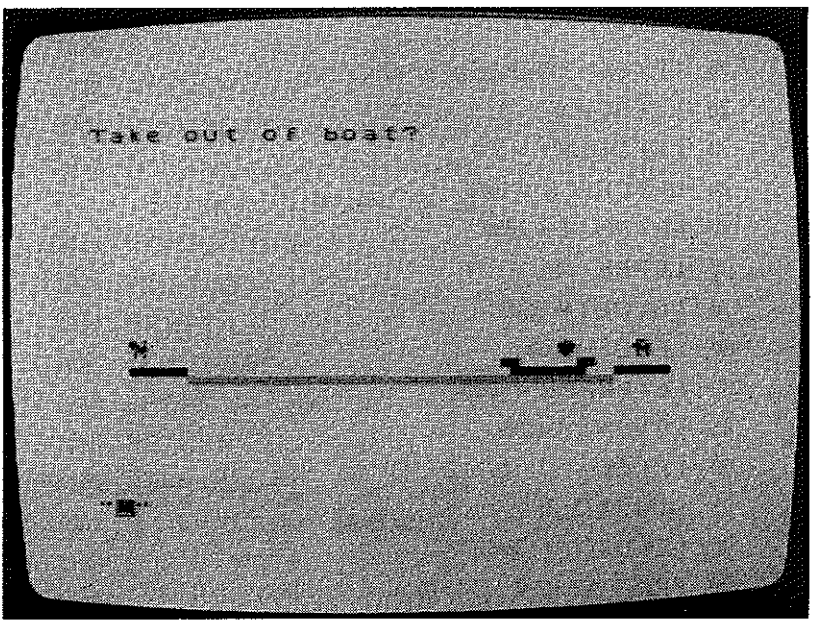
10 — E chega de conversa fiada *(sic)* — ao trabalho!

Atravesse com os seus bens o rio — mas tenha cuidado!
Senão eles comem-se uns aos outros.



O LOBO, A CABRA E A COUVE

the wolf, the goat and the cabbage



Imagine que você é um fazendeiro do século XVII que está levando os seus bens ao mercado. Eles são os seguintes:

1 lobo;
1 cabra;
1 couve.

Não me pergunte por que é que o fazendeiro tem um lobo; eu sou apenas o autor do programa.

Chega a um rio, que é preciso atravessar. Existe um barco que o transportará a si e apenas um dos seus bens (lobo, cabra ou couve).

Enquanto se aproxima os seus bens portam-se admiravelmente. *Mas:* deverá deixar o lobo sozinho com a cabra numa margem enquanto atravessa o rio... bem, os lobos são doidos por carne de cabra. E similarmente as cabras adoram couves. Felizmente que o lobo detesta couves (ele frequentou um internato quando era mais novo) e a couve é muito pacífica; portanto pode deixar estes dois juntos se quiser.

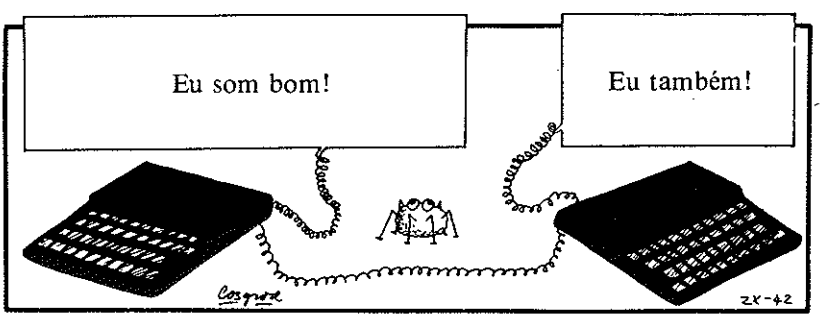
Quais deles devem atravessar o rio e em que ordem?

VERSÃO ZX81

```
→   10  LET B$ = "□ gR gF gF gF gE □"
    20  LET BC = 0
    30  LET QUERY = 500
    40  LET CHECK = 1000
    50  LET BOAT = 2000
    60  LET BK = 1
    70  LET OUT = 2500
    80  LET IN = 3000
    82  LET ERROR = 5000
    85  LET EAT = 4000
    90  LET C$ = "□□□□□□□"
    93  DIM D$ (2, 3)
→   95  LET D$ (1) = "WGC"
→   100 PRINT AT 15, 0; "□ □ g6 g6 g6 [22 espaços] g6 g6 g6"
→   110 PRINT AT 16, 0; "□ □ ■ ■ ■ gS [22 vezes] ■ ■ ■"
    120 PRINT AT 14, 2; D$ (1)
→   130 PRINT AT 15, 5; B$
    200 REM  MOVE
    210 GOSUB QUERY
    220 GOSUB BOAT
    230 GOTO 200
    500 REM  QUERY
    505 IF BC = 0 THEN GOTO 540
    510 PRINT AT 2, 0; "TAKE OUT OF BOAT?"
    520 INPUT O$
```

```
    530  GOSUB OUT
    535  IF D$ (BK) = "□ □ □" THEN GOTO 570
    540  PRINT AT 2, 0; "PUT INTO BOAT? □ □ □ □"
    550  INPUT I$
    560  GOSUB IN
    570  RETURN
   1000  REM  CHECK
→  1010  IF D$ (2) = "WGC" THEN PRINT AT 2, 0; "□ □ □ □ □ □ □
         WELL DONE  □ □ □ □ □ □ □"
→  1020  IF D$ (2) = "WGC" THEN STOP
   1030  RETURN
   2000  REM  BOAT
   2010  FOR T = 5 * (BK = 1) + 20 * (BK < > 1) TO 20 * (BK = 1)
         + 5 * (BK < > 1) STEP 1 − 2 * (BK < > 1)
→  2020  PRINT AT 15, T; B$
   2025  PRINT AT 14, T; C$
   2030  NEXT T
   2035  GOSUB EAT
   2040  LET BK = 3 − BK
   2050  RETURN
   2500  REM  OUT
   2505  IF O$ = " " THEN RETURN
   2510  FOR F = 3 TO 5
→  2520  IF C$ (F) = O$ THEN GOTO 2540
   2530  NEXT F
   2540  LET C$ (F) = "□"
→  2550  LET D$ (BK, F −2) = O$
   2560  PRINT AT 14, 2 * (BK = 1) + 27 * (BK < > 1); D$ (BK)
   2570  PRINT AT 14, 5 * (BK = 1) + 20 * (BK < > 1); C$
   2580  LET BC = BC − 1
   2590  GOSUB CHECK
   2600  RETURN
   3000  REM  IN
   3005  IF I$ = " " THEN RETURN
```

```
    3010  FOR F = 1 TO 3
→   3020  IF D$ (BK, F) = I$ THEN GOTO 3040
    3030  NEXT F
    3040  LET D$ (BK, F) = "□"
→   3050  LET C$ (F + 2) = I$
    3060  PRINT AT 14, 2 * (BK = 1) + 27 * (BK < > 1); D$ (BK)
    3070  PRINT AT 14, 5 * (BK = 1) + 20 * (BK < > 1); C$
    3080  LET BC = BC +1
    3085  IF BC = 2 THEN GOTO ERROR
    3090  GOSUB CHECK
    4000  REM  EAT
    4010  LET X$ = D$ (BK)
→   4020  IF X$ = "WG□" OR X$ = "□GC" THEN GOTO 4040
    4030  RETURN
    4040  PRINT AT 2, 0; "□□□□□□□ WOOPS . . . EATEN □□□□□□"
    4050  STOP
    5000  REM  ERROR
    5010  PRINT AT 2, 0; "SORRY, TOO MANY IN BOAT. TRY"
    5020  PRINT "AGAIN"
    5030  STOP
```



SE USAR A VERSÃO DO ZX81

Introduza tudo e *verifique:* tecle RUN. Verá o rio e o barco; o lobo, a cabra e a couve representadas pelas suas iniciais, situam-se na margem. (Eu não o desenhei a si mas está lá: você move-se com o barco).

Em cada passo ser-lhe-á perguntado que coisa quer levar no barco. Para responder deve teclar "W", "G", "C" ou □ conforme o que queira levar correspondente à inicial ou nada. Seguidamente pede-se-lhe o que quer tirar do barco: terá de responder com as teclas atrás descritas.

O barco atravessa o rio e verificará se fez uma boa escolha.

Se conseguir transportar tudo, o computador assinalá-lo-á.

VERSÃO ZX SPECTRUM

Modifique as linhas seguintes:

```
10   LET B$ = "□ g7 g3c g3c g3c g4c □"
95   LET D$ (1) = "gA gG gC"
100  PRINT AT 15, 0; "□ □ g3c g3c g3c [22 espaços] g3c g3c g3c"
110  PRINT AT 16, 0; "□ □ □ □ □"; INK 5; "g3 [22 vezes] "; INK 0; "□ □ □"
130  PRINT AT 15, 5; INK 3; B$
1010 IF D$ (2) = "gA gG gC" THEN PRINT AT 2, 0; "□ □ □ □ □ □ □
     Well done! □ □ □ □ □ □ □": STOP
1020 [apague esta linha]
2020 PRINT INK 3; AT 15, T; B$
2520 IF CODE C$ (F) = CODE O$ + 47 THEN GOTO 2540
2550 LET D$ (BK, F − 2) = CHR$ (47 + CODE O$)
3020 IF CODE D$ (BK, F) = CODE I$ + 47 THEN GOTO 3040
3050 LET C$ (F + 2) = CHR$ (47 + CODE I$)
4020 IF X$ = "gA gG □" OR X$ = "□ gG gC" THEN GO TO 4040
```

Adicione estas linhas:

```
525  IF O$ = "W" THEN LET O$ = "A"
555  IF I$ = "W" THEN LET I$ = "A"
2045 INK 0
```

Adicione as linhas para definir as silhuetas do lobo, da cabra e da couve (com A G C na secção de definição de caracteres pelo utilizador):

```
6000 LET Y$ = "AGC"
6010 DATA 162, 226, 242, 94, 30, 36, 34, 34, 56, 40, 226, 254, 62, 34, 34, 34, 16,
     84, 254, 254, 124, 124, 56, 16
6020 FOR X = 1 TO 3
6030 FOR N = 0 TO 7
6040 READ M
6050 POKE USR Y$ (X) + N, M
6060 NEXT N
6070 NEXT X
```

Se quiser efeitos de som para a solução adicione estas linhas:

```
1015 IF D$ (2) = "gA gG gC" THEN BEEP .3, 0: BEEP .3, 4: BEEP .3, 7:
     BEEP 1.5, 10: STOP
```

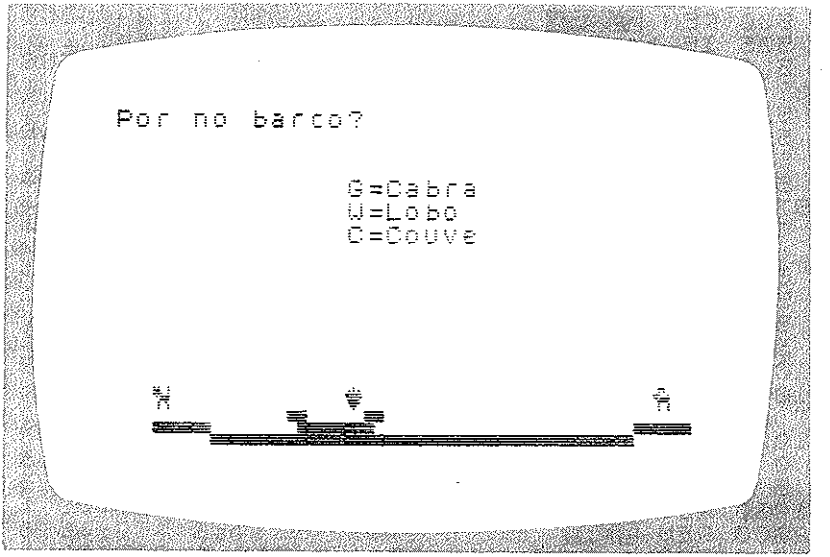
E apague:

STOP em 1010

SE USAR A VERSÃO DO ZX SPECTRUM

1 — Inicialize, após a ligação, com GO TO 6000 para definir os caracteres gráficos.

2 — Depois faça RUN e proceda como na versão ZX81.






```
  1 GO SUB 6000
 10 LET b$="          "
 20 LET bc=0
 30 LET query=500
 40 LET check=1000
 50 LET boat=2000
 60 LET bk=1
 70 LET out=2500
 80 LET in=3000
 82 LET eat=4000
 90 LET c$="          "
 93 DIM d$(2,3)
 95 LET d$(1)="
100 PRINT AT 15,0;"
110 PRINT AT 16,0;"          "; INK
5;"                                "; INK
0;"    "
115 PRINT AT 5,0; INK 0; PAPER
4;"                G=Cabra
                   W=Lobo
                   C=Couve
   "
117 BORDER 6
120 PRINT AT 14,2;d$(1)
130 PRINT AT 15,5; INK 3;b$
200 REM
210 GO SUB query
220 GO SUB boat
230 GO TO 200
```

```
500 REM [illegible]
505 IF bc=0 THEN GO TO 540
510 PRINT AT 2,0;"Tirar do barc
o? "
520 INPUT o$
525 IF o$="w" THEN LET o$="a"
530 GO SUB out
535 IF d$(bk)="    " THEN GO TO
570
540 PRINT AT 2,0;"Por no barco?
"
550 INPUT i$
555 IF i$="w" THEN LET i$="a"
560 GO SUB in
570 RETURN
1000 REM [illegible]
1010 IF d$(..)="[illegible]" THEN PRINT A
T 2,0;"    Conseguiu! parabens"
1020 IF d$(2)="[illegible]" THEN BEEP .3
,0: BEEP .3,4: BEEP .3,7: BEEP 1
,5,10: STOP
1030 RETURN
2000 REM [illegible]
2010 FOR t=5*(bk=1)+20*(bk<>1) T
O 20*(bk=1)+5*(bk<>1) STEP 1-2*(
bk<>1)
2020 PRINT INK 3;AT 15,t;b$
2025 PRINT AT 14,t;c$
2030 NEXT t
2035 GO SUB eat
2040 LET bk=3-bk
2050 RETURN
2500 REM [illegible]
2505 IF o$="" THEN RETURN
2510 FOR f=3 TO 5
2520 IF CODE c$(f)=CODE o$+47 TH
EN GO TO 2540
2530 NEXT f
2540 LET c$(f)=" "
2550 LET d$(bk,f-2)=CHR$ (47+COD
E o$)
2560 PRINT AT 14,2*(bk=1)+27*(bk
<>1);d$(bk)
2570 PRINT AT 14,5*(bk=1)+20*(bk
<>1);c$
2580 LET bc=bc-1
2590 GO SUB check
2600 RETURN
3000 REM [illegible]
3005 IF i$="" THEN RETURN
3010 FOR f=1 TO 3
3020 IF CODE d$(bk,f)=CODE i$+47
 THEN GO TO 3040
3030 NEXT f
3040 LET d$(bk,f)=" "
3050 LET c$(f+2)=CHR$ (47+CODE i
$)
```

```
3060 PRINT AT 14,2*(bk=1)+27*(bk
<>1);d$(bk)
3070 PRINT AT 14,5*(bk=1)+20*(bk
<>1);c$
3080 LET bc=bc+1
3085 IF bc=2 THEN GO TO error
3090 GO SUB check
4000 REM [illegible]
4010 LET x$=d$(bk)
4020 IF x$="[illegible]" OR x$=" [illegible]" THE
N GO TO 4040
4030 RETURN
4040 PRINT AT 2,0;"      Que pena
, perdeu!... "
4050 STOP
5000 REM [illegible]
5010 PRINT AT 2,0;"Demasiada car
ga, tente outra vez"
5030 STOP
6000 LET y$="agc"
6010 DATA 162,226,242,94,30,38,3
4,34,56,40,226,254,62,34,34,34,1
6,84,254,254,124,56,16
6020 FOR x=1 TO 3
6030 FOR n=0 TO 7
6040 READ m
6050 POKE USR y$(x)+n,m
6060 NEXT n
6070 NEXT x
6080 RETURN
```

NOTAS

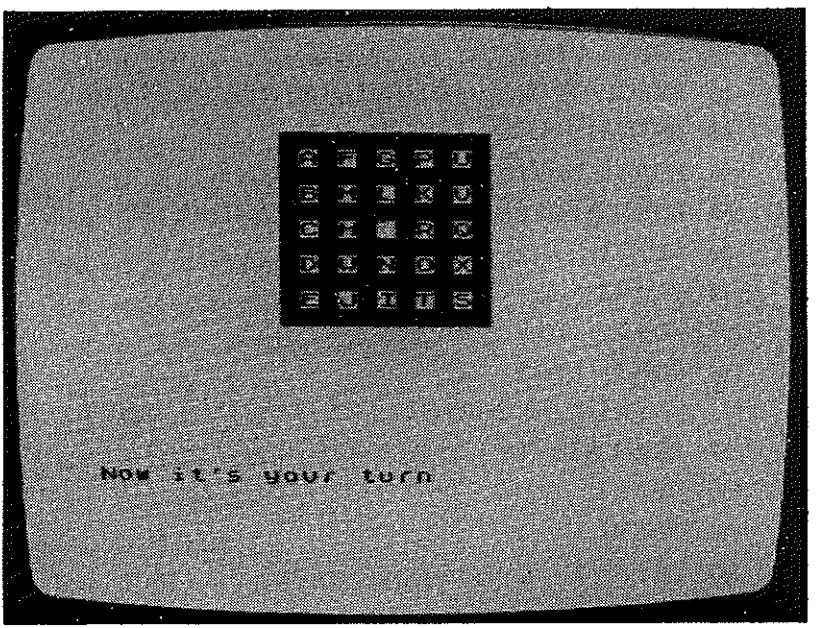
NOTAS

Baralhe o alfabeto (onde faltam o Y e o Z)
Volte ao mesmo:

2

DESORDEM ALFABÉTICA

alphabetical disorder



Este programa começa com o alfabeto ordenado da forma seguinte:

A F K P U
B G L Q V
C H M R W
D I N S X
E J O T □

e a primeira coisa a fazer é misturar as letras todas.

O seu trabalho é conseguir ordená-las novamente como estavam no início. Para o conseguir terá de trocar qualquer letra adjacente ao espaço em branco □ por ele próprio. Evidentemente que o espaço ficará agora onde estava a letra. Por outras palavras, imagine que rodopia pelo quadrado de letras.

Para mover o espaço use as quatro "setas": teclas 5, 6, 7 e 8.

VERSÃO ZX81

```
    10  FOR I = 1 TO 6
→   20  PRINT AT 2* I, 10; "gH gH gH gH gH gH gH gH gH gH"
    30  NEXT I
    40  FOR I = 1 TO 5
→   50  PRINT AT 2 * I + 1, 10; "gH□gH□gH□gH□gH□gH"
    60  NEXT I
    70  DIM A$ (5, 5)
   100  FOR I = 1 TO 5
   110  FOR J = 1 TO 5
→  115  LET A$ (I, J) = CHR$ (32 + I + 5 * J − 62 * (I = 5 AND J = 5) )
   120  PRINT AT 2 * I + 1, 2 * J + 9; A$ (I, J)
   130  NEXT J
   140  NEXT I
   200  LET P = 5
   210  LET Q = 5
   215  FOR T = 1 TO 50
   220  LET W = 2 * INT ( 2 * RND) − 1
   230  LET P0 = P + W
   240  IF P0 > 5 OR P0 < 1 THEN GOTO 220
   250  LET W = 2 * INT (2 * RND) − 1
   260  LET Q0 = Q + W
   270  IF Q0 > 5 OR Q0 < 1 THEN GOTO 250
   275  GOSUB 290
   280  NEXT T
   285  GOTO 400
```

```
290  LET Y$ = A$ (P0, Q0)
300  LET A$ (P0, Q0) = A$ (P, Q)
310  LET A$ (P, Q) = Y$
320  PRINT AT 2 * P0 + 1, 2 * Q0 + 9; A$ (P0, Q0)
330  PRINT AT 2 * P + 1, 2 * Q + 9; A$ (P, Q)
340  LET P = P0
350  LET Q = Q0
360  RETURN
400  PRINT AT 21, 0; "NOW IT " " S YOUR TURN"
410  IF INKEY$ < > " " THEN GOTO 410
420  IF INKEY$ = " " THEN GOTO 420
430  LET I$ = INKEY$
440  IF I$ = "5" THEN LET Q0 = Q - 1
450  IF I$ = "6" THEN LET P0 = P + 1
460  IF I$ = "7" THEN LET P0 = P - 1
470  IF I$ = "8" THEN LET Q0 = Q + 1
480  IF P0 > = 1 THEN GOTO 483
481  LET P0 = 1
482  GOTO 410
483  IF P0 < = 5 THEN GOTO 486
484  LET P0 = 5
485  GOTO 410
486  IF Q0 > = 1 THEN GOTO 489
487  LET Q0 = 1
488  GOTO 410
489  IF Q0 < = 5 THEN GOTO 492
490  LET Q0 = 5
491  GOTO 410
492  GOSUB 290
500  GOTO 410
```

SE USAR A VERSÃO DO ZX81

Comece com RUN; espere até finalizar o baralhar; comece agora teclando as setas como quiser!

VERSÃO ZX SPECTRUM

Modifique as linhas que se seguem:

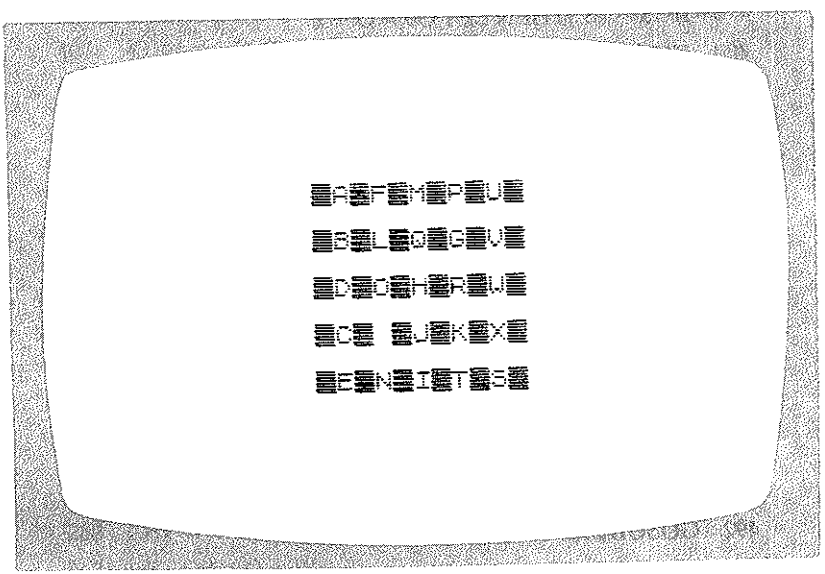
```
2Ø  PRINT AT 2 * I, 1Ø; PAPER 3; " [11 espaços ] "
5Ø  PRINT AT 2 * I + 1, 1Ø; INK 3; "■ □ ■ □ ■ □ ■ □ ■ □ ■"
115 LET A$ (I, J) = CHR$ (I + 5 * J + 59 – (I = 5 AND J = 5) * 57)
```

Adicione esta linha:

```
355 BEEP .1, P + 5 * Q
```

SE USAR A VERSÃO DO ZX SPECTRUM

Comece com RUN; espere (agora menos tempo) pelo baralhar das letras; comece teclando as setas como quiser.





```
Agora é a sua vez.
  10 FOR i=1 TO 8
  20 PRINT AT 2*i,10; PAPER 3;"
          "
  30 NEXT i
  40 FOR i=1 TO 5
  50 PRINT AT 2*i+1,10; INK 3;"▒
 ▒ ▒ ▒ ▒ ▒"
  60 NEXT i
  70 DIM a$(5,5)
 100 FOR i=1 TO 5
 110 FOR j=1 TO 5
 115 LET a$(i,j)=CHR$ (i+5*j+59-
(i=5 AND j=5)*57)
 120 PRINT AT 2*i+1,2*j+9;a$(i,j
)
 130 NEXT j
 140 NEXT i
 200 LET p=5
 210 LET q=5
 215 FOR t=1 TO 50
 220 LET w=2*INT (2*RND)-1
 230 LET p0=p+w
 240 IF p0>5 OR p0<1 THEN GO TO
220
 250 LET w=2*INT (2*RND)-1
 260 LET q0=q+w
 270 IF q0>5 OR q0<1 THEN GO TO
250
 275 GO SUB 290
 280 NEXT t
 285 GO TO 400
 290 LET y$=a$(p0,q0)
 300 LET a$(p0,q0)=a$(p,q)
 310 LET a$(p,q)=y$
 320 PRINT AT 2*p0+1,2*q0+9;a$(p
0,q0)
 330 PRINT AT 2*p+1,2*q+9;a$(p,q
)
 340 LET p=p0
 350 LET q=q0
 355 BEEP .1,p+5*q
 360 RETURN
 400 PRINT AT 21,0;"Agora é a su
a vez."
 410 IF INKEY$<>"" THEN GO TO 41
0
 420 IF INKEY$="" THEN GO TO 420
 430 LET i$=INKEY$
 440 IF i$="5" THEN LET q0=q-1
 450 IF i$="6" THEN LET p0=p+1
 460 IF i$="7" THEN LET p0=p-1
 470 IF i$="8" THEN LET q0=q+1
 480 IF p0>=1 THEN GO TO 483
 481 LET p0=1
 482 GO TO 410
 483 IF p0<=5 THEN GO TO 486
```

```
484 LET p0=5
485 GO TO 410
486 IF q0>=1 THEN GO TO 489
487 LET q0=1
488 GO TO 410
489 IF q0<=5 THEN GO TO 492
490 LET q0=5
491 GO TO 410
492 GO SUB 290
500 GO TO 410
```

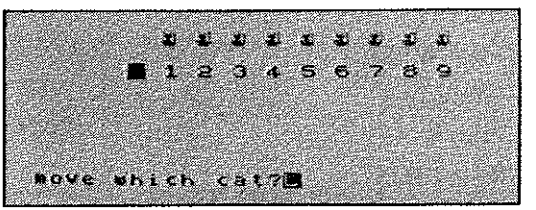
NOTAS

Vários gatos todos seguidos estão sentados num muro. Consegue inverter a sua sequência?

3

TROCA-GATOS

cat-swap



O programa apresenta-lhe uma série de gatos numerados e ordenados 1 2 3 4 5 6 7 (por exemplo). Consegue reordená-los para 7 6 5 4 3 2 1? Há um espaço em branco do lado esquerdo da série. Um gato pode mudar-se para o espaço se estiver adjacente a ele ou pode saltar apenas sobre um outro gato (como nas damas) para o espaço em branco. Você escolhe que gato quer mover e continua por aí em diante até pô-los na ordem pretendida. O espaço em branco deve ficar no fim onde estava no início, ou seja, do lado esquerdo.

VERSÃO ZX81

```
    10  LET P = 0
    20  DIM W (9)
    30  FOR I = 1 TO 9
    40  LET W (I) = I
    50  NEXT I
→   100 PRINT AT 0, 0; "HOW MANY CATS"
→   105 INPUT C
    110 LET N$ = "□1□2□3□4□5□6□7□8□9"
    120 PRINT AT 16, 6; "■" + N$ (TO 2 * C)
    130 FOR I = 1 TO C
```

```
→  140  PRINT AT 14, 6 + 2 * I; "*"
   150  NEXT I
→  200  PRINT AT 0, 0; "MOVE WHICH CAT"
→  205  INPUT A
   210  IF A < 1 OR A > C THEN GOTO 200
   212  PRINT AT 0, 0; " [14 espaços ] "
   215  IF ABS (W (A) − P) > 2 THEN GOTO 200
→  220  PRINT AT 16, 6 + 2 * P; A; AT 14, 6 + 2 * P; "*"; AT 16, 6 + 2 * W (A);
        "■"; AT 14, 6 + 2 * W (A); "□"
   224  LET R = W (A)
   225  LET W (A) = P
   230  LET P = R
   240  FOR J = 1 TO C
   250  IF W (J) + J < > C + 1 THEN GOTO 200
   260  NEXT J
   265  IF P > 0 THEN GOTO 200
   270  PRINT AT 8, 12; "CATS SWAPPED"
   280  STOP
```

SE USAR A VERSÃO DO ZX81

Tecle RUN. Os gatos são os asteriscos. Escolha os números e veja como eles saltam. Se conseguir ordená-los da maneira pretendida o computador assinalá-lo-á.

VERSÃO ZX SPECTRUM

Modifique as linhas seguintes:

```
100  INPUT "How many cats?"; C
105  [apague esta linha]
140  PRINT AT 14, 6 + 2 * I; INK 4; "gC"; INK 0
200  INPUT "Move which cat?"; A
205  [apague esta linha]
```

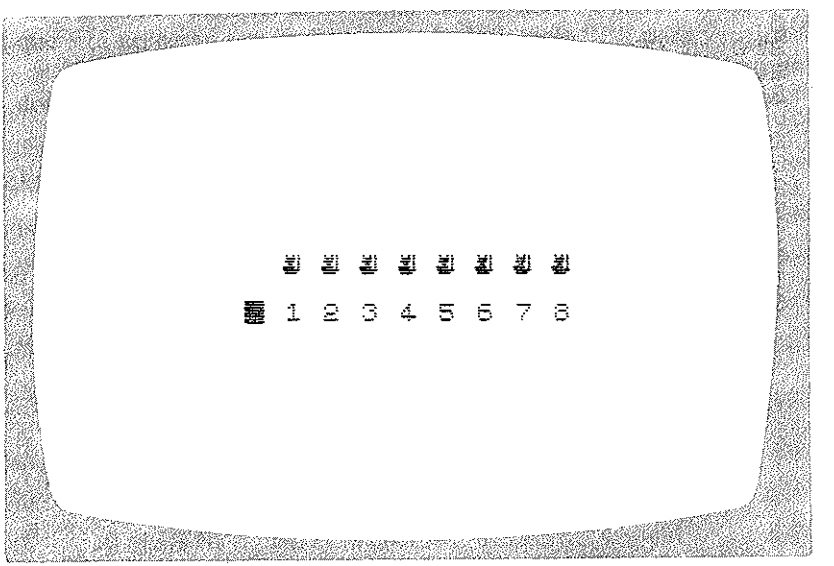
```
220 PRINT AT 16, 6 + 2 * P; A; AT 14, 6 + 2 * P; INK 4; "gC"; INK Ø;
    AT 16, 6 + 2 * W (A); "■"; AT 14, 6 + 2 * W (A); "□"; AT 16,6 +
    + 2*P; A;: BEEP .1, A
```

Adicione a rotina para definir os caracteres gráficos:

```
  1 GOSUB 5ØØ
5ØØ DATA 74, 122, 122, 5Ø, 25Ø, 254, 124, Ø
51Ø FOR T = Ø TO 7
52Ø READ X
53Ø POKE USR "C" + T, X
54Ø NEXT T
55Ø RETURN
```

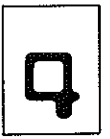
SE USAR A VERSÃO DO ZX SPECTRUM

Logo após a ligação, inicie com GO TO 5ØØ. Quando os caracteres gráficos foram definidos pode teclar RUN. Escolha qual o gato que quer mover tal como no ZX81.



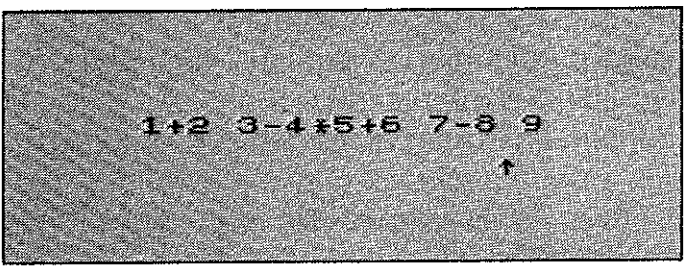
```
  1 GO SUB 500
 10 LET p=0
 20 DIM w(9)
 30 FOR i=1 TO 9
 40 LET w(i)=i
 50 NEXT i
100 INPUT "Quantos gatos? ";c
110 LET n$=" 1 2 3 4 5 6 7 8 9"
120 PRINT AT 16,6;"▒"+n$(TO 2*
c)
130 FOR i=1 TO c
140 PRINT AT 14,6+2*i; INK 4;"▒
"; INK 0
150 NEXT i
200 INPUT "Mover qual gato? ";a
210 IF a<1 OR a>c THEN GO TO 20
0
212 PRINT AT 0,0;"
"
215 IF ABS (w(a)-p)>2 THEN GO T
O 200
220 PRINT AT 14,6+2*p;a;AT 14,6
+2*p; INK 4;"▒"; INK 0;AT 16,6+2
*w(a);"▒";AT 14,6+2*w(a);" ";AT
16,6+2*p;a: BEEP .1,a
224 LET r=w(a)
225 LET w(a)=p
230 LET p=r
240 FOR j=1 TO c
250 IF w(j)+j<>c+1 THEN GO TO 2
00
260 NEXT j
265 IF p>0 THEN GO TO 200
270 PRINT AT 8,5;"ACERTOU! Troc
ou os gatos."
280 STOP
500 DATA 74,122,122,50,250,254,
124,0
510 FOR t=0 TO 7
520 READ x
530 POKE USR "c"+t,x
540 NEXT t
550 RETURN
```

Use os algarismos de 1 a 9 para fazer 100:



O CENTO

stringing the ton



O écran mostrar-lhe-á os nove algarismos 1-9, com espaços entre eles e uma seta a apontar um desses espaços. Você pode mover a seta e fazer inserir no espaço qualquer sinal de operação + , −, *, /; ou então removê-lo mais tarde se tomar essa decisão. Quando achar que o que está no écran tem como resultado 100, tecle "P" para verificá-lo.

Os espaços entre os algarismos são ignorados quando o computador calcula a resposta. Se, por exemplo, você tivesse respondido:

1+2 3+4 5+6+7+8−9

isso seria interpretado pelo computador como:

1+23+45+6+7+8−9

que é 81.

Muito perto da resposta correcta, portanto tente outra vez!

VERSÃO ZX81

```
    10  LET N$ = "1□2□3□4□5□6□7□8□9"
→   20  LET C$ = "□□>□□"
    30  LET H = 1
```

```
100  PRINT AT 16, 10; N$
110  PRINT AT 18, 7 + 2 * H; C$
120  IF INKEY$ < > " " THEN GOTO 120
130  IF INKEY$ = " " THEN GOTO 130
140  LET I$ = INKEY$
150  IF I$ = "5" THEN GOSUB 300
160  IF I$ = "8" THEN GOSUB 400
170  IF I$ = "B" THEN LET N$ (2 * H) = "*"
180  IF I$ = "K" THEN LET N$ (2 * H) = "+"
190  IF I$ = "J" THEN LET N$ (2 * H) = "-"
200  IF I$ = "V" THEN LET N$ (2 * H) = "/"
210  IF I$ = "X" THEN LET N$ (2 * H) = "□"
220  IF I$ = "P" THEN GOTO 1000
230  GOTO 100
300  LET H = H - 1 + (H = 1)
310  PRINT AT 18, 7 + 2 * H; C$
320  RETURN
400  LET H = H + 1 - (H = 8)
410  GOTO 310
1000 LET A$ = " "
1010 FOR T = 1 TO 17
1020 IF N$ (T) < > "□" THEN LET A$ = A$ + N$ (T)
1030 NEXT T
1040 PRINT AT 1, 5; "THE VALUE IS □"; VAL A$
1050 IF VAL A$ = 100 THEN PRINT AT 3, 5; "WELL DONE"
1060 IF VAL A$ = 100 THEN STOP
1070 FOR J = 1 TO 50
1080 NEXT J
1090 PRINT AT 1, 0; " [32 espaços] "
1100 PRINT AT 3, 0; " [32 espaços] "
1110 GOTO 100
```

SE USAR A VERSÃO DO ZX81

RUN; use as teclas 5 e 8 para mover o cursor; as introduções dos sinais aritméticos são feitas com as teclas onde esses sinais estão escritos; use o espaço para introduzir X. Quando estiver contente, prima P: veja se atingiu o cento.

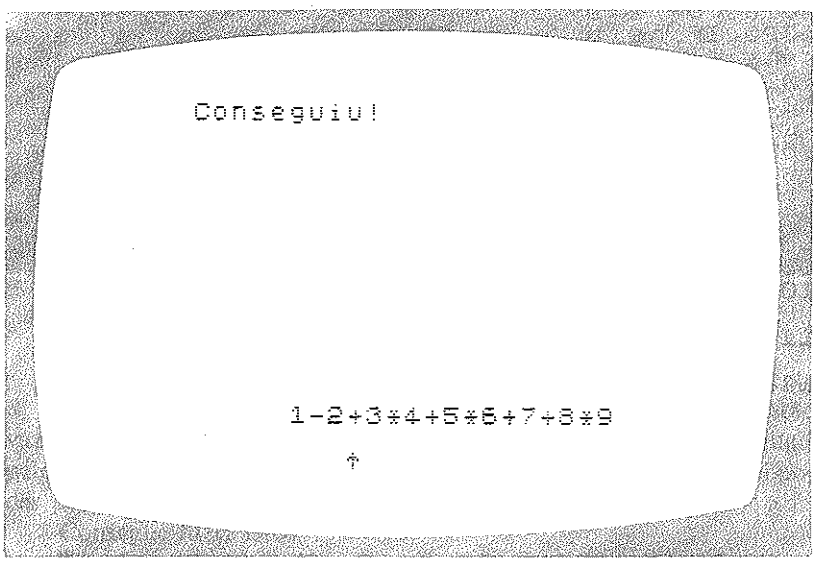
VERSÃO ZX SPECTRUM

Modifique as linhas seguintes:

```
20  LET C$ = "□□↑□□"
1050  IF VAL A$ = 100 THEN PRINT AT 3, 5; "Well done!": BEEP .3, 0:
      BEEP .3, 4: BEEP . 3, 7: BEEP 1.5, 10
```

SE USAR A VERSÃO DO ZX SPECTRUM

Faça tal como para o ZX81

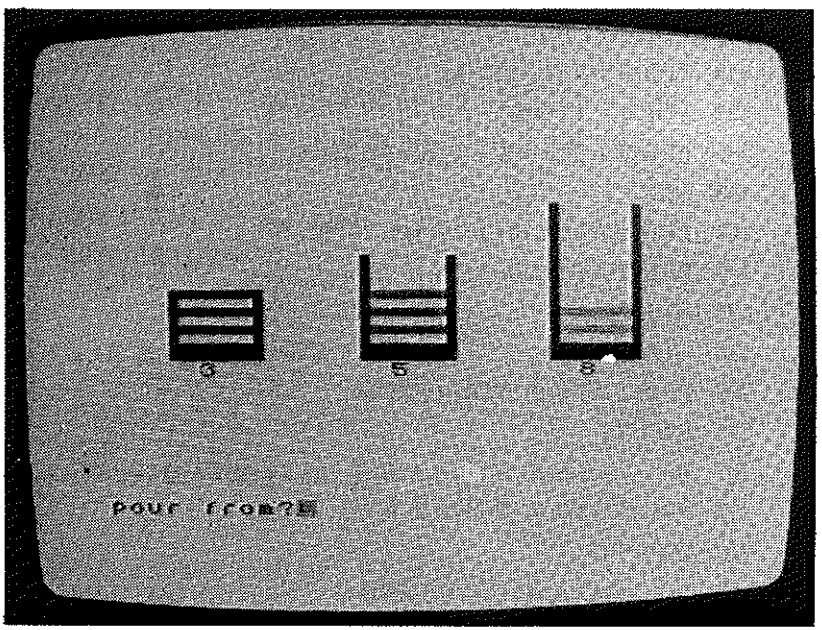


```
  10 LET n$="1 2 3 4 5 6 7 8 9"
  20 LET c$="  ↑  "
  30 LET h=1
 100 PRINT AT 16,10;n$
 110 PRINT AT 18,7+2*h;c$
 120 IF INKEY$<>"" THEN GO TO 12
0
 130 IF INKEY$="" THEN GO TO 130
 140 LET i$=INKEY$
 150 IF i$="5" THEN GO SUB 300
 160 IF i$="8" THEN GO SUB 400
 170 IF i$="b" THEN LET n$(2*h)=
"*"
 180 IF i$="k" THEN LET n$(2*h)=
"+"
 190 IF i$="j" THEN LET n$(2*h)=
"-"
 200 IF i$="v" THEN LET n$(2*h)=
"/"
 210 IF i$="x" THEN LET n$(2*h)=
" "
 220 IF i$="p" THEN GO TO 1000
 230 GO TO 100
 300 LET h=h-1+(h=1)
 310 PRINT AT 18,7+2*h;c$
 320 RETURN
 400 LET h=h+1-(h=8)
 410 GO TO 310
1000 LET a$=""
1010 FOR t=1 TO 17
1020 IF n$(t)<>" " THEN LET a$=a
$+n$(t)
1030 NEXT t
1040 PRINT AT 1,5;"O valor e ";V
AL a$
1050 IF VAL a$=100 THEN PRINT AT
 3,5;"Conseguiu!": BEEP .3,0: BE
EP .3,4: BEEP .3,7: BEEP 1.5,10
1060 IF VAL a$=100 THEN STOP
1070 FOR j=1 TO 50
1080 NEXT j
1090 PRINT AT 1,0;"
                  "
1100 PRINT AT 3,0;"
                  "
1110 GO TO 100
```

5

JUSTAS REPARTIÇÕES

fair shares



Você tem três copos com vinho rosé que contêm respectivamente 8, 5 e 3 medidas cada um. O copo de 8 está cheio mas os outros estão vazios. Você deve dividir o vinho em duas porções exactamente iguais pelos outros dois, vertendo o vinho de um copo para outro. Só vai poder parar quando um copo ficar cheio e o outro vazio.

Consegue fazer isto?

Este é um problema famoso que existe pelo menos desde o ano 14ØØ. Outro famoso problema foi pegar na garrafa para começar o problema propriamente dito...

```
    10  LET A$ = "□□□□"
→   20  LET B$ = "gG gG gG gG"
→   30  LET C$ = "g8□□□□g5"
→   40  LET D$ = "g8□□□□g5"
    50  DIM H (3)
    60  LET H (3) = 8
   100  FOR I = 1 TO 3
   110  LET K = 3 * I - (I > 1)
   120  FOR T = 1 TO K
   130  PRINT AT 13 - K + T, 10 * I - 7; C$
   140  NEXT T
   150  PRINT AT 14, 10 * I - 7; D$
   160  PRINT AT 15, 10 * I - 5; K
   170  NEXT I
   180  GOSUB 1000
→  200  PRINT AT 0, 0; "POUR FROM?"
→  205  INPUT F
   207  IF F < > 3 AND F < > 5 AND F < > 8 THEN GOTO 200
→  210  PRINT AT 0, 0; "POUR INTO?"
→  215  INPUT G
   217  IF G < > 3 AND G < > 5 AND G < > 8 THEN GOTO 210
   220  LET I = INT ( (F + 1) / 3)
   230  LET J = INT ( (G + 1) / 3)
   240  LET Q = H (I) + H (J)
   300  IF Q > G THEN GOSUB 400
   310  IF Q < = G THEN GOSUB 500
   320  GOSUB 1000
   330  IF H (2) = 4 AND H (3) = 4 THEN GOSUB 600
   340  GOTO 200
   400  LET H (J) = G
   410  LET H (I) = Q - G
```

```
420  RETURN
500  LET H (J) = Q
510  LET H (I) = 0
520  RETURN
600  PRINT AT 2, 10; "WELL DONE"
610  STOP
1000 FOR I = 1 TO 3
1010 FOR T = 1 TO H (I)
1020 PRINT AT 14 - T, 10 * I - 6; B$
1030 NEXT T
1040 FOR T = H (I) + 1 TO 3 * I - (I > 1)
1050 PRINT AT 14 - T, 10 * I - 6; A$
1060 NEXT T
1070 NEXT I
1090 RETURN
```

SE USAR A VERSÃO DO ZX81

RUN; escolha o copo para encher e esvaziar, usando o algarismo que lhe está por baixo; repita tudo até que tenha 4 medidas no copo de 5 e 4 medidas no copo de 8.

VERSÃO ZX SPECTRUM

Modifique as linhas seguintes:

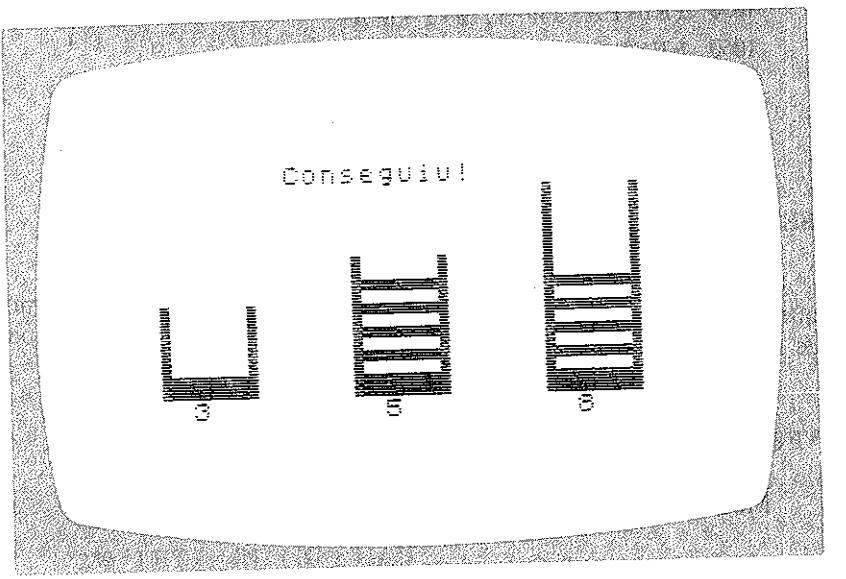
```
20  LET B$ = "g3 g3 g3 g3"
30  LET C$ = "g5 □ □ □ □ g5c"
40  LET D$ = "g5 g8c g8c g8c g8c g5c"
200 INPUT "Pour From?"; F
205 [apague esta linha]
210  INPUT "Pour Into?"; G
215 [apague esta linha]
```

Adicione estas:

```
 70  LET G = 8
1005  INK 2 + I
1080  INK 0
1025  IF I = INT ( (G + 1) / 3) THEN BEEP .05, 5 - T:
      BEEP .05, 5 - T - .3: BEEP .05, 5 - T - .7
```

SE USAR A VERSÃO DO SPECTRUM

Faça tal como para o ZX 81. O vinho muda de cor evidentemente...






```
 10  LET  a$="        "
 20  LET  b$="▀▀▀▀▀▀▀▀"
 30  LET  c$="▀▀  ▀▀  "
 40  LET  d$="▀▀▀▀▀▀▀▀"
 50  DIM  h(3)
 60  LET  h(3)=8
 70  LET  g=8
100  FOR  i=1  TO  3
110  LET  k=3*i-(i>1)
120  FOR  t=1  TO  k
130  PRINT  AT  13-k+t,10*i-7;c$
```

```
 140 NEXT t
 150 PRINT AT 14,10*i-7;d$
 160 PRINT AT 15,10*i-5;k
 170 NEXT i
 180 GO SUB 1000
 200 INPUT "Despejar de? ";f
 207 IF f<>3 AND f<>5 AND f<>8 T
HEN GO TO 200
 210 INPUT "Para onde? ";g
 217 IF g<>3 AND g<>5 AND g<>8 T
HEN GO TO 210
 220 LET i=INT ((f+1)/3)
 230 LET j=INT ((g+1)/3)
 240 LET q=h(i)+h(j)
 300 IF q>g THEN GO SUB 400
 310 IF q<=g THEN GO SUB 500
 320 GO SUB 1000
 330 IF h(2)=4 AND h(3)=4 THEN G
O SUB 600
 340 GO TO 200
 400 LET h(j)=g
 410 LET h(i)=q-g
 420 RETURN
 500 LET h(j)=q
 510 LET h(i)=0
 520 RETURN
 600 PRINT AT 5,10;"Conseguiu!"
 610 STOP
1000 FOR i=1 TO 3
1005 INK 2+i
1010 FOR t=1 TO h(i)
1020 PRINT AT 14-t,10*i-6;b$
1025 IF i=INT ((g+1)/3) THEN BEE
P .05,5-t: BEEP .05,5-t-.3: BEEP
 .05,5-t-.7
1030 NEXT t
1040 FOR t=h(i)+1 TO 3*i-(i>1)
1050 PRINT AT 14-t,10*i-6;a$
1060 NEXT t
1070 NEXT i
1080 INK 0
1090 RETURN
```

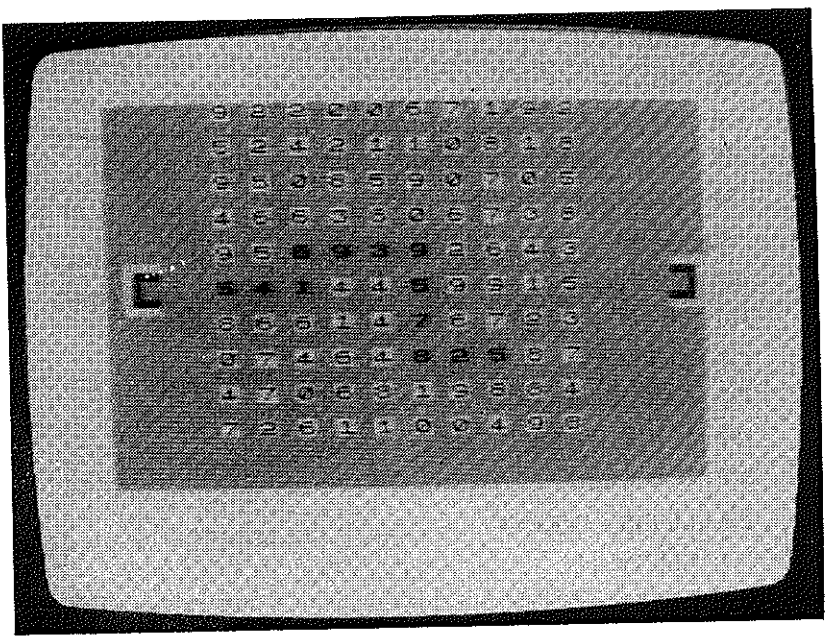
NOTAS

À medida que caminha pela floresta você vai acumulando uma carga mágica: para sair dela tem que conseguir ter uma carga mágica igual a zero. Consegui-lo-á?

6

A FLORESTA MÁGICA

the magic forest



Neste «puzzle» o computador desenha uma floresta quadrada com quadrados numerados. Cada um destes números representa o valor da «carga mágica» nesse local específico da floresta. Você vai movendo um cursor através da floresta, com o objectivo de chegar ao santuário do outro lado: o santuário é a baliza do outro lado. Á medida que vai passando por cada quadrado a carga mágica dele vai sendo *adicionada* à sua. Contudo, por ser mágica, a adição ignora as dezenas; se por exemplo 5 + 7 = 12 ele ignora o 1 e fica 2; se 8 + 6 é 14 ele ignora o 1 e fica 4. Deste modo você pode perder carga mágica já que, por exemplo, 3 + 7 = 1Ø e portanto é Ø, e lá se foi toda a carga mágica.

Consegue sair da floresta com uma carga mágica igual a zero?

```
→      1 FAST
       2 LET C = Ø
       5 DIM D (1Ø, 1Ø)
      1Ø FOR I = 1 TO 1Ø
      2Ø FOR J = 1 TO 1Ø
      3Ø LET D (I, J) = INT (1Ø * RND)
      35 PRINT AT 2 * I − 2, 2 * J + 4; D (I, J)
      4Ø NEXT J
      5Ø NEXT I
→     6Ø PRINT AT 9, 1; "g3 g6"; AT 1Ø, 1; "g8"; AT 11, 1; "g2 g7"
→     7Ø PRINT AT 9, 3Ø; "g6 g4"; AT 1Ø, 31; "g5"; AT 11, 3Ø; "g7 g1"
→     8Ø SLOW
     1ØØ LET P = 1Ø
     1Ø5 LET Q = 2
     11Ø LET N = Ø
     12Ø GOSUB 1ØØØ
     2ØØ IF INKEY$ < > " " THEN GOTO 2ØØ
     21Ø IF INKEY$ = " " THEN GOTO 21Ø
     22Ø LET I$ = INKEY$
     224 LET PØ = P
     226 LET QØ = Q
     23Ø IF I$ = "5" THEN LET Q = Q − 2
     24Ø IF I$ = "6" THEN LET P = P + 2
     25Ø IF I$ = "7" THEN LET P = P − 2
     26Ø IF I$ = "8" THEN LET Q = Q + 2
     262 LET C = C + 1
     265 IF P > 18 THEN GOTO 5ØØ
     27Ø IF Q < 6 OR Q > 24 THEN GOTO 3ØØ
     28Ø LET N = N + D (1 + P/2, Q/2 − 2)
     29Ø IF N > 9 THEN LET N = N − 1Ø
     3ØØ GOSUB 1ØØØ
```

```
 310  GOSUB 1020
 320  IF P = 10 AND Q = 30 THEN GOTO 2000
 400  GOTO 200
 500  LET P = P - 2
 510  PRINT AT 21, 0; "OUT OF BOUNDS"
 520  FOR T = 1 TO 50
 530  NEXT T
 540  PRINT AT 21, 0; "□□□□□□□□□□□□□□"
 550  GOTO 200
→ 1000  PRINT AT P, Q; CHR$ (156 + N)
 1010  RETURN
 1020  IF Q0 < 6 OR Q > = 24 THEN PRINT AT P0, Q0; "□"
 1030  IF Q0 > 5 AND Q0 < 25 THEN PRINT AT P0, Q0;
       D (1 + P0/2, Q0/2 - 2)
 1040  RETURN
 2000  IF N= 0 THEN PRINT AT 20, 0; "CONGRATULATIONS.
       YOU TOOK □", C; "□ MOVES"
 2010  IF N < > 0 THEN PRINT AT 20, 0; "SORRY, WRONG TOTAL - "
       "PLEASE TRY AGAIN."
```

SE USAR A VERSÃO DO ZX81

RUN; espere que a imagem do écran apareça. A floresta está no meio e você está do lado esquerdo, sentado nessa baliza. Há outra baliza do lado oposto onde você terá que chegar. Os números de floresta são as cargas mágicas.

Usando as quatro teclas de setas você pode mover-se através da floresta analizando ao mesmo tempo as adições que vai fazendo. Se escolher o caminho cuidadosamente vai conseguir com certeza atingir o objectivo com uma carga mágica apenas de 0.

O programa indica-lhe quantos passos teve que dar. Veja se consegue dar o menor número possível de passos.

VERSÃO ZX SPECTRUM

Modifique

```
 1  PAPER 7: INK Ø: CLS
60  PRINT AT 9, 1; "g4 g3c"; AT 1Ø, 1; "g5"; AT 11, 1; "g1 g3"
70  PRINT AT 9, 3Ø; "g3c g7c"; AT 1Ø, 31; "g5c"; AT 11, 3Ø; "g3 g2"
8Ø  [apague esta linha]
1ØØØ  PRINT AT P, Q; FLASH 1; CHR$ (48 + N)
```

Dê mais colorido ao seu «puzzle» fazendo:

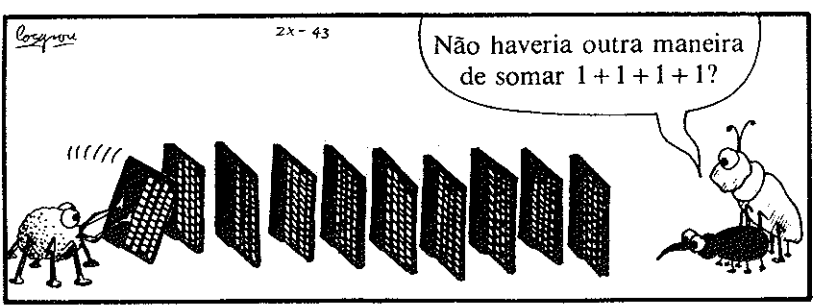
```
 8  PAPER 4
55  INK 2
75  INK Ø
77  PAPER 4
```

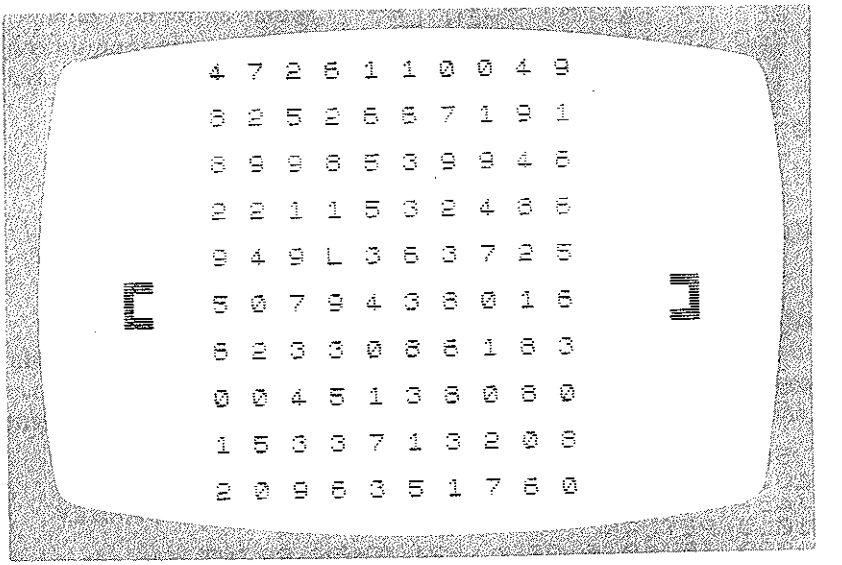
E também algum som com:

```
222  BEEP .3, 2 * VAL I$
```

SE USAR A VERSÃO DO ZX SPECTRUM

Faça o mesmo que para o ZX81.








```
  1 PAPER 7: INK 0: CLS
  2 LET c=0
  5 DIM d(10,10)
  8 PAPER 4
 10 FOR i=1 TO 10
 20 FOR j=1 TO 10
 30 LET d(i,j)=INT (10*RND)
 35 PRINT AT 2*i-2,2*j+4;d(i,j)
 40 NEXT j
 50 NEXT i
 55 INK 2
 60 PRINT AT 9,1;"▬";AT 10,1;"
▬";AT 11,1;"▬"
 70 PRINT AT 9,30;"▬";AT 10,31
;"▬";AT 11,30;"▬"
 75 INK 0
 77 PAPER 4
100 LET p=10
105 LET q=2
110 LET n=0
120 GO SUB 1000
200 IF INKEY$<>"" THEN GO TO 20
0
210 IF INKEY$="" THEN GO TO 210
220 LET i$=INKEY$
222 BEEP .3,2*VAL i$
224 LET p0=p
226 LET q0=q
230 IF i$="5" THEN LET q=q-2
240 IF i$="6" THEN LET p=p+2
```

```
 250 IF i$="7" THEN LET p=p-2
 260 IF i$="8" THEN LET q=q+2
 262 LET c=c+1
 265 IF p>18 THEN GO TO 500
 270 IF q<6 OR q>24 THEN GO TO 3
00
 280 LET n=n+d(1+p/2,q/2-2)
 290 IF n=19 THEN LET n=n-10
 300 GO SUB 1000
 310 GO SUB 1020
 320 IF p=10 AND q=30 THEN GO TO
 2000
 400 GO TO 200
 500 LET p=p-2
 510 PRINT AT 21,0;"Falhou!"
 520 FOR t=1 TO 50
 530 NEXT t
 540 PRINT AT 21,0;"
 "
 550 GO TO 200
1000 PRINT AT p,q; FLASH 1;CHR$
(48+n)
1010 RETURN
1020 IF q0<6 OR q>=24 THEN PRINT
 AT p0,q0;" "
1030 IF q0>5 AND q<25 THEN PRINT
 AT p0,q0;d(1+p0/2,q0/2-2)
1040 RETURN
2000 IF n=0 THEN PRINT AT 20,0;"
                Parabens!
Conseguiu em ";c;" movimentos"
2010 IF n<>0 THEN PRINT AT 19,0;
"              Que pena!...
       O total estava errado","
     Tente outra vez
```

Quantas rainhas de xadrez precisa para atacar, simultâneamente, todo o tabuleiro?

7

RAINHAS DOMINADORAS

queens dominant

Este programa permite que você distribua rainhas de xadrez num tabuleiro, mostrando quais os quadrados que elas estão a atacar. O objectivo é conseguir atacar todos os quadrados com o menor número possível de rainhas. É possível consegui-lo com apenas cinco rainhas.

VERSÃO ZX81

```
 10  DIM B (8, 8)
 20  FOR I = 1 TO 8
 30  FOR J = 1 TO 8
→ 40  PRINT AT 2 * I, 2 * J + 8; "gH"
 50  NEXT J
 60  NEXT I
 70  LET P = 2
 80  LET Q = 9
 90  GOSUB 2000
 92  LET P0 = P
 94  LET Q0 = Q
100  IF INKEY$ < > " " THEN GOTO 100
110  IF INKEY$ = " " THEN GOTO 110
120  LET I$ = INKEY$
130  IF I$ = "Q" THEN GOTO 200
140  IF I$ = "5" THEN LET Q = Q - 2
150  IF I$ = "6" THEN LET P = P + 2
160  IF I$ = "7" THEN LET P = P - 2
170  IF I$ = "8" THEN LET Q = Q + 2
180  GOSUB 1000
```

```
    185 GOSUB 1020
    190 GOTO 92
→   200 PRINT AT P, Q + 1; " Q "
    205 LET I = P/2
    207 LET J = (Q − 7) / 2
    210 LET B (I, J) = 1
    220 FOR T = 1 TO 8
    230 IF B (I, T) = 0 THEN LET B (I, T) = −1
    240 IF B (T, J) = 0 THEN LET B (T, J) = −1
    250 NEXT T
    260 FOR F = −1 TO 1 STEP 2
    270 FOR T = −8 TO 8
    280 LET U = I + F * T
    290 LET V = J + T
    300 IF U < 1 OR U > 8 OR V < 1 OR V > 8 THEN GOTO 320
    310 IF B (U, V) = 0 THEN LET B (U, V) = −1
    320 NEXT T
    330 NEXT F
    340 GOSUB 2000
    350 GOTO 92
→  1000 PRINT AT P, Q; " > "
   1010 RETURN
→  1020 PRINT AT P0, Q0; "□"
   1030 RETURN
   2000 FOR I = 1 TO 8
   2010 FOR J = 1 TO 8
   2020 IF B (I, J) = −1 THEN PRINT AT 2 * I, 2 * J + 8; "+"
   2030 NEXT J
   2040 NEXT I
   2050 RETURN
```

SE USAR A VERSÃO ZX81

RUN. Aparece o tabuleiro. Use as teclas das setas para mover o cursor sobre o tabuleiro. Se premir "Q" aparece uma rainha no qua-

drado, mostrando ao mesmo tempo que quadrados está a atacar. Adicione outra rainha... continue até conseguir atacar todos os quadrados.

VERSÃO ZX SPECTRUM

Modifique as linhas seguintes:

```
  40  PRINT AT 2 * I, 2 * J + 8; "□"
 200  PRINT AT P, Q + 1; "gQ"
1000  [apague esta linha]
1020  [apague esta linha]
```

Adicione algum colorido com:

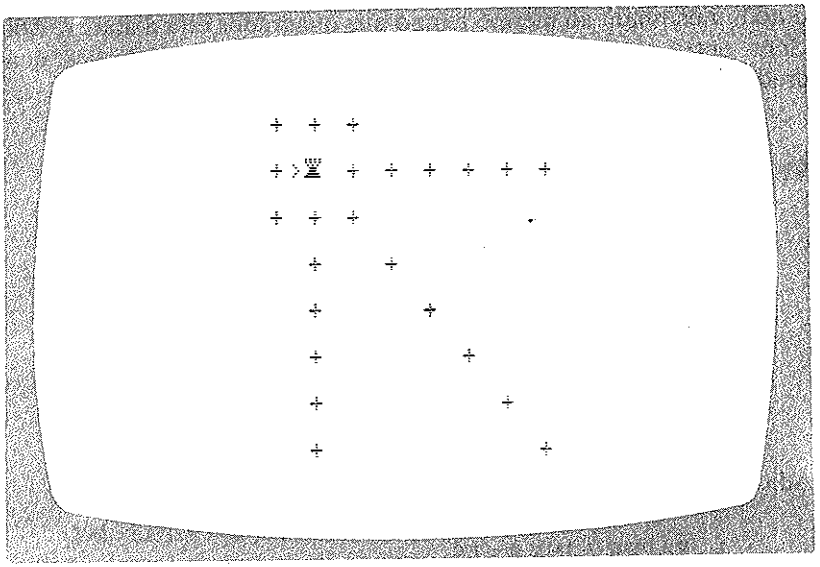
```
   1  PAPER 7: INK 0: CLS
  35  PAPER 5 + I + J - 2 * INT ( (I + J) / 2)
1000  PRINT PAPER 7; AT P, Q; ">"
1020  PRINT PAPER 7; AT P0, Q0; "□"
```

Defina o gráfico da rainha com:

```
   1  GOSUB 3000
3000  DATA 170, 170, 84, 56, 56, 124, 254, 0
3010  FOR N = 0 TO 7
3020  READ X
3025  POKE USR "Q" + N, X
3030  NEXT N
3080  RETURN
```

SE USAR A VERSÃO DO ZX SPECTRUM

Logo após ligar o computador faça GO TO 3000 para definir os gráficos e depois RUN. Depois disto faça tal como para o ZX81.






```
  1 PAPER 7: INK 0: CLS
  2 GO SUB 3000
 10 DIM b(8,8)
 20 FOR i=1 TO 8
 30 FOR j=1 TO 8
 35 PAPER 5+i+j-2*INT ((i+j)/2)
 40 PRINT AT 2*i,2*j+8;" "
 50 NEXT j
 60 NEXT i
 70 LET p=2
 80 LET q=9
 90 GO SUB 2000
 92 LET p0=p
 94 LET q0=q
100 IF INKEY$<>"" THEN GO TO 10
0
110 IF INKEY$="" THEN GO TO 110
120 LET i$=INKEY$
130 IF i$="q" THEN GO TO 200
140 IF i$="5" THEN LET q=q-2
150 IF i$="6" THEN LET p=p+2
160 IF i$="7" THEN LET p=p-2
170 IF i$="8" THEN LET q=q+2
180 GO SUB 1000
185 GO SUB 1020
190 GO TO 92
200 PRINT AT p,q+1;"♛"
205 LET i=p/2
207 LET j=(q-7)/2
210 LET b(i,j)=1
```

```
 220 FOR t=1 TO 8
 230 IF b(i,t)=0 THEN LET b(i,t)
=-1
 240 IF b(t,j)=0 THEN LET b(t,j)
=-1
 250 NEXT t
 260 FOR f=-1 TO 1 STEP 2
 270 FOR t=-8 TO 8
 280 LET u=i+f*t
 290 LET v=j+t
 300 IF u<1 OR u>8 OR v<1 OR v>8
 THEN GO TO 320
 310 IF b(u,v)=0 THEN LET b(u,v)
=-1
 320 NEXT t
 330 NEXT f
 340 GO SUB 2000
 350 GO TO 92
1000 PRINT PAPER 7;AT p,q;">"
1010 RETURN
1020 PRINT PAPER 7;AT p0,q0;" "
1030 RETURN
2000 FOR i=1 TO 8
2010 FOR j=1 TO 8
2020 IF b(i,j)=-1 THEN PRINT AT
2*i,2*j+8;"+"
2030 NEXT j
2040 NEXT i
2050 RETURN
3000 DATA 170,170,84,56,56,124,2
54,0
3010 FOR n=0 TO 7
3020 READ x
3025 POKE USR "q"+n,x
3030 NEXT n
3040 RETURN
```

NOTAS

NOTAS

8

BISPOS RUIDOSOS

bishops blatant

Este programa é semelhante ao das RAINHAS DOMINADORAS mas com bispos de xadrez.

VERSÃO ZX81

Comece como na versão do ZX81 das RAINHAS DOMINADORAS fazendo as modificações seguintes:

```
13Ø  IF I$ = "B" THEN GOTO 2ØØ
2ØØ  PRINT AT P, Q + 1; " B "
22Ø-25Ø [apague estas linhas]
```

SE USAR A VERSÃO DO ZX81

Faça tal como fez para as RAINHAS DOMINADORAS excepto que agora terá que premir "B" para fixar um bispo.

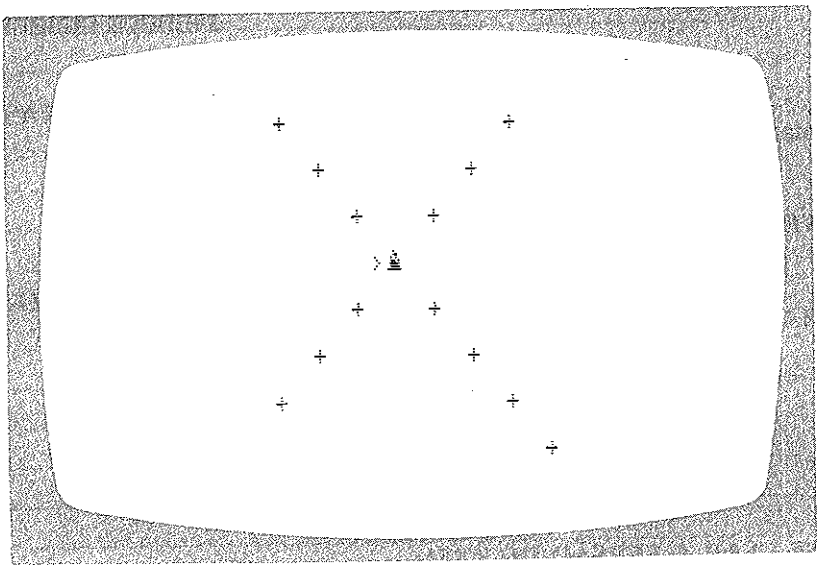
VERSÃO ZX SPECTRUM

Modifique as seguintes linhas na versão do Spectrum das RAINHAS DOMINADORAS:

```
13Ø  IF I$ = "B" THEN GOTO 2ØØ
2ØØ  PRINT AT P, Q + 1; "gB"
22Ø-25Ø [apague estas linhas]
3ØØØ  DATA 16, 24, 4Ø, 52, 6Ø, 6Ø, 126, Ø
3Ø25  POKE USR "B" + N, X
```

SE USAR A VERSÃO DO ZX SPECTRUM

Faça GO TO 3000 para definir os gráficos. Após isso, tudo funciona como nas RAINHAS DOMINADORAS excepto que usa "B" para fixar os bispos.






```
  1 PAPER 7: INK 0: CLS
  2 GO SUB 3000
 10 DIM b(8,8)
 20 FOR i=1 TO 8
 30 FOR j=1 TO 8
 35 PAPER 5+i+j-2*INT ((i+j)/2)
 40 PRINT AT 2*i,2*j+8;" "
 50 NEXT j
 60 NEXT i
 70 LET p=2
 80 LET q=9
 90 GO SUB 2000
 92 LET p0=p
 94 LET q0=q
100 IF INKEY$<>"" THEN GO TO 10
0
110 IF INKEY$="" THEN GO TO 110
120 LET i$=INKEY$
130 IF i$="b" THEN GO TO 200
140 IF i$="8" THEN LET q=q-2
150 IF i$="6" THEN LET p=p+2
```

```
 160  IF  i$="7" THEN LET  p=p-2
 170  IF  i$="8" THEN LET  q=q+2
 180  GO SUB  1000
 185  GO SUB  1020
 190  GO TO  92
 200  PRINT AT  p,q+1;"å"
 205  LET  i=p/2
 207  LET  j=(q-7)/2
 210  LET  b(i,j)=1
 260  FOR  f=-1 TO  1 STEP  2
 270  FOR  t=-8 TO  8
 280  LET  u=i+f*t
 290  LET  v=j+t
 300  IF  u<1  OR  u>8 OR  v<1  OR  v>8
 THEN  GO TO  320
 310  IF  b(u,v)=0 THEN LET  b(u,v)
=-1
 320  NEXT  t
 330  NEXT  f
 340  GO SUB  2000
 350  GO TO  92
1000  PRINT PAPER  7;AT  p,q;">"
1010  RETURN
1020  PRINT PAPER  7;AT  p0,q0;"  "
1030  RETURN
2000  FOR  i=1 TO  8
2010  FOR  j=1 TO  8
2020  IF  b(i,j)=-1 THEN PRINT AT
2*i,2*j+8;"+"
2030  NEXT  j
2040  NEXT  i
2050  RETURN
3000  DATA  16,24,40,52,60,60,126,
0
3010  FOR  n=0 TO  7
3020  READ  x
3025  POKE  USR  "b"+n,x
3030  NEXT  n
3040  RETURN
```

NOTAS

9

TORRES DOMINANTES

rooks rampant

Este programa funciona tal como o das RAINHAS DOMINADORAS (RD) ou o dos BISPOS RUIDOSOS (BR) — mas com torres.

VERSÃO ZX81

Comece como com as RAINHAS DOMINADORAS na versão do ZX81 mas modifique para:

```
13Ø  IF I$ = "R" THEN GOTO 2ØØ
2ØØ  PRINT AT P, Q + 1; " R"
26Ø-33Ø [apague estas linhas]
```

SE USAR A VERSÃO DO ZX81

Faça como para RD e BR mas prima "R" para fixar as torres.

VERSÃO ZX SPECTRUM

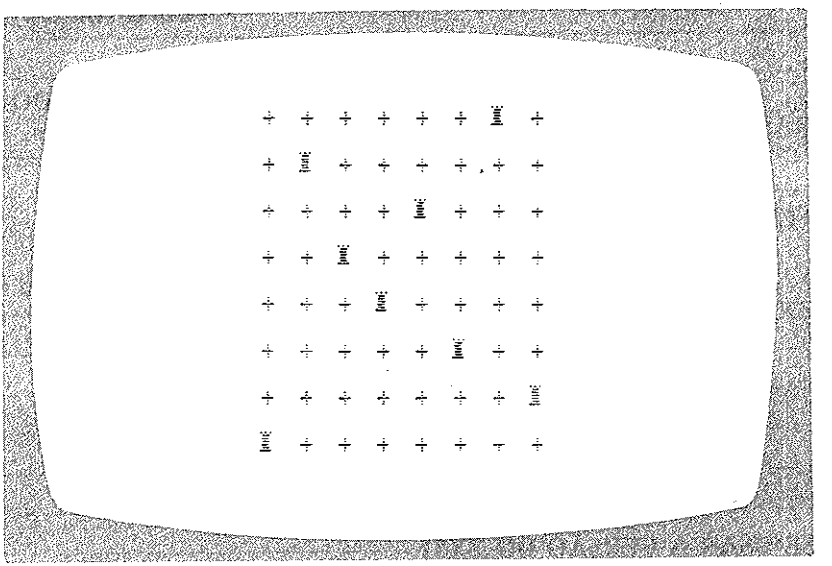
Modifique as seguintes linhas da versão ZX Spectrum das RAINHAS DOMINADORAS:

```
13Ø  IF I$ = "R" THEN GOTO 2ØØ
2ØØ  PRINT AT P, Q + 1; "gR"
26Ø-33Ø [apague estas linhas]
3ØØØ DATA 84, 56, 56, 56, 56, 56, 124, Ø
3Ø25 POKE USR "R" + N, X
```

SE USAR A VERSÃO DO ZX SPECTRUM

Logo após a ligação faça GO TO 3000 para definir os gráficos. A seguir RUN e use "R" para fixar as torres.

Vai precisar de pelo menos oito delas.






```
  1 PAPER 7: INK 0: CLS
  2 GO SUB 3000
 10 DIM b(8,8)
 20 FOR i=1 TO 8
 30 FOR j=1 TO 8
 35 PAPER 5+i+j-2*INT ((i+j)/2)
 40 PRINT AT 2*i,2*j+8;" "
 50 NEXT j
 60 NEXT i
 70 LET p=2
 80 LET q=9
 90 GO SUB 2000
 92 LET p0=p
 94 LET q0=q
100 IF INKEY$<>"" THEN GO TO 10
0
110 IF INKEY$="" THEN GO TO 110
120 LET i$=INKEY$
130 IF i$="r" THEN GO TO 200
140 IF i$="5" THEN LET q=q-2
150 IF i$="6" THEN LET p=p+2
```

```
 160 IF i$="7" THEN LET p=p-2
 170 IF i$="8" THEN LET q=q+2
 180 GO SUB 1000
 185 GO SUB 1020
 190 GO TO 92
 200 PRINT AT p,q+1;"▮"
 205 LET i=p/2
 207 LET j=(q-7)/2
 210 LET b(i,j)=1
 220 FOR t=1 TO 8
 230 IF b(i,t)=0 THEN LET b(i,t)
=-1
 240 IF b(t,j)=0 THEN LET b(t,j)
=-1
 250 NEXT t
 340 GO SUB 2000
 350 GO TO 92
1000 PRINT PAPER 7;AT p,q;">"
1010 RETURN
1020 PRINT PAPER 7;AT p0,q0;" "
1030 RETURN
2000 FOR i=1 TO 8
2010 FOR j=1 TO 8
2020 IF b(i,j)=-1 THEN PRINT AT
2*i,2*j+8;"+"
2030 NEXT j
2040 NEXT i
2050 RETURN
3000 DATA 84,56,56,56,56,56,124,
0
3010 FOR n=0 TO 7
3020 READ x
3025 POKE USR "r"+n,x
3030 NEXT n
3040 RETURN
```

NOTAS

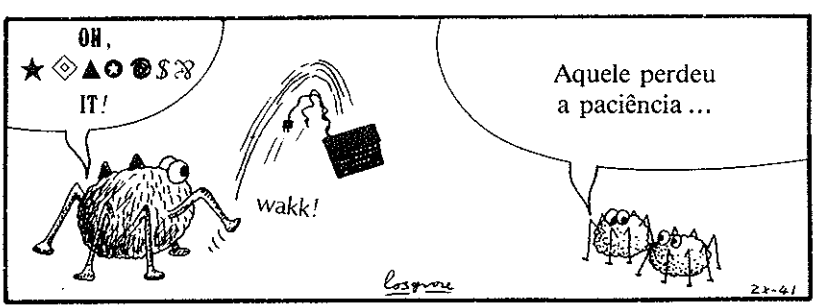
Prometo que esta é a última variação dos programas anteriores...

SENHORAS TÍMIDAS

shy ladies

Neste «puzzle» não há mais trabalho. O programa é o mesmo do das RAINHAS DOMINADORAS mas o objectivo é diferente: quantas rainhas se podem colocar no tabuleiro sem que se ataquem umas às outras? Consegue pôr oito?

Podemos fazer o mesmo para os bispos e para as torres... mas acho que agora é melhor fazer algo de novo...

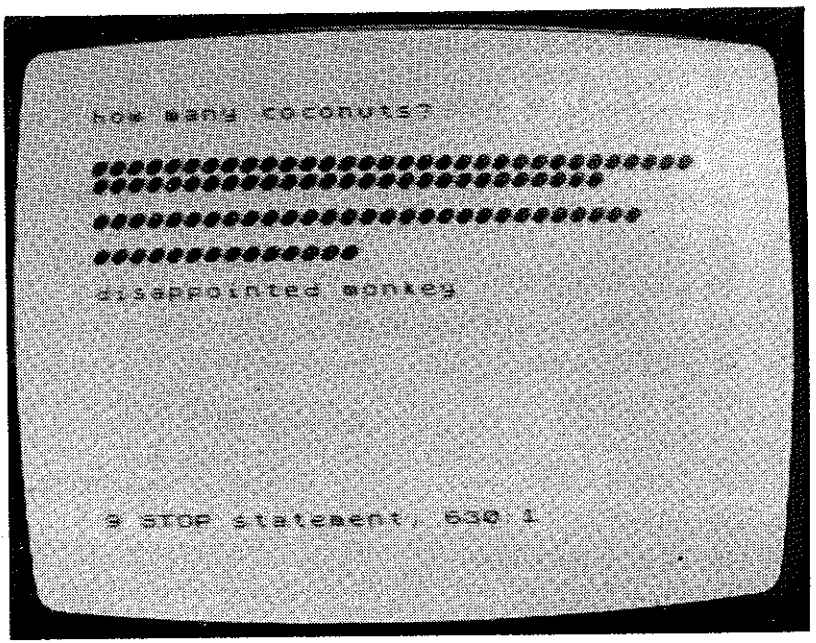


NOTAS



«PUZZLE» DO MACACO

monkey puzzle



Cinco astronautas caíram num asteróide deserto onde apenas habita um macaco chamado NYS. O que há para comer são inúmeros cocos.

Os astronautas pegam em determinada quantidade de cocos e fazem uma pilha com eles.

Durante a noite o primeiro astronauta acorda e decide comer metade dos cocos da pilha. Só que na divisão sobra um coco que ele decide dar ao macaco.

Pouco tempo depois o segundo astronauta acorda e faz exactamente o mesmo. De novo existe um coco a mais que ele dá ao macaco.

Os terceiro e quarto astronautas fazem exactamente o mesmo. O macaco continua a ter a mesma sorte; mas quando chega a vez do quinto astronauta ele fica um pouco intrigado porque sobrou apenas um coco para ele. Ele acha que havia mais cocos quando foi para a cama, na noite anterior, mas como ainda é madrugada a sua cabeça não está a funcionar muito bem.

Quantos cocos havia no início?

VERSÃO ZX81

```
→   10 PRINT "HOW MANY COCONUTS"
    20 INPUT C
    30 GOSUB 500
    40 FOR T = 1 TO 5
    50 IF C = 1 THEN GOTO 700
    60 LET C = C - 1
    70 LET K = INT (C/2)
    80 IF 2 * K < > C THEN GOTO 600
    90 LET C = K
   100 GOSUB 500
   110 NEXT T
   120 PRINT
   130 PRINT
   140 PRINT "TOO MANY"
   150 STOP
   500 PRINT
   510 PRINT
   520 FOR I = 1 TO C
   530 PRINT "*";
   540 NEXT I
   550 RETURN
   600 PRINT
   610 PRINT
   620 PRINT "DISAPPOINTED MONKEY"
   630 STOP
```

```
700 PRINT
710 PRINT
720 IF T = 5 THEN PRINT "THAT " " S RIGHT"
730 IF T < > 5 THEN PRINT "NOPE, TOO SMALL"
```

SE USAR A VERSÃO DO ZX81

RUN. Pode começar a adivinhar o número de cocos. O computador percorre todos os acontecimentos da noite, mostrando no écran as pilhas de cocos e verifica se toda a gente fica com a sua parte. Se você tiver acertado o computador avisa. Se não tiver acertado o computador responde-lhe com um incentivo para continuar.

VERSÃO ZX SPECTRUM

Modifique estas linhas:

```
10 PRINT "How many coconuts?"
720 IF T = 5 THEN PRINT FLASH 1; "THAT S RIGHT!!!!":
    GOTO 800
```

Adicione estas:

```
740 STOP
800 FOR I = 1 TO 50
810 BEEP .1, 40 * (.5 - RND): BORDER 7 * RND
820 NEXT I
```

Modifique também esta:

```
530 PRINT "gN";
```

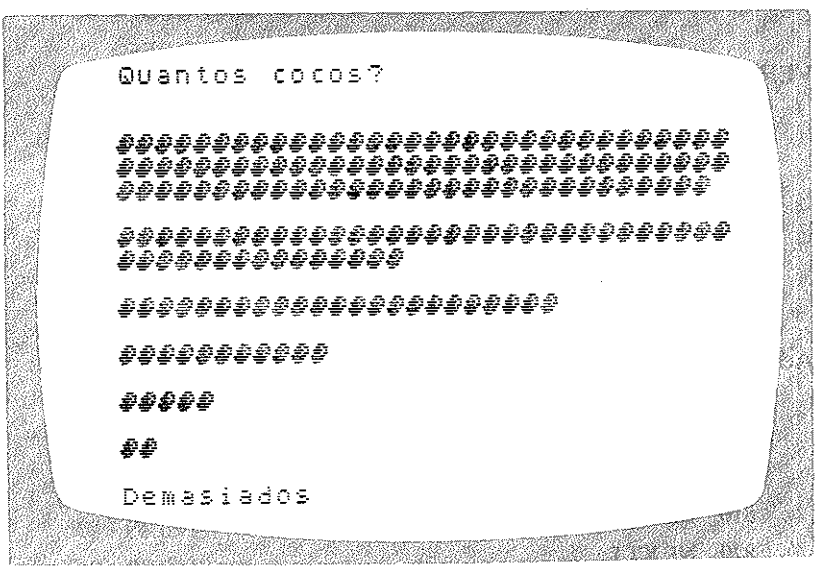
E adicione também estas:

```
  1 GOSUB 1000
1000 DATA 28, 54, 122, 118, 252, 248, 112, 0
1010 FOR N = 0 TO 7
```

```
1020 READ X
1030 POKE USR "N" + N, X
1040 NEXT N
1050 RETURN
```

SE USAR A VERSÃO DO ZX SPECTRUM

Faça GO TO 1000 para definir graficamente os cocos e depois RUN. Além disso as instruções são as mesmas que para a versão do ZX81.






```
  1 GO SUB 1000
 10 PRINT "Quantos cocos? "
 20 INPUT c
 30 GO SUB 500
 40 FOR t=1 TO 5
 50 IF c=1 THEN GO TO 700
 60 LET c=c-1
 70 LET k=INT (c/2)
 80 IF 2*k<>c THEN GO TO 600
 90 LET c=k
100 GO SUB 500
110 NEXT t
120 PRINT
```

```
 130 PRINT
 140 PRINT "Demasiados"
 150 STOP
 500 PRINT
 510 PRINT
 520 FOR i=1 TO c
 530 PRINT "#";
 540 NEXT i
 550 RETURN
 600 PRINT
 610 PRINT
 620 PRINT "Macaco desapontado"
 630 STOP
 700 PRINT
 710 PRINT
 720 IF t=5 THEN PRINT FLASH 1;"
Esta CORRECTISSIMO!!!!!": GO TO
800
 730 IF t<>5 THEN PRINT "Poucos"
 740 STOP
 800 FOR i=1 TO 50
 810 BEEP .1,40*(.5-RND): BORDER
 7*RND
 820 NEXT i
1000 DATA 28,54,122,118,252,248,
112,0
1010 FOR n=0 TO 7
1020 READ x
1030 POKE USR "n"+n,x
1040 NEXT n
1050 RETURN
```

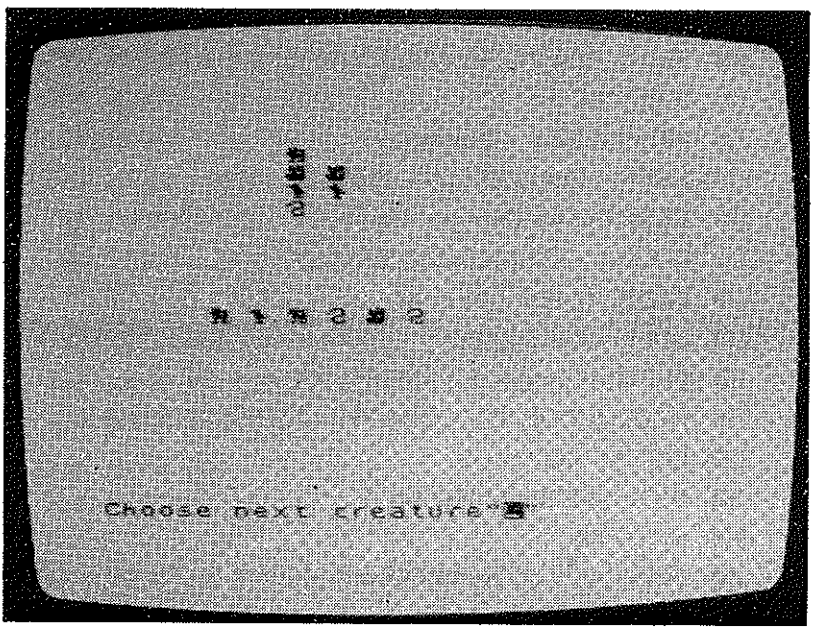
NOTAS

NOTAS

12

ENCADEAMENTO ECOLÓGICO

ecological chain



Você tem três cães, três gatos, três pássaros e três minhocas. Alinhe-os de modo que nenhum cão esteja a seguir a um gato, nenhum gato a seguir a um pássaro e nenhum pássaro a seguir a uma minhoca (por razões óbvias se assim não for eles zangam-se) — *e* não podem estar dois gatos juntos, nem dois cães, nem dois pássaros, nem duas minhocas. Mesmo uma minhoca tem o seu orgulho.

VERSÃO ZX81

```
    10  DIM K (4)
→   20  PRINT AT 3, 10; "D □ D □ D"
→   30  PRINT AT 4, 10; "C □ C □ C"
→   40  PRINT AT 5, 10; "B □ B □ B"
→   50  PRINT AT 6, 10; "W □ W □ W"
    60  LET C = 0
    70  LET Y$ = "G"
→  100  PRINT AT 0, 0; "CHOOSE WHICH CREATURE"
→  102  INPUT X$
→  104  PRINT AT 0, 0; " [21espaços] "
→  110  LET N = CODE X$ - 37 - 22 * (X$ = "W")
   120  LET K (N) = K (N) + 1
   130  IF K (N) > 3 THEN GOTO 500
→  140  PRINT AT 12, 6 + 2 * C; X$
   150  LET C = C + 1
   160  PRINT AT 7 - N, 16 - 2 * K (N); "□"
   180  IF ABS (CODE X$ - CODE Y$) < 2 THEN GOTO 600
   190  LET Y$ = X$
   200  IF C = 12 THEN GOTO 400
   210  GOTO 100
→  400  PRINT AT 0, 8; "WELL DONE"
   420  STOP
   500  PRINT AT 0, 8; "ILLEGAL MOVE"
   510  PAUSE 50
   520  PRINT AT 0, 8; " [12espaços] "
   530  GOTO 100
   600  PRINT AT 0, 8; "AAAAAAAAAGGGGGGGHHHHH"
```

SE USAR A VERSÃO DO ZX81

RUN. Os animais disponíveis são mostrados pelas seguintes iniciais C — gato, D — cão, A — minhoca e B — pássaro. Escolha com qual quer começar premindo a respectiva letra. Isso aparecerá no écran. Continue: tenha atenção às mensagens que vão aparecendo.

VERSÃO ZX SPECTRUM

Modifique as linhas seguintes:

```
20  FOR I = 3 TO 6
30  PRINT AT I, 10; INK I; CHR$ (150 - I) + "□" +
    CHR$ (150 - I) + "□" + CHR$ (150 - I)
40  NEXT I
50  [apague estas linhas]
100 INPUT "Choose which creature"; X$
102 [apague estas linhas]
104 [apague estas linhas]
110 LET N = CODE X$ - 96
140 PRINT AT 12, 6 + 2 * C; INK 7 - N; CHR$ (CODE X$ + 47)
400 PRINT AT 0, 8; FLASH 1; "Well Done!"
```

Adicione estas:

```
410 BEEP .3, 0: BEEP .3, 4: BEEP .3, 7: BEEP 1.5, 10
610 FOR I = 1 TO 10
620 BEEP I/15, - I
630 NEXT I
```

Defina os caracteres gráficos dos animais com:

```
  1 GOSUB 1000
1000 DATA 60, 66, 130, 28, 32, 34, 28, 0, 16, 112, 56, 28,
     30, 8, 28, 0, 74, 122, 122, 50, 250, 254, 124,
     0,80, 118, 242, 126, 62, 34, 102, 0
1010 FOR T = 1 TO 4
1020 FOR N = 0 TO 7
1030 READ X
1040 POKE USR CHR$ (T + 143) + N, X
1050 NEXT N
1060 NEXT T
1070  RETURN   610 STOP
```

SE USAR A VERSÃO DO ZX SPECTRUM

Inicie com GO TO 1000 para definir os gráficos dos animais. Depois faça RUN e quando lhe perguntar que animal quer alinhar introduza:

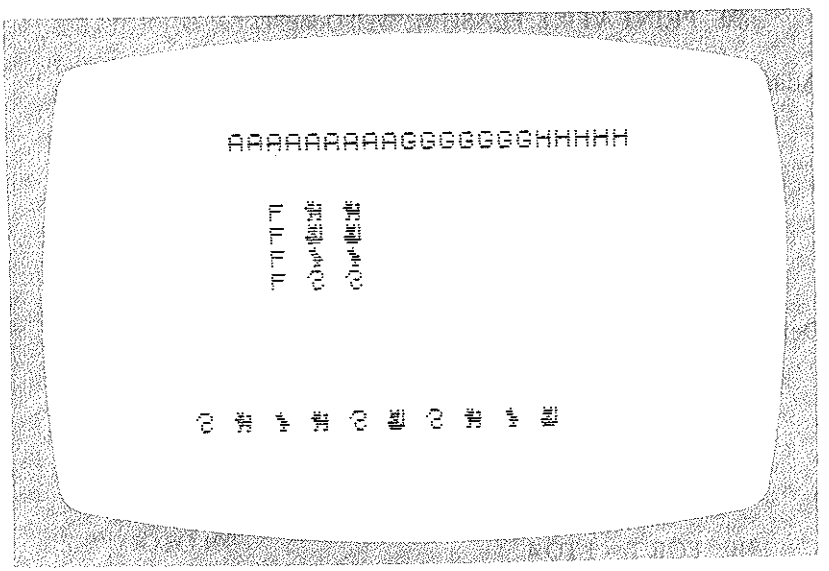
A para a minhoca

B para pássaro

C para gato

D para cão

tudo o resto é como na versão do ZX81.



```
  1 GO SUB 1000
 10 DIM k(4)
 20 FOR i=3 TO 6
 30 PRINT AT i,10; INK 1;CHR$ (
150-i)+" "+CHR$ (150-i)+" "+CHR$
 (150-i)
```

```
  40 NEXT i
  60 LET c=0
  70 LET y$="g"
 100 INPUT "Escolha um animal ";
x$
 110 LET n=CODE x$-96
 120 LET k(n)=k(n)+1
 130 IF k(n)>3 THEN GO TO 500
 140 PRINT AT 12,6+2*c; INK 7-n;
CHR$ (CODE x$+47)
 150 LET c=c+1
 160 PRINT AT 7-n,18-2*k(n);" "
 180 IF ABS (CODE x$-CODE y$)<2
THEN GO TO 600
 190 LET y$=x$
 200 IF c=12 THEN GO TO 400
 210 GO TO 100
 400 PRINT AT 0,8; FLASH 1;"Cons
eguiu!"
 410 BEEP .3,0: BEEP .3,4: BEEP
.3,7: BEEP 1.5,10
 420 STOP
 500 PRINT AT 0,8;"Movimento pro
ibido"
 510 PAUSE 0
 520 PRINT AT 0,8;"
                  "
 530 GO TO 100
 600 PRINT AT 0,8;"AAAAAAAAAGGGG
GGGHHHHHH"
 610 STOP
 620 BEEP i/15,-i
 630 NEXT i
1000 DATA 60,66,130,28,32,34,28,
0,16,112,56,28,30,8,28,0,74,122,
122,50,250,254,124,0,80,118,242,
126,62,34,102,0
1010 FOR t=1 TO 4
1020 FOR n=0 TO 7
1030 READ x
1040 POKE USR CHR$ (t+143)+n,x
1050 NEXT n
1060 NEXT t
1070 RETURN
```

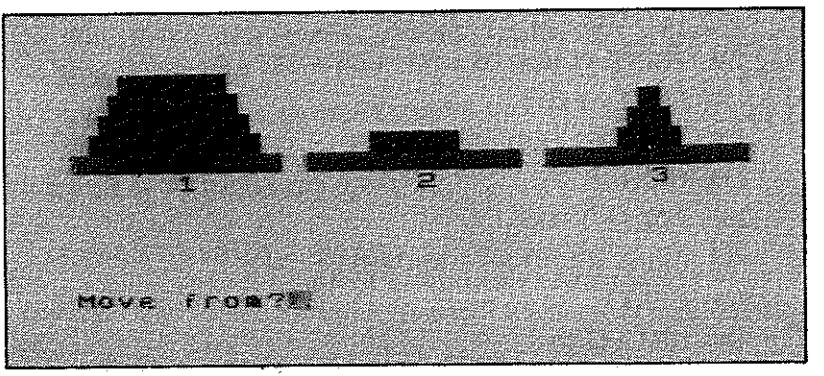
NOTAS

Este é muito velho, mas muito engraçado e graficamente muito bonito...

13

TORRE DE HANÓI COM BOLOS

tower of Hanoi with pancakes



Você tem três travessas. Numa tem um monte de bolos de diversos tamanhos 1, 2, 3, 4... amontoados de modo que os de baixo sejam maiores do que os de cima. O objectivo é transferir essa pilha de uma travessa para outra tendo em atenção o seguinte:

a) Movendo um bolo de cada vez;
b) Colocando o bolo numa travessa vazia ou em cima de um bolo *maior*. Àparte estas regras você pode fazer quantas tentativas quiser.

VERSÃO ZX81

```
    10  LET E$ = "■■■■■■■■■■"
    15  LET F$ = "□□□□□□□□□□"
→   18  LET G$ = "gH gH gH gH gH gH gH gH gH gH"
→   20  PRINT AT 0, 0; "INPUT SIZE"
→   22  INPUT S
→   24  PRINT AT 0, 0; " [10 espaços] "
```

```
    25  LET V = INT ( (11 − S) / 2 )
    30  DIM D$ (S, S + 2)
    40  FOR T = S TO 1 STEP − 2
    50  LET D$ (T) = F$ (TO (S − T) / 2 ) + E$ (TO T)
→   60  IF T < S THEN LET D$ (T + 1) = F$ (TO (S − T) / 2 − 1 ) + "g8" +
        E$ (TO T) + "g5"
    70  NEXT T
→   80  IF S = 2 * INT (S/2) THEN LET D$ (1) =
        F$ (TO (S − 2) / 2 ) + "g8 g5"
→  100  PRINT AT 16, 0; G$ + "□" + G$ + "□" + G$
   110  FOR I = 1 TO 3
   120  PRINT AT 17, 11 * I − 6; I
   130  NEXT I
   200  FOR I = 1 TO S
   210  PRINT AT 15 − S + I, V; D$ (I)
   220  NEXT I
   300  DIM A (3, S)
   310  FOR I = 1 TO S
   320  LET A (1, I) = S − I + 1
   330  NEXT I
   340  DIM H (3)
   350  LET H (1) = S
→  500  PRINT AT 0, 0; "MOVE FROM"
→  502  INPUT F
→  504  PRINT AT 0, 0; "MOVE TO □ □"
→  506  INPUT G
→  508  PRINT AT 0, 0; " [9 espaços ] "
   511  GOTO 1000
   512  LET R = A (F, H (F) )
   514  LET A (F, H (F) ) = 0
   516  LET A (G, H (G) + 1) = R
   520  LET H (F) = H (F) − 1
   530  LET H (G) = H (G) + 1
   540  LET X = R + 1 + .5 * (10 − R)
```

```
    545  IF X > S + 2 THEN LET X = S + 2
    550  PRINT AT 15 - H (F), 11 * F - 11 + V; F$ (TO X);
         AT 16 - H (G), 11 * G - 11 + V; D$ (R) (TO X)
→   555  PRINT AT 16, 0; G$
    557  IF H (2) = S OR H (3) = S THEN GOTO 1200
    560  GOTO 500
   1000  IF H (G) = 0 THEN GOTO 512
   1005  IF H(F) = 0 THEN GOTO 500
   1010  IF A (F, H (F) ) > = A (G, H (G) ) THEN GOTO 500
   1020  GOTO 512
→  1200  PRINT AT 1, 10; "SUCCESS"
```

SE USAR A VERSÃO DO ZX81

Comece com RUN. Espere que a torre fique desenhada. Escolha o bolo da pilha que vai transferir e para onde premindo os respectivos números. Continue até conseguir transferir a pilha toda para outra travessa.

VERSÃO DO ZX SPECTRUM

Modifique as seguintes linhas

```
 18  [apague esta linha]
 20  INPUT "Size?"; S
 22  [apague esta linha]
 24  [apague esta linha]
 60  IF T < S THEN LET D$ (T + 1) = F$ (TO (S - T) / 2 - 1) + "g5" +
     E$ (TO T) + "g5c"
 80  IF S = 2 * INT (S/2) THEN LET D$ (1) =
     F$ (TO (S - 2) / 2 ) + "g5 g5c"
100  PRINT AT 16, 0; INK 4; E$ + "□" + E$ + "□" + E$
500  INPUT "Move from?"; F
502  [apague esta linha]
504  INPUT "Move to?"; G
506  [apague esta linha]
```

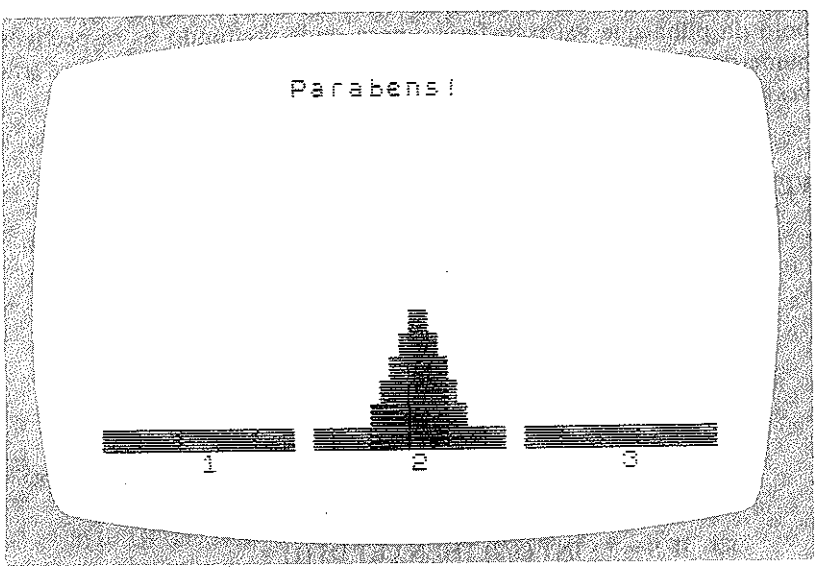
```
5Ø8 [apague esta linha]
555  PRINT AT 16, Ø; INK 4; E$
12ØØ  PRINT AT 1, 1Ø; FLASH 1; "Success!"
```

Adicione estas agora:

```
1210  FOR I = 1 TO 8Ø
1220  BORDER 7 * RND
1230  BEEP 2/(2Ø + I), I/2 + 1Ø * (RND - .5)
1240  NEXT I
```

SE USAR A VERSÃO DO ZX SPECTRUM

Proceda como na versão do ZX81.





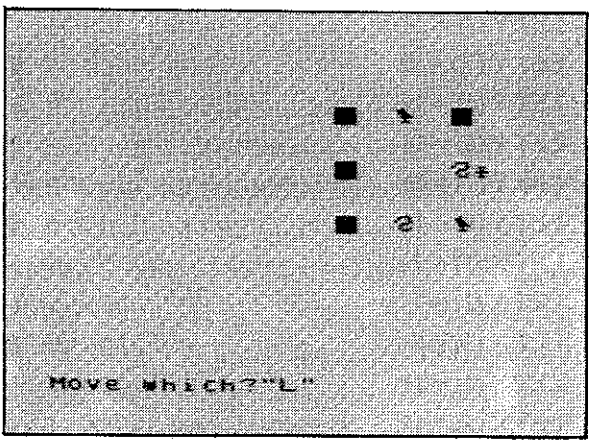

```
10 LET e$="                    "
15 LET f$="                    "
20 INPUT "Tamanho? ";s
25 LET v=INT ((11-s)/2)
30 DIM d$(s,s+2)
40 FOR t=s TO 1 STEP -2
```

```
  50 LET d$(t)=f$( TO  (s-t)/2)+e
$( TO t)
  60 IF t<s THEN LET d$(t+1)=f$(
 TO (s-t)/2-1)+"▒"+e$( TO t)+"▒"
  70 NEXT t
  80 IF s=2*INT (s/2) THEN LET d
$(1)=f$( TO (s-2)/2)+"▒"
 100 PRINT AT 16,0; INK 4;e$+" "
+e$+" "+e$
 110 FOR i=1 TO 3
 120 PRINT AT 17,11*i-6;i
 130 NEXT i
 200 FOR i=1 TO s
 210 PRINT AT 15-s+i,v;d$(i)
 220 NEXT i
 300 DIM a(3,s)
 310 FOR i=1 TO s
 320 LET a(1,i)=s-i+1
 330 NEXT i
 340 DIM h(3)
 350 LET h(1)=s
 500 INPUT "De onde? ";f
 504 INPUT "Para onde? ";g
 511 GO TO 1000
 512 LET r=a(f,h(f))
 514 LET a(f,h(f))=0
 516 LET a(g,h(g)+1)=r
 520 LET h(f)=h(f)-1
 530 LET h(g)=h(g)+1
 540 LET x=r+1+.5*(10-r)
 545 IF x>s THEN LET x=s+2
 550 PRINT AT 15-h(f),11*f-11+v;
f$( TO x);AT 16-h(g),11*g-11+v;d
$(r)( TO x)
 555 PRINT AT 16,0; INK 4;e$
 557 IF h(2)=s OR h(3)=s THEN GO
 TO 1200
 560 GO TO 500
1000 IF h(g)=0 THEN GO TO 512
1005 IF h(f)=0 THEN GO TO 500
1010 IF a(f,h(f))>=a(g,h(g)) THE
N GO TO 500
1020 GO TO 512
1200 PRINT AT 1,10; FLASH 1;"Par
abens!"
1210 FOR i=1 TO 80
1220 BORDER 7*RND
1230 BEEP 2/(20+i),i/2+10*(RND-.
5)
1240 NEXT i
```

NOTAS

CAVALGANDO

several times knightly



Para começar, duas espécies de animais estão colocados nos cantos de um quadrado de 3 por 3:

A □ A
□ □
B □ B

O objectivo é trocar os As pelos Bs. Qualquer animal pode ir para um *espaço branco* movendo-se tal como o cavalo num tabuleiro de *xadrez* (dois passos para a frente e um para o lado).

VERSÃO ZX81

```
10 DIM A (3, 3)
20 LET A (1, 1) = 1
30 LET A (1, 3) = 1
```

```
  40  LET A (3, 1) = 2
  50  LET A (3, 3) = 2
  60  LET A (2, 2) = 3
  70  LET C = 1
 100  DIM X (8)
 110  DIM Y (8)
 121  LET Y (1) = 14
 122  LET Y (2) = 18
 123  LET Y (3) = 22
 124  LET Y ( 4) = 22
 125  LET Y (5) = 22
 126  LET Y (6) = 18
 127  LET Y (7) = 14
 128  LET Y (8) = 14
 131  LET X (1) = 7
 132  LET X (2) = 7
 133  LET X (3) = 7
 134  LET X (4) = 11
 135  LET X (5) = 15
 136  LET X (6) = 15
 137  LET X (7) = 15
 138  LET X (8) = 11
 200  GOSUB 1000
 210  PRINT AT X (1), Y (1); "*"
→ 700  PRINT AT 0, 0; "MOVE WHICH"
→ 702  INPUT I$
 710  IF I$ < > " " THEN GOTO 719
 715  GOSUB 1100
 718  GOTO 700
 719  LET C0 = C
→ 720  PRINT AT 0, 0; "WHERE □ □ □ □ □"
→ 722  INPUT I$
→ 725  PRINT AT 0, 0; " [10 espaços ] "
 730  IF I$ < > " " THEN GOTO 750
```

```
  735  GOSUB 1100
  740  GOTO 720
  750  LET C1 = C
  760  IF ABS (C0 - C1) = 3 OR ABS (C0 - C1) = 5 THEN GOTO 2000
  770  PRINT AT 0, 0; "ILLEGAL"
  780  PAUSE 50
  790  PRINT AT 0, 0; " [7 espaços ] "
  800  GOTO 700
 1000  FOR I = 1 TO 3
 1010  FOR J = 1 TO 3
 1020  IF A (I, J) = 0 THEN LET Q$ = "■"
→1030  IF A (I, J) = 1 THEN LET Q$ = "A"
→1040  IF A (I, J) = 2 THEN LET Q$ = "B"
 1050  IF A (I, J) = 3 THEN LET Q$ = "□"
 1060  PRINT AT 5 + 3 * I, 12 + 3 * J; Q$
 1070  NEXT J
 1080  NEXT I
 1090  RETURN
 1100  PRINT AT X (C), Y (C);"□"
 1110  LET C = C + 1
 1120  IF C > 8 THEN LET C = 1
 1130  PRINT AT X (C), Y (C); "*"
 1140  RETURN
 2000  LET X 0 = (X (C0) - 3) / 4
 2010  LET Y0 = (Y (C0) - 10) / 4
 2020  LET X1 = (X (C1) - 3) / 4
 2030  LET Y1 = (Y (C1) - 10) / 4
 2040  LET R = A (X0, Y0)
 2045  IF R = 0 THEN GOTO 770
 2050  LET A (X0, Y0) = A (X1, Y1)
 2060  LET A (X1, Y1) = R
 2070  GOSUB 1000
 2080  IF A (1, 1) = 2 AND A (1, 3) = 2 AND
       A (3, 1) = 1 AND A (3, 3) = 1 THEN GOTO 3000
```

```
2100 GOTO 700
→ 3000 PRINT AT 0, 8; "GOTCHA"
```

SE USAR A VERSÃO DO ZX81

RUN. O quadrado aparece desenhado com os animais designados por A e B. Um asterisco actua como cursor. Se premir "NEWLINE" movimenta-se no sentido dos ponteiros do relógio repetindo-se consecutivamente até conseguir colocar o asterisco no lugar de um. A posição do cursor define que animal vai ser movimentado. Você seleccionará para onde o animal vai, usando novamente o cursor. O movimento está completo e pode continuar.

VERSÃO ZX SPECTRUM

Modifique estas linhas

```
700 INPUT "Move which?"; I$
702 [apague esta linha]
720 INPUT "Where?"; I$
722 [apague esta linha]
725 [apague esta linha]
1030 IF A (I, J) = 1 THEN LET Q$ = "gA"
1040 IF A (I, J) = 2 THEN LET Q$ = "gB"
3000 PRINT AT 0, 8; FLASH 1; "Gotcha!!!!"
```

E adicione estas:

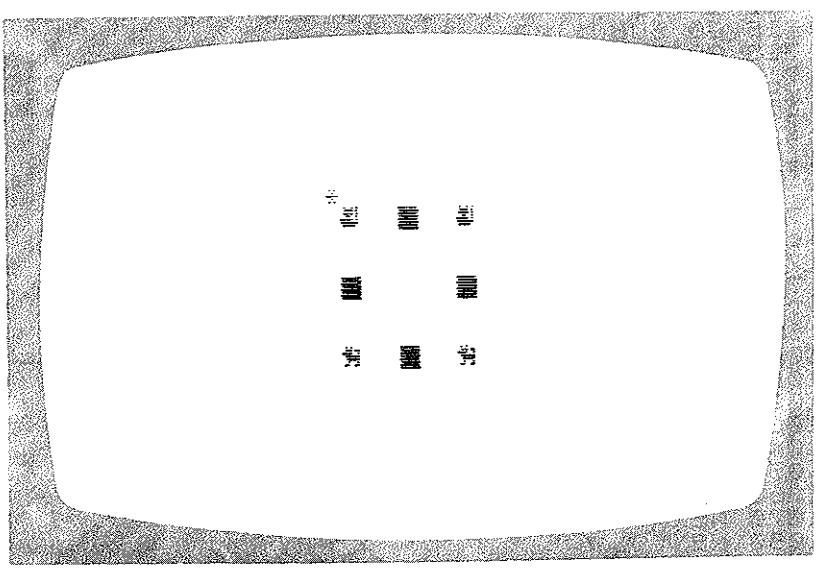
```
3010 FOR I =1 TO 80
3020 BORDER 7 * RND
3030 BEEP 1/(10 + I), I/2 + 7 * (RND − .3)
3040 NEXT I
3050 STOP
```

Defina os caracteres gráficos para "gA"e "gB" com os seus animais favoritos. Pode usar o cão e o gato do «puzzle» "ENCADEAMENTO ECOLÓGICO" onde eles estão em POKE como "C" e "D" mas pode modificá-los para "A" e "B" como segue:

```
 1 GOSUB 4000
4000 DATA 74, 122, 122, 50, 250, 254, 124, 0,
     80, 118, 242, 126, 62, 34, 102, 0
4010 FOR N = 0 TO 7
4020 READ X
4030 POKE USR "A" + N, X
4040 NEXT N
4050 FOR N = 0 TO 7
4060 READ X
4070 POKE USR "B" + N, X
4080 NEXT N
5000 RETURN
```

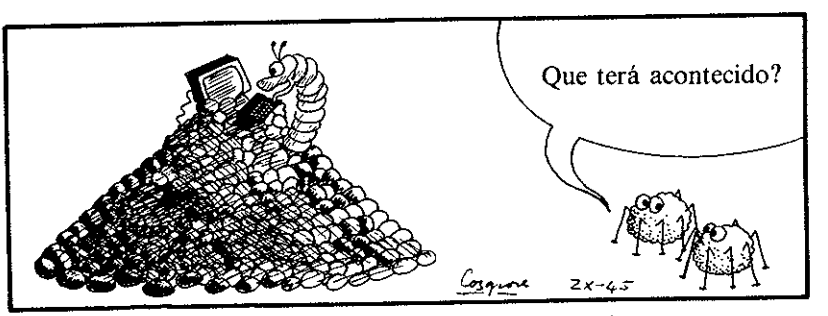
SE USAR A VERSÃO DO ZX SPECTRUM

GO TO 4000. Depois RUN. Funciona tal como na versão do ZX81 mas é muito mais bonito.



```
  1 GO SUB 4000
 10 DIM a(3,3)
 20 LET a(1,1)=1
 30 LET a(1,3)=1
 40 LET a(3,1)=2
 50 LET a(3,3)=2
 60 LET a(2,2)=3
 70 LET c=1
100 DIM x(8)
110 DIM y(8)
121 LET y(1)=14
122 LET y(2)=18
123 LET y(3)=22
124 LET y(4)=22
125 LET y(5)=22
126 LET y(6)=18
127 LET y(7)=14
128 LET y(8)=14
131 LET x(1)=7
132 LET x(2)=7
133 LET x(3)=7
134 LET x(4)=11
135 LET x(5)=15
136 LET x(6)=15
137 LET x(7)=15
138 LET x(8)=11
200 GO SUB 1000
210 PRINT AT x(1),y(1);"*"
700 INPUT "Mover qual? ";i$
710 IF i$<>"" THEN GO TO 719
715 GO SUB 1100
718 GO TO 700
719 LET c0=0
720 INPUT "Onde? ";i$
730 IF i$<>"" THEN GO TO 750
735 GO SUB 1100
740 GO TO 720
750 LET c1=c
760 IF ABS (c0-c1)=3 OR ABS (c
-c1)=5 THEN GO TO 2000
770 PRINT AT 0,0;"Ilegal!"
780 PAUSE 50
790 PRINT AT 0,0;"          "
800 GO TO 700
1000 FOR i=1 TO 3
1010 FOR j=1 TO 3
1020 IF a(i,j)=0 THEN LET q$="▒
1030 IF a(i,j)=1 THEN LET q$="▒
1040 IF a(i,j)=2 THEN LET q$="▒
1050 IF a(i,j)=3 THEN LET q$="
1060 PRINT AT 5+3*i,12+3*j;q$
1070 NEXT j
1080 NEXT i
1090 RETURN
1100 PRINT AT x(c),y(c);" "
1110 LET c=c+1
1120 IF c>8 THEN LET c=1
```

```
1130 PRINT AT x(c),y(c);"*"
1140 RETURN
2000 LET x0=(x(c0)-3)/4
2010 LET y0=(y(c0)-10)/4
2020 LET x1=(x(c1)-3)/4
2030 LET y1=(y(c1)-10)/4
2040 LET r=a(x0,y0)
2045 IF r=0 THEN GO TO 770
2050 LET a(x0,y0)=a(x1,y1)
2060 LET a(x1,y1)=r
2070 GO SUB 1000
2080 IF a(1,1)=2 AND a(1,3)=2 A
ND a(3,1)=1 AND a(3,3)=1 THEN GO
TO 3000
2100 GO TO 700
3000 PRINT AT 0,8; FLASH 1;"Rai
s!!"
3010 FOR i=1 TO 80
3020 BORDER 7*RND
3030 BEEP 1/(10+i),i/2+7*(RND-.
)
3040 NEXT i
3050 STOP
4000 DATA 74,122,122,50,250,254
124,0,80,118,242,126,62,34,102,
4010 FOR n=0 TO 7
4020 READ x
4030 POKE USR "a"+n,x
4040 NEXT n
4050 FOR n=0 TO 7
4060 READ x
4070 POKE USR "b"+n,x
4080 NEXT n
5000 RETURN
```



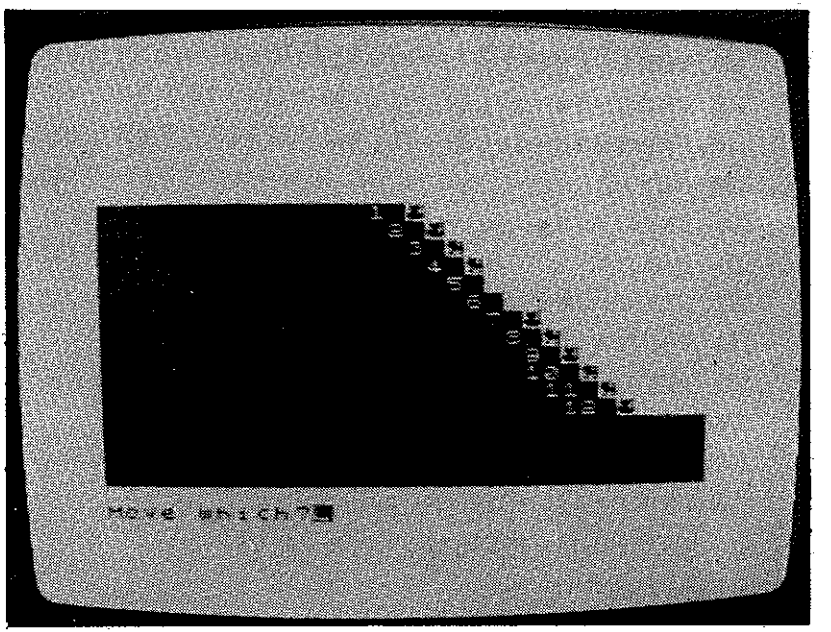
NOTAS

E finalmente outra permuta de animais:

15

ANIMAIS EM DEGRAUS

beasts on the steps



Cinco gatos e cinco cães sentados em doze degraus de uma escada. Você pode mover qualquer par de animais adjacentes a um espaço vazio para esse mesmo espaço. Repita isto até conseguir colocar os cinco gatos nos cinco degraus de cima e os cinco cães logo a seguir, deixando os últimos dois degraus de baixo vazios.

VERSÃO ZX81

```
    10  LET A$ = " [32 inverse spaces] "
→   15  LET B$ = "ABABABAB □ □"
```

```
  20  PRINT AT 5, 0
  30  FOR I = 1 TO 12
  40  PRINT A$ (TO 15 + I)
  50  NEXT I
  60  PRINT A$ + A$ + A$ + A$
  70  FOR I = 1 TO 12
  80  PRINT AT 5 + I, 12 + I + (1 < 10); I
  90  NEXT I
  95  LET P = 11
 100  FOR I = 1 TO 5
→ 110  PRINT AT 4 + 2 * I, 14 + 2 * I; "A"
→ 120  PRINT AT 5 + 2 * I, 15 + 2 * I; "B"
 130  NEXT I
→ 200  PRINT AT 0, 0; "MOVE WHICH"
→ 201  INPUT N
 202  IF N < 1 OR N > 11 THEN GOTO 200
→ 203  PRINT AT 0, 0; " [10 espaços] "
 205  LET X$ = B$ (N)
 206  LET Y$ = B$ (N + 1)
 208  IF X$ = "□" OR Y$ = "□" THEN GOTO 500
 210  PRINT AT 5 + N, 15 + N; "□"; AT 6 + N, 16 + N; "□";
      AT 5 + P, 15 + P; X$; AT 6 + P, 16 + P; Y$
 211  LET B$ (N) = "□"
 212  LET B$ (N + 1) = "□"
 213  LET B$ (P) = X$
 214  LET B$ (P + 1) = Y$
 220  LET P = N
→ 225  IF B$ = "AAAAABBBBB □ □" THEN GOTO 1000
 230  GOTO 200
 500  PRINT AT 0, 0; "ILLEGAL"
 510  PAUSE 50
 520  PRINT AT 0, 0; " [7 espaços] "
 530  GOTO 200
→ 1000  PRINT AT 1, 8; "DONE IT"
```

SE USAR A VERSÃO DO ZX81

RUN. Escolha o *primeiro* dos animais a ser movimentado pelo número do degrau correspondente e assim sucessivamente. Tudo o resto é automático.

VERSÃO ZX SPECTRUM

Modifique estas linhas:

```
 15 LET B$ = "gA gB gA gB gA gB gA gB gA gB □ □"
110 PRINT AT 4 + 2 * I, 14 + 2 * I; "gA"
120 PRINT AT 5 + 2 * I, 15 + 2 * I; "gB"
200 INPUT "Move which?"; N
201 [apague esta linha]
203 [apague esta linha]
225 IF B$ = "gA gA gA gA gA gB gB gB gB gB □ □"
    THEN GOTO 1000
1000 PRINT AT 1, 8; FLASH 1; "DONE IT!"
```

Adicione estas:

```
1020 FOR I = 1 TO 30
1030 BORDER 4: BEEP .1, 12: BORDER 2: BEEP .1, 0
1040 NEXT I
```

Defina as caracteres gráficos dos animais com:

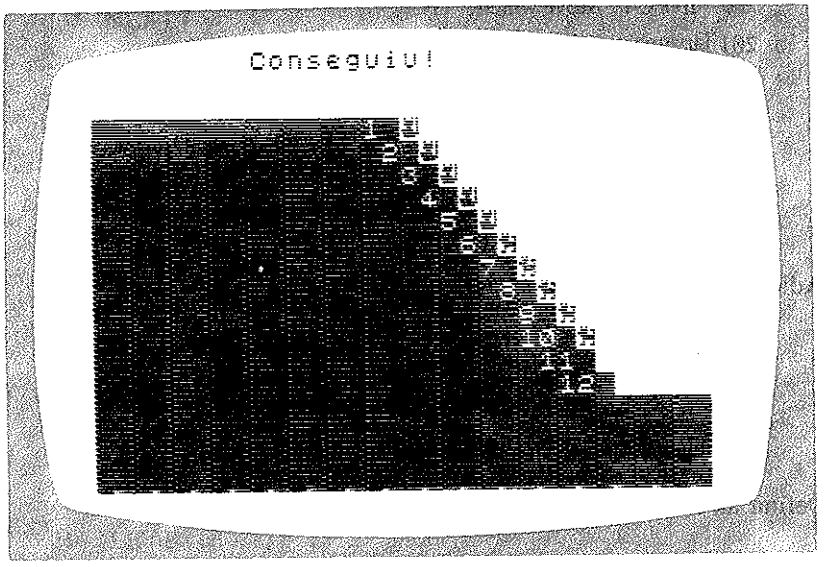
```
2000 DATA 37, 61, 61, 25, 125, 127, 62, 0,
     40, 59, 121, 63, 31, 17, 51, 0
2010 FOR N = 0 TO 7
2020 READ X
2030 POKE USR "A" + N, X
2040 NEXT N
2050 FOR N = 0 TO 7
2060 READ X
```

```
2070 POKE USR "B" + N, X
2080 NEXT N
```

(Estes números de linhas são precisamente metade dos utilizados para o «puzzle» CAVALGANDO. Se não se importa utilize os números de linha de CAVALGANDO e veja o que acontece.)

SE USAR A VERSÃO DO ZX SPECTRUM

Após a ligação faça GO TO 2000 ou RUN 2000. RUN. Tudo o resto é como na versão do XZ81.






```
 1 GOSUB 2000
10 LET a$="
                    "
15 LET b$="                  "
20 PRINT AT 5,0
30 FOR i=1 TO 12
40 PRINT a$( TO 19+i)
50 NEXT i
60 PRINT a$+a$+a$+a$
65 INVERSE 1
```

```
  70 FOR i=1 TO 12
  80 PRINT AT 5+i,12+i+(i<10);i
  90 NEXT i
  92 INVERSE 0
  95 LET p=11
 100 FOR i=1 TO 5
 110 PRINT AT 4+2*i,14+2*i;"▟"
 120 PRINT AT 5+2*i,15+2*i;"▙"
 130 NEXT i
 200 INPUT "Mower qual? ";n
 202 IF n<1 OR n>11 THEN GO TO 2
00
 205 LET x$=b$(n)
 206 LET y$=b$(n+1)
 208 IF x$=" " OR y$=" " THEN GO
 TO 500
 210 PRINT AT 5+n,15+n;" ";AT 6+
n,16+n;" ";AT 5+p,15+p;x$;AT 6+p
,16+p;y$
 211 LET b$(n)=" "
 212 LET b$(n+1)=" "
 213 LET b$(p)=x$
 214 LET b$(p+1)=y$
 220 LET p=n
 225 IF b$="▟▟▟▟▟▟▟▟▟▟▟▟▟▟▟▟▟▟▟▟  " THEN G
O TO 1000
 230 GO TO 200
 500 PRINT AT 0,0;"Ilegal"
 510 PAUSE 50
 520 PRINT AT 0,0;"       "
 530 GO TO 200
1000 PRINT AT 1,8; FLASH 1;"Cons
eguiu!"
1020 FOR i=1 TO 30
1030 BORDER 4: BEEP .1,12: BEEP
.1,0
1040 NEXT i
1050 STOP
2000 DATA 37,61,61,25,125,127,62
,0,40,59,121,63,31,17,51,0
2010 FOR n=0 TO 7
2020 READ x
2030 POKE USR "a"+n,x
2040 NEXT n
2050 FOR n=0 TO 7
2060 READ x
2070 POKE USR "b"+n,x
2080 NEXT n
2090 RETURN
```

NOTAS

ATENÇÃO!

ESTES «PUZZLES» SÃO SÓ PARA O SPECTRUM

Aqui estão alguns programas que funcionam apenas no ZX Spectrum. Bem, acho que devo uma explicação por ter posto neste livro alguns programas que só correm no ZX Spectrum. A justificação é que talvez os possuidores de ZX81 por causa destes programas vão já a correr comprar um ZX Spectrum o que me vai permitir vender muito mais livros destes. Pode sonhar-se não pode?

Aqui está um velho «puzzle» favorito
com alguns pormenores
que talvez ainda não tenha visto

16

SOLITÁRIO

solitaire

Este é o jogo normal do solitário. As cavilhas são mostradas como asteriscos e os quatro cantos onde não se pode jogar aparecem com Os maiúsculos. As linhas estão referenciadas da **a** a **g** no écran e as colunas de 1 a 7.

O computador pergunta-lhe qual o movimento de e para onde. Responda-lhe introduzindo primeiro a letra e depois o algarismo. Deste modo pode mover-se de f4 para o centro do quadrado, d4.

Para isso premiria:

f4 (ENTER) d4 (ENTER)

e o computador movimentaria as cavilhas por si. Atenção! ele dá conta quando você fizer batota!

Para o caso de ainda não estar familiarizado com este jogo aqui fica um pequeno resumo das suas regras:

1 — O objectivo do jogo é conseguir retirar todas as cavilhas menos uma do tabuleiro.

2 — Um movimento permitido consiste em pegar numa cavilha e comer uma cavilha adjacente (verticalmente ou horizontalmente mas não diagonalmente) saltando para um espaço em branco. A cavilha que foi comida sai do tabuleiro.

```
10  LET n = 32
20  LET getmove = 1000: LET testmove = 2000
30  LET makemove = 3000
40  FOR r = 4 TO 16 STEP 2
50  FOR c = 9 TO 21 STEP 2
60  PRINT PAPER 5; AT r, c; "O"
70  NEXT c
80  NEXT r
100 FOR c = 13 TO 17 STEP 2
```

```
110  FOR r = 4 TO 16 STEP 2
120  PRINT PAPER 6; AT r, c; "*"
130  NEXT r
140  NEXT c
150  FOR r = 8 TO 12 STEP 2
160  FOR c = 9 TO 21 STEP 2
170  PRINT PAPER 6; AT r, c; "*"
180  NEXT c
190  NEXT r
200  PRINT PAPER 6; OVER 1; AT 10, 15; "*"
210  FOR r = 4 TO 16 STEP 2
220  PRINT AT r, 7; CHR$ (r/2 + 95)
230  NEXT r
240  FOR c = 9 TO 21 STEP 2
250  PRINT AT 18, c; CHR$ (c/2 + 44)
260  NEXT c
270  GO SUB getmove
280  GO SUB testmove
290  IF NOT legal THEN PRINT AT 0, 0; "Illegal Move!!": GO TO 270
300  PRINT AT 0, 0; "□□□□□□□□□□□□□□"      [14 espaços]
310  GO SUB makemove: LET n = n - 1
320  IF n > 1 THEN GO TO 270
330  PRINT AT 0, 0; "You've done it!"
340  GO TO 9999

1000  INPUT "from to:"; f$, t$
1010  LET rf = 2 * (CODE f$ (1) - 95)
1020  LET cf = 2 * (CODE f$ (2) - 45) + 1
1030  LET rt = 2 * (CODE t$ (1) - 95)
1040  LET ct = 2 * (CODE t$ (2) - 45) + 1
1050  RETURN

2000  LET legal = 1
2010  IF SCREEN$ (rf, cf) < > "*" THEN LET legal = 0
2020  IF SCREEN$ (rt, ct) < > "□" THEN LET legal = 0
```

```
2Ø25  IF SCREEN$ ( (rf + rt) / 2, (cf + ct) / 2) < > "*"
      THEN LET legal = Ø
2Ø3Ø  IF ABS (rf - rt) = 4 AND cf < > ct THEN LET legal = Ø
2Ø4Ø  IF ABS (cf - ct) = 4 AND rf < > rt THEN LET legal = Ø
2Ø5Ø  IF ABS (rf - rt) < > 4 AND rf < > rt THEN LET legal = Ø
2Ø6Ø  IF ABS (cf - ct) < > 4 AND cf < > ct THEN LET legal = Ø
2Ø7Ø  RETURN

3ØØØ  PRINT PAPER 6; OVER 1; AT rf, cf; "*"
3Ø1Ø  PRINT PAPER 6; AT rt, ct; "*"
3Ø2Ø  PRINT PAPER 6; OVER 1; AT (rf + rt) / 2, (cf + ct) / 2; "*"
3Ø3Ø  RETURN
```

Existem outras versões deste jogo nas quais as regras para remover as cavilhas são as mesmas, mas o número delas e as suas posições de partida diferem.

Uma das mais simples é a «ESTRADA LATINA». Você pode gerar esta versão modificando as linhas 1Ø, 12Ø e 17Ø como segue:

```
 1Ø  LET n = 6
12Ø  PRINT PAPER 6; AT r, c; "□"
17Ø  PRINT PAPER 6; AT r, c; "□"
```

E substitua a linha 2ØØ por:

```
2ØØ  GO SUB 4ØØØ
```

E adicione esta sub-rotina

```
4ØØØ  FOR r = 6 TO 12 STEP 2
4Ø1Ø  PRINT PAPER 6; AT r, 15; "*"
4Ø2Ø  NEXT r
4Ø3Ø  PRINT PAPER 6; AT 8, 13; "*"
4Ø4Ø  PRINT PAPER 6; AT 8, 17; "*"
4Ø5Ø  RETURN
```

Para fazer a versão da «ESTRADA GREGA» faça exactamente as mesmas modificações mas substitua a totalidade da sub-rotina da linha 4ØØØ por esta:

```
4000 FOR r = 6 TO 14 STEP 2
4010 PRINT PAPER 6; AT r, 15; "*"
4020 NEXT r
4030 FOR c = 11 TO 19 STEP 2
4040 PRINT PAPER 6; AT 10, c; "*"
4050 NEXT c
4060 RETURN
```

E ponha assim a linha 10:

```
10 LET n = 9
```

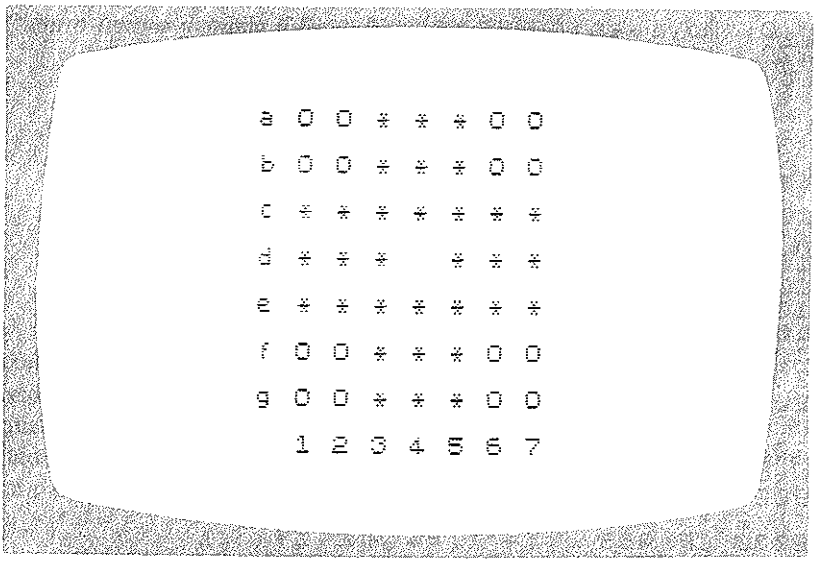
Finalmente aqui está o «quadrado inclinado» como é conhecido. Você verá quando ele estiver desenhado!

Pegue no programa *original* e adicione as linhas seguintes:

```
201 PRINT PAPER 6; OVER 1; AT 4, 13; "*"
202 PRINT PAPER 6; OVER 1; AT 4, 17; "*"
203 PRINT PAPER 6; OVER 1; AT 8, 21; "*"
204 PRINT PAPER 6; OVER 1; AT 12, 21; "*"
205 PRINT PAPER 6; OVER 1; AT 16, 17; "*"
206 PRINT PAPER 6; OVER 1; AT 16, 13; "*"
207 PRINT PAPER 6; OVER 1; AT 12, 9; "*"
208 PRINT PAPER 6; OVER 1; AT 8, 9; "*"
```

Agora modifique a linha 10 para:

```
10 LET n = 24
```






```
  10 LET n=32
  20 LET getmove=1000: LET testm
ove=2000
  30 LET makemove=3000
  40 FOR r=4 TO 16 STEP 2
  50 FOR c=9 TO 21 STEP 2
  60 PRINT PAPER 5;AT r,c;"O"
  70 NEXT c
  80 NEXT r
 100 FOR c=13 TO 17 STEP 2
 110 FOR r=4 TO 16 STEP 2
 120 PRINT PAPER 6;AT r,c;"*"
 130 NEXT r
 140 NEXT c
 150 FOR r=8 TO 12 STEP 2
 160 FOR c=9 TO 21 STEP 2
 170 PRINT PAPER 6;AT r,c;"*"
 180 NEXT c
 190 NEXT r
 200 PRINT PAPER 6; OVER 1;AT 10
 15;"*"
 210 FOR r=4 TO 16 STEP 2
 220 PRINT AT r,7;CHR$ (r/2+95)
 230 NEXT r
 240 FOR c=9 TO 21 STEP 2
 250 PRINT AT 18,c;CHR$ (c/2+44)
 260 NEXT c
 270 GO SUB getmove
 280 GO SUB testmove
 290 IF NOT legal THEN PRINT AT
```

```
0,0;"Movimento ilegal!!": GO TO
270
 300 PRINT AT 0,0;"
          "
 310 GO SUB makemove: LET n=n-1
 320 IF n>1 THEN GO TO 270
 330 PRINT AT 0,0;"Conseguiu!"
 340 GO TO 9999
1000 INPUT "De onde e para onde?
 ";f$,t$
1010 LET rf=2*(CODE f$(1)-95)
1020 LET cf=2*(CODE f$(2)-48)+1
1030 LET rt=2*(CODE t$(1)-95)
1040 LET ct=2*(CODE t$(2)-48)+1
1050 RETURN
2000 LET legal=1
2010 IF SCREEN$ (rf,cf)<>"*" THE
N LET legal=0
2020 IF SCREEN$ (rt,ct)<>" " THE
N LET legal=0
2025 IF SCREEN$ ((rf+rt)/2,(cf+c
t)/2)<>"*" THEN LET legal=0
2030 IF ABS (rf-rt)=4 AND cf<>ct
 THEN LET legal=0
2040 IF ABS (cf-ct)=4 AND rf<>rt
 THEN LET legal=0
2050 IF ABS (rf-rt)<>4 AND rf<>r
t THEN LET legal=0
2060 IF ABS (cf-ct)<>4 AND cf<>c
t THEN LET legal=0
2070 RETURN
3000 PRINT PAPER 6; OVER 1;AT rf
,cf;"*"
3010 PRINT PAPER 6;AT rt,ct;"*"
3020 PRINT PAPER 6; OVER 1;AT (r
f+rt)/2,(cf+ct/2);"*"
3030 RETURN
```

17

ANAGRAMAS CONTRA-RELÓGIO

anagrams against the clock

O computador mostra um anagrama de uma palavra de 1Ø letras e começa a contar desde 9999 até 0. Demora cerca de 50 segundos a fazer isso. Você tem que resolver o anagrama neste tempo. Logo que introduza a primeira letra o relógio pára. Mas não tente utilizar esta facilidade para aumentar o tempo de pensar porque tem apenas uma hipótece para colocar a primeira letra e não poderá apagá-la depois.

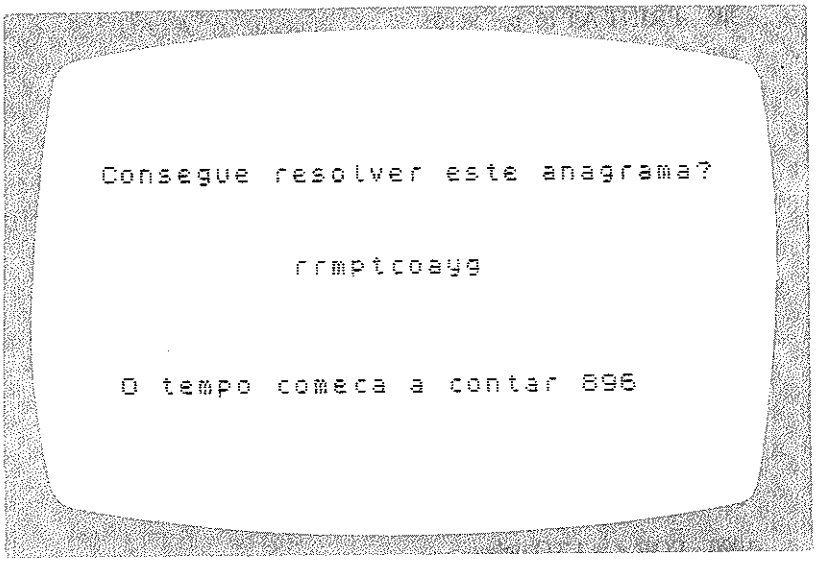
```
 1 LET cont = 35Ø: LET end = 37Ø
 2 LET delay = 3Ø
 3 LET pause = 1ØØØ
 4 LET begin = 9Ø
 5 LET clear = 2ØØØ
1Ø DATA "disability"
11 DATA "indefinite"
12 DATA "perfidious"
13 DATA "improbable"
14 DATA "practising"
15 DATA "caretakers"
16 DATA "medicament"
17 DATA "reciprocal"
18 DATA "procedures"
19 DATA "elementary"
2Ø DATA "regardless"
21 DATA "algorithms"
22 DATA "traversing"
23 DATA "represents"
24 DATA "preciously"
```

```
 25 DATA "completing"
 26 DATA "subroutine"
 27 DATA "inversions"
 28 DATA "cryptogram"
 29 DATA "statements"
 30 DIM w$ (10)
 90 RANDOMIZE
100 RESTORE (INT (RND * 20) + 10)
110 READ w$
115 LET x$ = w$
120 FOR i = 1 TO 20
130 LET p = INT (RND * 10) + 1
140 LET q = INT (RND * 10) + 1
150 LET t$ = w$ (p)
160 LET w$ (p) = w$ (q)
170 LET w$ (q) = t$
180 NEXT i
190 PRINT AT 2, 2; "Can you solve this anagram?"
200 PRINT AT 6, 10; w$
210 PRINT AT 10, 2; "Your time starts now"
215 PRINT AT 17, 10; "----------"
220 FOR i = 999 TO 0 STEP -1
225 IF i > 99 THEN PRINT AT 10, 25; i: GO TO 230
226 IF i > 9 THEN PRINT AT 10, 25; "□"; i: GO TO 230
227 PRINT AT 10, 25; "□ □"; i
230 LET c$ = INKEY$
240 IF c$ = " " THEN GO TO cont
250 LET w$ (1) = c$
260 PRINT AT 16, 10; w$ (1)
265 GO SUB pause
270 FOR p = 2 TO 10
280 LET c$ = INKEY$
290 IF c$ = " " THEN GO TO 280
300 LET w$ (p) = c$
```

```
305  PRINT AT 16, p + 9; c$
310  LET c$ = " "
315  GO SUB pause
320  NEXT p
330  IF w$ = x$ THEN PRINT AT 20, 2; "Well done!!!": GO TO end
340  PRINT AT 20, 2; "Wrong! □□ The word was . . . . . □□□□□□□";
     FLASH 1; x$: GO TO end
350  NEXT i
360  PRINT AT 20, 2; "Out of time . . . The word was . . . . □□□";
     FLASH 1; x$
370  INPUT "Want another one?"; q$
380  IF q$ = "yes" THEN GO SUB clear: GO TO begin
390  GO TO 9999

1000 FOR x = 1 TO delay
1010 NEXT x
1020 RETURN

2000 PRINT AT 16, 10; "□□□□□□□□□□"                    [10 espaços]
2005 PRINT AT 6, 10; "□□□□□□□□□□"                     [10 espaços]
2010 PRINT AT 20, 2; "□□□□□□□□□□□□□□□□□□□□
     □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"                 [42 espaços]
2015 PRINT AT 10, 2; "□□□□□□□□□□□□□□□□□□□□
     □□□□□□□□□□"                                      [28 espaços]
2020 RETURN
```






```
  1 LET cont=350: LET end=370
  2 LET delay=30
  3 LET pause=1000
  4 LET begin=90
  5 LET clear=2000
 10 DATA "disability"
 11 DATA "indefinite"
 12 DATA "perfidious"
 13 DATA "improbable"
 14 DATA "practising"
 15 DATA "caretakers"
 16 DATA "medicament"
 17 DATA "reciprocal"
 18 DATA "procedures"
 19 DATA "elementary"
 20 DATA "regardless"
 21 DATA "algorithms"
 22 DATA "traversing"
 23 DATA "represents"
 24 DATA "preciously"
 25 DATA "completing"
 26 DATA "subroutine"
 27 DATA "inversions"
 28 DATA "cryptogram"
 29 DATA "statements"
 30 DIM w$(10)
 90 RANDOMIZE
100 RESTORE (INT (RND*20)+10)
110 READ w$
115 LET x$=w$
```

```
 120 FOR i=1 TO 20
 130 LET p=INT (RND*10)+1
 140 LET q=INT (RND*10)+1
 150 LET t$=w$(p)
 160 LET w$(p)=w$(q)
 170 LET w$(q)=t$
 180 NEXT i
 190 PRINT AT 2,0;"Consegue reso
lver este anagrama?"
 200 PRINT AT 6,10;w$
 210 PRINT AT 10,2;"O tempo come
ca a contar "
 215 PRINT AT 17,10;"           "
 220 FOR i=999 TO 0 STEP -1
 225 IF i>99 THEN PRINT AT 10,25
;i: GO TO 230
 226 IF i>9 THEN PRINT AT 10,25;
" ";i: GO TO 230
 227 PRINT AT 10,25;"  ";i
 230 LET c$=INKEY$
 240 IF c$="" THEN GO TO cont
 250 LET w$(1)=c$
 260 PRINT AT 16,10;w$(1)
 265 GO SUB pause
 270 FOR p=2 TO 10
 280 LET c$=INKEY$
 290 IF c$="" THEN GO TO 280
 300 LET w$(p)=c$
 305 PRINT AT 16,p+9;c$
 310 LET c$=""
 315 GO SUB pause
 320 NEXT p
 330 IF w$=x$ THEN PRINT AT 20,2
;"Parabens!!!": GO TO end
 340 PRINT AT 20,2;"Errado! A pa
lavra era...          "; FLASH 1
;x$: GO TO end
 350 NEXT i
 360 PRINT AT 20,2;"Acabou o tem
po. A palavra era          "; FL
ASH 1;x$
 370 INPUT "Outra vez? ";q$
 380 IF q$="" THEN GO SUB clear:
 GO TO begin
 390 GO TO 9999
1000 FOR x=1 TO delay
1010 NEXT x
1020 RETURN
2000-2015
2020 RETURN
```

NOTAS

Aqui está um «puzzle» favorito dos livros da especialidade com alguns pormenores adicionados pelo Spectrum

18

PALAVRAS CRUZADAS

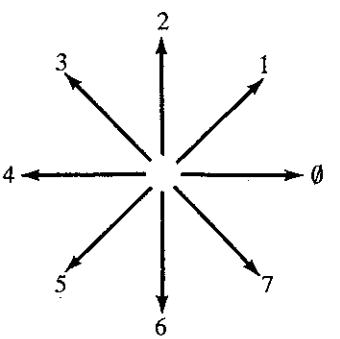
hidden words

Dê ao computador até dez palavras que não podem ter mais de dez letras (uma palavra de dez letras ficará encostada ao lado direito do seu écran, e portanto a regra é que as palavras não poderão saltar de linha).

O computador organizará as palavras horizontalmente, verticalmente ou diagonalmente, para trás e para a frente num conjunto quadrado aleatório de letras. Quando achar que deve começar diga-o ao computador. Por exemplo se é no quadrado g12 prima:

g (ENTER) 12 (ENTER)

Introduza depois o número que irá dar a direcção à palavra. Os valores para essa direcção são os seguintes:



Agora diga qual a palavra. Se estiver certo o computador fará desaparecer a palavra da lista de palavras que faltam e colori-la-á no quadrado.

Se introduzir uma palavra que está no quadrado mas que não está na lista de palavras que faltam o computador fica a saber que você está a fazer batota e recusa continuar a jogar consigo!

```
  1  LET fillin = 1000: LET test = 2000
  2  LET colour = 3000: LET rubout = 4000:
     LET list = 5000: LET delete = 6000
  3  LET n = 0
  4  RANDOMIZE
  5  CLS
  6  LET pc = 1
 10  DIM w$ (11, 11)
 20  DIM s$ (18, 18)
100  INPUT "How many words"; nw
110  FOR i = 1 TO nw
120  INPUT "Give me a word please"; w$ (i)
130  NEXT i
140  FOR i = 1 TO nw
150  LET d = INT (RND * 8)
160  LET r = INT (RND * 18) + 1
170  LET c = INT (RND * 18) + 1
180  GO SUB fillin
190  IF NOT filled THEN GO TO 150
200  NEXT i
210  FOR r = 1 TO 18
220  FOR c = 1 TO 18
230  IF s$ (r, c) = "□" THEN LET s$ (r, c) = CHR$ (INT (RND * 26) + 97)
240  NEXT c
250  NEXT r
260  FOR r = 1 TO 18
270  PRINT AT r, 0; CHR$ (r + 96)
280  PRINT AT r, 2; s$ (r)
290  NEXT r
300  PRINT AT 20, 2; "□□□□□□□□□111111111"
310  PRINT AT 21, 2; "123456789012345678"
315  GO SUB list
320  INPUT "Word start"; b$, bn
322  PRINT AT 16, 25; "□□□□□□"
```

```
325  INPUT "and direction"; d
330  INPUT "Which word is it"; f$
340  GO SUB test
350  IF valid THEN GO SUB colour
360  IF n < nw THEN GO TO 320
361  PRINT FLASH 1; AT 16, 22; "Well done"
370  GO TO 9999

1000 LET p = 1
1010 IF w$ (i, p) = "□" THEN LET filled = 1: RETURN
1020 IF r > 18 OR r < 1 OR c > 18 OR c < 1 THEN GO SUB rubout:
     LET filled = 0: RETURN
1030 IF s$ (r, c) < > "□" THEN GO SUB rubout: LET filled = 0: RETURN
1035 LET s$ (r, c) = w$ (i, p)
1040 LET p = p + 1
1050 GO TO 1100 + d * 10
1100 LET c = c + 1: GO TO 1010
1110 LET c = c + 1: LET r = r - 1: GO TO 1010
1120 LET r = r - 1: GO TO 1010
1130 LET c = c - 1: LET r = r - 1: GO TO 1010
1140 LET c = c - 1: GO TO 1010
1150 LET c = c - 1: LET r = r + 1: GO TO 1010
1160 LET r = r +1: GO TO 1010
1170 LET r = r + 1: LET c = c + 1: GO TO 1010

2000 LET r = CODE b$ - 96
2010 LET c = bn
2020 FOR p = 1 TO LEN f$
2030 IF f$ (p) < > s$ (r, c) THEN PRINT FLASH 1; AT 16, 25; "Wrong!":
     LET valid = 0: RETURN
2040 GO TO 2100 + 10 * d
2100 LET c = c + 1: GO TO 2180
2110 LET c = c + 1: LET r = r - 1: GO TO 2180
2120 LET r = r - 1: GO TO 2180
2130 LET c = c - 1: LET r = r - 1: GO TO 2180
```

```
2140  LET c = c - 1: GO TO 2180
2150  LET c = c - 1: LET r = r + 1: GO TO 2180
2160  LET r = r + 1: GO TO 2180
2170  LET r = r + 1: LET c = c + 1: GO TO 2180
2180  NEXT p
2190  LET valid = 1
2195  LET n = n + 1
2197  GO SUB delete
2200  RETURN

3000  LET r = CODE b$ - 96
3005  LET pc = pc + 1
3006  IF pc > 6 THEN LET pc = 1
3010  LET c = bn + 1
3020  FOR p = 1 TO LEN f$
3030  PRINT PAPER pc; AT r, c; f$ (p)
3040  GO TO 3100 + 10 * d
3100  LET c = c + 1: GO TO 3180
3110  LET c = c + 1: LET r = r - 1: GO TO 3180
3120  LET r = r - 1: GO TO 3180
3130  LET c = c - 1: LET r = r - 1: GO TO 3180
3140  LET c = c - 1: GO TO 3180
3150  LET c = c - 1: LET r = r + 1: GO TO 3180
3160  LET r = r + 1: GO TO 3180
3170  LET r = r + 1: LET c = c + 1: GO TO 3180
3180  NEXT p
3185  PAPER 7
3190  RETURN

4000  IF p = 1 THEN RETURN
4010  LET p = p - 1
4020  GO TO 4100 + 10 * d
4100  LET c = c - 1: GO TO 4180
4110  LET c = c - 1: LET r = r + 1: GO TO 4180
4120  LET r = r + 1: GO TO 4180
```

```
4130  LET c = c + 1: LET r = r + 1: GO TO 4180
4140  LET c = c + 1: GO TO 4180
4150  LET c = c + 1: LET r = r - 1: GO TO 4180
4160  LET r = r - 1: GO TO 4180
4170  LET r = r - 1: LET c = c - 1: GO TO 4180
4180  LET s$ (r, c) = "□"
4190  GO TO 4000

5000  PRINT AT 0, 22; "Missing"
5010  PRINT AT 1, 22; "words . ."
5020  FOR i = 1 TO nw
5030  PRINT PAPER 6; AT i + 2, 21; w$ (i)
5040  NEXT i
5050  RETURN

6000  LET w$ (11) = f$
6005  FOR i = 1 TO nw
6010  IF w$ (11) = w$ (i) THEN PRINT AT i + 2, 21;
      "□□□□□□□□□□□": RETURN
6020  NEXT i
```




```
    1 LET fillin=1000: LET test=2
    2 LET color=3000: LET rubout=
000
    3 LET n=0
    4 RANDOMIZE
    5 CLS
    6 LET pc=1
   10 DIM w$(11, 11)
   20 DIM s$(18,18)
  100 INPUT "Quantas palavras? ";
nw
  110 FOR i=1 TO nw
  120 INPUT "Palavra? ";w$(i)
```

```
 130 NEXT i
 140 FOR i=1 TO nw
 150 LET d=INT (RND*8)
 160 LET r=INT  RND*18)+1
 170 LET c=INT  RND*18)+1
 180 GO SUB fillin
 190 IF NOT filled THEN GO TO 15
0
 200 NEXT i
 210 FOR r=1 TO 18
 220 FOR c=1 TO 18
 230 IF s$(r,c)=" " THEN LET s$(
r,c)=CHR$ (INT  RND*26)+97)
 240 NEXT c
 250 NEXT r
 260 FOR r=1 TO 18
 270 PRINT AT r,0;CHR$ (r+96)
 280 PRINT AT r,2;s$(r)
 290 NEXT r
 300 PRINT AT 20,2;"          111
111111"
 310 PRINT AT 21,2;"123456789012
3456?8"
 315 GO SUB list
 320 INPUT "Primeira palavra? ";
b$,bn
 322 PRINT AT 16,25;"      "
 325 INPUT "Direccao? ";d
 330 INPUT "Qual e a palavra? ";
w$
 340 GO SUB test
 350 IF valid THEN GO SUB colour
 360 IF n<nw THEN GO TO 320
 361 PRINT FLASH 1;AT 16,22;"Con
seguiu!"
 370 GO TO 9999
1000 LET p=1
1010 IF w$(i,p)=" " THEN LET fil
led=1: RETURN
1020 IF r>18 OR r<1 OR c>18 OR c
<1 THEN GO SUB rubout: LET fille
d=0: RETURN
1030 IF s$(r,c)<>"" THEN GO SUB
rubout: LET filled=0: RETURN
1035 LET s$(r,c)=w$(i,p)
1040 LET p=p+1
1050 GO TO 1100+d*10
1100 LET c=c+1: GO TO 1010
1110 LET c=c+1: LET r=r-1: GO TO
 1010
1120 LET r=r-1: GO TO 1010
1130 LET c=c-1: LET r=r-1: GO TO
 1010
1140 LET c=c-1: GO TO 1010
1150 LET c=c-1: LET r=r+1: GO TO
 1010
1160 LET r=r+1: GO TO 1010
1170 LET r=r+1: LET c=c+1: GO TO
```

```
 1010
2000 LET r=CODE b$-96
2010 LET c=bn
2020 FOR p=1 TO LEN f$
2030 IF f$(p)<>s$(r,c) THEN PRIN
T FLASH 1;AT 16,25;"ERROU!!": LE
T valid=0: RETURN
2040 GO TO 2100+10*d
2100 LET c=c+1: GO TO 2180
2110 LET c=c+1: LET r=r-1: GO TO
 2180
2120 LET r=r-1: GO TO 2180
2130 LET c=c-1: LET r=r-1: GO TO
 2180
2140 LET c=c-1: GO TO 2180
2150 LET c=c-1: LET r=r+1: GO TO
 2180
2160 LET r=r+1: GO TO 2180
2170 LET r=r+1: LET c=c+1: GO TO
 2180
2180 NEXT p
2190 LET valid=1
2195 LET n=n+1
2197 GO SUB delet
2200 RETURN
3000 LET r=CODE b$-96
3005 LET pc=pc+1
3006 IF pc>6 THEN LET pc=1
3010 LET c=bn+1
3020 FOR p=1 TO LEN f$
3030 PRINT PAPER pc;AT r,c;f$(p)
3040 GO TO 3100+10*d
3100 LET c=c+1: GO TO 3180
3110 LET c=c+1: LET r=r+1: GO TO
 3180
3120 LET r=r-1: GO TO 3180
3130 LET c=c-1: LET r=r-1: GO TO
 3180
3140 LET c=c-1: GO TO 3180
3150 LET c=c-1: LET r=r+1: GO TO
 3180
3160 LET r=r+1: GO TO 3180
3170 LET r=r+1: LET c=c+1: GO TO
 3180
3180 NEXT p
3185 PAPER 7
3190 RETURN
4000 IF p=1 THEN RETURN
4010 LET p=p-1
4020 GO TO 4100+10*d
4100 LET c=c-1: GO TO 4180
4110 LET c=c-1: LET r=r+1: GO TO
 4180
4120 LET r=r+1: GO TO 4180
4130 LET c=c+1: LET r=r+1: GO TO
 4180
4140 LET c=c+1: GO TO 4180
```

```
4150 LET c=c+1: LET r=r-1: GO TO
 4180
4160 LET r=r-1: GO TO 4180
4170 LET r=r-1: LET c=c-1: GO TO
 4180
4180 LET s$(r,c)=" "
4190 GO TO 4000
5000 PRINT AT 0,22;"Palavras"
5010 PRINT AT 1,22;"Esquecidas..
"
5020 FOR i=1 TO nw
5030 PRINT PAPER 6;AT i+2,21;w$(
i)
5040 NEXT i
5050 RETURN
6000 LET w$(11)=f$
6005 FOR i=1 TO nw
6010 IF w$(11)=w$(i) THEN PRINT
AT i+2,21;"                ": RETURN
6020 NEXT i
```

NOTAS

19

CRIPTOGRAMAS

cryptograms

O computador seleciona uma frase e mostra uma versão codificada da mesma (sem espaços entre as palavras) no écran. Você deve descodificá-la até chegar à frase original. Pode colocar espaços na fase codificada para a tornar mais legível. Mas pense bem — só tem duas tentativas!

Não pense que quando conseguir descodificar a frase o «puzzle» deixou de ter interesse, porque o computador modifica o sistema de codificação sempre que você pede uma nova frase.

NOTA (1) Durante a 1.ª Grande Guerra os Serviços Secretos Navais Britânicos inventaram uma grande quantidade de sistemas de código. Para testar um deles, do qual eles estavam muito orgulhosos, enviaram uma pequena frase codificada aos seus colegas Americanos para que a decifrassem. No dia seguinte receberam uma mensagem com a sua descodificação:

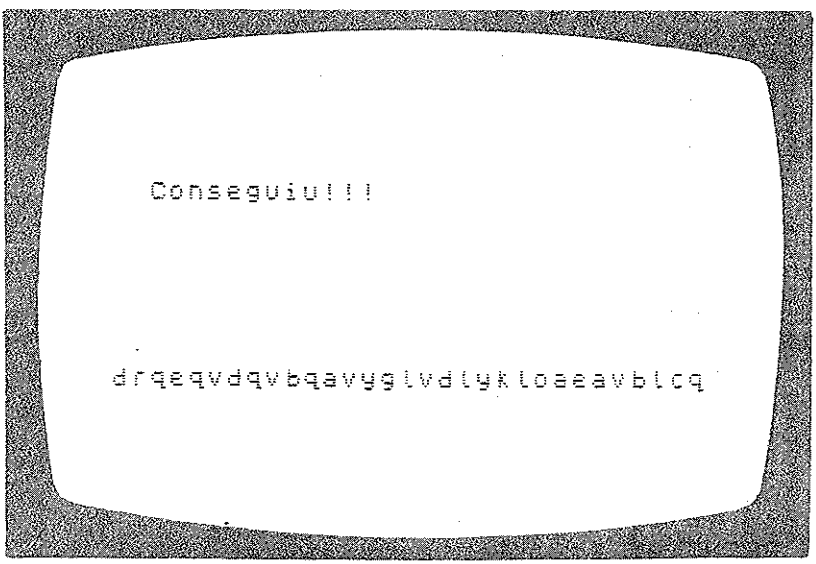
«Este código é absolutamente indecifrável»
«This code is absolutely unbreakable»

Por curiosa coincidência esta frase sem espaços ocupa precisamente 31 espaços do écran e por isso escolhemo-la como uma das mensagens para ser descodificada. Todas as outras mensagens têm também 31 letras e portanto nem sequer é necessário contar as letras da mensagem codificada.

NOTA (2) Não vai ser muito difícil resolver estes criptogramas porque você tem as respostas nas linhas de DATA e portanto não o vou ajudar muito. No entanto dir-lhe-ei que o código que o computador usa é sempre de substituição o que quer dizer que se o computador troca o «a» por «g» no início da fase também o trocou no fim da mesma. Se quer tornar as coisas um pouco mais difíceis para si porque não pede a um amigo para lhe modificar as linhas do programa DATA. Após isso nada de batota!

```
  1 RANDOMIZE
  2 DIM w$ (5, 31): DIM t (26): DIM n$ (31)
  3 DATA " this code is absolutely unbreakable"
  4 DATA "the sentence in front of you is in code"
  5 DATA "translating codes is a black art form"
  6 DATA "cryptologists are men who study code"
  7 DATA "a cryptographer deciphers messages"
 10 LET trys = 0
 70 FOR p = 1 TO 26
 80 LET t (p) = p
 90 NEXT p
100 FOR r = 1 TO 5
110 LET p = 1
120 READ m$
130 FOR c = 1 TO 31
140 IF m$ (p) = "□" THEN LET p = p + 1: GO TO 140
150 LET w$ (r, c) = m$ (p)
160 LET p = p + 1
170 NEXT c
180 NEXT r
190 FOR i = 1 TO 50
200 LET p = INT (RND * 26) + 1
210 LET q = INT (RND * 26) + 1
220 LET temp = t (p)
230 LET t (p) = t (q)
240 LET t (q) = temp
250 NEXT i
252 FOR i = 1 TO 26
253 LET t (i) = t (i) + 96
254 NEXT i
260 LET r = INT (RND * 5) + 1
270 FOR c = 1 TO 31
280 LET n$ (c) = CHR$ (t (CODE w$ (r, c) − CODE "a" + 1) )
290 NEXT c
```

```
300 PRINT AT 2, 2; "Here's the message . . ."
310 PRINT AT 10, 0; n$
320 INPUT "What's your decode?"; d$
330 LET c = 1
340 FOR p = 1 TO LEN d$
350 IF d$ (p) = "□" THEN GO TO 380
360 LET n$ (c) = d$ (p)
370 LET c = c + 1
380 NEXT p
385 LET trys = trys + 1
390 IF n$ < > w$ (r) AND trys < 2 THEN PRINT AT 2, 2;
    "Try again [15espaços]": GO TO 320
400 IF n$ = w$ (r) THEN PRINT AT 2, 2; "You've got it!!!! □ □ □ □ □ □ □"
410 INPUT "Want another one?"; q$
420 IF q$ = "yes" THEN CLS: RESTORE: GO TO 1
```




```
    1 RANDOMIZE
    2 DIM w$(5,31): DIM t(26): DI
M n$(31)
    3 DATA "thiscodeisabsolutelyu
nbreakable"
```



```
    4 DATA "thesentenceinfrontofy
ouisincode"
    5 DATA "translatingcodesisabl
ackartform"
    6 DATA "cryptologistsaremenwh
ostudycode"
    7 DATA "acryptographerdeciphe
rsmessages"
   10 LET trys=0
   70 FOR p=1 TO 26
   80 LET t(p)=p
   90 NEXT p
  100 FOR r=1 TO 5
  110 LET p=1
  120 READ m$
  130 FOR c=1 TO 31
  140 IF m$(p)=" " THEN LET p=p+1
: GO TO 140
  150 LET w$(r,c)=m$(p)
  160 LET p=p+1
  170 NEXT c
  180 NEXT r
  190 FOR i=1 TO 50
  200 LET p=INT (RND*26)+1
  210 LET q=INT (RND*26)+1
  220 LET temp=t(p)
  230 LET t(p)=t(q)
  240 LET t(q)=temp
  250 NEXT i
  252 FOR i=1 TO 26
  253 LET t(i)=t(i)+96
  254 NEXT i
  260 LET r=INT (RND*5)+1
  270 FOR c=1 TO 31
  280 LET n$(c)=CHR$ (t(CODE w$(r
,c)-CODE "a"+1))
  290 NEXT c
  300 PRINT AT 2,2;"Aqui esta a m
ensagem..."
  310 PRINT AT 10,0;n$
  320 INPUT "Como descodifica? ";
d$
  330 LET c=1
  340 FOR p=1 TO LEN d$
  350 IF d$(p)=" " THEN GO TO 380
  360 LET n$(c)=d$(p)
  370 LET c=c+1
  380 NEXT p
  385 LET trys=trys+1
  390 IF n$<>w$(r) AND trys<2 THE
N PRINT AT 2,2;"Tente outra vez
                ": GO TO 320
  400 IF n$=w$(r) THEN PRINT AT 2
,2;"Conseguiu!!!
    "
  410 INPUT "Outra vez? ";q$
  420 IF q$="" THEN CLS : RESTORE
  : GO TO 1
```

Consegue:

- Atravessar com êxito a floresta mágica?
- Transportar o lobo, a cabra e a couve através do rio?
- Distribuir igualmente o vinho sem derramá-lo?

Todos estes programas foram concebidos para funcionarem no Spectrum, sendo os 15 primeiros concebidos para o ZX 81 e depois modificados com alguns pormenores do Spectrum para lhes dar mais beleza.

NOTA: Necessita de 16K RAM no seu ZX 81.

Boa sorte!

GO TO Informática / 3
061900385
EDIÇÕES CETOP