Este livro destina-se tanto aos professores como aos pais que gostariam de ver os seus filhos fazer algo mais com o computador do que jogos de aventuras ou repelir extra-terrestres. O seu autor, Tim Hartnell, é um nome já conhecido dos leitores da colecção Tempos Livres, desde a publicação de *Como Programar o seu ZX Spectrum,* que ele escreveu conjuntamente com Dilwyn Jones. Neste volume, ele e os seus colaboradores apresentam inúmeras ideias e programas aplicáveis tanto ao ensino das matemáticas, como ao da geografia, química, linguagens e outras matérias. Christine Johnson contribui com um relato pormenorizado e ilustrado da sua experiência de trabalho com o computador, nas classes pré-primárias. Os numerosos apêndices incluem sugestões para leituras mais avançadas, uma introdução às linguagens LOGO, PROLOG e aos gráficos Turtle.

EDITORIAL PRESENÇA

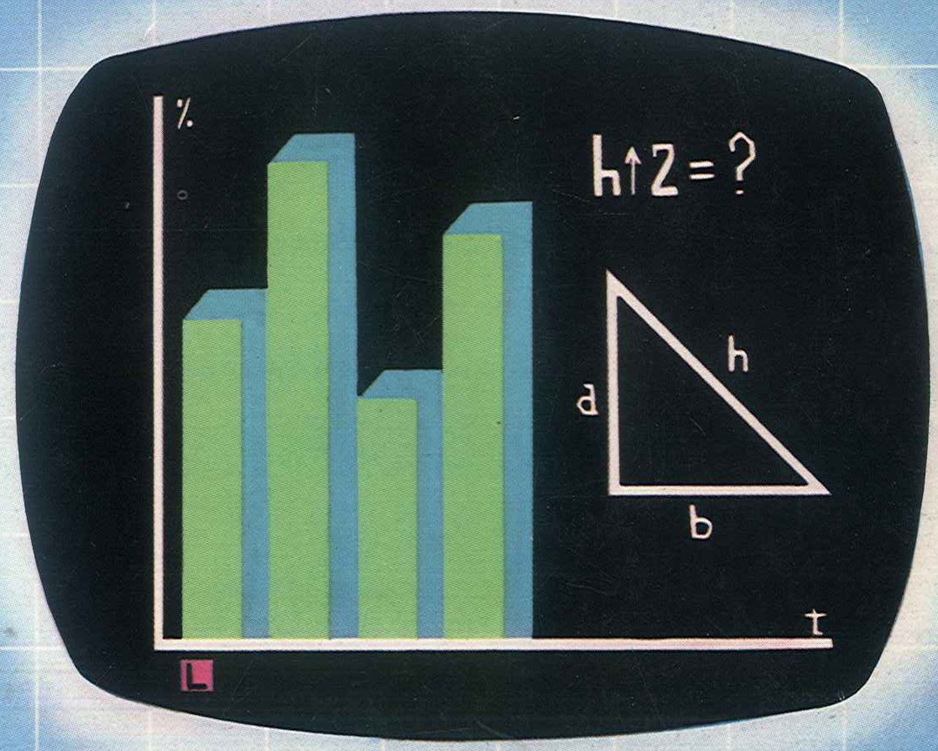
ZX Spectrum na Educação

177

TIM HARTNELL

CHRISTINE JOHNSON DAVID VALENTINE

ZX SPECTRUM NA EDUCAÇÃO



TEMPOS LIVRES

CULTURA E TEMPOS LIVRES

1. ABC do Xadrez, *Petar Trifunovitch* e *Sava Vukovitch*
4. ABC do Bridge, *Pierre Jais* e *H. Lahana*
5. Guia Prático de Fotografia, *W. D. Emanuel*
6. ABC do Judo, *E. J. Harrison*
7. Como Fazer Cinema, *Paul Petzold*
8. Bridge Moderno, *Pierre Jais* e *H. Lahana*
9. Fotografia — Técnicas e Truques I, *Edwin Smith*
10. ABC dos Estilos, da Arquitectura ao Mobiliário, *A. Aussel*
11. Fotografia — Técnicas e Truques II, *Edwin Smith*
12. A Pesca Submarina, *António Ribera*
13. Teoria dos Finais de Partida, *Yuri Averbach*
14. Aprenda Rádio, *B. Fighiera*
15. Guia do Cão, *Louise Laliberté-Robert e Jean-Pierre Robert*
16. ABC do Aquário, *Anthony Evans*
17. Iniciação à Electricidade e Electrónica, *Fernand Huré*
18. Os Transistores, *Fernand Huré*
19. Karaté I, *Albrecht Pflüger*
20. Iniciação ao Radiocomando dos Modelos Reduzidos, *C. Péricone*
21. Construa o seu Receptor, *B. Fighiera*
22. Montagens Electrónicas, *B. Fighiera*
23. O Berbequim Eléctrico, *Villy Dreier*
24. Cactos, *J. Nilaus Jensen*
25. Iniciação à Alta Fidelidade, *Peter Turner*
26. O Aquário de Água Doce, *Paulo de Oliveira*
27. ABC do Ténis, *Fonseca Vaz*
28. Karaté II, *Albrecht Pflüger*
29. ABC da Criação de Canários, *Curt Af Enehjelm*
30. Ginástica Feminina, *Sonja Helmer Jensen*
31. Cartomancia, *Rhea Koch*
32. Calculadoras Electrónicas de Bolso, *E. Dam Ravn*
33. O Pastor Alemão, *Gilles Legrand*
34. Xadrez — Teoria do Meio Jogo I, *Bondarevsky*
35. Manual do Super 8 — I, *Myron A. Matzkin*
36. ABC da Criação de Periquitos, *Cyril H. Rogers*
37. O Livro dos Gatos, *Barbel Gerber* e *Horst Bielfeld*
38. Manual do Super 8 — II, *Myron A. Matzkin*
39. ABC do Mergulho Desportivo, *Walter Mattes*
40. Circuitos Integrados/Aplicações Práticas, *F. Bergtold*
41. A Apicultura, *H. R. C. Riches*
42. ABC do Cultivo das Plantas, *H. G. Witham Fogg*
43. ABC da Criação de Pombos, *Kai R. Dahl*
44. Construção de Caixas Acústicas de Alta Fidelidade, *R. Brault*
45. Raças de Canários, *Klaus Speicher*
46. Jogos de Cartas, *Graciano Dolma*
47. Cocker Spaniels, *H. S. Lloyd*
48. ABC da Caça, *Fabián Abril*
49. Aprenda Televisão, *Gordon J. King*
50. Iniciação à Pesca, *Juan Nadal*
51. Basquetebol, *Marius Norregard*
52. Cães de Caça, *Santiago Pons*
53. Aprenda Electrónica, *T. L. Squires* e *C. M. Deason*
54. A Avicultura, *Jim Worthington*
55. A Produção de Coelhos, *P. Surdeau* e *R. Henaff*
56. ABC dos Computadores, *T. F. Fry*
57. Natação para Crianças, *John Idorn*
58. O Boxer, *Anni Mortensen*
59. Voleibol, *Ole Hansen* e *Per-Göran Pesson*
60. Iniciação à Vela, *Donald Law*
61. ABC da Filatelia, *Jacqueline Caurat*
62. A Pesca à Beira-Mar, *J.-M. Böelle* e *D. Doyen*
63. Enxerto de Árvores de Fruto, *Alejo Rigau*
64. A Cultura do Morangueiro, *Luís Alsina Grau*
65. Emissores-Receptores (Walkies-Talkies), *P. Duranton*
66. Iniciação à Fotoelectrónica, *Heinz Richter*
67. Doces e Conservas de Fruta, *Robin Howe*
68. A Criação de Hamsters, *C. F. Snow*
69. A Criação de Porcos, *Roy Genders*
70. Calendário do Horticultor, *Luís Alsina Grau*
71. Jogos Electrónicos, *F. G. Rayer*
72. Cultivo de Cogumelos e Trufas, *Alejo Rigau*
73. Aprenda Televisão a Cores, *Gordon J. King*
74. Gravação em Fita Magnética, *Ian R. Sinclair*
75. Poda de Árvores e Arbustos, *Roy Genders*
76. Como Treinar o Seu Cão, *E. Fitch Daglish*
77. Instrumentos de Medida e Verificação, *Heinrich Stöckle*
78. A Criação de Caracóis, *Matías Josa*
79. Rádio — Fundamentos e Técnicas, *Gordon J. King*
80. Como Fazer Gelados, *Sylvie Thiébault*
81. Iniciação à Jardinagem, *Noel Clarasó*
82. A Congelação dos Alimentos, *Suzanne Lapointe*
83. Windfsurf — Prancha à Vela, *Ernstfried Prade*
84. Raças de Cães, *O. Hasselfeldt*
85. Rummy e Canasta, *Claus D. Grupp*
86. A Encadernação, *Annie Persuy*
87. Aprenda Electricidade, *Heinz Richter*
88. Taxidermia, Embalsamamento de Aves e Mamíferos, *Harry Hjortaa*
89. Jogging — Correr para Manter a Forma, *Werner Sonntag*
90. ABC da Cozinha Chinesa, *Sonya Richmond*
91. Jogos T.V., *C. Tavernier*
92. Amplificadores de Som, *Richard Zierl*
93. O Livro do Poker, *Claus D. Grupp*
94. Aprenda a Desenhar, *Rose-Marie de Prémont* e *Nicole Philippi*
95. O Minitrampolim na Escola, *Sonja Helmer Jensen* e *Klaus Dano*
96. Jogos de Luzes e Efeitos Sonoros para Guitarras, *B. Fighiera*
97. O Cultivo do Tomate, *Louis N. Flawn*
98. Pilhas Solares, *F. Juster*
99. A Criação Doméstica de Coelhos, *C. F. Snow*
100. Iniciação ao Futebol, *Wieland Männle* e *Heinz Arnold*
101. Horóscopos Chineses, *Georg Haddenbach*
102. Guia Prático de Marcenaria, *Charles H. Hayward*
103. Andebol, *Fritz* e *Peter Hattig*
104. Dispositivos Anti-Roubo, *H. Schreiber*
105. Perus, Pintadas e Codornizes, *Jérome Sauze*
106. Crepes — Doces e Salgados, *Florence Arzel*
107. Aperitivos e Entradas, *Myrette Tiano*
108. Ténis de Mesa, *Leslie Woollard*
109. Aprenda Surf, *R. Abbott* e *M. Baker*
110. Futebol — Técnica e Táctica, *Kurt Lavall*
111. A Vaca Leiteira, *Colin T. Whittemore*
112. O Cubo Mágico, *Josef Trajber*
113. O Perdigueiro Português, *José M. Correia*
114. Pizzas e Massas à Italiana, *Marieanne Ränk*
115. O Cubo Para Quem Já o Faz, *Josef Trajber*
116. A Pirâmide Mágica, A Torre, O Barril do Diabo, *M. Mrowka-W. J. Weber*
117. Gansos e Patos, *Marie Mourthe*
118. Iniciação ao Kung-Fu, *A. P. Harrington*
119. Electrónica e Fotografia, *Hanns-Peter Siebert*
120. O Livro da Fortuna, *Douglas Hill*
121. Construção de um Alimentador de corrente, *Waldemar Baitinger*
122. Hóquei em Patins, *Francisco Velasco*
123. Técnicas de Tiro, *Anton Kovacic*
124. Aprenda a Tricotar, *Uta Mix*
125. ABC da Patinagem, *Christa-Maria* e *Richard Kerler*
126. A Pesca e os seus Segredos, *Armand Deschamps*
127. O Osciloscópio, *R. Rateau*
128. Guia Prático da Banda do Cidadão, *T. M. Normand*
129. Sumos e Batidos, *Manfred Donderski*
130. Introdução à Programação de Microcomputadores, *Peter C. Sanderson*
131. Aprenda Croché, *Uta Mix*
132. ABC do Microprocessador, *P. Mélusson*
133. Guia Prático de Basic, *Roger Hunt*
134. Introdução à Electrónica Digital, *Ian R. Sinclair*
135. ABC do Vídeo, *David K. Matthewson*
136. Fotografia em Movimento, *Don Morley*
137. Guia de Cobol, *Ray Welland*
138. Fotografia a Pequena Distância, *Sidney F. Ray*
139. Guia Moderno da Canaricultura, *Manuel Gonçalves*
140. Minielectrónica para Amadores, *Heinz Richter*

141. ABC da Programação de Computadores, *John Shelley*
142. Tarot — O Futuro Pelas Cartas, *Edwin J. Nigg*
143. ABC da Equitação, *Dorothy Johnson*
144. Como Programar o Seu ZX 81, *Patrick Gueulle*
145. 100 Avarias TV e a Maneira Prática de as Detectar, *P. Duranton*
146. ABC da Horticultura, *Louis Giordano*
147. Basic Para Microcomputadores, *A. P. Stephenson*
148. Como Programar o seu ZX Spectrum, *Tim Hartnell* e *Dilwyn Jones*
149. Iniciação aos Motores Diesel, *David S. Maclean*
150. 60 Jogos para o ZX Spectrum, *David Harwood*
151. As Linhas da Mão, *Rosé Hubert*
152. Cozinha Italiana, *Rotraud Degner*
153. Manual do ZX Spectrum, *R. J. Simpson* e *T. J. Terrel*
154. Z80 Assembler para o ZX Spectrum — Iniciação ao Código de Máquina, *João Paulo Fragoso*
155. Aeróbica, *H. Schulz*
156. ABC do Atletismo, *Denis Watts*
157. 26 Programas Basic para Microcomputadores, *Derrick Daines*
158. Aprenda Pascal no seu Microcomputador, *Jeremy Ruston*
159. Guia Moderno da Suinicultura, *Colin Whittemore*
160. O Bar em Sua Casa — 888 Cocktails, *Aladar von Wesendonk*
161. Código de Máquina para Principiantes, *James Walsh*
162. Código de Máquina para Programadores Avançados, *Paul Holmes*
163. ABC da Fruticultura, *Henri Gosselin*
164. ABC da Canoagem, *Alan Byde*
165. Guia de Fortran, *Philip Ridler*
166. Manual da Secretária, *Philippa Ramage*
167. ABC das Antenas, *Gordon J. King*
168. Programar Aventuras no Seu Computador, *Andrew Nelson*
169. Guia do Sinclair QL, *Boris Allan*
170. Novas Aventuras no seu ZX Spectrum, *Peter Shaw e James Mortleman*
171. O Computador no Escritório, *John Shelley*
172. Sobremesas, *Fred Timber*
173. Rádio — Do Circuito Oscilante ao Receptor de Ondas Curtas, *Richard Zierl*
174. ABC das Aberturas, *V. N. Panov*
175. A Construção de Pequenos Transformadores, *M. Dorian e F. Juster*
176. Guia de Pascal, *David Watt*
177. O ZX Spectrum na Educação, *Tim Hartnell, Christine Johnson e David Valentine*

TIM HARTNELL
CHRISTINE JOHNSON **DAVID VALENTINE**

O ZX SPECTRUM NA EDUCAÇÃO

EDITORIAL PRESENÇA

Título original:
EDUCATIONAL USES OF THE ZX SPECTRUM
Copyright © *1983* Tim Hartnell, Christine Johnson and David Valentine
Tradução de Conceição Jardim e Eduardo Nogueira
Capa de Rui Ligeiro

Reservados todos os direitos
para a língua portuguesa à
EDITORIAL PRESENÇA, LDA.
Rua Augusto Gil, 35-A — 1000 LISBOA

AGRADECIMENTOS

Várias pessoas nos ajudaram a escrever este livro, e gostaríamos de lhes agradecer o seu auxílio.

Em primeiro lugar, e em particular, devemos citar Mr. Crispin Hill, Director da Aldro School de Shackleford, que actuou como consultor neste nosso projecto. Muitos dos programas contidos neste volume baseiam-se nos da brochura de Mr. Hill intitulada "Notas Introdutórias e Programas Simples para o ZX 81".

Outros que contribuíram com programas para este volume foram Jeremy Ruston, David Perry, Jim Walsh, Gwyn Dewey, Paul Toland, Derek Cook e Gordon Armitt.

Esperamos que este livro ajude os leitores a usarem de modo mais eficaz o Spectrum em educação.

Tim Hartnell, Londres
Christine Johnson, Nottingham
David Valentine, Newthorpe

1

COMPUTADORES E EDUCAÇÃO

Neste livro vamos estudar algumas ideias para uso do ZX Spectrum em educação. Quer o leitor seja um professor interessado no uso do computador para ensinar alunos em disciplinas informáticas, ou para auxiliar ao ensino de um outro tema, encontrará certamente aqui algum material que lhe interessará.

Talvez seja um pai, que comprou o Spectrum para o filho com a esperança de ele o utilizar em apoio do seu trabalho escolar. Talvez tenha no entanto ficado um tanto desiludido ao verificar que a principal aplicação que o seu filho dá à máquina consiste em destruir extraterrestres. Se for este o caso, encontrará neste livro informações suficientes para garantir que pelo menos algumas vezes o Spectrum seja usado para aquilo que o comprou. Se bem que o livro seja principalmente dirigido a professores, grande parte do que será dito aplicar-se-á também ao seu caso. Leia a obra tendo isto em conta, adaptando-a às suas próprias necessidades.

Os microcomputadores começam hoje a estar bastante difundidos no sistema de educação de alguns países. A grande variedade de máquinas impediu até certo ponto a criação de uma biblioteca de programas universal (uma biblioteca de "software"). O material apresentado neste livro servirá até certo ponto para rectificar essa lacuna no caso do Spectrum. Os programas aqui incluídos servem principalmente para informarem o leitor sobre o modo como é possível escrever software educativo, constituindo programas-tipo que depois podem ser adaptados às suas necessidades.

As exigências dos temas específicos são claramente definidas. Não é realista esperar que um programa escrito a propósito de um dado tema se aplique a outro, ou que um programa criado para auxiliar os estudantes a um dado nível num campo tenha algum uso para estudantes que se ocupem do mesmo campo mas a um nível superior. No entanto, se utilizar este livro como guia, como fonte de

ideias, verificará que pode poupar bastante tempo na preparação de material para os seus estudantes ou filhos.

Um facto perturbador — mas que deve ser enfrentado desde início — é que muitas das crianças com quem o leitor trabalhará já sabem muito mais sobre computadores e programação destes do que você. Ser educado com um microcomputador já produziu, pelo menos em certos países, uma geração que em grande parte não se sente de modo algum ameaçada pela máquina. O jargão exótico — bits, bytes, RAM ou ROM — é um mar onde muitos jovens parecem nadar sem quaisquer problemas.

Incluí uma secção concebida para ensinar os rudimentos da programação em Basic (a linguagem utilizada pelo computador, e que significa Beginners All - Purpose Symbolic Instruction Code, Código de Instruções Simbólicas de Uso Geral para Principiantes, desenvolvida no Dartmouth College, nos Estados Unidos, em 1974). Depois de ter aprendido a programar nesta secção do livro, poderá usar o mesmo esquema para ensinar Basic nas suas aulas.

Este livro parte de certos pressupostos quanto aos métodos de ensino. Se não concordar com tudo o que dissermos, poderá mesmo assim adaptar o material às suas próprias necessidades. Neste livro, dirigimo-nos a uma vasta audiência, e é inevitável que algumas secções da obra sejam mais interessantes para si do que outras.

O leitor talvez discorde também de algumas das nossas afirmações. Elimine tudo o que lhe parecer incorrecto, não apropriado aos seus métodos de ensino, ou inaplicável no ambiente da sua escola. Como já dissemos, considere este livro como uma fonte de ideias. Ninguém pode dizer "Esta é a única forma de usar o Spectrum em educação". Não pense que sugerimos isto em relação às propostas deste livro, mesmo que em alguma das suas partes a nossa posição lhe pareça inflexível. Em vez de gastarmos o livro inteiro referindo-nos a cada sugestão dizendo "se lhe parecer apropriado", ou "se concordar com a sua filosofia da educação", etc., preferimos partir do princípio de que o leitor recusará o que não lhe interessar.

Apresentamos aqui uma vasta gama de material. Nem todo ele se aplicará directamente às suas necessidades, se bem que os autores gostassem de saber que todo ele despertará o seu interesse. A maior parte das informações deverão ajudá-lo a descobrir maneiras de usar o Spectrum. Mas utilize o livro como entender; a sequência de material aqui apresentada é apenas uma das possíveis, e poderá evidentemente aprofundar um qualquer ponto do livro que lhe pareça mais relevante.

No fim do livro encontra-se um glossário de termos. É certo que já foram publicados diccionários inteiros de termos de informática,

mas tentámos apresentar aqui os termos mais comuns, sem preocupações de exaustividade. Verificará que os termos técnicos cujo significado não seja evidente para si se encontrarão melhor explicados nesse glossário.

Existem portanto muitas maneiras de utilizar o Spectrum na educação. Vamos discutir apenas algumas delas, segundo uma lista que não é exaustiva e que deve levar o leitor a pensar noutros modos de recorrer à máquina para as suas necessidades.

O Spectrum pode ser usado para ajudar a ensinar um tema, apresentando os factos de um modo interessante, claro e, talvez, interactivo. Isto é, um programa que apresente, por exemplo, a tabela dos elementos químicos pode ser feito de tal modo que recorra à cor para sublinhar aqueles que não ocorrem naturalmente (usando talvez a cintilação — FLASH — para indicar os elementos instáveis); ou pode ser escrito de tal modo que permita isolar facilmente um grupo de elementos com uma dada característica específica. Quando começamos a pensar neste simples exemplo, apercebemo-nos de que tudo aquilo que pode ser apresentado sob a forma de uma tabela pode ser tornado interessante, convidando a uma interacção com os alunos. Não existe nenhuma maneira de uma tabela impressa num livro de estudo poder *envolver* os estudantes do mesmo modo que uma tabela de computador convenientemente apresentada.

Existe um vasto número de programas de cálculo que convidam os utilizadores a usarem dados num ambiente fictício. Esses programas permitem ao utilizador apresentar dados à máquina sobre coisas como custos, vendas, produzir folhas de balanço, e fazer depois perguntas do tipo: "E que aconteceria se duplicássemos as vendas?", "Que aconteceria se perdessemos determinado contrato?". O programa encarrega-se depois dos cálculos, alterando todos os valores afectados pela hipótese posta. Existem maneiras de utilizar o mesmo sistema para enriquecer a experiência dos estudantes que trabalham com material apresentado em tabelas; por exemplo, pode-se construir um programa simulador que permita aos alunos misturar produtos químicos em diferentes proporções — ao ponto de provocarem explosões...

O Spectrum pode ser usado em matemáticas, e mais adiante serão sugeridas várias aplicações neste campo. Se bem que o computador possa ser usado apenas como uma máquina de calcular, usá-lo desse modo equivale a perder grande parte das potencialidades da máquina. Como verá na secção sobre matemáticas, existem muitas maneiras de utilizar eficazmente o computador nesta área.

Já mencionámos o uso do computador para produzir tabelas de dados susceptíveis de atrair o interesse dos estudantes. Existem

muitas outras coisas, além das tabelas, que podem ser apresentadas pelo computador de um modo que encoraje o envolvimento dos estudantes, como sejam mapas no caso da geografia, e gráficos de todos os tipos para uma vasta variedade de temas. Algumas áreas do conhecimento humano, como por exemplo as ideias de Malthus relativas à interacção entre os recursos disponíveis e o crescimento da população, podem ser apresentadas de uma forma extremamente eficaz pelo computador. Neste tipo de áreas é difícil imaginar um modo mais eficaz de transmitir informação do que o uso do computador.

Voltando ao tema dos mapas, é possível dar vida de muitas formas ao visor da televisão. Os mapas animados, por exemplo, podem apresentar problemas como o efeito da altitude sobre a distribuição das chuvas ou o modo como a população de uma determinada área se modificou durante a Revolução Industrial. É possível armazenar no Spectrum um certo número de imagens completas sob a forma de cadeias *(strings)* que depois serão apresentadas no visor no momento adequado.

Um dos usos mais vulgares do Spectrum, no campo da educação, consiste em apresentar questões a que o estudante pode responder de diferentes maneiras. Mais adiante apresentaremos o esquema de um programa deste tipo, capaz de atrair o interesse do aluno para o tema que lhe queremos apresentar. Veremos igualmente um programa mais simples, que não apresenta uma série de possibilidades de escolha e se limita apenas a esperar pela resposta correcta.

O facto de a maioria dos computadores em uso doméstico parecerem ser usados pelo menos na maior parte do tempo para jogar, mostra o imenso fascínio que os jogos de computadores podem provocar. Em vez de lutar contra isto, parece-nos melhor aproveitar este fascínio, tentando, sempre que possível, incorporar certos aspectos dos jogos (por exemplo, dar pontuações exageradas às respostas dadas pelos alunos) em programas "sérios".

Levando esta ideia ainda mais longe, incluímos um capítulo sobre "jogos para a inteligência" que, apesar de não ensinarem um tema concreto, podem ser bastante úteis no desenvolvimento da capacidade de uso dos números ou da facilidade de leitura. A partir do que será dito nesse capítulo, o leitor compreenderá o modo de modificar muitos programas de jogos a fim de incluir neles suficientes dados educativos para os poder usar na escola. No mínimo, a inclusão desse tipo de material pode ser bastante útil para manter o interesse pelo computador, actuando como uma espécie de descanso na aprendizagem. Tais programas podem também ser muito úteis para quebrar o "gelo" inicial das crianças que entram em contacto com o computador.

2

IDEIAS BÁSICAS: O QUE É UM COMPUTADOR? COMO FUNCIONAM OS COMPUTADORES?

Se bem que já não seja necessário saber como funcionam os computadores para poder utilizá-los, tal como não somos forçados a conhecer as complexidades do motor de combustão interna para conduzirmos um automóvel, é inevitável que mais tarde ou mais cedo lhe seja feita a pergunta "Como é que o computador pensa?", ou algo do género. E de qualquer modo é sempre útil ter uma ideia do que se passa dentro do computador.

O American National Standards Institute (ANSI) definiu o computador como um "dispositivo capaz de executar sequências sistemáticas de operações sobre dados, incluindo numerosos procedimentos aritméticos e lógicos, sem intervenção de um operador humano". Esta definição não esclarece facilmente o interessado. Consideremo-la um pouco melhor.

A primeira frase importante é "sequências sistemáticas de operações sobre dados". O Spectrum não se assemelha a um receptor de rádio. Não basta ligá-lo à corrente e esperar que faça coisas. Se ligar o seu Spectrum a uma televisão, e sintonizar esta correctamente, a máquina imprimirá de facto uma mensagem de identificação no visor. Mas não fará mais nada. A menos que *programemos* o computador para fazer mais alguma coisa, ela limitar-se-á a ficar parada à nossa frente...

Um computador necessita de um programa para poder funcionar. Este livro contém muitos programas, e nas revistas da especialidade é possível encontrar muitos mais. Pode-se igualmente comprar programas gravados em cassettes, ou em microdrives, que devem ser simplesmente carregados no Spectrum e entrarão imediatamente em execução.

Como o leitor aprenderá rapidamente na parte de programação deste livro, cada linha de instruções dadas ao computador começa por um número — o *número de linha*. O computador, a menos que

receba ordens diferentes, executa as instruções sequencialmente, passando, de uma linha à seguinte por ordem dos respectivos números. Por outras palavras, realiza as instruções segundo uma "sequência sistemática". Os "dados", na definição do ANSI, referem-se a qualquer informação — quer se trate de números, símbolos ou caracteres, ou de uma combinação destes — que o computador seja capaz de processar.

Terminamos assim a primeira parte da definição. Os "numerosos procedimentos aritméticos e lógicos" que o computador pode realizar são fáceis de compreender. O adjectivo "aritméticos" cobre todas as manipulações normais de números — como seja a soma, a subtracção, a potenciação, a determinação da raiz quadrada, etc. — enquanto que os "procedimentos lógicos" que o computador pode executar incluem comparações (de dimensão, de comprimento ou de uma qualquer característica, como seja a posição numa série) e o grupo de actividades determinado pelas leis da *álgebra booleana*(*) que permitem a tomada de decisões partindo de relações AND ("e") ou OR ("ou") — se uma afirmação é verdadeira AND (e) outra também, então fazer...; ou se A for verdadeiro OR (ou) B for verdadeiro OR C for falso, realizar a sequência D. As manipulações numéricas, e a tomada de decisões baseada em conclusões AND/OR, governam todas as actividades do computador.

A parte final da definição do ANSI, "sem intervenção de um operador humano", é crucial. Um computador não é o mesmo que uma máquina de calcular. Estas exigem uma série de respostas a perguntas, enquanto estão em funcionamento. Um computador pode ir buscar o material de que necessita a outros locais (por exemplo a uma cassette, a um disco, ou a qualquer ponto do seu programa).

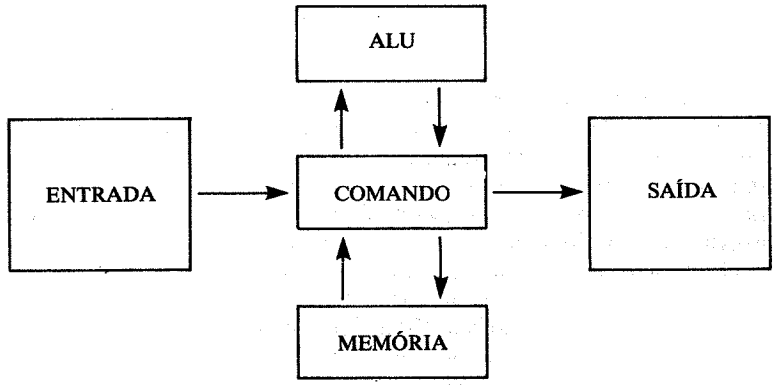
O Spectrum segue as instruções que lhe são dadas, realizando com rigor a tarefa de que foi incumbido. Se lhe fornecermos um programa para apresentação de questões de resposta múltipla, o Spectrum avaliará as diferentes respostas (desde que o programa tenha sido correctamente escrito). A menos que tenha sido programado para tal, o Spectrum não "se lembrará" de tocar uma música

(*) George Boole, um lógico e matemático inglês (1815-1864), foi o primeiro homem a formular as leis que permitem enquadrar as decisões lógicas no interior da estrutura matemática. A "álgebra booleana", que desenvolveu, permitiu a realização de *operações* (como a AND e a OR que já mencionámos, juntamente com as regras que governam as conclusões de tais comparações — *se* (IF) isto *então* (THEN) aquilo, de acordo com leis matemáticas estritas. Antes do desenvolvimento das ideias de Boole, considerara-se que a lógica era um ramo da filosofia. Através das suas obras *Mathematical Analysis of Logic* (1847) e *Investigation of the Laws of Thought* (1854), Boole mostrou claramente que a lógica era um ramo da matemática.

folclórica a meio da execução de um programa, nem encherá o visor de uma multidão de extraterrestres decididos a destruirem a Terra. O Spectrum executará as suas instruções metodicamente. Quaisquer erros aparentes que cometa serão de facto erros do próprio programador.

O leitor não deverá no entanto intimidar-se com este facto. Devido à natureza não ambígua da linguagem Basic do Spectrum, é extremamente fácil dizer à máquina exactamente o que deve fazer. Como compreenderá na parte dedicada à programação, a linguagem Basic usa palavras de instrução com um significado bastante claro (pelo menos para quem conhece o inglês...). A palavra PRINT (imprimir), por exemplo significa precisamente isso. Diz ao computador que deve imprimir qualquer coisa no visor da televisão. As outras palavras que o leitor encontrará significam exactamente o mesmo que quando utilizadas na conversação inglesa normal.

Em todos os computadores existem cinco partes básicas, que se pode observar na figura 1.



Todos os computadores devem dispor de um modo de obter informações do mundo exterior. É esta a tarefa da secção designada por "entrada" (INPUT), no lado esquerdo do diagrama. A entrada, no caso do Spectrum, é normalmente realizada pelo teclado, se bem que existam dispositivos que permitem a entrada de alguma informação na máquina através de um microfone. Além disso é possível ir buscar informação a um dispositivo de memória auxiliar, externa (o leitor de cassetes, ou um microdrive).

A informação que é comunicada à máquina através da secção IN-

PUT dirige-se em seguida para as três partes "pensantes" da máquina. Uma delas, a que chamamos CONTROLO, actua como uma espécie de polícia de trânsito e de cronómetro de todo o sistema, garantindo que as actividades do computador ocorram na sequência e no momento correctos. A ALU (Unidade aritmética e lógica) toma as decisões e realiza as somas, e a MEMÓRIA (que estudaremos um pouco mais em detalhe daqui a pouco) armazena não só os resultados intermédios de qualquer processamento que o computador esteja a realizar, como ainda toda a informação de que a máquina necessita para poder somar números ou tomar decisões.

Finalmente, depois de o computador ter chegado a uma conclusão, esta é enviada para o mundo exterior (isto é, para o utilizador) através da secção a que chamámos "saída" (OUTPUT). Esta saída é realizada para o visor de televisão, para a impressora, ou para o microdrive (ou para qualquer combinação destes dispositivos).

Se bem que o nosso diagrama sugira que todas as partes do computador são distintas umas das outras, não é esse o caso — como o leitor provavelmente compreenderá. No caso do Spectrum, existe um circuito integrado que combina a maior parte das actividades incluídas nos títulos CONTROLO, ALU e MEMÓRIA. No entanto, é conveniente pensar nas diferentes secções do computador separando-as em função das actividades que realizam.

Já dissemos que iríamos estudar a memória em maior detalhe. Vamos agora introduzir os primeiros exemplos "estarrecedores" de jargão informático. As palavras RAM e ROM encontram-se espalhadas por todos os livros e revistas de computadores. Apesar de estas palavras não serem utilizadas fora do mundo da informática, exprimem conceitos que não são difíceis de compreender.

Num computador existem dois tipos de memória (se bem que cada um deles possa assumir uma forma física diversa). O primeiro tipo é a ROM, o que significa "Memória apenas de leitura". Se bem que o Spectrum necessite de um programa para poder funcionar, contém já, no momento em que é adquirido, bastantes conhecimentos. No interior de um circuito integrado encontra-se toda a informação de que a máquina necessita para poder controlar o visor de televisão, compreender o que escrevemos no teclado, somar números, comparar quantidades diferentes, etc. Esta informação está gravada de uma forma definitiva nesse circuito integrado. Nestas condições, a memória que contém esta "inteligência" básica do Spectrum é chamada memória apenas de leitura (Read-only Memory), ou ROM.

Por outro lado, o computador necessita de uma área de memória menos permanente onde possa guardar o programa que nele introduzimos, assim como os resultados intermédios dos seus cálculos. Podemos aceder a qualquer posição desta memória aleatoriamente; devido a isto, esta memória não permanente é chamada Memória de acesso aleatório (Random Access Memory), ou RAM.

A única coisa que interessa saber para se poderem usar os termos ROM e RAM sem errar é que a ROM é uma memória fixa que diz ao Spectrum como deve realizar tarefas como a soma de dois números, enquanto a RAM é uma memória não fixa (as pessoas que trabalham em informática chamam-lhe muitas vezes "memória volátil") que contém o programa a executar e os resultados do processamento. Não podemos alterar a ROM, mas podemos alterar a RAM, e fazêmo-lo necessariamente.

3

PROGRAMAÇÃO DO SPECTRUM EM BASIC

Nesta secção do livro, vamos estudar a programação do Spectrum em BASIC. O material aqui apresentado foi concebido para uso do leitor, a fim de o ensinar a programar. Depois de ter aprendido (e ficará surpreendido ao constatar que se trata de uma tarefa bastante simples), poderá usar a informação contida aqui como base para ensinar os seus próprios alunos a programarem.

Verificará no entanto que terá de aprender um pouco mais do que se ensina aqui. Já foram escritos muitos livros sobre o modo de programar o Spectrum, e não podemos ter a pretensão — em apenas uma parte deste volume — de tratar o assunto com a mesma profundidade de todas essas obras.

O teclado

A primeira coisa que temos de aprender, infelizmente, é a utilizar o teclado... Digo infelizmente porque o teclado do Spectrum pode parecer bastante confuso àqueles que o utilizam pela primeira vez. No entanto, seria obviamente impossível programar sem dominar até certo ponto o teclado, e é portanto por aqui que devemos começar (a propósito, espero que o leitor tenha ligado o Spectrum antes de começar a ler, e que vá introduzir na máquina todas as instruções que iremos referir).

Depois de ligar o computador, seguindo as instruções do respectivo manual, e de ter ligado igualmente o receptor de televisão, verá a mensagem de "copyright" na parte inferior da imagem. Carregue agora na tecla ENTER. Aquela mensagem desaparecerá, surgindo em seu lugar um *cursor* "K" cintilando. Este cursor continuará presente enquanto estivermos a introduzir instruções, indicará os erros existentes na linha e, através da letra de que é formado (como K, neste caso), diz-nos ainda o *modo* em que o computador se encontra. É portanto bastante útil.

As três teclas mais importantes do computador são as designadas CAPS SHIFT, SYMBOL SHIFT e ENTER. Descubra-as no teclado que tem à frente. A tecla ENTER é usada depois de se escrever uma instrução, uma linha de programa ou uma resposta a uma pergunta que nos seja feita pela máquina. Diz ao Spectrum que deve executar aquilo que acabámos de escrever.

A CAPS SHIFT funciona como a tecla de maiúsculas de uma máquina de escrever. Normalmente o computador trabalha em minúsculas; no entanto, quando mantemos esta tecla premida, as letras são escritas em maiúsculas. Carregue nesta tecla e depois na tecla CAPS LOCK (a tecla da linha superior marcada com um 2); todas as letras escritas daí em diante sê-lo-ão em maiúsculas.

Se observar detidamente as teclas, verificará que a maior parte delas possuem uma palavra ou um símbolo (como ?,* ou +) escrito a vermelho. É possível escrever estes caracteres carregando simultaneamente na tecla SYMBOL SHIFT (legendada também a vermelho), no canto inferior direito do teclado.

Podemos "limpar" o computador em qualquer momento carregando na tecla A, que apresenta também a palavra NEW *(novo)* escrita a branco. NEW é uma ordem bastante drástica. Não existe qualquer modo de recuperar um programa depois de se ter carregado na tecla NEW, pelo que convém ter cuidado com o uso desta tecla. Carregue agora na tecla A, provocando o aparecimento da palavra NEW na janela inferior da imagem, e em seguida em ENTER. O visor tornar-se-á preto, aclarando depois a imagem e apresentando de novo a mensagem de copyright.

Agora que o computador está vazio, podemos iniciar a nossa exploração do teclado. Muitas das teclas possuem palavras escritas a branco. A estas palavras chama-se "palavras-chave". Uma das grandes qualidades do Spectrum é o seu sistema de escrever palavras de instrução carregando apenas numa das suas teclas. Em vez de escrever as palavras em causa letra a letra como acontece na grande maioria dos computadores, basta tocar por exemplo na tecla P (que possui a palavra PRINT escrita a branco) para ver a palavra-chave PRINT surgir no visor.

O leitor recorda ainda que o cursor é constituído por um K "cintilante". Sempre que carregamos numa tecla com uma palavra-chave escrita a branco, e o cursor é um K, a palavra-chave em causa surge automaticamente na parte inferior do visor.

No entanto, assim que a palavra é impressa, pode verificar que o cursor deixa de ser formado por um K e passa a sê-lo por um L. Esta inicial designa *letter*, letra. Carregue agora numa das teclas con-

tendo letras do alfabeto, e verá que o Spectrum imprime essa letra no visor. Quando se encontra neste modo "letra", o computador trabalha mais ou menos como uma máquina de escrever.

É chegado o momento de o leitor experimentar as teclas CAPS SHIFT e CAPS LOCK, a fim de mudar as letras escritas de minúsculas para maiúsculas. Verificará que ao fazê-lo o cursor L passará a C, indicando que carregou em CAPS LOCK (C indica Capitals, ou seja, maiúsculas).

O fundo do visor poderá já estar bastante povoado de caracteres escritos por si. Desligue a máquina, espere uns segundos, e volte a ligá-la.

Até agora estudámos os modos palavra-chave (K) e letra (L). O leitor já sabe como pode obter tudo aquilo que se encontra escrito *nas* teclas: palavras-chave, letras e os símbolos a vermelho (que, como recordará, se obtêm carregando na tecla SYMBOL SHIFT enquanto os escreve).

No entanto, existem mais coisas no teclado do que aquilo que vimos até agora. Existem por exemplo palavras a verde, acima das teclas, e a vermelho abaixo destas. Obtemos as palavras *acima* das teclas carregando simultaneamente nas teclas CAPS SHIFT e SYMBOL SHIFT, largando-as em seguida. Faça-o neste momento, e verificará que o cursor passou a ser representado por um E. Esta letra designa modo "extenso". Neste modo, o computador aceita as funções representadas acima das teclas. Carregue, por exemplo, na tecla K, e obterá RND (uma função que nos dá números aleatórios). O cursor voltará imediatamente a L depoik de se carregar numa tecla.

Por baixo das teclas existem também legendas a vermelho. Obtêm-se estas funções carregando nas teclas CAPS SHIFT e SYMBOL SHIFT ao mesmo tempo, ou seja, passando ao modo "extenso". No entanto, depois de o fazer, não largue a tecla da direita (SYMBOL SHIFT). Mantendo a pressão nesta tecla, carregue então em qualquer tecla e verificará que a palavra escrita abaixo dela surge escrita no visor. Carregando em Z, por exemplo, enquanto se está no modo "extenso" e se carrega simultaneamente em SYMBOL SHIFT, obtém-se a palavra BEEP.

Acabamos assim as generalidades sobre o uso do teclado, os seus modos e cursores. Mas existe ainda um outro cursor que devemos conhecer, o cursor G, correspondente ao modo Gráfico. Carregue na tecla CAPS SHIFT e na tecla 9 (que possui a palavra GRAPHICS escrita por cima), simultaneamente. O cursor passará a G. Em seguida carregue em qualquer das teclas existentes na fiada superior do teclado, e verá que o Spectrum escreve no visor o desenho

que se encontra representado na tecla em causa. Mantenha a tecla CAPS SHIFT premida e volte a carregar na mesma tecla, e obterá agora o símbolo gráfico inverso do anterior. Para abandonar o modo gráfico, deve carregar novamente em CAPS SHIFT e em 9.

Como dissémos no início, é pena que seja necessário estudar o teclado e os seus cursores antes de podermos prosseguir. No entanto, o leitor já conhece grande parte do que há para saber sobre o teclado. Basta-lhe agora notar a palavra DELETE (apagar) que está escrita por cima da tecla O (zero). DELETE serve para apagar as palavras escritas a partir do fim para o princípio. Pode usá-la carregando em CAPS SHIFT e na tecla O, simultaneamente. Proponho ao leitor que páre aqui por um momento, e reveja o que dissémos até agora. Depois de estar familiarizado com o teclado podemos continuar.

"Tijolos"

Todos os programas BASIC são formados por um certo número de palavras "standard", os "tijolos" de que nos servimos para a construção dos programas. Estes "tijolos" são palavras muito semelhantes às usadas em inglês, e às quais o computador atribui um sentido semelhante ao que têm nesta língua. Existem no entanto algumas palavras (como INKEY$, ATTR ou SCREEN$) que não se encontram nestas condições.

A palavra mais vulgar na maioria dos programas é a palavra PRINT. Esta palavra é usada, como o leitor certamente imagina, para escrever coisas no visor (PRINT = imprimir). Experimente fazê-lo. Limpe o computador desligando-o da rede durante alguns segundos, ou utilizando as funções DELETE e NEW.

Carregue na tecla P, a fim de obter a palavra PRINT, e depois escreva um número à frente, por exemplo:

```
PRINT 7
```

Carregue em seguida na tecla ENTER (não se esqueça de que deve carregar sempre nesta tecla para dar entrada a qualquer instrução que tenha escrito), e verá aparecer o número 7 na parte superior da imagem.

Para usar o Spectrum como uma máquina de calcular, escreva à frente de PRINT uma soma, por exemplo:

```
7+4−1
```

(daqui em diante parto do princípio de que o leitor não se esquece

de carregar em ENTER ao terminar uma instrução). O resultado da soma, 10, será imediatamente impresso no visor.

O leitor pode realizar os cálculos que deseje deste modo. No entanto, verificará que o teclado não possui um sinal de divisão; e, como pode também verificar facilmente, se utilizar X para realizar uma multiplicação a máquina não saberá o que fazer. De facto, para realizar multiplicações usa-se o símbolo asterisco (*), pelo que PRINT 7 * 3 dará 21; e o traço inclinado (/), na tecla V, para realizar divisões. PRINT 21/7 dará o resultado 3.

O outro símbolo que o leitor deve ainda conhecer é a seta da tecla H (que significa "potenciação"); 3↑2 é o modo de indicar ao Spectrum que deve calcular 3 ao quadrado.

Programa de teste

Começaremos por um simples programa que nos ensinará já bastante. Escreva as seguintes linhas de instruções no Spectrum, carregando em ENTER no final de cada uma delas. Se não tiver cometido erros ao escrever as instruções, o computador aceitará as linhas e escrevê-las-á no topo da imagem quando se carrega em ENTER. Se tiver cometido um erro, a máquina não aceitará a linha, e o cursor deslocar-se-á para o ponto desta onde o Spectrum julga que se encontra o erro, ajudando-o a descobrir o que escreveu mal.

De qualquer modo, escreva o que se segue, e carregue em seguida na tecla R (a fim de obter a palavra-chave RUN), que diz ao Spectrum que deve executar o programa:

```
10 REM programa de teste
20 PRINT "Escreva um numero"
30 INPUT a
40 PRINT a;" e escreva outro"
50 INPUT b
60 CLS
70 PRINT "Os seus numeros foram:
 ";a;" e ";b
80 PRINT
90 PRINT "E somam: ";a+b
```

Quando o leitor executar o programa, verá surgir no visor o seguinte:

```
Escreva um numero
123 e escreva outro
88

Os seus numeros foram 123 e 88

E somam 211
```


Vamos agora analisar este programa linha a linha. É extraordinário o que podemos aprender com ele. Em primeiro lugar, como pode observar, todas as linhas se iniciam por um número. Estes números de linha podem variar entre 1 e 9999. Em geral, o computador executa as linhas por ordem, a partir da que possui um número mais baixo (se bem que, como veremos mais adiante, esta regra não seja absoluta).

A primeira linha do programa começa pela palavra REM. Esta palavra significa REMARK (observação). Todas as linhas REM existentes num programa são ignoradas pelo computador; são incluídas apenas para dar informações às pessoas que lerem o programa. No início de um programa usa-se muitas vezes uma instrução REM, como acontece neste caso, para indicar qual a utilidade do programa. As REM's são também usadas no interior dos programas para indicar o que é feito na linha ou linhas seguintes. Estas declarações REM tornam muito mais fácil, quando se volta ao programa ao fim de algum tempo, descobrir o que o programa deve fazer e o modo como o faz.

Na linha 20, usa-se a instrução PRINT para escrever no visor as palavras "Escreva um número". O leitor verificará que estas palavras são encerradas dentro de aspas; é necessário fazê-lo para que o computador não confunda as letras com nomes de variáveis. Qualquer material inscrito dentro de aspas, como o que se encontra nesta linha de programa, é chamado *cadeia alfanumérica*.

A linha 30 permite-nos dar entrada ao computador de um número à nossa escolha. INPUT leva o computador a esperar por uma acção do utilizador, neste caso indicar um número. A letra "a" a seguir à palavra INPUT indica uma *variável numérica*. Uma variável numérica é uma letra (ou qualquer combinação de letras e números, desde que começada por uma letra) à qual o computador atribui um valor numérico. Daí em diante, o computador trata a variável numérica como se fosse um número. O leitor poderá verificar isto, mais adiante neste mesmo programa, onde o computador utiliza "a" (e "b", cujo valor é indicado na linha 50) nas linhas 70 e 90 para imprimir primeiro os números que o utilizador lhe fornece e depois a soma destes.

A palavra CLS na linha 60 significa "clear screen" (limpar o visor). O leitor terá provavelmente notado que o visor é sempre limpo depois de darmos entrada ao segundo número, antes de imprimir as duas frases finais.

Bastou este breve programa para nos ensinar já bastantes coisas. Note ainda (e isto é importante) que deve usar um ponto e vírgula para ligar a letra "a" na linha 40 ao resto do material a imprimir.

Adivinhar números

Elimine o programa que se encontra no computador usando NEW, e escreva depois o programa seguinte,

Neste novo programa existe um certo número de elementos ainda desconhecidos para o leitor que podem ser difíceis de escrever; sugerimos portanto que, antes de tentar fazê-lo, o leitor leia as notas que mais adiante se apresentam referentes a cada linha.

```
  10 REM numero do Spectrum
  20 LET a=INT (RND*50)+1
  30 LET adivinha=0
  40 INPUT "Como te chamas? ";b$
  50 PRINT "Ola', ";b$
  60 PRINT '"Estou a pensar num
numero"
  70 PRINT "entre 1 e 50 que tal
vez"
  80 PRINT "nao consigas adivinh
ar"
  90 LET adivinha=adivinha+1
 100 PRINT '''"Tentativa n. ";ad
ivinha
 110 INPUT "Escreve um numero: "
;c
 112 CLS
 115 PRINT "O teu numero, ";b$;"
,","foi ";c: PRINT
 120 IF c=a THEN PRINT "Consegui
ste acertar!": PRINT "Precisaste
 de ";adivinha;" adivinhas": STO
P
 130 IF c>a THEN PRINT "O teu nu
mero foi muito alto"
 140 IF c<a THEN PRINT "O teu nu
mero foi muito baixo"
 150 GO TO 90
```

Vejamos a aparência deste programa quando é executado:

```
Como te chamas? Tim
Ola', Tim

Estou a pensar num numero
entre 1 e 50 que talvez
nao consigas adivinhar

Tentativa n. 1
Escreve um numero: 38
O teu numero, Tim,
foi 38

O teu numero foi muito baixo


Tentativa n. 2
Escreve um numero: 50
O teu numero, Tim,
foi 50

O teu numero foi muito alto


Tentativa n. 3
Escreve um numero: 43
O teu numero, Tim,
foi 43

O teu numero foi muito alto


Tentativa n. 4
Escreve um numero: 40
O teu numero, Tim,
foi 40

Conseguiste acertar!
Precisaste de 4 adivinhas
```

Linha 10. Esta declaração REM é semelhante à que se encontra na primeira linha do programa anterior.

Linha 20. Esta linha utiliza a palavra LET (a palavra-chave que se encontra na tecla L) para atribuir um valor à variável "a". O leitor verificará que depois do sinal de igual se encontra a palavra INT (Inteiro, que reduz um número com uma parte fraccionária ao menor número inteiro *inferior*). INT é obtido na tecla R, carregando simultaneamente em ambas as teclas SHIFT e libertando-as em seguida, de modo a obter o modo extenso. Os parêntesis encontram-se nas teclas 8 e 9 respectivamente, e obtêm-se carregando na tecla SYMBOL SHIFT e na 8 ou 9. O asterisco (sinal de multiplicação) encontra-se na tecla B, sendo novamente necessário usar a tecla SYMBOL SHIFT para o obter. Esta linha utiliza a função do Spec-

trum que produz números aleatórios para produzir um número inteiro entre um e 50. Use uma declaração do mesmo tipo desta sempre que necessitar de um número ao acaso. O número que se encontra no interior dos parêntesis é o maior na gama que se deseja obter.

Linha 30. Esta linha dá a uma variável numérica chamada "adivinha" um valor inicial igual a zero. Como se recorda, uma variável numérica pode ter um nome constituído por qualquer combinação de letras e números, desde que comece por uma letra.

Linha 40. Esta declaração INPUT é um pouco diferente da primeira que estudámos. Como o leitor pode ver, existe uma frase no interior de aspas imediatamente a seguir a INPUT. Quando o programa é executado, a frase em causa é impressa na linha de entrada, no canto inferior esquerdo do visor. Este método é muito prático, pois permite ao programador indicar exactamente a informação que deseja que o utilizador dê à máquina. No fim desta linha encontra-se um nome de variável seguido de um sinal de cifrão ("b$"). Isto representa uma *variável de cadeia* ou *variável alfanumérica*, diferente das variáveis numéricas que temos usado até agora. Uma variável alfanumérica tem um nome constituído por uma qualquer letra do alfabeto seguida do cifrão, e pode conter qualquer caracter. Quando escrevemos o nosso nome em resposta ao pedido do computador, esse nome é guardado na variável alfanumérica b$.

Linha 50. Trata-se de uma linha de PRINT bastante vulgar, que imprime "Olá" e depois o nosso nome.

Linha 60. Outra instrução PRINT normal.

Linha 70. O mesmo acontece aqui...

Linha 80. ... e aqui.

Linha 90. O valor da variável numérica "adivinha" é aqui aumentado de um. É conveniente utilizar nomes explícitos para as variáveis numéricas, como por exemplo "adivinha". Se bem que seja muito fácil saber exactamente para que serve cada variável quando o programa é curto, o leitor verificará que tudo se torna extremamente confuso quando os programas aumentam de tamanho. Nesses casos, verificará que o emprego de nomes explícitos para as variáveis facilita bastante o trabalho. Num jogo de xadrez, por exemplo, P pode indicar peão, B o bispo, BP o bispo preto e BB o bispo branco. O uso destes nomes de variáveis pode tornar-se muito útil quando o programa é escrito, e facilita aos outros a compreensão do programa.

Linha 100. Esta linha assemelha-se a uma normal instrução PRINT, excepto no que se refere à presença de apóstrofos (''') no início dela. O apóstrofo encontra-se na tecla 7. Quando o Spectrum encontra um apóstrofo no início de uma instrução PRINT como esta, muda uma vez de linha no visor, deixando a linha de intervalo em branco. Nestas condições, os três apóstrofos aqui apresentados obrigam a máquina a deixar três linhas em branco, escrevendo a nova frase na quarta linha depois da última frase escrita, como o leitor poderá verificar executando o programa.

Linha 110. Esta linha utiliza um novo pedido de INPUT, terminando desta vez na variável "c", à qual é atribuído o valor numérico que indicamos ao computador em resposta à adivinha.

Linha 112. A instrução CLS limpa o visor depois de carregarmos em ENTER para dar entrada à nossa adivinha.

Linha 115. Esta linha serve apenas para recordar ao utilizador o número que indicou à máquina. Note que utiliza o nosso nome ("b$") e o número indicado ("c"), ligando ambas estas variáveis a material que se encontra na linha escrito entre aspas.

Linhas 120, 130 e 140. Vamos estudar estas três linhas em conjunto porque realizam funções semelhantes (e muito importantes). Note que todas se iniciam pela palavra IF *(Se)*. Quando o computador encontra uma declaração IF, observa a condição que se segue a essa palavra e se for verdadeira então (THEN) executa a instrução que se encontra a seguir à palavra THEN. IF e THEN encontram-se sempre juntas. IF o tempo está frio THEN ligue o aquecimento, IF a televisão está avariada THEN chame um técnico, etc. A linha 120 compara o nosso número ("c") com o produzido pelo computador no início do programa ("a"), e IF ambos forem iguais THEN passa a executar o resto da linha. O leitor verificará facilmente que a linha 120 contém três instruções completas. Podemos portanto introduzir mais do que uma instrução em cada linha de programa, desde que as separemos por dois pontos. A primeira parte da linha 120, depois de verificar a igualdade dos números, imprime uma mensagem de parabéns, e a segunda instrução PRINT diz-nos quantas foram as tentativas de que necessitámos. Finalmente, o programa pára ao encontrar a instrução STOP. Note que no caso de a condição verificada ser falsa, então nenhuma das instruções que se seguem a THEN nesta linha será executada. O computador passará imediatamente à linha seguinte. Nas linhas 130 e 140 o computador verifica qual dos números é maior e imprime as mensagens apropriadas. Note que a instrução THEN se encontra na tecla G.

Linha 150. O leitor recorda certamente o que já dissémos sobre o facto de um computador seguir normalmente os programas linha a linha. Duas das ordens que podem alterar isto são a GO TO e a GO SUB. Estudaremos a GO SUB um pouco mais adiante. A GO TO *(ir para)* pode ser encontrada na tecla G, e dirige neste caso o computador para a linha 90, onde recomeça o ciclo principal da execu-

ção do programa.

Introduza portanto o programa na máquina e execute-o diversas vezes; depois volte ao livro e leia novamente as notas anteriores. Procure compreender bem tudo o que dizemos antes de continuar.

Compreendemos que o material deste capítulo pode parecer aborrecido, e um tanto pesado. De facto, estamos a estudar — e em apenas algumas páginas — tópicos que noutros livros são estudados em vários capítulos. No entanto, se o leitor progredir lentamente, acabará por compreender tudo.

O programa que acabámos de apresentar pode ser muito bom para "quebrar o gelo" de estudantes que nunca utilizaram computadores anteriormente. Usa o nome do aluno (o que impressiona sempre as pessoas que usam um computador pela primeira vez) e é suficientemente fácil para garantir a "vitória" do jovem em pouco tempo, assegurando assim um primeiro contacto agradável com o computador.

Grave o programa numa cassete quando tiver a certeza de que sabe como trabalha o modo de utilizar as instruções SAVE — gravar em cassette — e VERIFY — verificar se ficou bem gravado — (este aspecto é descrito pormenorizadamente no manual do Spectrum). Voltaremos a este programa mais tarde, quando estivermos a estudar as cores, pelo que convirá tê-lo numa cassette a fim de não ter de o escrever novamente. Sugiro ao leitor que faça mais de uma gravação de todos os programas que lhe interessarem, por exemplo em cassetes diferentes, a fim de não os perder no caso de uma delas se corromper por alguma razão; e apesar de compreender que pode poupar espaço gravando programas em cassetes de 60 ou 90 minutos, as frustrações que sente ao procurar um programa no meio de uma fita com este programa são suficientes para aconselhar um outro processo: uma das gravações, a de "uso", deve ser feita em cassettes mais pequenas que conterão apenas um ou dois programas de cada lado. Se houver problemas com esta gravação "de uso" pode sempre recorrer à outra cópia do programa, numa cassete de suporte (de "back up").

Lançamento de dados

O nosso próximo programa ensinar-nos-á também muitas coisas. Neste programa, o utilizador e a máquina lançam alternadamente um par de dados. Aquele que obtiver mais ganha. A propósito, pensamos que a execução e a escrita de jogos constitui o modo mais simples e agradável de aprender programação. É por isso que todos os programas desta primeira parte são jogos.

Neste programa, estudaremos os seguintes acrescentos importantes ao nosso vocabulário BASIC:

— PAUSE (Pausa)
— BRIGHT (Brilho)
— O uso de subrotinas (através das instruções GO SUB — ir para subrotina — e RETURN — sair da subrotina).
— Ciclos FOR/NEXT
— BEEP

Vejamos os resultados de uma execução do programa:

```
Carrega numa tecla para lancar
os dados

Primeiro dado: 1

Segundo dado: 1


A tua pontuacao e' 2

Prepara-te! Vou rolar os dados!

Ca' vai...

Primeiro dado: 3

Segundo dado: 5

A minha pontuacao e' 8
e eu ganhei-te!
```

Introduza o programa na máquina, execute-o durante algum tempo, e depois volte ao livro e estude-o linha a linha.

```
  10 REM Lancamento de dados
  20 PRINT "Carrega numa tecla p
ara lancar  os dados"
  30 PAUSE 0
  40 GO SUB 1000
  70 LET pessoa=a+b
  80 PRINT "A tua pontuacao e'
";pessoa
  90 PAUSE 50
```

```
100 PRINT BRIGHT 1;'"Prepara-te
! Vou rolar os dados!"
 110 PAUSE 50
 120 PRINT '"Ca' vai..."
 130 PAUSE 50: GO SUB 1000
 140 LET computador=a+b
 150 PRINT "A minha pontuacao e'
";computador
 160 IF computador=pessoa THEN P
RINT "e estamos empatados!"
 170 IF computador>pessoa THEN P
RINT "e eu ganhei-te!"
 180 IF computador<pessoa THEN P
RINT "Ganhaste!"
 190 PAUSE 300
 200 RUN
1000 REM Isto e' uma subrotina
1010 LET a=INT (RND*6)+1
1020 LET b=INT (RND*6)+1
1030 FOR g=1 TO 30
1032 BEEP .03,g: BEEP .03,50-g
1036 NEXT g
1040 PRINT '"Primeiro dado: ";a
1045 PAUSE 50
1050 PRINT '"Segundo dado: ";b
1055 PAUSE 50
1060 PRINT
1070 RETURN
```

Linha 10. Encontramos aqui a declaração REM que identifica o programa.

Linha 20. Trata-se de uma instrução PRINT normal, como as que encontrámos até agora.

Linha 30. A ordem PAUSE (na tecla M) faz uma pausa na execução do programa durante um período de tempo que é definido pelo número que se segue a PAUSE. Se este número for 50 (PAUSE 50), o computador ficará "parado" durante um segundo (porque a unidade empregue nesta instrução é "ciclos por segundo", e a rede de alimentação eléctrica funciona a 50 ciclos por segundo). Se carregarmos em qualquer tecla a pausa fica sem efeito. A instrução PAUSE 0 (zero) fará uma pausa infinita, que só será terminada quando tocarmos numa tecla qualquer; daí a instrução na linha 20 para "Carregue numa tecla..."

Linha 40. — Quando o computador encontra uma instrução GO SUB, passa a execução para a linha cujo número se segue às palavras GO SUB (tal como acontece com a instrução GO TO). Até agora, portanto, ambas estas instruções são iguais. No entanto, há

uma diferença na medida em que quando o computador encontra a palavra RETURN ao executar uma subrotina, volta à linha de onde essa subrotina foi chamada (isto é, à linha onde se encontrava a instrução GO SUB, mas depois desta). Neste programa, a subrotina inicia-se na linha 1000. O Spectrum salta portanto para a linha 1000, seguindo depois as linhas uma a uma a partir desse ponto. Quando atinge a linha 1070 encontra a palavra RETURN já mencionada. Volta então para a instrução ou a linha que se seguem à instrução que invocou a subrotina (GO SUB). Neste caso volta portanto para a linha 70 (a primeira linha que se encontra depois da linha 40, que enviou a execução para a subrotina). As subrotinas são usadas quando existe uma parte do programa que deve ser executada várias vezes e chamada de diferentes pontos do programa. Em vez de reescrever toda essa parte várias vezes, podemos escrevê-la uma só vez e usá-la como subrotina. Neste programa o leitor pode verificar que os dados são lançados duas vezes, uma pelo computador e outra por si. É a mesma parte do programa que de facto "lança" os dados, sendo porém chamada de dois pontos diferentes.

Linha 70. Linha que se segue à chamada de subrotina; é portanto a ela que o computador volta depois de a subrotina em causa ser executada. No interior da subrotina (ver as linhas 1010 e 1020) são escolhidos números aleatórios entre 1 e 6 que depois são atribuídos às variáveis "a" e "b". Ao voltar da subrotina, estes números são somados e o resultado é atribuído a uma variável chamada "pessoa".

Linha 80. Esta linha imprime a pontuação.

Linha 90. A ordem PAUSE leva o computador a esperar um segundo.

Linha 100. É impressa a mensagem usando cores brilhantes. Encontrará a declaração BRIGHT na linha inferior de teclas, por baixo da tecla B. Existem várias maneiras de modificar as saídas impressas no visor, incluindo o uso de INVERSE (que inverte as cores, imprimindo letras brancas sobre um fundo negro) e de FLASH (que "cintila", isto é, que alternadamente muda as cores de fundo e da letra de preto para branco e vice-versa). Neste caso, como o leitor verificará quando executar o programa, BRIGHT torna mais claro o fundo sobre o qual são escritas as palavras. Se à frente de BRIGHT se escrever o número 1 (BRIGHT 1), tal significará que o Spectrum deve "ligar" a função BRIGHT, ou seja, escrever com brilho. Se for seguida de 0 (BRIGHT 0), "desliga" a função. Usa-se o mesmo método para os outros modos de tratamento da impressão, ou seja, para as funções FLASH, INVERSE ou OVER (que escreve numa dada posição sem apagar o que já lá estiver escri-

to). Todas estas palavras-chave, como se compreenderá no programa seguinte, podem ser usadas "globalmente" ou "localmente". Se a instrução BRIGHT aparecer sozinha numa linha de programa, altera tudo o que for impresso daí em diante. Diz-se neste caso que actua globalmente. Se no entanto, como acontece aqui, a mesma instrução surgir no interior de uma declaração PRINT, afecta apenas o que for impresso por essa instrução, sendo portanto usada localmente.

Linha 110. Outra pausa de um segundo. O uso de PAUSE deste modo constitui um modo muito eficaz de dar o ritmo desejado a um programa, em particular quando se pretende dar ao utilizador a ideia de que o computador está a "pensar", tentando resolver um problema. Muitas vezes usa-se PAUSE para dar tempo ao utilizador de ler o que é escrito no visor antes de a imagem ser limpa.

Linha 120. A mensagem "Cá vai..." é impressa deixando uma linha de intervalo devido ao emprego do apóstrofo.

Linha 130. Esta linha contém duas instruções, sendo ambas separadas, como o leitor pode verificar, por dois pontos. Depois da segunda pausa, o programa envia a máquina novamente para a subrotina.

Linha 140. Depois de executar a subrotina, o computador, volta a esta linha, a que se segue à instrução que invocou a subrotina. Aqui o resultado da soma de "a" e "b" é atribuída à variável "computador".

Linha 150. O resultado do lançamento de dados efectuado pelo computador é impresso no visor.

Linhas 160, 170 e 180. Estas três linhas IF/THEN comparam os valores das duas variáveis "pessoa" e "computador", determinando qual é maior e portanto quem ganhou o lançamento de dados.

Linha 190. O programa faz uma pausa de seis segundos.

Linha 200. O programa é executado novamente. Note que pode utilizar RUN, a palavra que faz executar um programa, do mesmo modo que qualquer outra palavra-chave no interior de um programa.

Linha 1000. Esta declaração REM introduz a subrotina.

Linha 1010. A variável "a" recebe um valor escolhido aleatoriamente entre 1 e 6.

Linha 1020. O mesmo se passa com a variável "b".

Linhas 1030, 1032, 1036. Estas três linhas formam um ciclo FOR/NEXT. Tal como as palavras GO SUB e RETURN têm de surgir sempre juntas (se bem que não necessariamente na mesma linha), o mesmo se passa com FOR e NEXT. Este ciclo é usado para repetir um número definido de vezes a parte do programa que se encontra entre as declarações FOR e NEXT. A quantidade de repetições é definida pela *diferença* entre os dois números indicados na declaração FOR. Na linha 1030, que inicia o ciclo, lemos:

FOR g = 1 TO 30

Isto significa "Desde (FOR) g = 1 Até (TO) 30". Garante-se assim que o ciclo será executado trinta vezes. Na linha 1036 (NEXT G, *seguinte* G) encontra-se o final do ciclo, indicado pela palavra NEXT que obriga a executar o que ficou para trás para o valor "seguinte" de G. Note que a letra usada no ciclo (a chamada "variável de comando de ciclo", "g" neste caso) deve ser igual em ambas as instruções FOR e NEXT. Quando o computador executa este ciclo FOR/NEXT, a variável de comando ("g") possui inicialmente o valor 1, depois o valor 2 na segunda vez que o ciclo é executado, etc., até 30. Quando atinge o valor 31 (por acção da instrução NEXT) passa para o resto do programa, dado que o ciclo só é repetido para valores de g inferiores a esse.

Neste caso, g é usada para definir o argumento da instrução BEEP (que faz soar o altifalante do Spectrum) que se encontra na linha 1032. Como de cada vez que o ciclo é executado g tem um valor diferente, o som produzido é também diferente, produzindo tons sucessivamente mais agudos e mais graves. A palavra-chave BEEP, que se encontra por baixo da tecla Z, é sempre seguida de dois números. O primeiro controla a duração da nota (sendo 0,007 o menor valor praticável), e o segundo a altura da nota (que deve encontrar-se na gama $^{-}60$ a $+60$). O tom varia porque "g", a variável usada na definição da altura da nota, se altera à medida que o ciclo é executado.

Linha 1040. Depois dos sons é impresso o resultado do primeiro lançamento do dado.

Linha 1045. Pausa de um segundo.

Linha 1050. É impresso o resultado do segundo lançamento do dado.

Linha 1055. Nova pausa de um segundo.

Linha 1060. O computador imprime uma linha em branco.

Linha 1070. Instrução de retorno, que termina a subrotina e reenvia o computador para a linha que se segue àquela onde a subrotina foi invocada.

Instruções adicionais

O nosso último programa desta secção baseia-se no segundo pro-

grama que apresentámos. Se o tiver gravado em cassette, pode carregá-lo (LOAD) no computador, e fazer em seguida as adaptações necessárias.

Neste programa iremos examinar as seguintes instruções novas:

— FLASH
— INK
— PAPER
— BORDER
— INVERSE
— OVER

Modifique o programa igualando-o à listagem que se segue, e volte em seguida ao livro para uma discussão destas novas instruções.

```
  10 REM numero do Spectrum
  20 LET a=INT (RND*50)+1
  30 LET adivinha=0
  40 INPUT FLASH 1; INK 2;"Como
te chamas? ";b$
  45 INK 2: PAPER 6: BORDER 6: C
LS
  50 PRINT INK 1; PAPER 7;"Ola";
AT 0,2; OVER 1;" , ";b$
  60 PRINT '"Estou a pensar num
nu"; OVER 1;AT 2,20;"'mero"
  70 PRINT "entre 1 e 50 que tal
vez"
  80 PRINT "na"; OVER 1;AT 4,1;"
"o consigas adivinhar"
  90 LET adivinha=adivinha+1
 100 PRINT '''"Tentativa n. ";ad
ivinha
 110 INPUT "Escreve um numero: "
;c
 112 CLS
 115 PRINT "O teu nu"; OVER 1;AT
 0,7;"'mero, ";b$;",","foi ";c:
PRINT
 120 IF c=a THEN PRINT FLASH 1;"
Conseguiste acertar!": PRINT FLA
SH 1; INVERSE 1;"Precisaste de "
;adivinha;" adivinhas": STOP
 130 IF c>a THEN PRINT "O teu nu
"; OVER 1;AT 3,7;"'mero foi muit
o alto"
 140 IF c<a THEN PRINT "O teu nu
"; OVER 1;AT 3,7;"'mero foi muit
```



```
o baixo"
 150 GO TO 90

O teu numero, Tim,
foi 27

Conseguiste acertar!
Precisaste de 27 adivinhas
```

Como pode verificar, o programa tornou-se muito mais interessante do que no início. É extraordinário como a cor pode alterar tanto a apresentação de um programa. Vale a pena ter este facto em conta ao criar os seus próprios programas.

Como certamente já compreendeu, FLASH e INVERSE funcionam de um modo semelhante a BRIGHT: seguidas do número 1 "ligam"; seguidas do número 0 "desligam". O mesmo se passa com OVER.

Já referimos o facto de estas palavras poderem ser usadas localmente ou globalmente. O mesmo é verdadeiro para as palavras-chave INK (que define a cor dos caracteres), PAPER (que determina a cor de fundo) e BORDER (que indica a cor da margem). Na linha 40 a INK é usada localmente, afectando portanto apenas a instrução INPUT. Na linha seguinte, 45 a cor de INK é passada para vermelho (cor número 2), o papel para amarelo (cor número 6) e a margem também para amarelo. A instrução CLS no final da linha é necessária para provocar imediatamente a mudança da cor de fundo em todo o visor; sem ela (como verificará se não a empregar) o computador limitar-se-á a utilizar um fundo amarelo sob tudo aquilo que escrever daí em diante.

A instrução OVER é aqui usada localmente para imprimir os acentos em palavras como "Olá". Na linha 50, depois de ser impresso "Olá", puxa-se a posição de impressão um caracter atrás e imprime-se o resto começando pelo acento. A declaração AT, como a TAB, são discutidas mais abaixo.

Vamos observar brevemente algumas das outras palavras-chave que utilizaremos na programação do seu Spectrum em BASIC. Não as estudaremos em tanto detalhe como as anteriores. No entanto, partindo daquilo que já sabe e do que aprenderá nos outros programas que apresentaremos neste livro (deduzindo o modo como as instruções actuam a partir do contexto onde são usadas e dos resultados que produzem) não lhe será difícil educar-se a si próprio neste campo.

EDIT — Esta função serve para alterar rapidamente as linhas de um programa. Coloque o cursor das linhas (>) na linha que pretende modificar, e premindo em seguida na tecla CAPS SHIFT carregue na tecla 1; a linha em causa passará imediatamente para a parte inferior da imagem. Pode depois deslocar o cursor ao longo da linha utilizando CAPS SHIFT e as teclas com setas (5 e 8) para chegar ao ponto desejado, onde poderá acrescentar novo material ou remover o existente (usando CAPS SHIFT e a tecla 0, DELETE). O cursor das linhas é deslocado de uma linha para outra usando CAPS SHIFT e as teclas 6 e 7. Todas as funções escritas a branco acima das teclas da fiada superior são portanto obtidas usando CAPS SHIFT e a tecla correspondente.

STEP — Esta declaração é usada num ciclo FOR/NEXT quando não se pretende que o ciclo seja efectuado para valores seguidos da variável de comando. O passo (STEP) pode ser um número inteiro positivo (ou uma fracção) ou negativo no caso de se querer que a variável de comando diminua progressivamente de valor. O formato da instrução é:

```
FOR a=b TO c STEP d
```

LIST — Esta ordem provoca a impressão no visor da listagem de um programa, começando pela linha de número menor. Podemos escrever igualmente LIST n, caso em que a listagem se iniciará pela linha de número n.

TAB — Esta ordem desloca o ponto de impressão do início de uma linha para a coluna que se deseje. É usada no formato

```
PRINT TAB n;"cadeia"
```

PRINT AT — Especifica a coluna do visor onde se pretende imprimir (PRINT AT = Imprimir em). O seu formato é

```
PRINT AT x,y;"cadeia"
```

onde x é o número da linha e y o número da coluna onde o computador deve imprimir a cadeia (pelo que PRINT AT 0,0 corresponde ao canto superior esquerdo do visor).

PLOT — Esta ordem marca um ponto numa posição do visor de alta resolução. O seu formato é

```
PLOT x,y
```


PLOT 0,0 marca um ponto no canto inferior esquerdo do visor, dado que neste caso as fiadas de pontos são contadas verticalmente a partir de baixo.

INKEY$ — Lê o teclado de modo a receber entradas de um só caracter, e não obriga ao uso da tecla ENTER depois de aquele ter sido escrito. É usada normalmente do seguinte modo:

```
IF INKEY$="n" THEN...
```

ou

```
LET A$=INKEY$: IF A$="n" THEN...
```

DEF FN — Instrução usada para definir funções.

SCREEN$ — Indica o início de uma determinada posição de caracter; os seus parâmetros são semelhantes aos usados em PRINT AT.

VAL — Esta função converte o conteúdo de uma cadeia no seu equivalente numérico, pelo que VAL a$, onde a$ é igual a "12+3*4", será igual a 24.

STR$ — Esta função é o oposto de VAL, transformando um número numa cadeia alfanumérica.

LLIST, LPRINT, COPY — Estas ordens comandam a impressora, funcionando LLIST e LPRINT do mesmo modo que LIST e PRINT no caso do visor; COPY é usada para copiar o conteúdo de uma imagem do visor para a impressora.

4

TRATAMENTO DE CADEIAS E USO DESTAS EM MATEMÁTICA

Como o leitor provavelmente sabe, a versão da linguagem BASIC usada no Spectrum não é exactamente igual às versões da mesma linguagem utilizadas noutros computadores. Os diferentes "dialectos" da BASIC são suficientemente semelhantes para garantir que quando se sabe programar numa máquina se pode programar qualquer outra de um modo razoavelmente fácil. E ao fim de uma hora ou pouco mais de estudo do manual de um computador, determinando as diferenças que existem entre uma qualquer versão da BASIC que nos é estranha e aquela que conhecemos, conseguimos programar na nova máquina sem grandes problemas.

Muitos aspectos da Basic do Spectrum (por vezes designada "Basic Sinclair") são perfeitamente normalizados (apesar do facto de, paradoxalmente, não existir de facto uma versão da Basic que possa ser considerada "normalizada"). Os códigos utilizados para os caracteres numéricos e alfabéticos, assim como vários outros, correspondem aos códigos normais ASCII (American Standard Codes for Information Interchange) — ao contrário do ZX80 e do ZX81, que utilizam um sistema Sinclair original. A maioria das técnicas de programação que aprender no Spectrum podem ser facilmente transferidas para outros computadores.

No entanto, há uma parte do dialecto Basic Sinclair que é único — o relativo ao tratamento de cadeias alfanuméricas. Recorda certamente do capítulo anterior que qualquer material inscrito entre aspas — por exemplo "Olá" — é designado por *cadeia* alfanumérica em linguagem de informática. Ora, como já dissémos, o tratamento das cadeias no Spectrum é único. Em vez de utilizar instruções excêntricas de outros dialectos Basic como LEFT$, MID$ e RIGHT$, o Spectrum reduz praticamente todo o tratamento de cadeias ao uso de parêntesis e da palavra-chave TO (até). Depois de compreender o método (o que será bastante rápido) poderá fazer

praticamente tudo o que quiser com as cadeias alfanuméricas.

O leitor sabe já que uma variável de cadeia é uma letra (A a Z) maiúscula ou minúscula com um cifrão como sufixo (por exemplo A$ ou a$). As palavras associadas ao tratamento de cadeias incluem LEN (por exemplo em LEN A$), que indica o comprimento *(length)* da cadeia, ou seja, o número de caracteres, símbolos e espaços que contém; CODE, que indica o código do primeiro elemento da cadeia (PRINT CODE A$, quando A$="A", dará 65, tal como PRINT CODE X"A"); e TO, que discutiremos daqui a pouco.

Iremos ainda estudar outras palavras como VAL e STR$, mas introduzi-las imediatamente aumentaria apenas a confusão do leitor.

CHR$

A palavra-chave CHR$ (CHR provém de *character*, caracter) é usada para alterar o código (CODE, já mencionado) no caracter correspondente. Por exemplo, se escrever (e pode fazê-lo imediatamente no seu Spectrum) PRINT CHR$ 65, o computador imprimirá a letra A. Se depois escrever PRINT CODE "A", obterá, como já foi dito, 65. É portanto este o número que representa dentro do computador a letra *a* maiúscula.

Introduza agora o programa que se segue no computador, e execute-o algumas vezes até ter a certeza de entender o modo como actuam as duas funções CHR$ e CODE:

```
  10 REM uso de CHR$ e CODE
  20 INPUT "ESCREVA UM NUMERO EN
TRE 1 E 255 ";A
  30 PRINT '"CHR$ ";A;" = ";CHR$
 A
  40 INPUT "AGORA ESCREVA UMA CA
DEIA ";A$
  50 PRINT '"CODE ""A$;"" = ";C
ODE A$
  60 GO TO 20
```

Vejamos um curto exemplo de execução:

```
CHR$ 76 = L

CODE ";A$;" = 116

CHR$ 107 = k
```

```
CODE ";A$;" = 72
CHR$ 99 = c
CODE ";A$;" = 84
CHR$ 254 = RETURN
CODE ";A$;" = 116
CHR$ 34 = "
CODE ";A$;" = 102
```

A utilidade de "TO"

O mais valioso instrumento de manipulação de cadeias de que dispomos no Spectrum é o "corte". Recorrendo a ele, podemos extrair de uma cadeia a porção que desejarmos, somar diferentes partes da mesma cadeia por qualquer ordem pretendida, ou acrescentar pedaços de cadeias diferentes.

Estude o seguinte, e veja se consegue descobrir como actua a declaração TO:

LET A$ = "ABCDEF"

(LET é uma função que atribui à variável indicada a seguir o valor que se encontra depois do sinal de igual).

Depois verifique que:

PRINT A$(1) produz A
PRINT A$(1 TO 3) produz ABC
PRINT A$(2 TO 4) produz BCD
PRINT A$(4) produz D

Por outras palavras, um único número no interior dos parêntesis indicará o elemento da cadeia cuja ordem corresponde a esse número, como pode observar no exemplo anterior no caso de A$(1) e de A$(5). Encontramos depois duas variantes: enquanto que

PRINT A$(1 TO 5)

produz ABCDE, como seria de esperar, se quisermos utilizar a cadeia inteira não necessitamos de especificar o último número (6 neste caso). Isto é, PRINT A$(1 TO) produzirá exactamente o mesmo resultado que PRINT A$(1 TO 6). Mais ainda, se quisermos toda a cadeia desde o início até um dado ponto, basta-nos mencionar este último ponto. Portanto, PRINT A$(TO 3) imprimirá ABC, tal como PRINT A$(1 TO 3). A propósito, a palavra-chave TO encontra-se na tecla F.

LEN

O programa que se segue escolhe aleatoriamente elementos de uma cadeia por nós indicada. Note, na linha 40, a referência a LEN, o comprimento da cadeia. Introduza o programa na máquina, e execute-o durante algum tempo utilizando uma cadeia qualquer de sua preferência, por exemplo o seu nome:

```
 10 REM TRATAMENTO DE CADEIAS
 20 INPUT "ESCREVA UMA CADEIA "
A$
 30 PRINT "A SUA CADEIA E' ";A$
 40 PRINT '"LEN A$ = ";LEN A$
 50 PRINT '"A$(1) = ";A$(1)
 60 PRINT '"A$(2 TO ) ='";A$(2
TO )
 80 LET B=INT (RND*(LEN A$)+1)
 90 LET C=INT (RND*(LEN A$)+1)
100 IF C>B THEN GO TO 80
110 PRINT "A$(";C;" TO ";B;") =
";A$(C TO B)
120 GO TO 80
```

Vejamos dois exemplos de execução do programa:

```
A SUA CADEIA E' PRACA DO AREEIRO

LEN A$ = 16

A$(1) = P

A$(2 TO ) = RACA DO AREEIRO
A$(4 TO 5) = CA
A$(3 TO 4) = AC
A$(2 TO 12) = RACA DO ARE
A$(9 TO 10) = A
A$(9 TO 13) = AREE
A$(1 TO 11) = PRACA DO AR
A$(3 TO 13) = ACA DO AREE
A$(3 TO 6) = ACA
A$(2 TO 5) = RACA
A$(3 TO 13) = ACA DO AREE
A$(7 TO 10) = DO A
```

```
A$(11 TO 16)  = REEIRO
A$(5 TO 6)  = A

A SUA CADEIA E' CLIVE SINCLAIR

LEN A$ = 14

A$(1)  = C

A$(2 TO )  = LIVE SINCLAIR
A$(3 TO 4)  = IV
A$(2 TO 3)  = LI
A$(2 TO 10)  = LIVE SINC
A$(8 TO 9)  = IN
A$(8 TO 11)  = INCL
A$(1 TO 10)  = CLIVE SINC
A$(3 TO 12)  = IVE SINCLA
A$(3 TO 5)  = IVE
A$(2 TO 4)  = LIV
A$(3 TO 11)  = IVE SINCL
A$(9 TO 12)  =  ARE
A$(6 TO 10)  =  DO A
A$(3 TO 14)  = ACA DO AREEI
A$(11 TO 13)  = REE
A$(2 TO 11)  = RACA DO AR
A$(1 TO 16)  = PRACA DO AREEIRO
A$(15 TO 15)  = R
A$(14 TO 16)  = IRO
A$(6 TO 12)  =  DO ARE
A$(4 TO 5)  = CA
A$(14 TO 16)  = IRO
A$(2 TO 11)  = RACA DO AR
A$(2 TO 8)  = RACA DO
A$(2 TO 3)  = RA
A$(11 TO 15)  = REEIR
A$(2 TO 11)  = RACA DO AR
A$(7 TO 7)  = D
A$(11 TO 16)  = REEIRO
A$(5 TO 16)  = A DO AREEIRO
A$(9 TO 12)  =  ARE
A$(9 TO 14)  =  AREEI
A$(4 TO 4)  = C
A$(2 TO 3)  = RA
A$(4 TO 6)  = CA
A$(7 TO 10)  = DO A
A$(7 TO 11)  = DO AR
A$(10 TO 13)  = AREE
A$(8 TO 14)  = O AREEI
A$(4 TO 14)  = CA DO AREEI
A$(5 TO 9)  = A DO
A$(7 TO 7)  = D
A$(4 TO 4)  = C
```



```
A$(2 TO 10)  = RACA DO A
A$(9 TO 15)  =  AREEIR
A$(15 TO 16)  = RO
A$(2 TO 16)  = RACA DO AREEIRO
A$(2 TO 5)  = RACA
A$(6 TO 16)  =  DO AREEIRO
```

Uma cadeia pode ser cortada em partes e cada uma destas acrescentada a outras partes da mesma cadeia ou de qualquer outra. O acrescento ou soma de cadeias é designado por *concatenação* (não se pode, a propósito, subtrair cadeias umas das outras).

O programa seguinte escolhe partes de uma cadeia por nós indicada e "concatena-as". A secção que inclui as linhas 45, 46 e 47 constitui um outro exemplo da facilidade de manipulação das cadeias. Neste caso a cadeia é impressa do fim para o princípio, como o leitor poderá constatar no exemplo de execução que se segue à listagem do programa. Para parar este programa o leitor deverá carregar na tecla **BREAK**.

```
  10 REM MANUSEAMENTO DE CADEIAS
  20 INPUT "ESCREVA UMA CADEIA "
;A$
  30 PRINT "A CADEIA E' ";A$
  40 LET L=LEN A$
  45 FOR G=L TO 1 STEP -1
  46 PRINT A$(G);
  47 NEXT G
  50 LET B=INT (RND*L)+1
  60 LET C=INT (RND*L)+1
  70 IF B=C THEN GO TO 60
  80 PRINT 'B;" ";C
  90 IF B>C THEN PRINT A$( TO C)
+A$(B TO )
 100 IF B<C THEN PRINT A$( TO B)
+A$(C TO )
 110 GO TO 50


A CADEIA E' CLIVE<>SINCLAIR
RIALCNIS<>EVILC
14  10
CLIVE<>SINCR

5  4
CLIVE<>SINCLAIR

10  8
CLIVE<>SICLAIR
```

```
6 9
CLIVE<>NCLAIR

6 13
CLIVE<>IR

8 6
CLIVE<>INCLAIR

7 10
CLIVE<>SCLAIR

1 11
CLAIR

4 9
CLIVNCLAIR
```

VAL

Uma outra função muito útil para o tratamento de cadeias em Basic é a VAL. Esta função determina o *equivalente numérico* da cadeia. Actua assim:

LET A$ = "23 + 3"
PRINT A$ dará, como seria de esperar, 23 + 3
PRINT VAL A$ imprimirá 26

Pode-se usar a função VAL para produzir uma simples máquina de calcular usando o Spectrum, capaz de calcular praticamente tudo o que quisermos:

```
  10 REM CADEIAS COMO NUMEROS
  20 INPUT "INDIQUE A OPERACAO "
;A$
  30 PRINT 'A$;" = ";VAL A$
  40 GO TO 20
```

Vejamos um exemplo da execução deste programa:

```
33+6-19 = 20

2↑5-7 = 25

LN 7-SQR 12 = -1.5181915

(79-4↑2)-COS 3 = 83.989992

(45/7+SGN 7)↑34.789 = 1.9857502E
+30
```


O oposto de VAL é STR$, que transforma um número numa cadeia. Ou seja, se dissermos ao computador para tornar A$ igual a STR$ 123 este atribuirá a A$ o valor "123".

O aspecto mais interessante da discussão que fizemos até agora neste capítulo reside precisamente na extraordinária utilidade do tratamento de cadeias para trabalho com números. Como é possível extrair facilmente *partes* das cadeias de um modo que não é possível utilizar usando números, e como se podem transformar números em cadeias e estas naqueles usando STR$ e VAL, dispomos de um poderoso instrumento para tratamento de números.

Para verificar isto o leitor poderá introduzir o seguinte programa na máquina:

```
  10 REM NUMEROS COMO CADEIAS
  20 INPUT "ESCREVA UM NUMERO ";
A
  25 PRINT "O SEU NUMERO E' ";A
  30 LET A$=STR$ A
  40 LET L=LEN A$
  50 LET B=INT (RND*L)+1
  60 LET C=INT (RND*L)+1
  70 IF B>=C THEN GO TO 50
  80 PRINT '"A$ (";B;" TO ";C;")
= ";A$(B TO C)
  90 GO TO 50
```

Como pode observar no seguinte exemplo de execução, grande parte das saídas são razoavelmente inúteis:

```
O SEU NUMERO E' 12.345

A$ (3 TO 5) = .34

A$ (4 TO 6) = 345

A$ (4 TO 6) = 345

A$ (5 TO 6) = 45

A$ (4 TO 6) = 345

A$ (4 TO 5) = 34

A$ (3 TO 4) = .3

A$ (3 TO 6) = .345
```

```
A$ (3 TO 5) = .34
A$ (1 TO 5) = 12.34
A$ (4 TO 6) = 345
A$ (3 TO 6) = .345
A$ (1 TO 5) = 12.34
A$ (3 TO 6) = .345
A$ (1 TO 2) = 12
A$ (2 TO 3) = 2.
A$ (2 TO 5) = 2.34
A$ (1 TO 3) = 12.
A$ (3 TO 5) = .34
A$ (1 TO 6) = 12.345
A$ (1 TO 2) = 12
```

No entanto, encontra entre elas A$(1 TO 2). Poderá notar que esta tem o valor 12, ou seja o número original sem a parte fraccionária. Na linha 13 encontra A$(1 TO 5). Neste caso obteve-se 12,34, o número original sem a terceira casa decimal. Se pudermos controlar os cortes executados, teremos à nossa disposição um instrumento muito útil para manipular números.

Correcção para N algarismos significativos

O resto deste capítulo tentará demonstrar o rigor da anterior afirmação. O nosso primeiro programa, como o leitor pode verificar, define números com a quantidade de algarismos significativos por nós especificadas:

```
  10 REM PARA CORRIGIR NUMEROS
       POSITIVOS < 10E7
     PARA N ALGARISMOS
      SIGNIFICATIVOS
  20 INPUT "ESCREVA UM NUMERO ";
X$
  25 INPUT "PARA QUANTOS ALGARIS
MOS SIGNIFI-CATIVOS QUER CORRIGI
R ";(X$);" ? ";N
  30 LET Y=VAL X$*(10↑(LEN X$-1)
)
  40 PRINT 'X$;" com ";N;" algar
ismos","significativos e' ";INT
(VAL STR$ Y/10↑(LEN STR$ Y-N)+.5
)*(10↑(LEN STR$ Y-N))/(10↑(LEN X
$-1))
  50 GO TO 20

54637 com 2 algarismos
significativos e' 55000

.054672 com 3 algarismos
significativos e' .0547

1/674 com 3 algarismos
significativos e' 0

674/45 com 3 algarismos
significativos e' 10

674/45 com 7 algarismos
significativos e' 14.978
```

A versão seguinte do programa especifica melhor o que está a acontecer (no caso de números negativos, a propósito, indique apenas os algarismos). A linha 30 transforma o número num número inteiro, no caso de o número original ser decimal. A linha 70 produz os algarismos significativos. A linha 90 produz o número inteiro corrigido, e a linha 110 apresenta a resposta:

```
   5 REM CORRIGIR NUMEROS PARA N
     ALGARISMOS SIGNIFICATIVOS
  10 INPUT "ESCREVA UM NUMERO ";
X$
  20 PRINT "O NUMERO E' ";X$
  30 LET Y=VAL X$*(10↑(LEN X$-1)
)
  40 PRINT '"Y = ";Y
  50 INPUT "QUANTOS ALGARISMOS S
IGNIFICATI- VOS? ";N
  60 PRINT '"VOU CORRIGI-LO PARA
 ";N,"ALGARISMOS SIGNIFICATIVOS"
  70 LET Z=INT (VAL STR$ Y/10↑(L
EN STR$ Y-N)+.5)
  80 PRINT TAB 6;"Z = ";Z
```

```
  90 LET U=Z*(10↑(LEN STR$ Y-N))
 100 PRINT TAB 10;"Y = ";Y
 110 PRINT 'X$;" = ";U*VAL X$/Y'
"      com ";N;" algarismos","
  significativos"
 120 GO TO 10
```

```
Y = 54672

VOU CORRIGI-LO PARA 3
ALGARISMOS SIGNIFICATIVOS
      Z = 547
          Y = 54672

.054672 = .0547
     com 3 algarismos
     significativos

Y = 1497777.8

VOU CORRIGI-LO PARA 7
ALGARISMOS SIGNIFICATIVOS
      Z = 14978
          Y = 1497777.8

674/45 = 14.978
     com 7 algarismos
     significativos
```

Conversão para notação científica

Finalmente, veremos um programa que converte números para notação científica:

```
   5 REM CONVERSAO PARA NOTACAO
       CIENTIFICA
  10 INPUT "ESCREVA UM NUMERO ";
X$
  15 IF X$="P" OR X$="p" THEN ST
OP
  20 PRINT 'X$;" = ";
  30 LET Y=VAL X$*10↑(LEN X$-1)
  40 IF VAL X$>=10 THEN PRINT VA
L X$/10↑(LEN X$-1);"E+";LEN X$-1
  50 IF VAL X$<10 THEN PRINT Y/1
0↑(LEN STR$ Y-1);"E";LEN STR$ Y-
LEN X$
  60 GO TO 10
```



```
.00000345 = 3.45E-6

.0345 = 3.45E-2

.345 = 3.45E-1

345 = 3.45E+2

3450000000000000 = 3.45E+15
```

5

USO DO SPECTRUM PARA CÁLCULOS MAIS AVANÇADOS

O Spectrum parece ter sido construído de propósito para execução de cálculos matemáticos. A sua vasta gama de funções matemáticas, e a facilidade com que pode ser programado, convidam os estudantes (e professores) a criarem programas para resolver problemas específicos ou demonstrar procedimentos matemáticos.

O Spectrum é ideal para situações em que um dado problema se coloca do mesmo modo durante algum tempo, mas em que os dados a manipular se alteram. O leitor verificará por outro lado que o computador é óptimo para produzir exemplos visuais de procedimentos matemáticos.

Neste capítulo do livro, observaremos alguns programas que incluímos principalmente para dar ao leitor uma ideia dos procedimentos matemáticos que podem ser executados no Spectrum. Alguns deles poderão ser aplicados num curso dado pelo leitor.

Ao introduzir e executar estes programas, o leitor deve ficar com uma ideia da flexibilidade do Spectrum neste tipo de trabalho. Além disso, ganhará algumas sugestões relativas à eficácia que as apresentações visuais podem ter quando usadas em programas de matemática ou de outros temas.

Série de números

A relação existente entre os elementos sucessivos de uma série é muitas vezes simples de descobrir, mas o cálculo da série pode ser bastante moroso, mesmo quando o termo geral usado para a produção de cada número não é particularmente complexo. O Spectrum torna muito mais rápido este tipo de trabalho, como o leitor poderá verificar nos cinco programas que se seguem.

O primeiro imprime a série 1, 1*2, 2*3, 6*7,..., depressa atingindo o limite numérico superior do Spectrum (7*10↑38).

Vejamos a listagem:

```
10 REM SERIES DE NUMEROS
20 LET A=1
30 PRINT A
40 LET A=A*(A+1)
50 GO TO 30
```

E observemos um exemplo de execução:

```
1
2
6
42
1806
3263442
1.0650057E+13
1.1342371E+26
```

Série mistério

Vamos deixar ao leitor o cuidado de descobrir o termo geral da série que se segue. Vejamos a saída do programa:

```
1
2
6
21
88
445
2676
18739
149920
1349289
13492900
1.4842191E+8
1.7810629E+9
2.3153818E+10
3.2415346E+11
4.8623018E+12
7.7796829E+13
1.3225461E+15
2.380583E+16
4.5231077E+17
9.0462153E+18
```

Conseguiu? Vejamos agora a listagem do programa:

```
20 LET A=1: PRINT A
30 FOR N=1 TO 20
40 LET A=N*(A+1)
50 PRINT A
60 NEXT N
```

Números de Fibonnaci

Fibonnaci é o santo patrono dos coelhos. O seu nome embeleza a série de números produzida por este breve programa:

```
10 PRINT "NUMEROS DE FIBONNACI

20 LET M=0
30 LET N=1
40 LET P=M+N
50 PRINT P
60 LET M=N
70 LET N=P
80 GO TO 40
```

Início da série de Fibonnaci:

```
NUMEROS DE FIBONNACI
1
2
3
5
8
13
21
34
55
89
144
233
377
610
987
1597
2584
4181
6765
10946
17711
28657
46368
```



```
75025
121393
196418
317811
514229
832040
13462269
2178309
3524578
57028887
9227465
14930352
241157817
390088169
632245986
1.0233416E+8
1.6558014E+8
2.679143E+8
4.3349444E+8
7.0140873E+8
```

Soma de séries

Observaremos agora dois programas que produzem a soma de uma série. O primeiro dá-nos a soma da série

$$1/X^m + 1/X^2 + 1/X^3 + \ldots + 1/X^N$$

```
  10 REM SOMA DE SERIES
  15 INPUT "INDIQUE VALOR DE X "
;X
  20 LET S=0
  30 FOR N=1 TO 40
  40 LET S=S+1/(X↑N)
  50 PRINT S,
  60 NEXT N
```

Vejamos a série para X = 1,7:

```
0.58823529      0.93425606
1.1377977       1.2557528
1.3279577       1.36938869
1.393757        1.40809923
1.41165249      1.42148852
1.4244031       1.42611195
1.4271291       1.427723
1.42807724      1.42822779
1.42839987      1.42846598
1.42865117      1.42865363
```

```
1.4285503          1.4285593
1.4285643          1.4285672
1.428569           1.42857
1.4285706          1.4285709
1.4285711          1.4285713
1.4285713          1.4285714
1.4285714          1.4285714
1.4285714          1.4285714
1.4285714          1.4285714
1.4285714          1.4285714
```

A segunda soma de uma série utiliza a série

$1^2+2^2+3^2+\ldots+N^2$

```
  10 REM SOMA DE SERIES
  15 INPUT "INDIQUE VALOR DE N "
;N
  20 LET X=0
  30 FOR M=1 TO N
  40 LET X=X+(M↑2)
  50 PRINT X
  60 NEXT M
```

Vejamos a execução para N = 16:

```
1
5
14
30
55
91
140
204
285
385
506
650
819
1015
1240
1496
1785
2109
2470
2870
```


As raízes quadradas e Sir Isaac

Inicialmente, observaremos três programas que têm a ver com a dedução de raízes quadradas. O primeiro converge para a raiz quadrada de dois, usando fracções contínuas, e o segundo exemplifica este processo. O terceiro programa, baseado no primeiro, é pensado para determinar a raiz quadrada de N por fracções contínuas.

Dê a R qualquer valor arbitrário ao executar este programa:

```
  20 INPUT "INDIQUE UM NUMERO: "
;R
  30 PRINT "RAIZ QUADRADA DE 2 =
"
  40 PRINT TAB 23;R
  50 LET R=1+1/(1+R)
  60 GO TO 40
```

```
RAIZ QUADRADA DE 2 =
                              1000
                              1.000999
                              1.4997504
                              1.4000399
                              1.4166597
                              1.4137943
                              1.4142855
                              1.4142012
                              1.4142157
                              1.4142132
                              1.4142136
                              1.4142136
                              1.4142136
                              1.4142136
                              1.4142136
                              1.4142136
                              1.4142136
                              1.4142136
                              1.4142136
                              1.4142136
                              1.4142136
```

O programa que se segue, como poderá verificar pelo exemplo de execução correspondente, exemplifica o processo:

```
  5 REM FRACCOES CONTINUAS
 10 REM FAZER R=SQR DE 2
 20 REM R↑2-1=(R+1)(R-1)
 30 REM PORTANTO (R+1)(R-1)=1
 40 REM DONDE R-1=1/(R+1)
```

```
 50 REM OU R=1+1/(1+R)
 60 PRINT AT 1,0;"R=1 +"
 70 FOR N=0 TO 18 STEP 2
 80 PRINT AT N,N+8;"1";AT N+1,N+6;"-----";AT N+2,N+6;"2 +"
 90 NEXT N
100 PRINT AT 20,28;"etc."
```

```
           1
R=1 +   -----
        2 + 1
          -----
          2 + 1
            -----
            2 + 1
              -----
              2 + 1
                -----
                2 + 1
                  -----
                  2 + 1
                    -----
                    2 + 1
                      -----
                      2 + 1
                        -----
                        2 + 1
                          -----
                          2 + etc.
```

Raízes quadradas por fracções contínuas

O nosso terceiro programa converge para a raiz quadrada de N usando fracções contínuas. Depois da listagem, o leitor poderá observar como determina a raiz quadrada de 4 (execução número um) e de 9 (execução número dois):

```
 4 REM    RAIZ QUADRADA DE N
 5 REM  POR FRACCOES CONTINUAS
10 INPUT "INDIQUE N ";N
20 LET R=1
30 LET R=1+(N-1)/(1+R)
40 PRINT R
50 GO TO 30
```

```
2.5
1.8571429
2.05
1.9836066
2.0054945
1.9981718
2.0006098
1.9997968
2.0000677
1.9999774
2.0000075
1.9999975
2.0000008
1.9999997
2.0000001
2
2
2
2
2
2
2
```

```
5
2.3333333
3.4
2.8181818
3.0952381
2.9534884
3.0235294
2.9883041
3.0058651
2.9970717
3.0014652
2.9992677
3.0003662
2.9998169
3.0000916
2.9999542
3.0000229
2.9999886
3.0000057
2.9999971
3.0000014
2.9999993
3.0000004
2.9999998
3.0000001
3
3
```

Raízes quadradas pela fórmula de Newton

No caso de querer saber porque razão Sir Isaac Newton mereceu

uma menção nos nossos subtítulos, eis a resposta. O programa que se segue foi concebido para determinar raízes quadradas recorrendo à fórmula de Newton. Depois de escrever o número (N), indica-se um valor provável da raiz:

```
  10 REM    DETERMINACAO DA
            RAIZ QUADRADA PELA
            FORMULA DE NEWTON
  20 INPUT "INDIQUE UM NUMERO: "
;N
  25 INPUT "O NUMERO E' ";(N),"I
NDIQUE UM VALOR APROXIMADO DA","
RAIZ: ";G
  30 LET S=(G+N/G)/2
  40 PRINT S,S↑2
  50 IF ABS (S-G)<S/1000000 THEN
 STOP
  60 LET G=S
  70 GO TO 30
```

Eis o programa em acção procurando a raiz quadrada de 1000 com um valor provável de 17:

```
37.911765       1437.3019
32.144401       1033.2625
31.627009       1000.2677
31.622777       1000
31.622777       1000
```

Desta vez indiquei um valor provável de 1, para ver qual seria a diferença:

```
500.5           250500.25
251.249         63126.06
127.61456       16285.475
67.725327       4586.72
41.245426       1701.1852
32.745269       1072.2527
31.642016       1001.2172
31.622782       1000.0004
31.622777       1000
```

É muitas vezes instrutivo, depois de se ter um programa correcto, perder algum tempo a analisar as saídas do programa numa tentativa de mostrar ao utilizador o que se está a passar. A partir da listagem dada mais atrás produzimos a seguinte versão, que imprime

claramente o procedimento usado:

```
  10 PRINT "RAIZ QUADRADA PELA
            FORMULA DE NEWTON"
  20 INPUT "INDIQUE UM NUMERO: "
;N
  25 INPUT "O NUMERO E' ";(N),"I
NDIQUE UM VALOR APROXIMADO DA","
RAIZ: ";G
  27 PRINT '"O numero e' ";N
  28 PRINT '"O valor aproximado
e' ";G: PRINT
  30 LET S=(G+N/G)/2
  40 PRINT S,S↑2
  50 IF ABS (S-G)<S/1000000 THEN
 STOP
  60 LET G=S
  70 GO TO 30
```

Vejamos um exemplo de execução. Como o leitor poderá verificar, o programa torna-se mais explícito. Por outro lado, são convenientemente registados os valores indicados para N e G (um último melhoramento consistiria em introduzir uma linha do tipo "A raiz quadrada de ... é ..."):

```
RAIZ QUADRADA PELA
FORMULA DE NEWTON

O numero e' .09894

O valor aproximado e' .03

1.664           2.768896
0.86172957      0.74257785
0.48827259      0.23841012
0.34545265      0.11933753
0.31592974      .0998116O2
0.31455032      .0989419O3
0.31454729      .09894
0.31454729      .09894
```

Resolução de equações pelo método de Newton

A secção que se segue estuda programas que resolvem equações gerais usando o método de Newton. É surpreendente como os computadores convergem tão rapidamente para a solução.

A primeira rotina, escrita por Jeremy Ruston, espera que o leitor

escreva uma função de X (linha 10). Depois, em resposta ao pedido da linha 40, o leitor deve introduzir um valor aproximado a partir do qual o computador realizará as iterações. Este valor deve (como acontece no caso dos exemplos da raiz quadrada apresentados anteriormente) ser próximo daquilo que o leitor pensa ser a resposta correcta, ou no caso de existirem várias respostas correctas, um número próximo da resposta que deseja. Para experimentar o programa, escreva X*X-5 (que determina a raiz quadrada de cinco) ou X↑3—27.6 (que determina a raiz quadrada de 27,6):

```
  5 REM   METODO DE NEWTON PARA
          RESOLUCAO DE EQUACOES
 10 REM    Por JEREMY RUSTON
 20 INPUT "INDIQUE UMA FUNCAO D
E X: ";F$
 30 PRINT F$
 40 INPUT "INDIQUE UM VALOR APR
OXIMADO: ";S
 50 PRINT S
 70 INPUT "INDIQUE ERRO MAXIMO:
";ERR
 90 PRINT ERR
100 PRINT AT 10,10;S
110 LET X=S
120 IF ABS (VAL F$)<ERR THEN ST
OP
130 LET T=VAL F$
140 LET X=X+0.00001
150 LET B=((VAL F$)-T)/0.00001
160 LET S=S-T/B
170 GO TO 100
```

A versão que se segue é mais flexível, permitindo-lhe indicar ambos os membros da equação, sendo novamente X o valor a determinar (note que o erro aceitável — no final da linha 80 — é tão pequeno que algumas equações não produzirão soluções aceitáveis; talvez convenha então aumentar este erro para 0,0001):

```
  5 REM   SOLUCAO DE EQUACOES
          PELA FORMULA DE NEWTON
 10 INPUT "INDIQUE PRIMEIRO MEM
BRO DA","EQUACAO: ";E$
 20 INPUT "INDIQUE SEGUNDO MEMB
RO DA","EQUACAO: ";D$
 30 PRINT AT 1,10;E$;"=";D$
 40 INPUT "INDIQUE UM VALOR APR
```



```
OXIMADO: ";S
 45 PRINT AT 5,11;".";AT 5,10;"
. . X="
 50 PRINT AT 5,16;S
 60 LET X=S
 70 LET F=VAL E$-VAL D$
 80 IF ABS F<.000001 THEN STOP
 90 LET X=X+1
100 LET S=S-F/(VAL E$-VAL D$-F)
110 GO TO 50
```

Vejamos os resultados da execução do programa:

```
X↑2-4*X=X↑6-44

                . .  X=1.8494857


X↑3=100

                . .  X=4.6415888


X↑2=100

                . .  X=10.000002
```

Como já dissemos anteriormente, é muitas vezes boa ideia reescrever um programa depois de estar terminado de modo a tornar as imagens mais agradáveis e tão informativas quanto possível. Um outro modo de modificar um programa consiste em introduzir instruções que enviam informações para a impressora e não para o visor. Para exemplificar isto, vamos apresentar agora o programa anterior reescrito de modo a enviar as saídas para a impressora:

```
  5 REM   SOLUCAO DE EQUACOES
          PELA FORMULA DE NEWTON
  6 REM VERSAO PARA IMPRESSORA
 10 INPUT "INDIQUE PRIMEIRO MEM
BRO DA","EQUACAO: ";E$
```

```
  20 INPUT "INDIQUE SEGUNDO MEMB
RO DA","EQUACAO: ";D$
  30 LPRINT "'E$;"=";D$
  40 INPUT "INDIQUE UM VALOR APR
OXIMADO: ";S
  42 LPRINT : LPRINT "Valor apro
ximado: ";S
  50 LPRINT : LPRINT S
  60 LET X=S
  70 LET F=VAL E$-VAL D$
  80 IF ABS F<.0001 THEN GO TO 1
20
  90 LET X=X+1
 100 LET S=S-F/(VAL E$-VAL D$-F)
 110 GO TO 50
 120 LPRINT INVERSE 1;">>"; INVE
RSE 0;" X = ";S
```

Vejamos um exemplo da execução do programa, neste caso procurando uma raiz de 1000:

```
X↑5=1000
Valor aproximado: 4
4
3.9885769
3.984028
3.9822331
3.9815275
3.9812505
3.9811418
3.9810992
3.9810825
3.9810759
3.9810734
3.9810724
3.981072
3.9810718
3.9810717
>> X = 3.9810717
```

Um outro exemplo, resolvendo uma equação bastante estranha (o que demonstra que é preciso muito para deixar mal o nosso Spectrum...):

```
(2*PI/X)↑2=LN X
Valor aproximado: 1
1
2.3028338
3.881935
4.9413868
4.9694722
4.9624762
4.9642559
4.9638055
>> X = 4.9638055
```

O computador resolve agora duas vezes uma outra equação, partindo de valores iniciais muito diferentes (o que mostra que, em alguns casos, o valor de partida pode ser praticamente qualquer):

```
17*X-COS X/PI=SQR X
Valor aproximado: .02
.02
.027349178
.028491787
.028652158
.028674357
>> X = .028674357
```

```
17*X-COS X/PI=SGR X

Valor aproximado: 100

100

0.256304

.042804712

.030397919

.028911325

.028710073

.028682355
[illegible] X = .028682355
```

Determinação de factores

Esquecendo agora Sir Isaac, vamos examinar algumas aplicações matemáticas do Spectrum. A primeira, escrita por David Perry, determina factores. O programa fará as perguntas necessárias, pelo que o leitor não terá qualquer problema na sua utilização:

```
  10 BORDER 1: PAPER 1: INK 7: B
RIGHT 8: CLS
  20 PRINT AT 0,6; PAPER 2;" CAL
CULAR FACTORES. "
  30 INPUT "PRODUTO? ";P: PRINT
AT 2,2; PAPER 0;" Produto: ";P;"
  "
  40 INPUT "SOMA? ";S: PRINT AT
4,4; PAPER 0;" Soma: ";S;"  "
  50 FOR a=1 TO ABS (P)
  60 FOR b=ABS (P) TO 1 STEP -1
  70 IF (a*b)=ABS (P) THEN GO SU
B 120
  80 NEXT b
  90 NEXT a
 100 PRINT : PRINT "Carregue em
[illegible] para continuar"
 110 PAUSE 0: PAUSE 0: RUN
 120 IF (a*b)=P AND (a+b)=S THEN
 PRINT " Factores: +";a;",+";b
 130 IF ((-a)+b)=S AND ((-a)*b)=
P THEN PRINT " Factores: -";a;",
+";b
```



```
 140 IF ((-b)+a)=S AND ((-b)*a)=
P THEN PRINT " Factores: +";a;",
-";b
 150 IF ((-a)+(-b))=S AND ((-a)*
(-b))=P THEN PRINT " Factores: -
";a;",-";b
 160 PRINT a,b
 170 RETURN
```

Cálculo de N! (factorial)

O programa que se segue calcula N! (N factorial). A listagem é seguida de alguns exemplos de execução:

```
  10 REM Imprimir N! (factorial)
  15 INPUT "Indique valor de N:
";N
  20 PRINT N;" factorial ="
  30 LET x=1
  40 FOR p=1 TO N
  50 LET x=x*p
  60 NEXT p
  70 PRINT x


5 factorial =
120

12 factorial =
4.790016E+8

23 factorial =
2.5852017E+22

32 factorial =
2.6313084E+35
```

Relacionado ainda com o último programa, o que se segue imprime factoriais entre 1 e M. Apresenta-se a seguir um exemplo de execução:

```
  10 REM Imprimir factoriais
         de 1 a m
  15 INPUT "Indique valor de M:
";M
  20 FOR n=1 TO M
  30 LET x=1
  40 FOR p=1 TO N
  50 LET x=x*p
```

```
60 NEXT P
70 PRINT N;" factorial = ";X
80 NEXT n
```

```
1  factorial  =  1
2  factorial  =  2
3  factorial  =  6
4  factorial  =  24
5  factorial  =  120
6  factorial  =  720
7  factorial  =  5040
8  factorial  =  40320
9  factorial  =  362880
10  factorial  =  3628800
11  factorial  =  39916800
12  factorial  =  4.7900016E+8
13  factorial  =  6.2270208E+9
14  factorial  =  8.7178291E+10
15  factorial  =  1.3076744E+12
16  factorial  =  2.0922279E+13
17  factorial  =  3.5568743E+14
18  factorial  =  6.4023737E+15
19  factorial  =  1.2164451E+17
20  factorial  =  2.4329002E+18
21  factorial  =  5.1090942E+19
22  factorial  =  1.1240007E+21
```

Resolução de equações quadráticas

Entre os muitos recursos do Spectrum encontra-se a capacidade para resolver equações quadráticas mais rapidamente do que o leitor:

```
  10 REM  PROGRAMA PARA RESOLVER
          EQUACOES QUADRATICAS
  20 INPUT "INDIQUE A: ";A
  30 INPUT "INDIQUE B: ";B
  35 INPUT "INDIQUE C: ";C
  37 PRINT "Se A = ";A;", B = ";
B;" e C = ";C
  40 LET X1=(-B+SQR (B*B-4*A*C))
/(2*A)
  50 LET X2=(-B-SQR (B*B-4*A*C))
/(2*A)
  60 PRINT '"X = ";X1;" ou ";X2
```

```
Se A = 1, B = 9 e C = 2

X = -0.22799813 ou -8.7720019
```


Série triangular

O programa que se segue produz elementos da Série Triangular de ordem compreendida entre 1 e 100 como se pode verificar consultando a saída do programa, apresentada a seguir à listagem:

```
  10 PRINT "SERIE TRIANGULAR"
  20 FOR N=1 TO 100
  30 PRINT N*(N+1)/2;"   ";
  40 NEXT N
```

```
SERIE TRIANGULAR
1    3    6    10    15    21    28
36    45    55    66    78    91    10
5    120    136    153    171    190
  210    231    253    276    300
325    351    378    406    435    46
5    496    528    561    595    630
  666    703    741    780    820
861    903    946    990    1035    1
081    1128    1176    1225    1275
  1326    1378    1431    1485    15
40    1596    1653    1711    1770
  1830    1891    1953    2016    208
0    2145    2211    2278    2346
2415    2485    2556    2628    2701
  2775    2850    2926    3003    3
081    3160    3240    3321    3403
  3486    3570    3655    3741    38
28    3916    4005    4095    4186
  4278    4371    4465    4560    465
5    4753    4851    4950    5050
```

Se preferir ordenar a saída de um modo mais compreensível, talvez lhe interesse esta segunda versão:

```
   5 REM     VERSAO ALTERNATIVA
  10 PRINT "SERIE TRIANGULAR"
  20 FOR N=1 TO 21
  30 PRINT N;TAB 5;N*(N+1)/2;TAB
 10;(N+23)*(N+24)/2;TAB 15;(N+45
)*(N+46)/2;TAB 20;(N+67)*(N+68)/
2
  40 NEXT N
```

```
SERIE TRIANGULAR
1     1     300   1081 2346
```

```
 2      3      325    1128   2415
 3      6      351    1176   2485
 4      10     378    1225   2556
 5      15     406    1275   2628
 6      21     435    1326   2701
 7      28     465    1378   2775
 8      36     496    1431   2850
 9      45     528    1485   2926
 10     55     561    1540   3003
 11     66     595    1596   3081
 12     78     630    1653   3160
 13     91     666    1711   3240
 14     105    703    1770   3321
 15     120    741    1830   3403
 16     136    780    1891   3486
 17     153    820    1953   3570
 18     171    861    2016   3655
 19     190    903    2080   3741
 20     210    946    2145   3828
 21     231    990    2211   3916
```

Triângulo de Pascal

O triângulo de Pascal pode ser impresso por este programa, no qual a linha 30 dá o primeiro número em cada fiada, a linha 50 o resto dos números dessa fiada, e a linha 60 determina a "recorrência":

```
  5 REM Triangulo de Pascal
 10 INPUT "QUANTAS LINHAS (<8)?
";M
 15 IF M>8 THEN LET M=8
 20 FOR N=0 TO M
 30 LET C=1
 40 PRINT AT 2*N,(15-2*N);C
 50 FOR R=1 TO N
 60 LET C=C*(N-R+1)/R
 70 PRINT AT N*2,(15-N*2+R*4);C
 80 NEXT R
 90 NEXT N
```

Como poderá verificar pelos exemplos de execução apresentados em seguida, a saída do programa é limitada pela largura da imagem produzida pelo Spectrum (talvez o leitor queira modificar o programa de tal modo que produza os números mas os imprima de outra forma, sendo depois convertidos "manualmente" para uma imagem triangular):


```
                1
              1   1
            1   2   1
          1   3   3   1
        1   4   6   4   1
      1   5   10  10  5   1
```

```
                1
              1   1
            1   2   1
          1   3   3   1
        1   4   6   4   1
      1   5   10  10  5   1
    1   6   15  20  15  6   1
  1   7   21  35  35  21  7   1
1 8   28  56  70  56  28  8   1
```

Conversão da base decimal para a binária

Este programa demonstra o modo de converter números na base decimal para a base binária. Leia os restos das divisões de baixo para cima para encontrar a solução na base dois:

```
 10 INPUT "INDIQUE UM NUMERO NA
BASE DEZ:  ";N
 20 PRINT N;" a dividir por 2 d
a' ";INT (N/2);'TAB 20;"Resto: "
;N-2*INT (N/2)
 30 LET N=INT (N/2)
 40 IF N>=1 THEN GO TO 20
```

Vejamos o programa em acção:

```
16 a dividir por 2 da' 8
                  Resto: 0
8 a dividir por 2 da' 4
                  Resto: 0
4 a dividir por 2 da' 2
                  Resto: 0
2 a dividir por 2 da' 1
                  Resto: 0
1 a dividir por 2 da' 0
                  Resto: 1

19 a dividir por 2 da' 9
                  Resto: 1
9 a dividir por 2 da' 4
                  Resto: 1
4 a dividir por 2 da' 2
                  Resto: 0
2 a dividir por 2 da' 1
                  Resto: 0
1 a dividir por 2 da' 0
                  Resto: 1

94 a dividir por 2 da' 47
                  Resto: 0
47 a dividir por 2 da' 23
                  Resto: 1
23 a dividir por 2 da' 11
                  Resto: 1
11 a dividir por 2 da' 5
                  Resto: 1
5 a dividir por 2 da' 2
                  Resto: 1
2 a dividir por 2 da' 1
                  Resto: 0
1 a dividir por 2 da' 0
                  Resto: 1
```

A listagem que se segue converte números na base dez para os seus equivalentes binários:

```
   5 REM CONVERSAO DE DECIMAL PA
RA        BINARIO
  10 INPUT "INDIQUE UM NUMERO NA
 BASE DEZ:  ";N
  20 PRINT ''N;" (base 10)";
  30 FOR J=0 TO INT (LN N/LN 2)
  40 PRINT AT 2,28-J;N-2*INT (N/
2);
  50 LET N=INT (N/2)
  60 NEXT J
```

Vejamos como actua este programa:

```
123 (base 10) =            1111011

3 (base 10) =                    11

452 (base 10) =          111000100

1234 (base 10) =       10011010010

255 (base 10) =           11111111.
```

Arredondar para o inteiro mais próximo

Esta curta rotina, como se pode observar pelo exemplo de execução que se segue à listagem, arredonda números para o valor inteiro mais próximo:

```
  10 REM ARREDONDAR PARA INTEIRO
       MAIS PROXIMO
  20 INPUT "INDIQUE UM NUMERO: "
;X
  30 PRINT "O arredondamento de
";X;"   da' ";INT (X+.50001)
  40 PRINT
  50 GO TO 20

O arredondamento de 12.3
  da' 12

O arredondamento de 12.75
  da' 13

O arredondamento de 0.6
  da' 1

O arredondamento de 0.3
  da' 0

O arredondamento de 0.5
  da' 1

O arredondamento de .099
  da' 0

O arredondamento de 0.9
  da' 1
```

Exprimir números com N casas decimais

Esta rotina permite-lhe exprimir um número com N casas decimais:

```
   5 REM EXPRIMIR UM NUMERO COM
N         CASAS DECIMAIS
  10 INPUT "INDIQUE UM NUMERO: "
;X
  20 INPUT "INDIQUE NUMERO DE CA
SAS DECIMAIS";N
  30 PRINT X;" com ";N;" casas",
"  decimais e' ";
  40 PRINT INT (X*10↑N+.5)/10↑N

24.689751 com 4 casas
  decimais e' 24.6898

.00034 com 4 casas
  decimais e' .0003

12399.99 com 0 casas
  decimais e' 12400
```

Geração de números primos

Este programa produzirá tantos números primos quantos o leitor desejar:

```
   5 REM NUMEROS PRIMOS
  10 LET X=1: LET Y=2: LET D=3
  20 INPUT "QUANTOS NUMEROS PRIM
OS DESEJA?  ";A
  40 PRINT "Primeiros ";A;" nume
ros primos:"''1,2,3,
  60 FOR B=1 TO A-3
  70 LET D=D+Y
  80 LET C=Y+X
  90 LET E=INT (D/C)
 100 LET F=D-E*C
 110 IF NOT F THEN GO TO 70
 120 IF C>=E THEN GO TO 150
 130 LET C=C+Y
 140 GO TO 90
 160 PRINT D,
 170 NEXT B
```


O programa que se segue faz praticamente a mesma coisa, mas utilizando de uma forma muito mais viva a imagem produzida pelo computador. Primeiramente imprime os números entre 1 e 100 da forma seguinte:

```
1  2  3  4  5  6  7  8  9  10
11 12 13 14 15 16 17 18 19 20
21 22 23 24 25 26 27 28 29 30
31 32 33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48 49 50
51 52 53 54 55 56 57 58 59 60
61 62 63 64 65 66 67 68 69 70
71 72 73 74 75 76 77 78 79 80
81 82 83 84 85 86 87 88 89 90
91 92 93 94 95 96 97 98 99 100
```

Em seguida considera novamente todos os números, verificando se cada um deles é ou não um número primo:

```
1  2  3     5     7
11    13          17    19
      23                29
31                37
41 ▒ 43 44 45 46 47 48 49 50
51 52 53 54 55 56 57 58 59 60
61 62 63 64 65 66 67 68 69 70
71 72 73 74 75 76 77 78 79 80
81 82 83 84 85 86 87 88 89 90
91 92 93 94 95 96 97 98 99 100
```

Todos os números não-primos são apagados, deixando no final, esta linha dos números primos entre 1 e 100:

```
1  2  3     5     7

11    13          17    19
      23                29

31                37

41    43          47

      53                59

61                67

71    73                79
      83                89

                  97
```

Finalmente, eis o programa que foi usado para obter este efeito:

```
    5 REM NUMEROS PRIMOS, USANDO
      PENEIRA DE ERATOSTENES
   10 FOR R=0 TO 9
   20 FOR N=1 TO 10
   30 PRINT AT 2*R+3,3*N-1;10*R+N
   40 NEXT N
   50 NEXT R
   60 FOR R=0 TO 9
   70 FOR N=1 TO 10
   75 LET P=10*R+N: LET X=2*R+3:
LET Y=3*N-1
   80 FOR M=2 TO 7
  100 IF P/M=INT (P/M) AND P<>2 A
ND P<>3 AND P<>5 AND P<>7 THEN P
RINT AT X,Y; INVERSE 1;P: PAUSE
10: PRINT AT X,Y; INVERSE 0;AT X
,Y;"  "
  110 NEXT M
  120 NEXT N
  130 NEXT R
```


Testes de matemáticas

Encontramos agora dois programas completos, contendo instruções suficientes nas respectivas listagens, que produzem testes de matemáticas. O primeiro, escrito por Jim Walsh, permite-nos especificar o número de perguntas a que queremos responder, produz estas e verifica as respostas dadas, atribuindo ao utilizador uma pontuação percentual no fim. O programa usa de uma forma bastante eficaz instruções como BRIGHT a fim de dar maior vivacidade à apresentação visual:

```
    5 REM              **Decimais
                     por Jim Walsh
   10 CLS : PRINT AT 11,4;"DECIMA
IS, por Jim Walsh.";''''
 Outubro 192": PRINT AT 20,2;"Ca
rregue numa tecla qualquer": PAU
SE 300
   15 GO SUB 1000
   50 LET l=10000: LET m=100: LET
 o=1000: LET go=go+1
   51 PAUSE 200: LET nova=INT (RN
D*4)+1: IF nova=2 THEN GO SUB 30
0
   52 IF nova=1 THEN GO SUB 100
   53 IF nova=3 THEN GO SUB 400
   55 IF nova=4 THEN GO SUB 500
   59 IF go=ultima THEN GO TO 200
0
   60 FOR n=1 TO 100: NEXT n: CLS
  : PRINT AT 10,5;INT (per/go*100
);"% certo!!";AT 15,3;"Depois de
";go;" somas.";AT 20,3;"Restam
";ultima-go: PAUSE 100: GO TO 50
   99 REM ********SOMA***********
  100 LET rnd=INT (RND*5)+2
  105 DIM x(rnd): DIM y(rnd): DIM
 z(rnd)
  110 LET total=0: FOR n=1 TO rnd
: GO SUB 220: LET x(n)=tot: LET
total=total+tot: NEXT n: LET z$=
STR$ total
  115 CLS : PRINT AT 0,27;"No.";g
o: PRINT AT 0,5;"Soma de Decimai
s": PRINT OVER 1;AT 0,5;"_______
_______": PRINT AT 3,3;"Some o
 seguinte com 3 casas": PRINT AT
 5,0;" ";: FOR n=1 TO rnd: PRINT
 x(n);: IF n<>rnd THEN PRINT " +
 ";
```

```
 120 NEXT n: LET v$=STR$ total:
PRINT " = ??": INPUT "Escreva re
sposta: "; LINE i$
 125 IF i$=v$ THEN GO TO 180
 129 REM errado
 130 CLS : PRINT AT 11,0; BRIGHT
 1;"[redacted]
[redacted]": FOR k=1 TO 300: NEXT k:
CLS : PRINT AT 5,0;" "
 135 FOR n=1 TO rnd: LET p$=STR$
 INT x(n): PRINT TAB 10-LEN p$;x
(n): PAUSE 30: NEXT n: LET p$=ST
R$ INT total: PRINT TAB 9-LEN p$
;" ";: FOR n=1 TO LEN z$: PRINT
"_";: NEXT n: PAUSE 30: PRINT TA
B 10-LEN p$;total: PRINT TAB 9-L
EN p$;" ";: FOR n=1 TO LEN z$: P
RINT OVER 1;"_";: NEXT n: PRINT
"
 140 FOR k=1 TO 500: NEXT k: PRI
NT AT 19,27;"O.K.": PAUSE 0: PRI
NT AT 21,3;"Um momento!": RETURN

 179 REM certo
 180 LET per=per+1: CLS : PRINT
AT 11,4;"Acertou!!": PAUSE 0: PR
INT AT 21,3;"Um momento!": RETUR
N
 219 REM ***numero aleatorio****
 220 LET a=INT (RND*l)+m
 225 LET b=INT (RND*o)+m: LET a$
=STR$ b: LET dec=b/10↑LEN a$: LE
T tot=a+dec
 226 IF b=10 OR b=100 OR b=1000
THEN GO TO 225
 230 RETURN
 299 REM *****subtraccao******
 300 DIM x(2): FOR n=1 TO 2: GO
SUB 220: LET x(n)=tot: NEXT n: I
F x(1)<=x(2) THEN GO TO 300
 302 LET total=x(1)-x(2): IF tot
al<100 THEN GO TO 300
 305 CLS : PRINT AT 0,27;"No.";g
o: PRINT AT 0,0;"   Subtraccao D
ecimal";AT 0,3; OVER 1;"_________
_____"
 310 PRINT AT 11,3;x(1);" - ";x(
2);" = ??"
 315 INPUT "Escreva resposta: ";
 LINE i$
 320 IF i$=STR$ total THEN GO TO
```



```
 380
 325 CLS : PRINT "ERRADO!!!"'''''
  Vou mostrar-lhe a resposta": P
AUSE 250: LET z$=STR$ INT x(1):
LET y$=STR$ INT x(2): LET w$=STR
$ INT total
 330 PRINT '''';TAB 10-INT LEN z$
;x(1)
 331 PAUSE 30
 332 PRINT TAB 10-INT LEN y$;x(2
);" -": PAUSE 30
 334 PRINT TAB 9-INT LEN w$;" ";
: LET v$=STR$ total: FOR n=1 TO
LEN v$: PRINT "_";: NEXT n: PRIN
T " "
 336 PRINT TAB 10-INT LEN w$;tot
al: PRINT TAB 9-INT LEN w$;" ";:
 FOR n=1 TO LEN v$: PRINT "_";:
NEXT n: PRINT " "
 340 FOR k=1 TO 400: NEXT k: PRI
NT AT 20,25;"O.K.": PAUSE 0: PRI
NT AT 21,3;"Um momento!"
 345 RETURN
 380 LET per=per+1: CLS : PRINT
AT 11,0;"Optimo!"''''"ACERTOU!!":
 PAUSE 0: PRINT AT 21,3;"Um mome
nto!": RETURN
 399 REM *****multiplicacao*****
 400 DIM i(2): DIM x(2): DIM d(2
): LET len=0: LET l=100: LET m=1
: LET o=100
 405 FOR n=1 TO 2: GO SUB 220: L
ET i(n)=a: LET x(n)=tot: LET d(n
)=VAL a$: LET len=len+LEN a$: NE
XT n
 410 LET total=x(1)*x(2)
 415 CLS : PRINT AT 0,27;"No.";g
o: PRINT AT 0,9;"MULTIPLICACAO":
 PRINT AT 0,9; OVER 1;"_________
_____"
 420 PRINT AT 5,0;"Quanto e'-"''
'TAB 5;x(1);" X ";x(2);" ?"''''"E
screva a resposta.": INPUT LINE
s$
 425 IF s$=STR$ total THEN GO TO
 490
 427 LET u$="Vou mostrar-lhe a r
esposta": PRINT '''; BRIGHT 1;"[redacted]
[redacted]": PAUSE 100: PRINT '''';
: FOR n=1 TO LEN u$: PRINT u$(n
TO n);: PAUSE 10: NEXT n: PAUSE
100: CLS
```

```
 430 PRINT i(1);".";: BRIGHT 1;d(
1): PRINT i(2);".";: BRIGHT 1;d(2
);: BRIGHT 0;" x": FOR n=1 TO 6:
PRINT "_";: NEXT n: PRINT " ": P
AUSE 100: PRINT AT 2,10;"=": PAU
SE 50: PRINT AT 2,12;len;" casas
 decimais.": PAUSE 200: PRINT AT
 2,12;"
 432 LET m$=STR$ i(1)+STR$ d(1):
 LET n$=STR$ i(2)+STR$ d(2): PRI
NT TAB 20-LEN m$;m$: PRINT TAB 2
0-LEN n$;n$;" x": PRINT TAB 19-L
EN n$;" ";: FOR n=1 TO LEN n$: P
AUSE 5: PRINT " ";: NEXT n: LET
p=0: PRINT " ";: FOR n=LEN n$ TO
 1 STEP -1: PAUSE 20: LET ans=VAL
 m$*VAL n$(n TO n)*10↑p: LET b$=
STR$ ans: PRINT TAB 20-LEN b$;b$
: LET p=p+1: NEXT n
 435 LET ans=VAL m$*VAL n$: LET
b$=STR$ ans: PRINT TAB 19-LEN b$
;" ";: FOR n=1 TO LEN b$: PAUSE
5: PRINT " ";: NEXT n: PRINT "
: PRINT TAB 20-LEN b$;b$: PRINT
TAB 19-LEN b$;" ";: FOR n=1 TO L
EN b$: PAUSE 5: PRINT "_";: NEXT
 n: PRINT " "
 440 PAUSE 100: PRINT AT 21,0;"
=  ";total;" resp.": FOR k=1 TO
 250: NEXT k: PRINT AT 21,28;"O.
K.": PAUSE 0: PRINT AT 18,24;"Um
": PRINT AT 19,24;"momento!": RE
TURN
 490 LET per=per+1: PRINT '"Opti
mo - [illegible]": PAUSE 0: PRINT
'"Um momento!": RETURN
 499 REM ******divisao**********
 500 REM
 505 GO SUB 220: LET l1=LEN a$:
LET b$=STR$ tot
 510 LET l=100: LET m=1: LET o=1
00: GO SUB 220: LET l2=LEN a$: L
ET c$=STR$ tot: LET d$=STR$ (VAL
 b$/VAL c$)
 515 LET d=VAL d$: LET d1=INT (d
): LET d2=d-d1: LET e$=STR$ d2
 520 LET d3=VAL e$(6 TO 6): LET
d4=VAL e$(5 TO 5): IF d3>4 THEN
LET d4=d4+1
 525 LET e$(5 TO 5)=STR$ d4: LET
 e$=e$(1 TO 5): LET d1=d1+VAL e$
: LET f$=STR$ d1
```



```
 530 CLS : PRINT AT 0,0;"   DIVI
SAO";AT 0,0; OVER 1;"        "
;AT 0,27;"No.";go
 534 REM na linha seguinte, A e'
       caracter grafico
 535 PRINT AT 4,3;"Quanto e' ";b
$;" A ";c$;" ?";AT 15,0;"Escreva
 a resposta com 3 casas  decimai
s": INPUT LINE i$
 540 IF i$=f$ THEN GO TO 490
 545 CLS : PRINT "[illegible]": PAUS
E 100: PRINT AT 12,0;"Vou mostra
r-lhe como deve fazer.": FOR n=1
 TO 200: NEXT n
 549 REM na linha seguinte, B e
       C sao caracteres
       graficos
 550 CLS : PRINT AT 0,0;c$;"┌";b
$;: PAUSE 15: PRINT AT 0,LEN c$;
 OVER 1;" ";: FOR n=1 TO LEN b$:
 PAUSE 15: PRINT OVER 1;"─";: NE
XT n: PRINT "  = "
 551 IF l1>l2 THEN LET l3=l1
 552 IF l2>l1 THEN LET l3=l2
 553 IF l2=l1 THEN LET l3=l1
 555 LET x=VAL c$*10↑l3: LET g$=
STR$ x: LET x=VAL b$*10↑l3: LET
h$=STR$ x
 558 PAUSE 100
 559 REM na linha seguinte, B e
       C sao caracteres
       graficos
 560 PRINT AT 0,LEN b$+LEN c$+4;
g$;"B";h$;AT 0,LEN b$+LEN c$+LEN
 g$+4; OVER 1;" ";: FOR n=1 TO L
EN h$: PAUSE 15: PRINT OVER 1;"C
";: NEXT n: PRINT "  ="
 565 LET d=INT VAL d$: LET d1=VA
L d$=d: LET t$=STR$ d1: LET d$=S
TR$ d+"."+t$(3 TO 6)
 570 LET t=INT VAL d$: LET t$=ST
R$ t: PRINT AT 3,0;TAB 18-LEN t$
;d$
 580 PRINT AT 4,0;TAB 18-LEN h$;
h$
 584 REM nas linhas seguintes, A
       e B sao caracteres
       graficos
 585 PRINT AT 4,0;TAB 17-LEN h$;
"B";
 590 FOR n=1 TO LEN h$+LEN d$: P
AUSE 10: PRINT OVER 1;"C";: NEXT
```

```
 n
 595 PRINT AT 4,0;TAB 17-LEN h$-
LEN 9$;9$
 600 PRINT AT 7,10;"=";AT 10,14;
" ";: FOR n=1 TO LEN f$: PRINT f
$(n TO n);: PAUSE 30: NEXT n: PR
INT " resp.": PAUSE 300: PRINT
AT 21,27;"O.K."
 605 PAUSE 0: RETURN
1000 DATA 24,24,0,255,255,0,24,2
4
1001 DATA 31,16,8,8,8,8,8,16
1002 DATA 255,0,0,0,0,0,0,0
1004 CLEAR 32599
1005 RESTORE : FOR n=0 TO 23: RE
AD a: POKE 32600+n,a: NEXT n: LE
T per=0: LET go=0
1010 CLS : PRINT AT 5,2;"Quantas
 somas deseja fazer?";AT 21,2;"I
ndique numero:": INPUT LINE i$:
FOR n=1 TO LEN i$: LET a=CODE i$
(n TO n): IF a<48 OR a>57 THEN G
O TO 1010
1015 NEXT n: LET ultima=VAL i$:
CLS : PRINT "Um momento!": RETUR
N
2000 CLS : PRINT TAB 8;"Acabou..
.";AT 5,3;"Em ";ultima;" somas o
bteve",INT (per/ultima*100);" %"
: PAUSE 200: PRINT AT 21,25;"O.K
.": PAUSE 0: RUN
```

O programa que se segue, da autoria de Gwyn Dewey, assemelha-se mais a um jogo, sendo no entanto bastante valioso para o desenvolvimento da habilidade no campo da matemática. O jogo foi concebido para quatro jogadores. São feitas dez perguntas a cada um deles, e o vencedor será aquele que der mais respostas certas. O programa emprega os nomes dos jogadores, e imprime os resultados dos testes de uma forma particularmente eficaz.

O leitor verificará na linha 506 que existem dez tipos possíveis de perguntas neste programa:

```
  1 REM © 24/7/82 G.Dewey Maths
  2 LET d$="Matematicas"
  3 GO SUB 8000
  4 CLS
  5 DIM a(4)
  6 DIM c$(4,10)
  7 DIM a$(4,15)
 10 PRINT "Matematicas"
 20 PRINT '"para 4 jogadores"
 30 PRINT '"E' necessario respo
nder a 10   perguntas, sendo ap
resentada umatabela de respostas
 certas"
 40 PRINT '"Cada pessoa realiza
 um teste ateTodos terminarem, s
endo depois  determinado quem ve
nceu"
 50 PRINT ' FLASH 1;"Carregue e
m 's' para comecar"
 60 IF INKEY$="" THEN GO TO 60
 70 CLS
 80 PRINT "Indicar nomes"
 90 FOR i=1 TO 4
100 INPUT "Jogador ";(i);"? ";
LINE a$(i)
110 NEXT i
130 CLS
135 IF c$(1,10)<>" " AND c$(2,1
0)<>" " AND c$(3,10)<>" " AND c$
(4,10)<>" " THEN GO TO 2000
140 PRINT "Matematicas"''"Menu"
145 LET m=96+i*16
150 PRINT '"1=Ficha de trabalho
"
160 PRINT "2=Percentagem"
170 PRINT "3=Grafico"
175 PRINT "4=Desistir"
180 INPUT "Opcao? "; LINE e$: I
F e$<"1" OR e$>"4" THEN GO TO 18
0
190 CLS
191 IF e$="4" THEN GO TO 2000
200 GO TO VAL e$*500
500 INPUT "Jogador (1/4)? "; LI
NE s$: IF s$<"1" OR s$>"4" THEN
GO TO 500
505 LET f=VAL s$
506 PRINT "0=soma facil"'"1=som
a media"'"2=soma dificil"'"3=sub
traccao facil"'"4=subtraccao med
ia"'"5=subtraccao dificil"
507 PRINT "6=multiplicacao faci
l"'"7=multiplicacao dificil"'"8=
divisao facil  }arredondar"'"9=
divisao dificil }para inteiro"'T
AB 19;"mais proximo"
510 INPUT "Ficha de trabalho (0
-9)? "; LINE g$: IF g$="d" THEN
GO TO 130
511 IF LEN g$>1 THEN GO TO 510
```

```
512 IF g$<"0" OR g$>"9" THEN GO
 TO 510
 515 FOR i=1 TO 10: IF c$(f,i)=p
$ THEN GO TO 130
 515 NEXT i
 520 LET b$="+++----*$//"
 530 FOR i=1 TO 10: IF c$(f,i)<>
" " THEN NEXT i
 540 LET c$(f,i)=g$
 570 LET z=VAL g$+1
 580 IF z=1 OR z=4 OR z=7 THEN L
ET h=1
 590 IF z=2 OR z=5 OR z=8 OR z=9
 THEN LET h=2
 600 IF z=3 OR z=6 OR z=10 THEN
LET h=3
 610 CLS
 620 PRINT "Ficha de trabalho ";
VAL g$
 630 PRINT ' FLASH 1;"Carregue n
uma tecla"
 640 IF INKEY$="" THEN GO TO 640
 641 LET j=0
 645 FOR i=1 TO 10
 650 CLS
 655 PRINT "Pergunta ";i
 656 PRINT j;" de ";i-1
 657 PRINT "Jogador: ";a$(f)
 660 LET a=INT (RND*10^h)
 670 LET b=INT (RND*10^h)
 675 IF a<b THEN IF z=4 OR z=5 O
R z=6 THEN LET y=a: LET a=b: LET
 b=y
 680 IF z=9 OR z=10 THEN LET b=I
NT (RND*10^(h-1))
 681 IF b=0 THEN IF z=9 OR z=10
THEN GO TO 680
 690 LET z$=STR$ a+b$(VAL g$+1)+
STR$ b
 695 LET z$=STR$ (INT ((VAL z$)+
.5))
 696 GO TO 700
 697 PRINT AT 21,0;"Escreva um n
umero em resposta"
 700 INPUT "Quanto e' ";(a);(b$(
VAL g$+1));(b);"? "; LINE x$
 702 IF x$="" THEN GO TO 700
 705 FOR n=1 TO LEN x$: IF x$(n)
<"0" OR x$(n)>"9" THEN GO TO 697
 706 NEXT n
 707 PRINT AT 21,0;"
```



```
 710 LET k=VAL z$
 720 IF VAL x$=k THEN PRINT AT 2
1,0;"Certo.": LET j=j+1
 730 IF VAL x$<>k THEN PRINT AT
21,0;"Errado. Resposta certa: ";
k
 731 FOR l=1 TO 250
 732 NEXT l
 740 NEXT i
 750 LET a(f)=a(f)+j
 780 GO TO 130
1010 PRINT "------------"
1020 FOR i=1 TO 4
1030 PRINT a$(i);" ";a(i);"%"
1040 NEXT i
1050 PRINT ' FLASH 1;"Carregue e
m d para desistir"
1060 IF INKEY$<>"d" AND INKEY$<>
"D" THEN GO TO 1060
1070 GO TO 130
1500 FOR i=1 TO 4
1510 PRINT a$(i);" fichas: ";c$(
i)
1520 PRINT PAPER 2;"
                "
1530 NEXT i
1540 FOR i=1 TO 4
1545 LET m=176-i*16
1546 LET v=a(i)
1550 IF a(i)=0 THEN NEXT i
1560 FOR j=0 TO v*2-1
1565 FOR o=m TO m+7
1570 PLOT INK 4;j,o
1580 NEXT o
1590 NEXT j
1600 NEXT i
1610 PRINT AT 21,0; FLASH 1;"Car
regue em d para desistir"
1620 IF INKEY$<>"d" AND INKEY$<>
"D" THEN GO TO 1620
1630 GO TO 130
2000 CLS : PRINT "Terminado"
2010 FOR i=1 TO 4
2020 IF a(i)>=a(1) AND a(i)>=a(2
) AND a(i)>=a(3) AND a(i)>=a(4)
THEN PRINT "Vencedor: ";a$(i)
2030 NEXT i
2040 GO TO 9000
8000 BORDER 7: PAPER 7: INK 0: C
LS
8001 FOR g=1 TO LEN d$
8005 LET f=USR "a"
```

```
8010 LET e=((CODE d$(g))-32)*8+1
5516
8020 FOR y=e TO e+7
8030 FOR z=1 TO 2
8040 POKE f,PEEK y
8050 LET f=f+1
8060 NEXT z
8070 NEXT y
8080 PRINT AT 9,g;"^"
8090 PRINT AT 10,g;"="
8100 NEXT g
8200 PAUSE 300
8400 RETURN
9910 INPUT "Quer passar a outro
programa    (s/n)? ";z$
9920 IF z$="n" THEN RUN
9925 PRINT "Ligue o gravador"
9930 LOAD ""
```

Testes de matemáticas: correlação/regressão

Finalmente, neste capítulo, encontramos um último programa gerador de testes matemáticos, escrito por Paul Toland. Se bem que o leitor talvez não necessite de um programa igual a este nas suas aulas, é incluído aqui por tratar-se de um óptimo exemplo de como é possível converter um procedimento matemático bem definido num programa. A regressão é o processo de definição da melhor linha recta que passa por vários pontos de um gráfico. Esta linha recta apresentará uma direcção média entre todos os pontos. Se a linha for estendida para além dos dados conhecidos, permitirá realizar uma previsão dos valores ainda desconhecidos.

Conforme os dados fornecidos ao programa, certas linhas de regressão constituirão um reflexo mais rigoroso dos valores originais do que outras. Este rigor é medido pelos coeficientes de correlação.

Quando se executa o programa, é-nos perguntado se desejamos enviar os resultados para a impressora durante o programa. Se respondermos "s" (sim), teremos a oportunidade de copiar (COPY) a imagem do visor em todas as passagens importantes do programa. Se escrevermos "n" (não), o programa não voltará a perguntar-nos se queremos imprimir a página apresentada.

Em seguida o leitor deve indicar os dados. Isto será feito em duas fases, indicando primeiro os valores de X e depois os valores de Y. Os valores de X são quantidades independentes. Por exemplo, para os totais das vendas anuais os dados relativos a X podem ser 1984, 1985, 1987, etc. Mas é possível utilizar outros valores equivalentes

sem afectar os resultados, como por exemplo os números 1, 2, 4. Escolhe-se o valor de partida (1 neste caso) e adoptam-se depois incrementos idênticos aos que figuram nos dados originais.

Quando se introduzem os dados para os cálculos de séries temporais (este tipo de cálculos serve para traduzir as flutuações de dados causadas por factores sasonais; só são úteis para aplicações em que os dados obedecem a um padrão repetido) é essencial usar esta notação alternativa para os valores de X. Por exemplo, para o caso de vendas trimestrais, viria:

```
Ano Trimestre
'84 1 deve escrever-se 1
    2 deve escrever-se 2
    3 deve escrever-se 3
    4 deve escrever-se 4
'85 1 deve escrever-se 5
    2 deve escrever-se 6 ... etc.
```

Note-se que nos cálculos de séries temporais não pode haver omissão de estações. Por isso, no exemplo anterior os trimestres não podem ser 1, 2, 4, embora a omissão integral de um período não afecte muito desfavoravelmente os resultados.

Depois de introduzidos todos os valores de X, escreva 9999 para terminar. A seguir introduzem-se os valores de Y na sua forma original. Estes dados são dependentes e podem reportar-se, por exemplo, ao volume de vendas ou aos totais de produção. No programa considerámos um máximo de 100 grupos de números.

Depois de introduzidos todos os dados, o programa imprime os resultados de correlação/regressão. Destes, os importantes são o coeficiente de Pearson, o coeficiente de determinação e a equação da linha de regressão.

Terá seguidamente a possibilidade de interpolar ou extrapolar com base na linha de regressão. Escreve um valor de X na mesma notação que foi originalmente adoptada ao introduzir os dados relativos a X, e obtém o valor de Y correspondente ao ponto sobre a linha. Este método pode ser usado para fazer previsões. Por exemplo, se tiver introduzido os valores das vendas relativos aos anos de 1 a 10, escrevendo 11 obterá uma previsão para os valores das vendas do ano seguinte.

O programa traça a seguir um gráfico com os dados originais e a linha de regressão sobreposta.

Depois de desenhado o gráfico, é-lhe perguntado se quer entrar na parte do programa relativa às séries temporais. Isso só faz senti-

do se os dados forem da natureza sazonal. Se decidir afirmativamente terá de escrever a duração do período, ou seja, o número de valores que formam um período (como por exemplo 4 no caso de vendas trimestrais).

A seguir declara qual o tipo de modelo que pretende usar, multiplicativo ou aditivo (M ou A). Feito isto, o programa traça o gráfico das médias dos dados. Como terá oportunidade de verificar, este acompanha as oscilações dos dados e é, portanto, mais flexível. Um valor extravagante próximo do início dos dados não afectará o final da linha.

São depois impressos os factores sazonais, seguidos dos dados originais sob uma forma ajustada. Por fim, o programa imprime os números relativos às previsões para os dois períodos seguintes, antes de lhe perguntar se deseja uma nova execução do programa.

Posto isto, segue-se a listagem (os esclarecimentos sobre as noções matemáticas envolvidas no programa são dados após a listagem, para o caso de querer refrescar a memória neste domínio):

```
   2 REM © P. Toland
   3 REM
   4 REM usar so maiusculas
   5 DEF FN Z$(X$)=(CHR$ (CODE (
X$)-32*(CODE X$>90)))
   6 REM ↑ SO MAIUSCULAS
  10 PRINT TAB 5;"CORRELACAO/PRO
GRESSAO                SERIES DE T
EMPO"
  20 INPUT "QUER USAR IMPRESSORA
? ";A$
  30 LET PCON=0
  40 IF FN Z$(A$)="S" THEN LET P
CON=1
  50 GO TO 100
  60 IF NOT PCON THEN INPUT "Car
regue em ENTER";A$: RETURN
  70 INPUT "IMPRIMIR ESTA PAGINA
? ";A$
  80 IF FN Z$(A$)="S" THEN COPY
  LPRINT "----------------------
----------"
  90 RETURN
 100 DIM X(100): DIM Y(100)
 110 PRINT "Escreva valores de X
 por ordem -      termine com 999
9"
 120 FOR I=1 TO 100
 130 INPUT X
 140 IF X=9999 THEN GO TO 180
 150 PRINT X: LET X(I)=X
 160 NEXT I
 170 PRINT "100 E' O MAXIMO PERM
ITIDO"
 180 LET I=I-1: CLS
 190 PRINT "     ";I;" VALORES
"'"Escreva valores de Y por orde
m "
 195 LET SM=1E38: LET LA=4*1E-39
 200 FOR J=1 TO I
 205 PRINT X(J)
 210 INPUT Y(J): PRINT TAB 9,Y(J
)
 214 IF Y(J)>LA THEN LET LA=Y(J)
 216 IF Y(J)<SM THEN LET SM=Y(J)
 220 IF PEEK 23689=2 THEN GO SUB
 60: CLS
 225 NEXT J
 230 GO SUB 60
 240 LET SX=0: LET SY=0
 250 LET SXQ=0: LET SYQ=0
 260 LET SXY=0
 270 FOR J=1 TO I
 280 LET SX=SX+X(J)
 290 LET SY=SY+Y(J)
 300 LET SXQ=SXQ+X(J)*X(J)
 310 LET SYQ=SYQ+Y(J)*Y(J)
 320 LET SXY=SXY+X(J)*Y(J)
 330 NEXT J
 340 LET R=(I*SXY-SX*SY)/SQR ((I
*SXQ-SX*SX)*(I*SYQ-SY*SY))
 345 CLS
 350 PRINT "SOMA DE X: ";SX'"SOM
A DE Y: ";SY'"SOMA DE X Q.: ";SX
Q'"SOMA DE Y Q.: ";SYQ'"SOMA DE
XY: ";SXY
 360 PRINT '"COEFIC. DE CORRELAC
AO PEARSON          ";R
 370 PRINT ''"COEFICIENTE DE DET
ERMINACAO           ";R*R
 380 LET B=(I*SXY-SX*SY)/(I*SXQ-
SX*SX)
 390 LET A=SY/I-SX/I*B
 400 PRINT ''"EQUACAO DE REGRESS
AO LINEAR:"''"Y=";A;("+" AND B>=
0);B;" X"
 410 GO SUB 60
 430 CLS
 440 PRINT "INTERPOLACAO/EXTRAPO
LACAO                9999 PARA TER
MINAR"'"     X          Y"
 450 INPUT "INDIQUE VALOR DE X "
```

```
;X
 460 IF X=9999 THEN GO TO 495
 470 LET Y=A+B*X
 480 PRINT X;TAB 8;Y
 485 IF PEEK 23689=2 THEN GO SUB
 60: CLS
 490 GO TO 450
 495 GO SUB 60
 500 LET SL=A+B*X(1): LET EL=A+B
*X(I)
 502 IF SL>LA THEN LET LA=SL
 504 IF EL>LA THEN LET LA=EL
 506 IF SL<SM THEN LET SM=SL
 508 IF EL<SM THEN LET SM=EL
 510 LET SCX=253/(X(I)-X(1))
 520 LET SCY=173/(LA-SM)
 525 CLS
 530 FOR J=1 TO I
 540 CIRCLE 1+(X(J)-X(1))*SCX,1+
(Y(J)-SM)*SCY,1
 550 NEXT J
 560 PLOT 0,(SL-SM)*SCY
 570 DRAW 255,(EL-SM)*SCY-(SL-SM
)*SCY
 580 GO SUB 60
 590 INPUT "CONTINUAR PARA SERIE
S DE TEMPO ?";A$
 600 IF FN Z$(A$)="N" THEN GO TO
 900
 610 INPUT "DURACAO DO PERIODO?
";PER
 620 IF PER<=0 OR PER<>INT PER T
HEN GO TO 610
 630 INPUT "MODELO MULTIPLICATIV
O OU ADITIVO(M/A)? ";M$
 640 LET M$=FN Z$(M$)
 650 IF M$<>"A" AND M$<>"M" THEN
 GO TO 630
 660 DIM T(PER): DIM A(I)
 670 LET IP=INT (I/PER)*PER
 680 LET FIM=IP-PER+1
 690 LET DIV=PER
 700 FOR J=1 TO FIM
 710 FOR K=J TO J+PER-1
 720 LET A(J)=A(J)+Y(K)
 730 NEXT K
 740 NEXT J
 750 LET ODD=(PER<>INT (PER/2)*2
)
 760 IF ODD THEN GO TO 820
 770 LET FIM=FIM-1
 780 FOR J=1 TO FIM
```



```
 790 LET A(J)=A(J)+A(J+1)
 800 NEXT J
 810 LET DIV=DIV*2
 830 FOR J=1 TO FIM
 840 LET A(J)=A(J)/DIV
 850 NEXT J
 860 LET STRT=INT ,(PER/2)+1
 870 LET X1=(X(STRT)-X(1))*SCX
 880 LET Y1=(A(1)-SM)*SCY
 890 PLOT X1,Y1
 900 FOR J=2 TO FIM
 910 LET X2=(X(STRT-1+J)-X(1))*S
CX
 920 LET Y2=(A(J)-SM)*SCY
 930 DRAW X2-X1,Y2-Y1
 940 LET X1=X2: LET Y1=Y2
 950 NEXT J
 960 GO SUB 60
 970 FOR J=1 TO FIM
 980 LET CP=STRT-1+J
 990 LET TP=(CP/PER-INT (CP/PER)
)*PER
 995 LET TP=TP+PER*(TP=0)
1000 LET VENDA=Y(CP)
1010 LET MA=A(J)
1020 LET T=VENDA-MA
1030 IF M$="M" THEN LET T=VENDA/
MA
1040 LET T(TP)=T(TP)+T
1050 NEXT J
1060 LET TOT=0
1070 IF M$="M" THEN LET TOT=FIM
1080 LET TTL=0
1090 FOR J=1 TO PER
1100 LET TTL=TTL+T(J)
1110 NEXT J
1120 LET ADJ=(TTL-TOT)/PER
1125 PRINT "OS ";PER;" FACTORES
SAZONAIS SAO:"
1130 FOR J=1 TO PER
1140 LET T(J)=(T(J)-ADJ)/(FIM/PE
R)
1150 PRINT J;"___";T(J)
1160 NEXT J
1170 PRINT '"VALORES AJUSTADOS S
AZONALMENTE:"'"X          Y
Y AJ
USTADO"
1175 LET TP=0
1180 FOR J=1 TO I
1190 LET TP=TP+1
1200 LET SY=Y(J)-T(TP)
1210 IF M$="M" THEN LET SY=Y(J)/
```

```
T(TP)
1220 PRINT X(J);TAB 6;Y(J);TAB 1
6;SY
1230 IF TP=PER THEN LET TP=0
1240 IF PEEK 23689=2 THEN GO SUB
 60: CLS
1250 NEXT J
1260 GO SUB 60
1270 CLS
1280 PRINT "PREVISAO PARA OS PRO
XIMOS          DOIS PERIODOS:
                 X   PERIODO PREVISAO
 Y"
1290 LET INCX=X(I)-X(I-1)
1300 LET TP=0
1310 LET INCY=(A(FIM)-A(FIM-2))/
2
1320 LET STX=X(I)+INCX
1330 LET STY=A(FIM)+INCY
1340 FOR J=FIM+STRT-1 TO I
1350 LET STY=STY+INCY
1360 NEXT J
1365 LET TP=TP+I-IP
1370 FOR J=STX TO STX+PER*2*INCX
-INCX STEP INCX
1380 LET TP=TP+1
1390 LET FY=STY+T(TP)
1400 IF M$="M" THEN LET FY=STY*T
(TP)
1410 PRINT J;TAB 6;TP;TAB 13;FY
1420 LET STY=STY+INCY
1430 IF TP=PER THEN LET TP=0
1440 NEXT J
1450 GO SUB 60
1460 PRINT ' FLASH 1;"          PRO
GRAMA TERMINADO"
1470 INPUT "DESEJA NOVA EXECUCAO
 (S/N)? ";A$
1480 IF FN Z$(A$)="S" THEN RUN
```

Como dissemos no início das notas introdutórias a esta secção, o processo chamado de regressão destina-se a traçar uma linha recta através de um conjunto de pontos num gráfico da forma que melhor traduza a sua posição relativa. Essa linha recta seguirá uma trajectória média entre todos os pontos. Prolongando a linha para além do final dos dados conhecidos, obtém-se uma previsão dos valores seguintes — ainda desconhecidos. Consoante os dados de partida, algumas linhas de regressão reflectem de forma mais rigorosa do que outras os valores originais. O grau de rigor do "ajustamento" de uma linha é dado pelos coeficientes de correlação.

O coeficiente de correlação de Pearson, representado pela letra R, dá o grau de correlação linear. Os valores deste coeficiente situam-se numa gama entre —1 e 1. Um valor próximo de —1 ou de 1 significa um elevado grau de correlação, um bom ajustamento. Os valores próximos de 0 significam um baixo grau de correlação, um mau ajustamento. Os valores positivos indicam uma linha em sentido crescente, e os valores negativos uma linha decrescente. Entre outras coisas, um elevado grau de correlação significa — naturalmente — que qualquer previsão baseada nos dados observados deverá ser razoavelmente rigorosa.

O coeficiente de determinação é muito idêntico ao de Pearson: é R ao quadrado. Situa-se numa gama de valores entre 0 e 1. Se $R = 0.8$, então $R\uparrow 2 = 0.64$. Isto significaria que 64% da variação nos valores de Y observados é explicada pelo modelo.

Naturalmente, quaisquer cálculos baseados apenas num pequeno conjunto de números não serão significativos. São necessários vários grupos de dados para se ter uma indicação de tendência geral.

Séries temporais

Os cálculos de séries temporais servem para traduzir as flutuações nos dados que são devidas a factores sazonais. Por isso, só são úteis em aplicações nas quais os dados obedeçam a um padrão repetido. Incluem-se neste caso valores de vendas (a cada ano corresponderão quatro valores) ou totais de produção diária (correspondendo a cinco, seis ou sete valores por semana). Neste último exemplo, o período seria a semana, e a sua duração seria de cinco (ou seis, ou sete).

A relação entre a dimensão de um determinado valor e a estação a que se refere pode ser calculada, e é designada por factor sazonal. A propósito, neste contexto, uma estação refere-se à parte do período a que o número corresponde, podendo portanto equivaler a um dia da semana ou a uma estação do ano.

Quando os dados são afectados por variações sazonais, os valores de um período não podem ser utilmente comparados entre si. Por exemplo, se vendemos brinquedos, esperamos provavelmente vender mais no Natal do que durante o Verão. Se num determinado ano tivermos um Natal de más vendas, e um Verão forte, a comparação dos dois valores não indicará isto dado que o valor no Natal será mesmo assim superior ao do Verão.

Se no entanto, pudermos eliminar o factor sazonal dos valores, será possível determinar qual a verdadeira situação. Numa expressão já vulgarizada, a modificação dos valores neste sentido será de-

signada por *ajustamento sazonal*. A previsão feita a partir de séries de tempo pode conduzir a resultados mais satisfatórios na medida em que os valores considerados sigam um mesmo padrão sazonal.

Note que existem dois tipos de modelo de série temporal: o multiplicativo e o aditivo. O primeiro é usado quando os desvios variam bastante de um período para outro. O modelo aditivo é usado quando ocorre uma variação sazonal relativamente constante.

Qualquer que seja a aplicação, os resultados de qualquer previsão feita usando a regressão linear ou a série de tempo não podem ser considerados isoladamente. As vendas podem diminuir devido ao aumento de preços, a produção pode aumentar devido à instalação de uma nova maquinaria automática. Estes acontecimentos afectarão obviamente os valores considerados, e como também é óbvio o computador não pode conhecê-los. É portanto, necessário avaliar as previsões realizadas pela máquina tendo em conta quaisquer outras informações de que se disponha e que não sejam reflectidas nos dados usados.

6

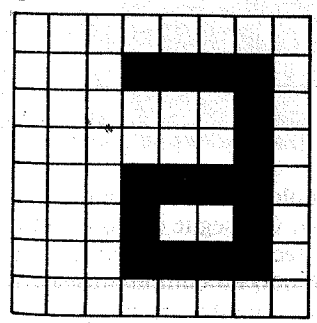
USO EFICAZ DOS GRÁFICOS DO SPECTRUM

PRIMEIRA PARTE

Uma das grandes vantagens do Spectrum relativamente a muitos outros computadores é a possibilidade de definir símbolos gráficos próprios. Isto tem um tremendo valor em aplicações educativas, quer se trate do ensino de uma linguagem, de química, de geografia ou música.

Os gráficos do Spectrum são formados por uma grelha de oito por oito "pixels", sendo alguns destes negros e outros brancos. Os padrões formados pelos pixels pretos e brancos podem corresponder por exemplo às letras, números e outros símbolos que observamos normalmente no visor. O Spectrum permite-nos construir os nossos próprios padrões, de modo a desenhar notas musicais, letras com acentos, ou quaisquer outros símbolos especiais de que necessitemos.

A informação é guardada sob a forma de uma série de zeros (pixels brancos) e uns (pixels negros). Para esclarecermos um pouco melhor esta questão, apresentamos na figura seguinte um exemplo desta rede de oito por oito pixels, na qual os pixels negros formam a letra "a". À direita podemos observar o padrão de bits um e zero que formam a letra:



Padrão de bits correspondente

```
0 0 0 0 0 0 0 0
0 0 0 1 1 1 1 0
0 0 0 0 0 0 1 0
0 0 0 0 0 0 1 0
0 0 0 1 1 1 1 0
0 0 0 1 0 0 1 0
0 0 0 1 1 1 1 0
0 0 0 0 0 0 0 0
```

Como o leitor pode verificar, cada um dos conjuntos de bits forma obviamente um número binário. O computador foi programado de tal modo que dando-lhe a conhecer oito números binários em que os 1 correspondem aos pontos que queremos a negro no caracter que estamos a criar, e os zeros a pontos em branco, podemos formar um caracter.

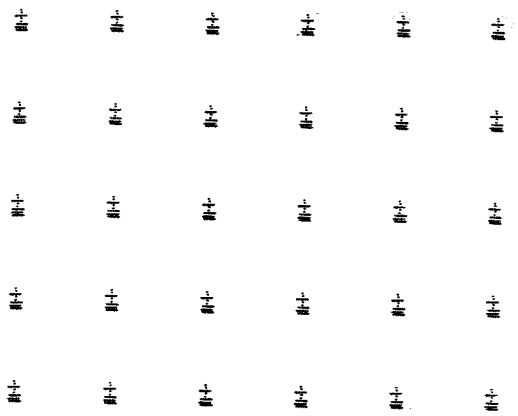
Como por outro lado cada linha de oito zeros ou uns constitui um número binário, é óbvio que o maior número que podemos obter é 11111111 ou 255 em decimal (128+64+32+16+8+4+2+1=255). Qualquer das combinações possíveis de zeros e uns corresponde a um número entre zero e 255. Fazendo um esboço sobre uma rede de oito por oito, e determinando depois o equivalente decimal para cada linha, podemos atribuir a forma criada a certas teclas do Spectrum (qualquer das teclas correspondentes às letras A a U; o caracter é acedido colocando o computador em modo gráfico antes de carregar na tecla relevante).

Cruz de igreja

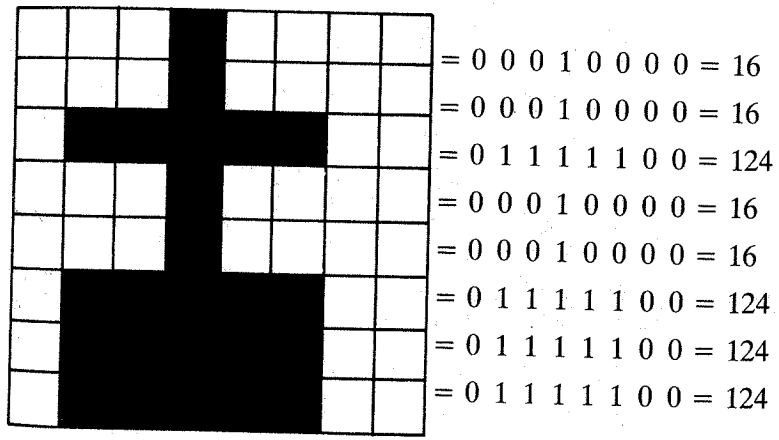
Define-se um caracter por nós concebido usando um programa simples como o que se segue, no qual os números decimais equivalentes aos binários que formam o padrão por nós pretendido (e nos apêndices poderá encontrar uma tabela de conversão de binário para decimal que facilitará esta tarefa) são colocados numa instrução DATA ao mesmo tempo que a letra que passará a designar o caracter em causa é colocada dentro de aspas depois da palavra-chave USR:

```
  5 REM IGREJA
 10 DATA 16,16,124,16,16,124,12
4,124
 20 FOR X=0 TO 7
 30 READ Z
 40 POKE USR "A"+X,Z
 50 NEXT X
```

Este programa cria uma pequeno símbolo (a cruz de uma igreja) que pode por exemplo utilizar num mapa. Consegue que o computador imprima o símbolo carregando na tecla "A" depois de colocar a máquina em modo gráfico, ou dizendo ao computador para imprimir CHR$ 144:



Vejamos agora o padrão de oito por oito pixels correspondente a este símbolo, apresentando à direita os números binários e os seus equivalentes decimais:

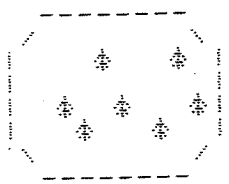


Floresta

É por vezes necessário imprimir um mesmo caracter por nós definido em dois ou mais locais diferentes. O programa que se segue de-

fine um caracter "árvore", utilizando-o em seguida, para criar um gráfico de maiores dimensões — indicando neste caso uma floresta:

```
Floresta de coniferas.
```



Vejamos agora o programa que produziu o efeito anterior, sendo as pequenas árvores da linha 150 obtidas colocando o computador em modo gráfico e carregando na tecla A:

```
  10 REM FLORESTA
  20 FOR X=0 TO 7
  30 READ Z: POKE USR "A"+X,Z
  40 NEXT X
 100 PAPER 5: INK 4: BORDER 1: C
LS
 105 PRINT INK 9;AT 3,5;"Floresta
 de coniferas."
 110 PRINT INK 9;AT 10,11;"-----
---";AT 11,10;"/";AT 11,19;"\"
 120 FOR x=1 TO 4
 130 PRINT INK 9;AT 11+x,9;"|";A
T 11+x,20;"|"
 135 NEXT x
 140 PRINT INK 9;AT 16,10;"\";AT
 16,19;"/"
 150 PRINT AT 12,14;"♣  ♣";AT 1
4,12;"♣  ♣  ♣";AT 15,13;"♣  ♣"
 160 PRINT INK 9;AT 17,11;"-----
---"
 170 DATA 16,16,56,84,146,56,84,
16
```

Estes gráficos podem ser incorporados num mapa simplificado, com uma letra identificadora ao lado, sendo pedido ao aluno que identifique cada caracter.

Química

O programa que se segue é um desenvolvimento da ideia anterior, mas aplicada a um campo diferente: a química. Neste caso, existem certos símbolos que só poderão ser obtidos usando os caracteres definíveis pelo utilizador. Como anteriormente, desenharam-se primeiro os caracteres numa rede de oito por oito pixels, e o número decimal resultante foi colocado em instruções DATA. Como é necessário definir vários caracteres diferentes, as instruções READ e DATA são colocadas num ciclo FOR/NEAT separado.

Os caracteres definidos são indicados na tabela que se segue, juntamente com as letras que lhes correspondem. Escolheram-se as teclas A a L, permitindo o uso de um pedaço de cartão a colocar sobre as teclas e no qual se podem identificar os símbolos, permitindo ao utilizador reconhecer imediatamente qual o símbolo atribuído a cada tecla.

TABELA 1x

Símbolo	Tecla	CHR$
+	a	144
—	b	145
2+	d	147
2–	e	148
3+	g	150
3–	h	151
2	j	153
3	k	154
4	l	155

Vejamos agora o programa que produz estes índices para uso em fórmulas químicas:

```
  10 REM formulas quimicas
  20 GO SUB 500
  25 INK 1: BORDER 1: PAPER 5: C
LS
  30 PRINT INK 9;"Passe ao modo
grafico e experi- mente as tecla
s A a L"
  40 STOP
 500 DATA "A",16,16,124,16,16,0,
0,0
```

```
 501 DATA "B",0,0,124,0,0,0,0,0
 502 DATA "D",34,82,23,34,114,0,
0,0
 503 DATA "E",32,80,23,32,112,0,
0,0
 504 DATA "G",114,18,55,18,114,0
,0,0
 505 DATA "H",112,16,55,16,112,0
,0,0
 506 DATA "J",0,0,0,32,80,16,32,
112
 507 DATA "K",0,0,0,112,16,48,16
,112
 508 DATA "L",0,0,0,16,48,80,120
,16
 510 FOR X=1 TO 9
 520 READ A$
 530 FOR A=0 TO 7
 540 READ B: POKE USR A$+A,B
 550 NEXT A
 560 NEXT X
 570 RETURN
```

Em seguida damos um exemplo do que é possível escrever depois de este programa ter sido executado. Note que o programa que cria os caracteres de nossa autoria pode ser eliminado depois de executado, dado que os novos caracteres gráficos se mantêm depois de executada a função NEW. Manter-se-ão até o computador ser desligado da alimentação:

```
COMPOSTO                    SIMBOLO

1-Sulfato de Cobre II       CuSO4

2-Sulfato de Ferro III      Fe2(SO4)3

3-Iao Aluminio              A(3+

REACCAO DE NEUTRALIZACAO:

H+(aq) + OH-(aq) -> H2O(l)
```

E se quiser repetir esta demonstração, eis o programa usado para a obter:

```
1000>PRINT ''''"COMPOSTO";TAB 23;
"SIMBOLO"
1010 PRINT ''"1-Sulfato de Cobre
II";TAB 23;"CuSO4"
1020 PRINT ''"2-Sulfato de Ferro
III";TAB 23;"Fe2(SO4)3"
1030 PRINT ''"3-Iao Aluminio";TAB
 23;"A(3+"
1040 PRINT ''''"REACCAO DE NEUTRA
LIZACAO:"
1050 PRINT ''"H+(aq) + OH-(aq) ->
 H2O(l)"
```


Pesos moleculares

O programa que se segue utiliza os símbolos já definidos pelo programa anterior. Guardam-se num quadro ("array") as fórmulas correspondentes a cinco substâncias. O valor do peso de uma mole de cada uma destas substâncias é guardado num outro quadro. Escolhe-se aleatoriamente uma das substâncias do quadro, e ao mesmo tempo um valor entre 0,5 e 2,5 como factor multiplicador dos pesos moleculares.

Ao estudante é apresentada a fórmula de uma dada substância, e o número de moles a considerar. É-lhe então pedido que indique a massa correcta em gramas.

Depois de ter executado o programa na sua forma actual, o leitor descobrirá provavelmente várias maneiras de o melhorar, como seja tornando-o mais "próximo" do aluno, perguntando-lhe o nome a usando-o; dando no final uma pontuação, e um eventual comentário dependendo do número armazenado; aumentando o número de substâncias; e imprimindo uma página de pesos atómicos.

Neste programa, usei os valores aproximados de alguns elementos:

Cálcio 40	Carbono 12	Hidrogénio 1
Ferro 56	Chumbo 207	Azoto 14
Oxigénio 16	Sódio 23	Enxofre 32

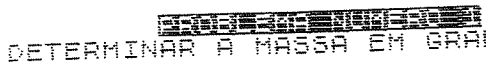
A fim de obtermos os índices há pouco definidos, incorporamos o programa anterior neste, como o leitor poderá verificar:

```
  10 REM CALCULO DE PESOS MOLECU
LARES
  15 LET SC=0
  20 GO SUB 500
  25 PAPER 5: INK 1: BORDER 1
 100 FOR J=1 TO 10
```

```
 105 LET X=INT (RND*5)+1
 110 LET Y=INT (RND*5)*.5+.5
 115 CLS
 120 PRINT INVERSE 1;AT 0,8;"PRO
BLEMA NUMERO ";J
 125 PRINT "DETERMINAR A MASSA E
M GRAMAS DE:"
 130 PRINT INVERSE 1;AT 4,0;"SUB
STANCIA          NO. DE MOLES"
 135 PRINT AT 7,3;A$(X);AT 7,22;
Y
 140 PRINT INK 9;AT 14,0;"ESCREV
A A RESPOSTA E CARREGUE EM"; INV
ERSE 1;"ENTER"
 150 LET RES=VAL (B$(X))*Y
 160 INPUT C$
 165 IF C$="" THEN GO TO 160
 170 IF CODE C$<49 OR CODE C$>58
 THEN GO TO 160
 180 IF VAL C$<RES-.05 OR VAL C$
>RES+.05 THEN GO TO 250
 200 CLS
 205 FOR X=1 TO 300
 210 PRINT FLASH 1;AT 10,5;"ACER
TOU!"
 220 NEXT X
 230 LET SC=SC+1
 235 NEXT J
 240 GO TO 700
 250 CLS
 260 PRINT INVERSE 1;AT 10,0;"ER
RADO!"
 265 PRINT '''"A RESPOSTA E' ";RE
S
 270 PAUSE 100
 280 GO TO 235
 500 DATA "A",16,16,124,16,16,0,
0,0
 501 DATA "B",0,0,124,0,0,0,0,0
 502 DATA "D",34,82,23,34,114,0,
0,0
 503 DATA "E",32,80,23,32,112,0,
0,0
 504 DATA "G",114,18,55,18,114,0
,0,0
 505 DATA "H",112,16,55,16,112,0
,0,0
 506 DATA "J",0,0,0,32,80,16,32,
112
 507 DATA "K",0,0,0,112,16,48,16
,112
 508 DATA "L",0,0,0,16,48,80,120
,16
 510 FOR X=1 TO 9
 520 READ A$
 530 FOR A=0 TO 7
 540 READ B: POKE USR A$+A,B
 550 NEXT A
 560 NEXT X
 600 DIM A$(5,12)
 605 DIM B$(5,3)
 610 DATA "CuSO4.5H2O","CaCO3","
FeIII(SO4)3","Pb(NO3)2","NaOH"
 615 FOR X=1 TO 5
 620 READ Z$: LET A$(X)=Z$
 625 NEXT X
 630 DATA "250","100","400","331
","40"
 635 FOR X=1 TO 5
 640 READ Z$: LET B$(X)=Z$
 645 NEXT X
 650 RETURN
 700 CLS
 705 PRINT AT 8,3;"PONTUOU ";SC;
" (MAX. 10)"
 710 PRINT '''"CARREGUE EM TECLA
PARA RECOMECAR"
 720 PAUSE 0
 730 RANDOMIZE
 740 GO TO 100
```



```
PROBLEMA NUMERO 1

DETERMINAR A MASSA EM GRAMAS DE:

SUBSTANCIA                NO. DE MOLES

   Pb(NO3)2                    0.5


ESCREVA A RESPOSTA E CARREGUE EM
ENTER
```



DETERMINAR A MASSA EM GRAMAS DE:

FEIII(SO4)3 1.5

ESCREVA A RESPOSTA E CARREGUE EM

DETERMINAR A MASSA EM GRAMAS DE:

CaCO3 0.5

ESCREVA A RESPOSTA E CARREGUE EM

PONTUOU 8 (MAX. 10)

CARREGUE EM TECLA PARA RECOMECAR

O leitor pode evidentemente aumentar o programa incorporando no quadro a$ um maior número de substâncias, acrescentadas no final da declaração DATA da linha 610. É obviamente necessário redimensionar o quadro a$ na linha 600. A massa de uma mole das novas substâncias deve ser indicada no quadro b$, através da declaração DATA da linha 630, e b$ deve ser redimensionado em função disso na linha 605.

Os dois ciclos FOR/NEXT deverão também ser modificados nas linhas 615 e 635. São estes os ciclos que lêem as declarações DATA, atribuindo valores aos quadros. A linha 105 deve também ser alterada, de tal modo que o número aleatório "X" seja aumentado a fim de ter em conta o maior número de substâncias químicas.

Tal como se encontra, o programa permite apenas usar 0,5 a 2,5 vezes a massa de uma mole, em passos de 0,5 moles. Para tornar o teste mais difícil é no entanto possível alterar o número Y na linha 110. Por exemplo, se forem necessários passos de 0,1, a linha deve ser alterada do seguinte modo:

```
Let Y = INT(RND*25)*0,1 + 0,1
```

As outras variáveis usadas neste programa são:

J — contador do número de problemas; recebe o valor 10 na linha 100, e pode ser modificado se assim o desejar.

RESPOSTA — Esta variável guarda a resposta correcta, e é criada na linha 150. Para evitar problemas com números que não são guardados rigorosamente no computador, a linha 180 permite um erro de 0,05 em qualquer dos sentidos.

C$ — É esta a resposta do aluno, aceite pela máquina na linha 160. A resposta é validada nas linhas 165 e 170, de tal modo que (1) o computador não aceitará apenas que o utilizador carregue na tecla ENTER sem indicar qualquer resposta, e (b) o primeiro elemento da resposta deve ser um número (a validação de entradas é discutida em pormenor no capítulo IX).

A última variável guarda o número de problemas resolvidos correctamente. Só é incrementada quando o aluno responde correctamente na linha 230.

7

USO EFICAZ DOS GRÁFICOS DO SPECTRUM

SEGUNDA PARTE

Neste segundo capítulo sobre gráficos, tentaremos responder a uma questão colocada por muitos professores: "Que se poderá fazer num microcomputador que não seja possível fazer num quadro?"

Com limitadas potencialidades gráficas, um computador monocromático raramente poderá competir com um bom professor. No entanto, já que dispomos do Spectrum, com a sua cor e os seus gráficos de alta resolução, existem modos de o usar com os quais o quadro negro nunca poderá competir. A área onde esta afirmação se torna mais obviamente verdadeira é aquela em que um visor móvel pode fornecer indicações de um modo muito mais eficaz do que uma imagem estática. O Spectrum pode mostrar em movimento uma vasta gama de objectos — desde um coração a bater até um motor de quatro tempos.

O Ciclo do azoto

O programa que se segue foi pensado para apresentar continuamente a mesma informação. Não temos qualquer controlo sobre o programa, na sua forma actual; é no entanto possível introduzir uma PAUSE O para aguentar a imagem até ser premida uma tecla em todos os pontos do programa que separam uma secção da seguinte:

```
  10 REM o ciclo do azoto
  20 BORDER 6: PAPER 5: INK 1
 100 CLS
 105 PRINT INK 2;AT 1,2;"*******
 O CICLO DO AZOTO *******"
 110 PRINT AT 3,3;"N2 (79% DO AR
)"
 120 PRINT AT 7,0;"Desazotan-";A
T 8,0;"tes"
 130 PRINT AT 7,19;"Fixadores"
 135 PRINT AT 8,19;"do azoto"
 140 FOR x=1 TO 5: PRINT INK 7;A
T 10-x,10;"*"
 150 PRINT INK 7;AT 4+x,18;"*":
PAUSE 20
 155 PRINT AT 10-x,10;" "
 160 PRINT AT 4+x,18;" "
 165 NEXT x
 170 PRINT INK 2; FLASH 1; INVER
SE 1;AT 10,10;"AZOTATOS"
 180 PRINT AT 11,0;"Relampago->"
 185 PRINT AT 11,16;"<-Fertiliza
dores"
 190 FOR x=1 TO 5: PRINT INK 7;A
T 10+x,16;"*"
 200 PAUSE 20: NEXT x
 210 FOR x=1 TO 5: PRINT AT 10+x
,16;" "
 215 PRINT INK 7;AT 15,15+x;"*"
 220 PAUSE 20: PRINT AT 15,15+x;
" "
 225 NEXT x
 230 PRINT AT 15,21;"Captados";A
T 16,21;"por raizes";AT 17,23;"!
"
 250 PRINT INK 2; INVERSE 1;AT 1
8,18;"PROTEINAS VEG."
 260 PRINT AT 19,18;"formando ao
";AT 20,18;"serem comidas"
 270 PRINT INK 2; INVERSE 1;AT 2
1,18;"PROTEINAS ANI."
 280 PRINT AT 19,2;"Corpos morto
s <"
 290 PRINT AT 21,3;" + fezes <<"
 295 PAUSE 50
 300 PRINT INK 7;AT 18,6;"↑"
 310 PRINT INK 2; INVERSE 1;AT 1
7,0;"Decomposicao"
 320 PRINT INVERSE 1;AT 15,3;"NH
3"
 330 PRINT INK 2; INVERSE 1;AT 1
3,0;"Azotantes"
 335 PRINT INK 7;AT 16,5;"*"
 340 PAUSE 50
 345 PRINT AT 16,5;" "
 350 PRINT INK 7;AT 14,5;"*"
 355 PAUSE 50
 360 PRINT AT 14,5;" "
 370 FOR x=1 TO 3: PRINT INK 7;A
T 14-x,13;"*"
```

```
375 PAUSE 60
380 PRINT AT 14-x,13;" ": NEXT
x
385 PAUSE 200
390 IF INKEY$="" THEN GO TO 100
```

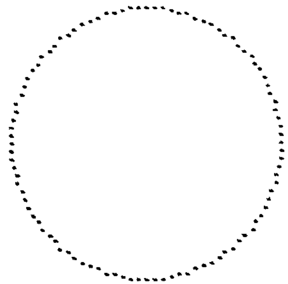
Tal como acontece em muitos programas para o Spectrum, este exemplo de execução não dá uma indicação muito viva da cor e da animação da imagem:

```
  ******* O CICLO DO AZOTO *****
**
   N2 (79% DO AR)


Desazotan-            Fixadores
tes                   do azoto

Relampago->        -Fertilizadores
              *


                     Captados
                     por raizes
                        |

      ↑
  Corpos mortos < formando ao
                  serem comidas
    + fezes <<
```

Desenho

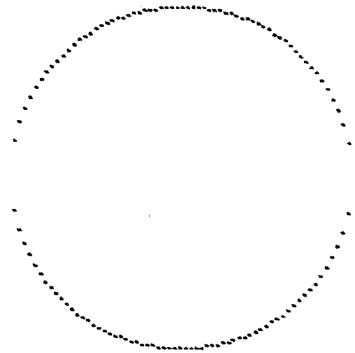
O Spectrum é também óptimo para representar curvas e funções. Ver uma forma desenvolver-se é bastante mais eficaz do que marcá-la a pouco e pouco em papel milimétrico, ou esboçá-la simplesmente num quadro negro.

O primeiro exemplo disto é um programa bastante curto demonstrando a equação polar de um círculo:



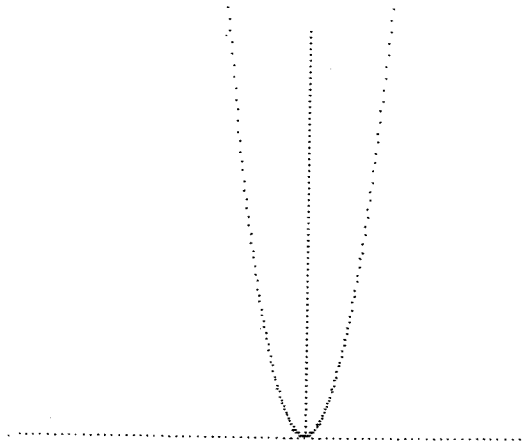
```
  5 REM EQUACAO POLAR DO CIRCUL
O
 10 FOR A=0 TO 2*PI STEP PI/50
 20 PLOT 115+50*COS A,85+50*SIN
A
 30 NEXT A
```

Em seguida ilustra-se a equação cartesiana do círculo:



```
  5 REM EQUACAO CARTESIANA DO C
IRCULO
 10 FOR X=-31 TO 31
 20 PLOT 115+2*X,85+2*SQR (1000
-X*X+1)
 30 PLOT 115+2*X,85-2*SQR (1000
-X*X+1)
 40 NEXT X
```

Um programa capaz de construir um gráfico de $Y = X^2$ produziu o seguinte resultado (encontrará a listagem a seguir):

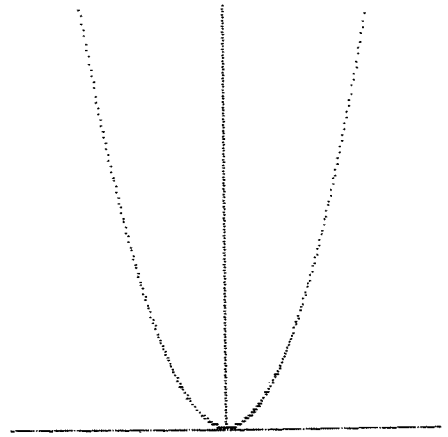


```
  10 REM GRAFICO DE Y=X AO QUADR
ADO
  15 FOR X=-7 TO 7 STEP .1
  20 PLOT 5*X+120,3*X*X+1
  30 PLOT 120,20*X
  40 PLOT 30*X,0
  50 NEXT X
```

Para mostrar que vale a pena experimentar um programa depois de estar em execução — produzimos aqui uma nova versão do programa anterior:

```
10 FOR N=0 TO 90 STEP .5
15 FOR X=-7 TO 7 STEP .1
20 PLOT 120,1.62*N
30 PLOT 30+2*N,0
40 NEXT N
50 FOR X=-12 TO 12 STEP .1
60 PLOT 5*X+120,1+X*X
70 NEXT X
```

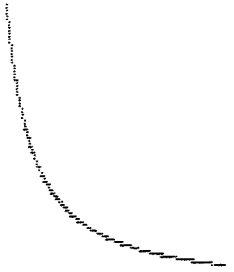

Vejamos o resultado:



A rotina que se segue desenha uma curva tangente:

```
  10 REM CURVA TANGENTE
  20 FOR X=0 TO 124
  30 PLOT 1.5*X,9*TAN (X/6.05)+2
0
  40 NEXT X
```

E esta um gráfico recíproco:

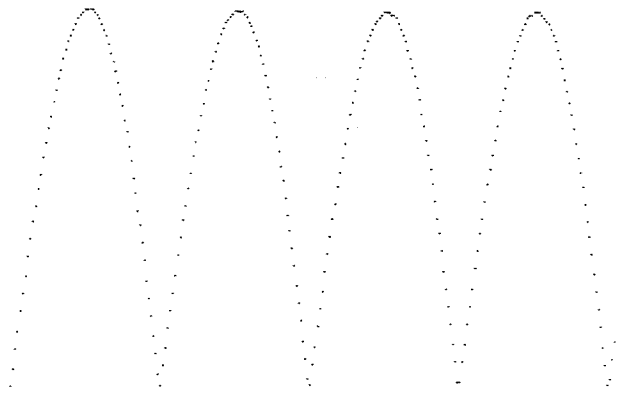


```
10 REM GRAFICO RECIPROCO
20 FOR X=1 TO 10 STEP .01
30 PLOT 10*X,100/X
40 NEXT X
```

Sinusóides

O programa aqui apresentado constrói um padrão com base numa onda sinusoidal:

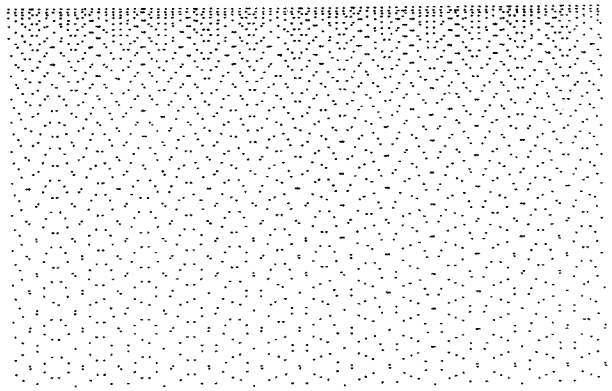
```
  10 REM SINUSOIDE
  15 FOR M=1 TO 2
  20 FOR N=0 TO 255
  30 IF M=1 THEN PLOT N,130+SIN
(N/20)
  35 IF M=2 THEN PLOT OVER 1;N,1
30+SIN (N/20)
  40 NEXT N
  50 NEXT M
```



Bola em ressalto

O programa que se segue é difícil de explicar, se bem que se torne bastante fácil compreender o que acontece quando se vê o resultado da execução. O leitor deverá imaginar uma bola (que possui uma notável tendência para não perder energia) ressaltando no visor. Em períodos de tempo bem definidos acende-se uma luz, projectando uma sombra da bola na parede atrás desta. A bola move-se ligeiramente para a direita quando ressalta, acabando por abandonar o visor pelo lado direito deste. O programa produz o padrão formado na parede pelas sombras:


```
10 REM TRACADO OBTIDO POR
       POSICOES CONSECUTIVAS
       DE UMA BOLA SALTANDO A
       INTERVALOS REGULARES
20 FOR N=0 TO 250 STEP .1
30 PLOT N,130+COS N
40 NEXT N
```

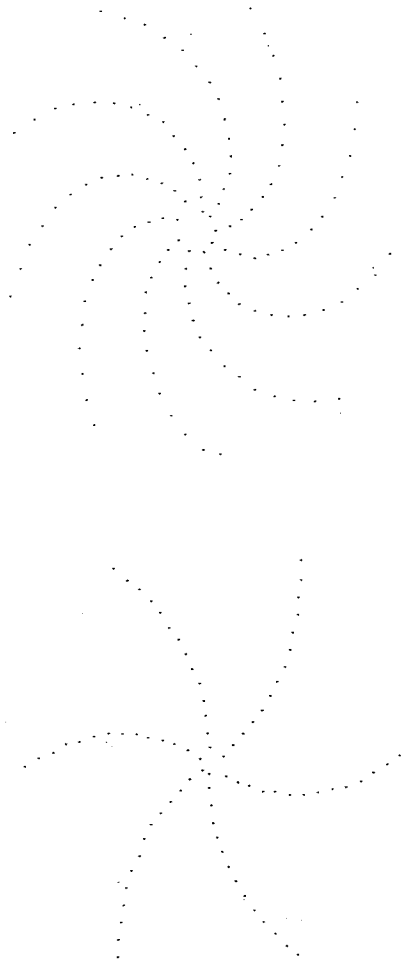


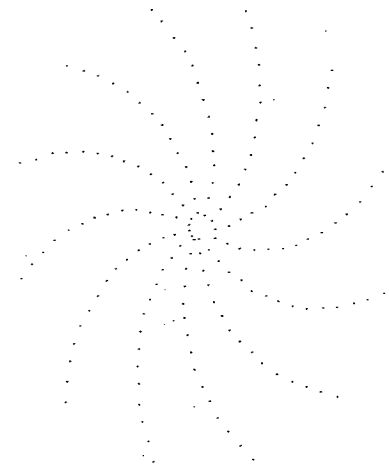
Se quiser ver a bola em movimento, acrescente a linha 35:

```
10 REM TRACADO OBTIDO POR
       POSICOES CONSECUTIVAS
       DE UMA BOLA SALTANDO P
       INTERVALOS REGULARES
20 FOR N=0 TO 250 STEP .1
30 PLOT N,130+COS N
35 PLOT OVER 1;N,130+COS N
40 NEXT N
```

Espirais

Este programa produz espirais com um diferente número de ramos. Usando um valor aleatório, começa por desenhar os ramos, apagando-os em seguida (usando a instrução PLOT OVER 1, ver a linha 25). O leitor compreenderá o que quero dizer executando o programa. Este pode ser mantido em execução durante bastante tempo:





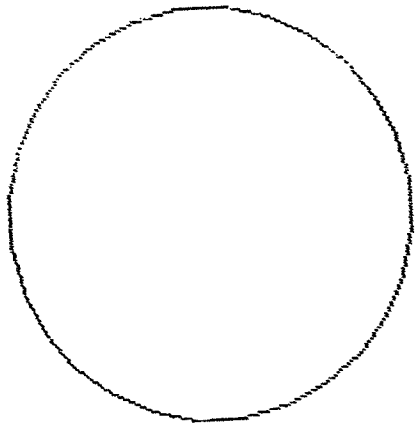
```
  5 REM ESPIRAL
  6 LET A=RND+.25
  7 FOR S=1 TO 2
 10 FOR A=PI TO 30*PI STEP A
 20 IF S=1 THEN PLOT 125+.9*A*S
IN A,80+.9*A*COS A
 25 IF S=2 THEN PLOT OVER 1;125
+.9*A*SIN A,80+.9*A*COS A
 30 NEXT A
 40 NEXT S
```

Formas

Observaremos em seguida como é possível construir cinco formas bastante úteis utilizando o Spectrum. Talvez o leitor verifique que pode usar estas formas para melhorar a apresentação gráfica dos seus programas:

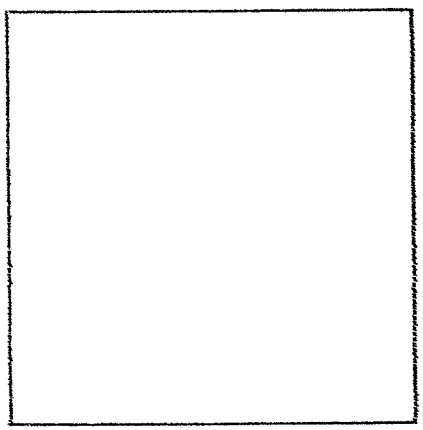
CIRCULO

```
10 CIRCLE 100,100,75
```

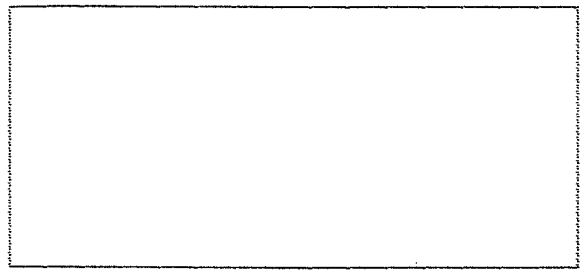


QUADRADO

```
10 DRAW 150,0
20 DRAW 0,150
30 DRAW -150,0
40 DRAW 0,-150
```

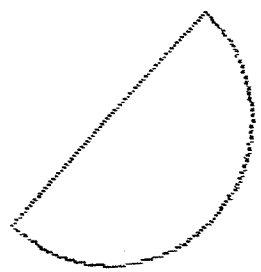



```
10 DRAW 240,0
20 DRAW 0,90
30 DRAW -240,0
40 DRAW 0,-90
```

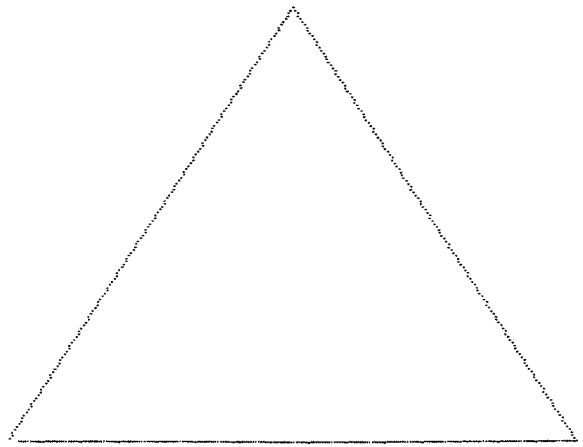


SEMICIRCULO

```
10 PLOT 50,50
20 DRAW 75,75,PI
30 DRAW -75,-75
```



```
TRIANGULO

10 DRAW 120,150
20 DRAW 120,-150
30 DRAW -235,0
```



Se estiver a trabalhar com crianças bastante novas, uma ficha, semelhante à da página seguinte, pode ajudá-las a programar (como se explica na secção deste livro dedicada ao uso do computador em infantários).

DESENHAR FIGURAS

CIRCLE - Carregar em H DRAW - Carregar em W

CÍRCULO ⊙

CAPS CAPS	SYMBOL CIRCLE	100	SYMBOL ,	100	SYMBOL ,	75 ENTER

TRIÂNGULO △

10 DRAW	120	SYMBOL ,	150		ENTER
20 DRAW	120	SYMBOL ,	SYMBOL —	150	ENTER
30 DRAW	SYMBOL —	235	SYMBOL ,	0	ENTER

QUADRADO □

10 DRAW	150	SYMBOL ,	0		ENTER
20 DRAW	0	SYMBOL ,	150		ENTER
30 DRAW	SYMBOL —	150	SYMBOL ,	0	ENTER
40 DRAW	0	SYMBOL ,	SYMBOL —	150	ENTER

RECTÂNGULO ▭

10 DRAW	240	SYMBOL ,	0		ENTER
20 DRAW	0	SYMBOL ,	100		ENTER
30 DRAW	SYMBOL —	240	SYMBOL ,	0	ENTER
40 DRAW	0	SYMBOL ,	SYMBOL —	100	ENTER

Manipulação de texto

Existem alguns truques que se podem aplicar ao tratamento do texto de modo a torná-lo mais interessante no Spectrum, como se demonstra nas rotinas seguintes, da autoria de David Perry.

A primeira rotina escreve tudo em caracteres invertidos:

```
  10 INPUT "ESCREVA UMA PALAVRA
";A$: IF LEN A$>30 THEN GO TO 10
  15 PRINT AT 0,0;A$
  20 LET Z=168: FOR A=167 TO 175
  30 FOR N=0 TO (LEN A$*8)
  40 IF POINT (N,A)=1 THEN PLOT
N,Z
  50 NEXT N: LET Z=Z-1: NEXT A
```

A seguinte permite escrever ao alto:

```
  10 INPUT "ESCREVA UMA PALAVRA
";A$: IF LEN A$>20 THEN GO TO 10
  15 PRINT AT 0,0;A$
  20 LET Z=0: FOR A=175 TO 168 S
TEP -1
  30 FOR N=(LEN A$*8) TO 0 STEP
-1
  40 IF POINT (N,A)=1 THEN PLOT
Z,N
  50 NEXT N: LET Z=Z-1: NEXT A
```

A terceira rotina coloca uma esquadria à volta das palavras:

```
  10 BORDER 0: PAPER 0: INK 7: C
LS
  20 INPUT "PALAVRA? ";A$
  30 INPUT "COORD. X? ";X: INPUT
 "COORD. Y? ";Y
  40 PRINT AT X,Y;A$
  50 LET C=((LEN A$)*8): LET A=(
8*Y)-2: LET B=(8*(21-X)-2)
  60 PLOT A,B: DRAW C+4,0: DRAW
0,12: DRAW -C-4,0: DRAW 0,-12
```

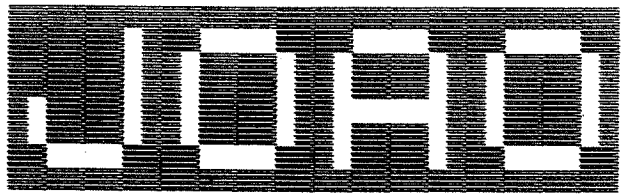
A quarta e última permite-nos escrever em letras de grandes dimensões:

```
   2 INPUT "QUER PALAVRAS NORMAI
S (1)
          OU SOBRE FUNDO
ESCURO (2)? ";Z
   3 IF Z>=2 THEN LET Z=0
  10 INPUT "PALAVRA (MAX. 4 LETR
AS? ";A$
  20 PRINT AT 0,0;A$
  25 LET L=LEN A$*8-1
  30 FOR X=168 TO 175: FOR Y=0 T
O L
  40 IF POINT (Y,X)=Z THEN PRINT
 AT X-166-20,Y;"■"
  50 NEXT Y: NEXT X
```


Como o leitor pode verificar observando o exemplo seguinte, os resultados podem ser bastante notáveis:

```
JOAO
```

```
JOAO
```



Gerador de caracteres

Finalmente apresentamos neste capítulo um programa gerador de caracteres, também escrito por David Perry, que deve tornar muito mais fácil o seu trabalho de criação de imagens.

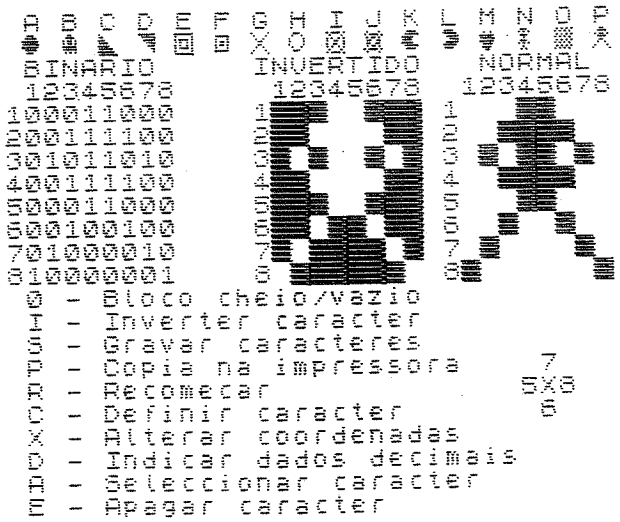
O programa possui um grande número de comandos. O cursor é movido com as teclas 5, 6, 7 e 8 (nas direcções indicadas nessas teclas). A tecla 0 enche o bloco onde o cursor se encontra, ou "apaga-o" se já estiver preenchido.

O leitor pode seleccionar e alterar qualquer caracter carregando numa tecla, inverter o caracter que se encontra na grelha e gravar os caracteres na cassette para uso em programas seus.

Ao executar o programa, verificará que este contém todos os detalhes sobre o modo como deve ser usado, juntamente com instruções claras quanto às teclas a utilizar.

```
  10 REM GERADOR DE CARACTERES
          DAVID PERRY 1983
  20 BORDER 1: PAPER 0: INK 7: B
RIGHT 1: CLS
  30 PRINT " A B C D E F G H I J
 K L M N O P"
  40 PRINT " [illegible]"
  50 PRINT BRIGHT 8; INK 5;" BIN
ARIO      INVERTIDO    NORMAL"
  60 PRINT " 12345678      123456
78  12345678"
  70 DIM Z(8,8)
  80 FOR N=1 TO 8
  90 PRINT N; INK 2;"00000000
 "; INK 7;N; INK 5;"████████ ";
INK 7;N
 100 NEXT N
 110 GO SUB 630
 120 INPUT "Eixo X (1 to 8)",B:
IF B>8 OR B<1 THEN GO TO 120
 130 INPUT "Eixo Y (1 to 8)",A:
IF A>8 OR A<1 THEN GO TO 130
 140 PRINT AT A+3,B; INK 4;"X";A
T A+3,B+13;"X";AT A+3,B+23;"X"
 150 LET K$=INKEY$: IF K$="" THE
N GO TO 150
 160 IF (K$="0" AND Z(A,B)=0) TH
EN LET Z(A,B)=1: GO TO 180
 170 IF (K$="0" AND Z(A,B)=1) TH
EN LET Z(A,B)=0
 180 IF Z(A,B)=0 THEN PRINT AT A
+3,B; INK 2;"0";AT A+3,B+13; INK
 5;"█";AT A+3,B+23; PAPER 0;" "
 190 IF Z(A,B)=1 THEN PRINT AT A
+3,B; INK 7;"1";AT A+3,B+13; PAP
ER 0;" ";AT A+3,B+23; INK 3;"█"
 200 LET A=A+(INKEY$="6" AND A<8
)-(INKEY$="7" AND A>1)
 210 LET B=B+(INKEY$="8" AND B<8
)-(INKEY$="5" AND B>1)
 220 IF INKEY$="p" THEN COPY
 230 IF INKEY$="s" THEN GO SUB 5
90
 240 IF INKEY$="g" THEN GO TO 10
50
 250 IF INKEY$="i" THEN GO TO 96
0
 260 IF INKEY$="r" THEN GO TO 33
0
 270 IF INKEY$="x" THEN GO TO 12
0
 280 IF INKEY$="c" THEN GO SUB 3
40
 290 IF INKEY$="d" THEN GO SUB 1
060
 300 IF INKEY$="a" THEN GO SUB 7
40
 310 IF INKEY$="e" THEN GO SUB 9
20
 320 GO TO 140
 330 CLS : RUN 30
 340 LET z$="": INPUT "Letra cor
respondente? ";D$: IF D$<"a" OR
D$>"u" THEN GO TO 340
 350 DIM a$(8,8)
 360 FOR X=1 TO 8
 370 FOR Y=1 TO 8
 380 LET A$(X,Y)=SCREEN$ (X+3,Y)
 390 NEXT Y: NEXT X
 400 FOR X=1 TO 8
 410 LET Z$=""
 420 FOR Y=1 TO 8
 430 LET Z$=Z$+A$(X,Y)
 440 NEXT Y
 450 LET NO=0
 460 IF Z$(1)="1" THEN LET NO=NO
+128
 470 IF Z$(2)="1" THEN LET NO=NO
+64
 480 IF Z$(3)="1" THEN LET NO=NO
+32
 490 IF Z$(4)="1" THEN LET NO=NO
+16
 500 IF Z$(5)="1" THEN LET NO=NO
+8
 510 IF Z$(6)="1" THEN LET NO=NO
+4
 520 IF Z$(7)="1" THEN LET NO=NO
+2
 530 IF Z$(8)="1" THEN LET NO=NO
+1
 540 POKE USR D$+X-1,NO
 550 PRINT AT X+3,9; FLASH 1;NO;
 FLASH 0;("  " AND NO<=99)
```

```
 560 NEXT X
 570 PRINT AT 1,0;"[illegible]
X O [illegible]"
 580 RETURN
 590 CLS : PRINT AT 20,8;"LIGUE
O GRAVADOR"
 600 INPUT "NOME? ";A$
 610 SAVE A$CODE USR "A",168
 620 RUN
 630 PRINT AT 13,1;"I - Inverter
 caracter"
 640 INK 6: PRINT AT 14,1;"S - G
ravar caracteres"
 650 PRINT AT 12,1;"0 - Bloco ch
eio/vazio"
 660 PRINT AT 15,1;"P - Copia na
 impressora        7"
 670 PRINT AT 16,1;"R - Recomeca
r               5X8"
 680 PRINT AT 17,1;"C - Definir
caracter          6"
 690 PRINT AT 18,1;"X - Alterar
coordenadas"
 700 PRINT AT 19,1;"D - Indicar
dados decimais"
 710 PRINT AT 20,1;"A - Seleccio
nar caracter"
 720 PRINT AT 21,1;"E - Apagar c
aracter"
 730 RETURN
 740 REM seleccionar caracter
 750 INPUT "Caracter (A a U)? ";
R$: IF R$>"u" OR R$<"a" THEN GO
TO 750
 760 LET R$=R$(1 TO 1)
 770 FOR Y=0 TO 7
 780 LET S=PEEK (USR R$+Y)
 790 LET L=128
 800 FOR X=0 TO 7
 810 LET T=INT (S/L)
 820 PRINT AT Y+4,X+1; INK 2;("0
" AND NOT T=1); INK 7;("1" AND N
OT T<>1)
 830 PRINT AT Y+4,X+14; INK 5;("
▒" AND NOT T=1); PAPER 0;(" " AN
D NOT T<>1)
 840 PRINT AT Y+4,X+24; PAPER 0;
(" " AND NOT T=1); INK 3;("▒" AN
D NOT T<>1)
 845 IF SCREEN$ (Y+4,X+1)="0" TH
EN LET Z(Y+1,X+1)=0
 846 IF SCREEN$ (Y+4,X+1)="1" TH
EN LET Z(Y+1,X+1)=1
 850 LET S=S-T*L
 860 LET L=L/2
 870 NEXT X: NEXT Y
 880 FOR N=1 TO 8: PRINT AT N+3,
9;"   ": NEXT N
 890 INPUT "Eixo X (1 to 8)",B:
IF B>8 OR B<1 THEN GO TO 120
 900 INPUT "Eixo Y (1 to 8)",A:
IF A>8 OR A<1 THEN GO TO 130
 910 RETURN
 920 INPUT "Caracter? ";D$: IF D
$<"a" OR D$>"u" THEN GO TO 920
 930 FOR F=0 TO 7
 931 POKE USR D$+F,0
 932 NEXT F
 940 PRINT AT 1,0;"[illegible]
X O [illegible]"
 950 RETURN
 960 FOR U=1 TO 8
 970 FOR F=1 TO 8
 980 IF SCREEN$ (U+3,F)="0" THEN
 GO TO 1010
 990 LET Z(U,F)=0: PRINT AT U+3,
F; INK 2;"0";AT U+3,F+13; INK 5;
"▒";AT U+3,F+23; PAPER 0;" "
1000 GO TO 1020
1010 LET Z(U,F)=1
1011 PRINT AT U+3,F; INK 7;"1";A
T U+3,F+13; PAPER 0;" ";AT U+3,F
+23; INK 3;"▒"
1020 NEXT F
1030 NEXT U
1040 GO TO 120
1050 CLS : PRINT "[illegible]": PRINT "[illegible]
[illegible]": PRINT "XO[illegible]": PAUSE 0: CLS :
 RUN
1060 INPUT "LETRA? ";L$: FOR F=0
 TO 7: INPUT "Decimal? ";S: POKE
 USR L$+F,S: NEXT F
1070 PRINT AT 1,0;"[illegible]
X O [illegible]"
1080 RETURN
```



8

USO DO SPECTRUM NO ENSINO DE LÍNGUAS

O computador é um pouco mais limitado nesta área do que no ensino de matemáticas, se bem que com alguma paciência seja possível definir programas com interesse.

Ortografia

Começaremos por um programa de ortografia que pode também ser usado para ensino de línguas estrangeiras. O programa armazena até 12 palavras normalmente mal escritas, como a indicada na declaração DATA da linha 250, e faz 10 perguntas sobre elas. Bastará ao leitor acrescentar as palavras desejadas depois da linha citada, subtituindo esta se o pretender.

Para usar o programa a fim de o adaptar ao ensino de línguas estrangeiras, bastará alterar a linha 110 de modo a imprimir algo do género: "Indique a palavra inglesa correspondente a..."

Vejamos o programa em acção:

```
Escolha a grafia correcta entre
as indicadas e escreva-a (maius-
culas)

COROBORAR

CORRUBORAR

CORROBORAR

A resposta foi: CORROBORAR

Optimo, CORROBORAR
esta' correcto
```

```
Tem 1 certas
num total de 1


Um momento...
```

E agora a listagem do programa:

```
  10 REM grafia
  15 RANDOMIZE
  20 LET pontos=0
  25 DIM f$(11,7)
  30 FOR h=1 TO 10
  40 RESTORE 250+10*INT (RND*9)
  60 READ a$
  65 FOR t=1 TO h
  66 IF RND>.5 THEN IF f$(t)( TO
 7)=a$( TO 7) THEN GO TO 40
  70 NEXT t
  75 LET f$(h)=a$
  80 READ b$
  90 READ c$
 100 READ d$
 110 PRINT "Escolha a grafia cor
recta entre as indicadas e escre
va-a (maius-culas)"
 120 PRINT '''b$
 130 PRINT '''c$
 140 PRINT '''d$
 150 INPUT e$
 155 PRINT '"A resposta foi: ";e
$
 160 IF e$=a$ THEN PRINT '"Optim
o, ";e$,"esta' correcto": LET po
ntos=pontos+1
 170 IF e$<>a$ THEN PRINT "mas e
sta' errada."''"A grafia correct
a e' ";a$
 180 IF pontos>0 THEN PRINT ''"T
em ";pontos;" certas""num total
 de ";h
 190 PRINT '' FLASH 1;"Um moment
o..."
 195 INPUT a$: IF a$<>"" THEN CO
PY
 200 PAUSE 200
 210 CLS
 220 NEXT h
 230 PRINT ''''"Escreveu ";pontos
;" palavras""correctamente."
 240 STOP
```



```
 250 DATA "CORROBORAR","COROBORA
R","CORRUBORAR","CORROBORAR"
```

O leitor poderá facilmente incluir no programa as palavras mais adequadas à idade dos alunos, graduando a dificuldade destas.

Anagramas

O programa que se segue foi concebido por Derek Cook. Contém todas as instruções necessárias, e foi pensado para uso por crianças entre seis e nove anos:

```
 200 PAPER 0: CLS : PAPER 2: INK
 6
 210 PRINT AT 10,3;"Sabe o que s
ao ANAGRAMAS ?"
 220 PAUSE 100: CLS
 230 BORDER 1: PAPER 7: INK 0: C
LS
 300 PRINT "Usa-se uma palavra:"
 310 FOR a=1 TO 4
 315 READ a,a$
 320 INK a: PRINT AT 5,12+a;a$
 330 DATA 1,"T",2,"A",3,"L",4,"O
"
 340 NEXT a
 345 PAUSE 100
 350 INK 0: PRINT AT 3,0;"e mist
uram-se as letras:"
 360 RESTORE 400
 370 FOR z=1 TO 4
 380 READ b,b$
 390 INK b: BEEP .1,z: PRINT AT
6,12+z;b$
 395 PAUSE 50
 400 DATA 3,"L",2,"A",1,"T",4,"O
"
 410 NEXT z
 420 RESTORE 450
 430 FOR y=1 TO 4
 435 READ c,c$
 440 INK c: BEEP .1,y: PRINT AT
7,12+y;c$
 445 PAUSE 50
 450 DATA 1,"T",4,"O",3,"L",2,"A
"
 460 NEXT y
 470 RESTORE 500
 475 FOR x=1 TO 4
 480 READ d,d$
```

```
 490 INK d: BEEP .1,x: PRINT AT
0,12+x;d$
 500 DATA 2,"A",1,"T",4,"O",3,"L
"
 510 NEXT x
 515 PAUSE 200: BORDER 1: PAPER
6: INK 2: CLS
 520 PRINT "O computador pode fa
zer estes   anagramas em seu lug
ar, mas em  geral as palavras na
o terao sen-tido."
 525 PRINT AT 5,0;"Por exemplo:"
 530 FLASH 1: PRINT AT 7,13;"AOT
L"
 535 FLASH 0: PRINT AT 9,0;"O qu
e sera' um TOAL? Procure no dicc
ionario."
 540 PRINT AT 11,0;"As palavras
sem sentido podem   ser divertid
as; arranje voce    mesmo sentid
o para elas. Se qui-ser que eu l
he de algumas ideiascarregue em
ENTER."
 550 INPUT j$
 560 CLS
 600 RANDOMIZE
 605 PAPER 0: CLS : PAPER 2: INK
6
 610 INPUT "Escreva uma palavra
e carregue  em ENTER: ";a$
 620 LET l=LEN a$
 630 DIM b$(2,l)
 640 FOR n=1 TO 40
 650 FOR c=1 TO l
 660 LET b$(1,c)=a$(c)
 670 LET b$(2,c)="1"
 680 NEXT c
 690 FOR d=1 TO l
 700 LET r=INT (RND*l)+1
 710 IF b$(2,r)="0" THEN GO TO 7
00
 720 LET b$(2,r)="0"
 730 PRINT b$(1,r);
 740 NEXT d
 750 PRINT
 760 NEXT n
 770 BORDER 1: PAPER 0: INK 7: F
LASH 1: PRINT "Carregue ENTER pa
ra instrucoes."
 780 FLASH 0: INPUT i$
 800 CLS : BORDER 3: PAPER 6: IN
K 0: PRINT AT 5,0;"Escreva:"
 810 PRINT AT 7,0;"1 para mais 4
0 anagramas da mes-  ma palavra
                    2 para anagra
mas de outra pala-  vra
                    3 para dizer
'adeus'"
 820 INPUT p
 830 IF p=1 THEN GO TO 620
 840 IF p=2 THEN GO TO 605
 850 CLS : BORDER 4: PAPER 3: IN
K 5: PRINT AT 11,13;"ADUSE!";AT
13,8;"... digo, ADEUS!"
```

Leitura rápida

O último programa deste capítulo, escrito por Gordon Armitt, foi concebido para desenvolver uma maior rapidez de leitura. Veja-mo-lo em acção:

```
ESCREVA AGORA AS LETRAS QUE VIU.

AS MINHAS LETRAS ERAM TXCD

AS SUAS FORAM TXCD

ESTA' CERTO. GANHA PORTANTO -▒▒-
PONTOS.

A SUA PONTUACAO E' 10

 PREPARE-SE PARA UM NOVO TESTE!


ESCREVA AGORA AS LETRAS QUE VIU.

AS MINHAS LETRAS ERAM YRDI

AS SUAS FORAM YRDI

ESTA' CERTO. GANHA PORTANTO -▒▒-
PONTOS.
```

```
A SUA PONTUACAO E' 20

 PREPARE-SE PARA UM NOVO TESTE!

ESCREVA AGORA AS LETRAS QUE VIU.

AS MINHAS LETRAS ERAM BTJK

AS SUAS FORAM BTLK

ESTA' ERRADO. PERDE PORTANTO ▒▒
PONTOS.

A SUA PONTUACAO E' 10

 PREPARE-SE PARA UM NOVO TESTE!

ESCREVA AGORA AS LETRAS QUE VIU.

AS MINHAS LETRAS ERAM ZGDJ

AS SUAS FORAM ZGDJ

ESTA' CERTO. GANHA PORTANTO -▒▒-
PONTOS.

A SUA PONTUACAO E' 20

 PREPARE-SE PARA UM NOVO TESTE!
```

E agora o programa:

```
  10 PRINT TAB 4;"TREINO DE LEIT
URA RAPIDA"
  15 PRINT TAB 4;"--------------
----------"
  20 LET V=0
  30 PRINT AT 10,0;" POR FAVOR I
NDIQUE SE DESEJA NU-MEROS OU LET
RAS";AT 15,9;"ESCREVA N OU L"
  40 INPUT D$
  60 INPUT "VELOCIDADE DO TESTE
(1/9)? ";Z
  65 IF Z<1 OR Z>9 THEN GO TO 60
  70 IF D$="L" THEN GO TO 600
 210 PAUSE 80: CLS : PAUSE 10
 230 LET M=INT (RND*((10↑5)))
 232 LET P=INT (RND*20)
 234 LET Q=INT (RND*20)
 240 PRINT AT P,Q;M
 250 PAUSE 25*Z
 260 CLS
 270 PRINT ''"ESCREVA O NUMERO Q
UE VIU"
 280 INPUT A
 290 PRINT AT 10,0;TAB 10;M
 300 IF A=M THEN LET V=V+10: PRI
NT '' INK 2; PAPER 6; FLASH 1;"E
STA' CERTO. GANHA PORTANTO -"; I
NVERSE 1;10; INVERSE 0;"-PONTOS.
"
 310 IF A<>M THEN LET V=V-10: PR
INT '' INK 1; PAPER 7; FLASH 1;"
ESTA' ERRADO. PERDE PORTANTO ";
INVERSE 1;10; INVERSE 0;" PONTOS
."
 370 PRINT AT 18,0;"A SUA PONTUA
CAO E' ";V
 380 IF V=-30 THEN PRINT "FALHOU
 DEMAIS!": STOP
 385 IF V=50 THEN PRINT "OPTIMO!
 TEVE BONS RESULTADOS!": STOP
 390 PRINT AT 20,1; BRIGHT 1;"PR
EPARE-SE PARA UM NOVO TESTE!"
 400 PAUSE 300
 410 GO TO 210
 600 REM LETRAS
 610 PAUSE 140: CLS : PAUSE 10
 620 LET A$=CHR$ (INT (RND*26)+6
5)+CHR$ (INT (RND*26)+65)+CHR$ (
INT (RND*26)+65)+CHR$ (INT (RND*
26)+65)
 630 LET P=INT (RND*20)
 640 LET Q=INT (RND*20)
 650 PRINT AT P,Q;A$
 660 PAUSE 25*Z
 670 CLS
 680 PRINT "ESCREVA AGORA AS LET
RAS QUE VIU."
 690 INPUT B$
 700 IF CODE B$<65 OR CODE B$>90
 OR LEN B$<>4 THEN GO TO 690
 710 PRINT ''"AS MINHAS LETRAS E
RAM ";A$''"AS SUAS FORAM ";B$
 720 IF A$=B$ THEN LET V=V+10: P
```

```
RINT '''' INK 2; PAPER 6; FLASH 1;
"ESTA' CERTO, GANHA PORTANTO -";
 INVERSE 1;10; INVERSE 0;"-PONTO
S."
 730 IF A$<>B$ THEN LET V=V-10:
PRINT '''' INK 1; PAPER 7; FLASH 1
;"ESTA' ERRADO, PERDE PORTANTO "
; INVERSE 1;10; INVERSE 0;" PONT
OS."
 735 PRINT AT 18,0;"A SUA PONTUA
CAO E' ";V
 740 IF V=-30 THEN PRINT "FALHOU
 DEMAIS!": STOP
 750 IF V=50 THEN PRINT "OPTIMO!
 TEVE BONS RESULTADOS!": STOP
 770 PRINT AT 20,1; BRIGHT 1;"PR
EPARE-SE PARA UM NOVO TESTE!"
 780 PAUSE 160
 900 GO TO 610
```


9

VALIDAÇÕES DE ENTRADAS

Os bons programas de computador contêm instruções que verificam se as entradas feitas pelo utilizador são aceitáveis, rejeitando as que não o forem.

Muitos programas de computador serão parados se devido a erro, ou por má vontade o utilizador indicar dados incorrectos. Por exemplo, se um programa espera uma entrada numérica, e for teclada uma letra que não tenha sido definida anteriormente como variável, o Spectrum parará com a mensagem "variável desconhecida".

É difícil prever todas as eventualidades porque os maldosos têm muitos recursos. Por muito grande que seja o cuidado em validar as entradas, é provável que o leitor descubra que um dos seus alunos descobriu uma maneira de parar o programa.

O programa sobre cálculo de pesos moleculares apresentado mais atrás contém um bom exemplo de validação de entradas nas linhas 160 a 170. Reproduzimo-las aqui:

```
 150 LET RESPOSTA=VAL (B$(X))*Y
 160 INPUT C$
 165 IF C$="" THEN GO TO 160
 170 IF CODE C$<49 OR CODE C$>58
 THEN GO TO 160
 180 IF VAL C$<RESPOSTA-.05 OR V
AL C$>RESPOSTA+.05 THEN GO TO 25
0
 200 CLS
```

Se bem que o computador pretenda de facto um dado numérico, pede uma cadeia alfanumérica. Como o leitor já aprendeu, o Spectrum pode converter uma cadeia no seu equivalente numérico recorrendo à função VAL. É mais fácil ao computador rejeitar uma en-

trada alfanumérica incorrecta do que uma numérica; enquanto que a entrada de uma letra quando a máquina espera um número possa parar imediatamente a execução ou deixá-la continuar usando valores completamente errados, é sempre possível verificar cuidadosamente uma entrada alfanumérica tentando evitar qualquer dos casos.

No fragmento de programa apresentado anteriormente, a linha 165 rejeita qualquer "ausência de entrada", isto é, a possibilidade de o utilizador carregar em ENTER sem indicar qualquer valor. A linha seguinte, 170, verifica o código da cadeia dada em resposta, e se este não variar entre 49 (código correspondente ao número 1) e 58 (código correspondente ao número 9), a entrada é rejeitada. Rejeita assim qualquer entrada que não se inicie por um número. Mas já não rejeita, e o facto pode ser aproveitado por um aluno maldoso, uma entrada do género "12z" ou "9truque1".

Fósforos

O programa seguinte, "Fósforos", é uma variante de um jogo bem conhecido em que os jogadores tiram fósforos alternadamente de uma pilha deles, com um limite quanto à quantidade que podem tirar de cada vez, perdendo aquele que é forçado a tirar o último fósforo.

Este jogo é incluído aqui porque contém algumas validações bastante boas das entradas indicadas pelo jogador. É um pouco mais difícil parar este programa do que o do cálculo de pesos moleculares.

Dê entrada ao programa na máquina, execute-o algumas vezes, e depois discutiremos os métodos usados para validar as entradas:

```
                FOSFOROS

 O maximo que pode tirar de cada
vez e' 4


1 2 3
4 5 6 7 8 9 10
11 12 13 14 15
16 17 18 19

OK, tira 3


Vou tirar 1
```



```
   10 REM FOSFOROS
   20 RANDOMIZE
   30 CLS
   40 LET M=0
   50 LET E=0
   60 LET Z=INT (RND*8)+16
   70 IF 2*INT (Z/2)=Z THEN LET Z
=Z+1
   80 LET M=3+INT (RND*4)
   90 CLS
  100 PRINT AT 3,12;"FOSFOROS"
  110 PRINT AT 5,1;"O maximo que
pode tirar de cadavez e' "; FLAS
H 1;M
  120 GO SUB 320
  130 IF e>0 THEN PRINT AT 7,0;"
Tirou "; FLASH 1;E; FLASH 0;", e
 eu "; FLASH 1;q
  135 PRINT
  140 FOR K=1 TO Z: BEEP .03,2*K
  150 PRINT K;" ";: IF RND>.8 THE
N PRINT
  160 NEXT K
  170 GO SUB 320
  180 INPUT "Quantos quer tirar?
";E$
  185 IF LEN E$>1 THEN GO TO 180
  186 IF E$<"1" OR E$>"9" THEN GO
 TO 180
  187 LET E=VAL E$
  190 IF E>M THEN PRINT "Nao pode
 tirar tantos!": GO TO 180
  200 IF E>Z THEN PRINT "Ja  nao
restam tantos.": GO TO 180
  210 PRINT : PRINT "OK, tira ";E
  220 LET Z=Z-E
  230 GO SUB 320
  240 IF Z<1 THEN CLS : PRINT AT
5,0;"Voce tirou o ultimo, e port
anto GANHEI!": PRINT : PRINT "Ob
rigado pelo jogo": STOP
  250 LET 0=Z-1-INT ((Z-1)/(M+1))
*(M+1)-INT (RND*2)+INT (RND*2)
*250 IF 0<1 OR 0>M THEN GO TO 25
0
  270 PRINT : PRINT "Vou tirar ";
0
  275 GO SUB 350
  280 LET Z=Z-0
  290 IF Z<1 THEN CLS : PRINT AT
5,0;"Tirei o ultimo, e portanto
voce GANHOU!": PRINT : PRINT FLA
```

```
SH 1;"PARABENS!": STOP
 300 GO TO 90
 320 REM rotina de atraso
 340 PRINT
 350 FOR R=1 TO 100: NEXT R
 360 RETURN
```

O leitor verificará, quando executar este programa, que cada nova imagem inclui a mensagem "O máximo que pode tirar é...". Vale a pena recordar aos utilizadores do programa os limites que existem para as respostas que deverão dar. A linha 110 imprime portanto esta mensagem repetidamente, com o objectivo de evitar os erros nas respostas.

A linha 180 pede uma entrada do utilizador sob a forma de uma cadeia alfanumérica. O computador sabe (porque o jogo é feito deste modo) que cada entrada deve ter apenas um algarismo, entre 1 e H (variável esta que recebe um valor definido na linha 80). A linha 185 verifica o comprimento da cadeia alfanumérica de resposta. Se for maior do que um (como aconteceria no caso de 2RD2, por exemplo), a entrada é rejeitada (se bem que, como vimos anteriormente, tivesse sido aceite pelo programa anterior). Tendo passado este obstáculo, a cadeia de resposta é analisada de novo. Se for menor do que "1" ou maior do que "9" será também rejeitada. Isto permite eliminar eficazmente qualquer entrada não numérica, tal como uma entrada nula (isto é, a entrada obtida carregando em ENTER sem ter escrito nada). Rejeitará igualmente a resposta obtida carregando na tecla SPACE/BREAK.

Finalmente, na linha 190 (depois de E$ ter sido transformada numa variável numérica, E, pela linha 187), verifica-se o valor de resposta relativamente ao limite máximo admissível, H.

Se bem que o programa esteja longe de ser perfeito, demonstra bem o grau de rigor que se deve tentar obter em qualquer programa desenvolvido para uso em escolas. Depois de o programa estar construído e dar os resultados pretendidos, e de a sua parte gráfica ter sido tornada tão agradável quanto possível, o leitor deverá prestar alguma atenção à validação de todas as entradas, evitando que qualquer resposta incorrecta vá afectar o funcionamento do programa.

10

PROGRAMAS DE ESCOLHA

Existe uma tendência para considerar o Spectrum e outros microcomputadores em escolas apenas em termos de fazerem aquilo que já é feito de outra maneira. É necessária uma reconversão mental para pensar em coisas que não são actualmente feitas, mas que o podem ser com relativa facilidade quando se dispõe de um Spectrum.

Os programas de escolha são dos mais fáceis de escrever, e devido a isto predominam entre o software existente comercialmente. Por outro lado, os programas de escolha são dos mais criticados por constituírem um desperdício das potencialidades do computador. Porém, isto não acontece necessariamente.

Grande parte do software deste tipo merece certamente uma crítica, porque é excessivamente limitado e talvez não esteja suficientemente ligado às matérias dadas na aula. Neste capítulo apresentamos um programa que produz até 100 perguntas (no caso de um Spectrum de 48 K, mas apenas 15 numa máquina de 16 K) cada uma das quais avaliando até seis respostas diferentes do utilizador.

O número de respostas a escolher pode variar de uma pergunta para outra. O programa foi deliberadamente escrito de modo a ser tão flexível quanto possível, permitindo ao utilizador usá-lo para qualquer tema. O programa suporta uma pergunta feita por frase completa em todos os casos, não tendo portanto uma das deficiências de muitos outros programas, em que a pergunta é reduzida a algumas palavras mais significativas.

Tendo sido já experimentado, o programa em causa revelou-se bastante útil na medida em que pode ser usado para praticamente qualquer tema e para estudantes de qualquer idade. É bastante fácil usá-lo, adaptando-o onde necessário, de modo a concordar com o tema que nos interessa. As imagens apresentadas durante a criação de um teste são concebidas para conduzir o utilizador por todos os

passos necessários, como se pode verificar pelos seguintes exemplos:

```
                CAPITAIS

QUANTAS PERGUNTAS (MAX. 100)?
5

ESCREVA PERGUNTAS, RESPOSTAS E A
ESCOLHA CORRECTA COMO PEDIDO, OS
ERROS PODERAO SER EMENDADOS DE-
POIS.

COMPRIMENTOS MAXIMOS:
 60 CARACTERES POR PERGUNTA,
 50 CARACTERES POR RESPOSTA
CARREGUE EM [ENTER]


PERGUNTA NO. 1

NO. DE ESCOLHAS (MAX. 6)?
QUAL A CAPITAL DA AUSTRALIA

RESPOSTA(A)MELBOURNE

RESPOSTA(B)SYDNEY

RESPOSTA(C)CANBERRA

ESCOLHA CORRECTA?


PERGUNTA NO. 2

NO. DE ESCOLHAS (MAX. 6)?
QUAL A CAPITAL DA ROMENIA

RESPOSTA(A)BUCARESTE

RESPOSTA(B)MADRID

ESCOLHA CORRECTA?
```



```
PERGUNTA NO. 3

NO. DE ESCOLHAS (MAX. 6)?
QUAL A CAPITAL DO UGANDA

RESPOSTA(A)KAMPALA

RESPOSTA(B)HANOI

RESPOSTA(C)KABUL

RESPOSTA(D)FEZ

RESPOSTA(E)CAIRO

ESCOLHA CORRECTA?
```

Depois de ter introduzido todas as perguntas, pode verificar o programa. Também esta função é apresentada de um modo fácil de compreender, facilitando o trabalho a quem elabora o teste:

```
VERIFICACAO DO PROGRAMA

PERGUNTA NO. 3
QUAL A CAPITAL DO UGANDA

(A)KAMPALA

(B)HANOI

(C)KABUL

(D)FEZ

(E)CAIRO

ESCOLHA CORRECTA=A

CARREGUE EM [ENTER] PARA CONTINUAR
OU 'E' PARA CORRIGIR
```

Finalmente, quando o estudante usa o programa, a imagem dificilmente se presta a confusões:

```
OLA', TIM VOU FAZER
ALGUMAS PERGUNTAS E QUERO QUE
```

ESCOLHA A RESPOSTA QUE PENSA SER CORRECTA, IDENTIFICADA POR UMA LETRA.

CARREGUE EM [ENTER]

Vejamos a listagem completa:

```
   5 REM escolha multipla
  10 LET m=1
  20 PRINT "TITULO (MAX. 20 CHR$
)? "
  30 INPUT A$
  40 LET A=LEN A$+1
  50 LET B=((30-A)/2)
  60 GO SUB 1130
  70 PRINT "QUANTAS PERGUNTAS (M
AX. 100)?"
  80 INPUT E: PRINT E
  90 DIM B$(E,60)
 100 DIM C$(E,50)
 110 DIM D$(E,50)
 120 DIM E$(E,50)
 130 DIM F$(E,50)
 140 DIM G$(E,50)
 150 DIM H$(E,50)
 160 DIM J$(E,50)
 170 DIM F(E)
 180 PRINT '"ESCREVA PERGUNTAS,
RESPOSTAS E ESCOLHA CORRECTA CO
MO PEDIDO, OS ERROS PODERAO SER E
MENDADOS DE- POIS."
 190 PRINT '"COMPRIMENTOS MAXIMO
S:                60 CARACTERES POR
PERGUNTA,         50 CARACTERES POR
RESPOSTA"
 200 GO SUB 1210
 210 INPUT Z$: CLS
 220 FOR D=1 TO E
 230 PRINT "PERGUNTA NO. ";D
 240 PRINT '"NO. DE ESCOLHAS (M
AX. 6)?"
 270 INPUT F(D): PRINT F(D)
 280 PRINT AT 4,0;"INDIQUE PERGU
NTA (MAX. 60 CHR)"
 290 INPUT B$(D)
 300 PRINT AT 4,0;B$(D);"     "
 310 FOR G=0 TO F(D)-1
 320 PRINT "RESPOSTA";
```



```
 330 GO SUB 710+40*G
 340 NEXT G
 350 PRINT "ESCOLHA CORRECTA?"
 360 INPUT J$(D)
 370 CLS : NEXT D
 380 FOR D=1 TO E
 390 GO SUB 1130
 400 PRINT "VERIFICACAO DO PROGR
AMA"
 410 PRINT "═══════════════════
═══"
 420 GO SUB 950
 430 PRINT "ESCOLHA CORRECTA=";J
$(D)
 440 GO SUB 1210
 450 PRINT "PARA CONTINUAROU 'E'
 PARA CORRIGIR";
 460 INPUT Z$
 470 IF Z$="E" OR Z$="e" THEN CL
S : GO TO 230
 480 NEXT D
 490 CLS
 500 PRINT "CORRECCOES TERMINADA
S, EXECUCAO DE ENSAIO EM SEGUIDA
."
 510 PAUSE 250
 520 LET S=0
 530 GO SUB 1130
 540 PRINT "OLA', SOU O SPECTRUM
. COMO SE      CHAMA?"
 550 INPUT K$
 560 CLS : PRINT "OLA', ";K$;" V
OU FAZER","ALGUMAS PERGUNTAS E O
UERO QUE   ESCOLHA A RESPOSTA OU
E PENSA SER CORRECTA, IDENTIFICAD
A POR UMA  LETRA."
 570 GO SUB 1210
 580 INPUT Z$
 590 FOR D=1 TO E
 600 GO SUB 1130
 610 PRINT AT 2,0;K$;" ";
 620 GO SUB 950
 630 PRINT "RESPOSTA=(?)"
 640 INPUT L$
 650 IF L$=J$(D) THEN GO TO 1270
 660 GO SUB 1230
 670 PRINT "POR FAVOR ESPERE UM
MOMENTO"
 680 PAUSE 500
 690 NEXT D
 700 GO SUB 1310
 710 PRINT "(A)";
```

```
 720 INPUT C$(D)
 730 PRINT C$(D)
 740 RETURN
 750 PRINT "(B)";
 760 INPUT D$(D)
 770 PRINT D$(D)
 780 RETURN
 790 PRINT "(C)";
 800 INPUT E$(D)
 810 PRINT E$(D)
 820 RETURN
 830 PRINT "(D)";
 840 INPUT F$(D)
 850 PRINT F$(D)
 860 RETURN
 870 PRINT "(E)";
 880 INPUT G$(D)
 890 PRINT G$(D)
 900 RETURN
 910 PRINT "(F)";
 920 INPUT H$(D)
 930 PRINT H$(D)
 940 RETURN
 950 PRINT "PERGUNTA NO. ";D
 960 PRINT B$(D)
 970 FOR G=0 TO F(D)-1
 980 GO SUB 1010+20*G
 990 NEXT G
1000 RETURN
1010 PRINT "(A)";C$(D)
1020 RETURN
1030 PRINT "(B)";D$(D)
1040 RETURN
1050 PRINT "(C)";E$(D)
1060 RETURN
1070 PRINT "(D)";F$(D)
1080 RETURN
1090 PRINT "(E)";G$(D)
1100 RETURN
1110 PRINT "(F)";H$(D)
1120 RETURN
1130 CLS
1140 PRINT AT 0,B;A$
1150 PRINT AT 1,B-1;"■";
1160 FOR C=1 TO A
1170 PRINT "■";
1180 NEXT C
1190 PRINT
1200 RETURN
1210 PRINT "CARREGUE EM [■■■■] ";
1220 RETURN
1230 PRINT "Errado, ";K$;",","A
resposta correcta e'",J$(D)
1240 GO SUB 1210
1250 INPUT Z$
1260 RETURN
1270 LET S=S+1
1280 PRINT "Optimo, ";K$;", essa
","e' a resposta correcta. Ganho
u um ponto."
1290 GO TO 670
1300 GO SUB 1130
1310 PRINT K$;", TEM ";S;" PERGU
NTAS","NUM TOTAL DE ";E;" "
1320 IF M=1 THEN GO TO 1440
1330 PRINT AT 21,0;"N = CICLO SE
GUINTE"
1340 FOR Y=1 TO 10
1350 IF INKEY$="N" OR INKEY$="n"
 THEN GO TO 520
1360 NEXT Y
1370 PRINT AT 21,0;"
   "
1380 FOR Y=1 TO 10
1390 IF INKEY$="N" OR INKEY$="n"
 THEN GO TO 520
1400 NEXT Y
1410 GO TO 1330
1420 SAVE "teste"
1430 PRINT "GRAVAR OUTRA VEZ (S/
N)?"
1440 INPUT W$
1450 IF W$="S" OR W$="s" THEN GO
 TO 1420
1460 LET M=0
1470 GO TO 520
```

11

OUTROS PROGRAMAS COM INTERESSE

Neste capítulo vamos observar diversos programas que, apesar de não estarem directamente relacionados com um qualquer tema, tem interesse no campo educativo. Podem perfeitamente ser usados nas suas aulas.

Histogramas e gráficos de barras

As linhas ou colunas de diferentes comprimentos constituem um modo muito prático de apresentar informações. É mais simples fazê-lo em linhas horizontais, e veremos primeiro um programa que o faz. No entanto, não é muito difícil — usando a instrução PRINT AT do Spectrum — produzir um gráfico em que as barras sejam impressas verticalmente, e é disso que trataremos no nosso segundo programa.

O primeiro indica a frequência de produção de números entre 1 e 20 pelo gerador de números aleatórios do Spectrum. Vejamos a listagem:

```
  10 REM HISTOGRAMAS
  20 DIM A(20)
  25 PRINT "Um momento por favor
"
  30 FOR D=1 TO 300
  40 LET C=INT (RND*20)+1
  50 IF A(C)<30 THEN LET A(C)=A(
C)+1
  60 NEXT D
  70 REM IMPRESSAO
  75 CLS
  80 FOR B=1 TO 20
  90 IF B<10 THEN PRINT " ";
 100 PRINT B;" ";
 110 FOR C=1 TO A(B)
 120 PRINT "=";
 130 NEXT C
 140 PRINT
 150 NEXT B
```

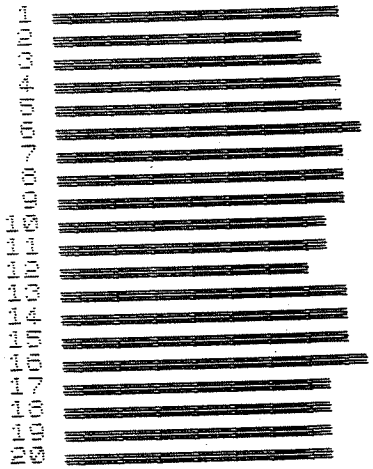
Consideremos agora um exemplo de execução deste programa na sua forma actual:

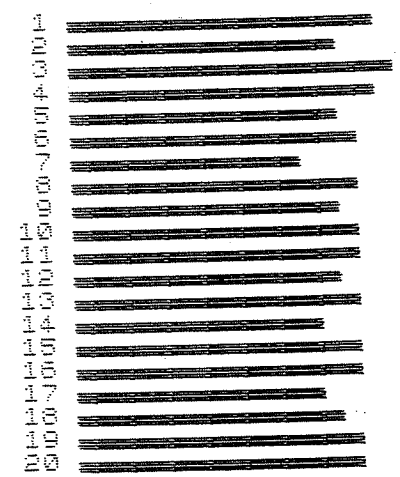
Como o leitor sabe, se o gerador de números aleatórios funcionasse perfeitamente, e realizássemos um número infinito de tentativas, a frequência de cada número seria igual. Reescrevi o programa de tal modo que, em vez de produzir apenas 300 números, produzisse 3000. Vejamos a nova listagem:

```
  10 REM HISTOGRAMAS
  15 REM AMOSTRAGEM MAIOR
  20 DIM A(20)
  25 PRINT "Um momento por favor
"
  30 FOR D=1 TO 3000
  40 LET C=INT (RND*20)+1
  50 IF A(C)<300 THEN LET A(C)=A
(C)+1
  60 NEXT D
  70 REM IMPRESSAO
  75 CLS
  80 FOR B=1 TO 20
  90 IF B<10 THEN PRINT " ";
```

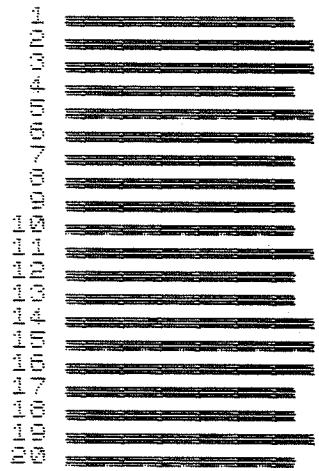
```
100 PRINT B;" ";
110 FOR C=1 TO A(B)/10
120 PRINT "■";
130 NEXT C
140 PRINT
150 NEXT B
```

Observemos agora o resultado de duas execuções do programa. Como se pode verificar, a distribuição dos números aproxima-se mais da distribuição teórica:





Finalmente (e isto exigiu alguma paciência para esperar pelos resultados), modifiquei uma vez mais o programa a fim de produzir 10 000 números, obtendo o histograma seguinte:



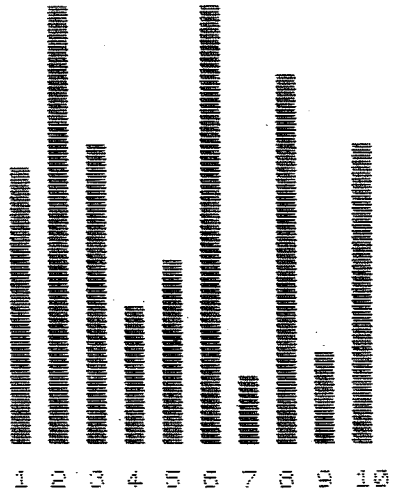
Gráficos de colunas

O programa que se segue produz um gráfico de barras com o aspecto que lhe é habitual, isto é, com a origem no canto inferior esquerdo. É possível escolher o número de colunas de dados que desejamos (até 15), introduzindo depois os dados de cada elemento no gráfico.

Vejamos a listagem:

```
  5 REM GRAFICO DE COLUNAS
  7 INPUT "QUANTAS COLUNAS (MAX
15)? ";Q
  9 IF Q>15 THEN GO TO 7
 10 DIM A(Q)
 20 FOR G=1 TO Q
 30 INPUT A(G)
 35 IF A(G)>19 THEN LET A(G)=19
 40 NEXT G
 50 FOR G=1 TO Q
 55 PRINT AT 21,2*G;G
 60 FOR N=1 TO A(G)
 70 PRINT AT 20-N,2*G;"■"
 80 NEXT N
 90 NEXT G
```

E agora um exemplo de execução:



Pode ter interesse modificar este programa de modo a produzir um gráfico da distribuição relativa dos números produzidos pelo gerador de números aleatórios do Spectrum.

Rotinas de ordenação

Vamos estudar agora três rotinas de ordenação. A primeira limita-se a ordenar números. Se desejamos inverter a ordem da sequência final, basta alterar o símbolo > = da linha 80 para < =:

```
  5 REM ORDENACAO DE NUMEROS
  7 INPUT "QUANTOS NUMEROS? ";N
 10 DIM A(N)
 20 FOR J=1 TO N
 30 INPUT "ESCREVA NUMERO ";(J)
;": ";A(J)
 40 PRINT A(J);" ";
 50 NEXT J
 55 CLS
 60 FOR Z=1 TO N
 70 FOR J=1 TO N-Z
 80 IF A(J)>=A(J+1) THEN GO TO
120
 90 LET T=A(J)
100 LET A(J)=A(J+1)
110 LET A(J+1)=T
120 NEXT J
150 PRINT A(J);" ";
160 NEXT Z
```

A segunda coloca cadeias alfanuméricas por ordem alfabética, como se ilustra no seguinte exemplo de execução:

```
ESTE                TESTE
SERVE               PARA
VERIFICAR           SE
O                   PROGRAMA
DE                  ORDENACAO
EXECUTA             CORRECTAMENTE


UM MOMENTO POR FAVOR


CORRECTAMENTE       DE
ESTE                EXECUTA
O                   ORDENACAO
PARA                PROGRAMA
SE                  SERVE
TESTE               VERIFICAR
```

Vejamos a listagem (e note que as entradas devem ser feitas todas em maiúsculas ou minúsculas; se a primeira letra de qualquer palavra for maiúscula, com o resto da palavra em minúsculas, todas as outras iniciais devem ser igualmente escritas em maiúsculas):

```
  5 REM ORDENACAO ALFABETICA
  7 REM TODAS AS ENTRADAS DEVEM
        SER EM MAIUSCULAS OU EM
        MINUSCULAS
 10 INPUT "Quantas palavras? ";
X
 20 INPUT "Qual e' o compriment
o maximo?    ";Y$: IF Y$="" THEN
LET Y$="15"
 25 LET Y=VAL Y$
 30 DIM A$(X,Y)
 40 FOR N=1 TO X
```

```
  50 INPUT INVERSE 1;"Escreva pa
lavra no. ";(N),A$(N)
  60 PRINT A$(N),
  70 NEXT N
  80 FOR N=1 TO X-1
  90 FOR M=1 TO X-N
 100 IF A$(M)<=A$(M+1) THEN GO T
O 140
 110 LET B$=A$(M)
 120 LET A$(M)=A$(M+1)
 130 LET A$(M+1)=B$
 140 NEXT M
 150 NEXT N
 160 PRINT '"UM MOMENTO POR FAVO
R"
 170 FOR N=1 TO X
 180 PRINT A$(N),
 190 NEXT N
```

Finalmente apresentamos um exemplo de ordenação por idades. São introduzidos no programa dados sobre nomes e idades, e este apresenta-os por ordem de idades. Este tipo de ordenação pode ser facilmente adaptado a qualquer situação em que duas variáveis estejam ligadas entre si e em que desejemos ordenar elementos em função de uma delas:

```
IDADE   NOME

14 Silva
16 Nogueira
12 Amaral
104 Nunes
Nunes                          104
Nogueira                       16
Silva                          14
Amaral                         12
```

```
  5 REM ORDENACAO POR IDADES
  7 REM DE X PESSOAS
 10 INPUT "Quantas pessoas? ";X
 20 DIM N$(X,26)
 30 DIM A(X)
 35 PRINT "IDADE  NOME"
 40 FOR J=1 TO X
 50 INPUT INVERSE 1;"Escreva no
me no. ";(J),N$(J)
 60 INPUT "Idade de ";(N$(J));A
(J)
 70 PRINT A(J);" ";N$(J)
 75 NEXT J
```



```
 80 FOR Z=1 TO X-1
 90 FOR J=1 TO X-Z
100 IF A(J)>=A(J+1) THEN GO TO
170
110 LET T$=N$(J)
120 LET N$(J)=N$(J+1)
130 LET N$(J+1)=T$
140 LET T=A(J)
150 LET A(J)=A(J+1)
160 LET A(J+1)=T
170 NEXT J
180 NEXT Z
190 CLS
200 FOR J=1 TO X
210 PRINT N$(J);A(J)
220 NEXT J
```

Comparação de qualidades diferentes

O programa que se segue surgiu a partir de uma pergunta apresentada na minha coluna de Correio dos Leitores da revista *Your Computer,* quando um leitor da Nova Zelândia afirmou desejar um programa que determinasse as diferenças de tempo entre duas horas do dia (em notação inglesa, por exemplo 9.17 *am* e 3.34 *pm*) e indicasse qual a que ocorre mais cedo (ou mais tarde). O programa final tem pouco valor intrínseco (porque de facto é apenas necessária uma rotina que interprete "AM" e "PM").

No entanto, incluímo-lo aqui porque nos dá uma ideia do modo como podem ser comparadas quantidades diferentes medidas em unidades que não se comparam facilmente. É importante recordar, em qualquer programa que compare quantidades medidas em unidades diferentes (por exemplo horas e minutos) que os valores devem ser reduzidos a uma unidade única. É também necessário definir um formato comum de entradas, a fim de a máquina poder interpretar os dados.

```
 10 DIM A$(2,8): DIM A(2)
 20 FOR Z=1 TO 2
 30 INPUT A$(Z)
 40 LET A(Z)=VAL (A$(Z)( TO 5))
 50 IF A$(Z,7)="P" THEN LET A(Z
)=A(Z)+12
 60 LET B=INT (A(Z))
 70 LET A(Z)=60*B+100*A(Z)-60*B
 80 NEXT Z
 90 IF A(1)>A(2) THEN PRINT A$(
1);" E' MAIS TARDE DO QUE ";A$(2
)
```

```
  100 IF A(1)<A(2) THEN PRINT A$(
1);" E' MAIS CEDO DO QUE ";A$(2)
```

Esta comparação pode ser realizada de um modo relativamente fácil no ZX Spectrum, devido ao seu modo simples de tratar cadeias alfanuméricas (já discutido num capítulo anterior). O programa aqui apresentado para comparar tempos necessita de entradas sob a forma 09.37 AM ou 12.04 PM. Depois (linha 40) determina o valor (sob a forma de um número decimal) do período de tempo considerado, acrescenta 12 ao valor inteiro, à esquerda da vírgula decimal, se o sétimo elemento da cadeia for um "P" e converte o número para minutos, comparando os dois totais em minutos. Usa então as cadeias originais para imprimir a informação final.

Desenho

O interessante programa que se segue, escrito por Gwyn Dewey, ajuda-nos a criar imagens no Spectrum utilizando as teclas indicadas nas linhas 60 a 130 da listagem. As teclas zero a seis comandam a cor. As teclas Q, W, E, A, D, Z, X e C imprimem um pixel, a tecla 0 termina a impressão, a tecla 8 apaga, B controla o brilho e 9 envia uma cópia da imagem para a impressora ZX Printer.

Abandona-se o programa carregando na tecla V, e pode-se gravar a nossa obra de arte em cassete carregando em K ou recuperá-la carregando em J. Note que o programa está preparado para entradas em minúsculas:

```
  1 REM DESENHO, POR G. DEWEY
 10 LET A=100
 15 LET AA=0
 20 LET B=100
 30 LET C=0
 32 LET D=0
 33 LET Z=0
 35 CLS
 40 PRINT "SUPER DESENHO"
 50 REM AUTOR G. DEWEY
 60 PRINT : PRINT "Este program
a ajuda-o a desenharimagens usan
do as seguintes te- clas, NOTA:
em qualquer 'quadradoso' pode hav
eruma cor."
 70 PRINT '"0-6 seleccao de cor
es"
 80 PRINT '"Q,W,E,A,D,Z,X,C des
enhar"
 90 PRINT '"8 apagar"
100 PRINT '"B brilho"
110 PRINT '"9 copiar na ZX Prin
ter"
120 PRINT '"V desistir"
130 PRINT '"K gravar imagem (J
le imagem)"
140 IF INKEY$="" THEN GO TO 140
150 CLS
151 POKE 23658,0
154 INVERSE 0: PLOT a,b
155 INVERSE aa: PLOT a,b
156 LET b$=INKEY$
157 IF b$="" THEN GO TO 154
160 LET a=a-(b$="q" OR b$="a" O
R b$="z")+(b$="e" OR b$="d" OR b
$="c")
170 LET b=b-(b$="z" OR b$="x" O
R b$="c")+(b$="q" OR b$="w" OR b
$="e")
210 IF b$>="0" AND b$<="7" THEN
INK (VAL b$)
215 IF b$="8" THEN LET aa=1
216 IF b$="0" THEN LET aa=0
220 IF b$="b" AND d=0 THEN LET
d=1
230 IF b$="b" AND d=1 THEN LET
d=0
240 BRIGHT d
250 IF b$="9" THEN COPY
260 IF b$="v" THEN GO TO 9100
270 IF b$="k" THEN SAVE "imagem
"SCREEN$
280 IF b$="j" THEN LOAD "imagem
"SCREEN$
290 GO TO 154
```

Ângulo interior de um polígono regular

O programa seguinte determina o valor do ângulo interior de um polígono regular, arredondando a resposta em caso de necessidade (veja o exemplo para o polígono de sete lados) para duas casas decimais. É interessante experimentá-lo para um polígono imaginário tendo apenas um ou dois lados:

```
Um poligono regular de 3 lados
tem angulos interiores de 60

Um poligono regular de 4 lados
tem angulos interiores de 90
```

```
Um poligono regular de 5 lados
tem angulos interiores de 108

Um poligono regular de 6 lados
tem angulos interiores de 120

Um poligono regular de 7 lados
tem angulos interiores de 128.57

Um poligono regular de 8 lados
tem angulos interiores de 135

Um poligono regular de 9 lados
tem angulos interiores de 140
```

```
  10 REM ANGULO INTERNO DE UM
       POLIGONO REGULAR
  30 INPUT "Quantos lados? ";LAD
OS
  40 LET LADOS=INT (LADOS)
  50 LET ANGULO=INT (100*(180-36
0/LADOS))/100
  60 PRINT '"Um poligono regular
 de ";LADOS;" lados"
  70 PRINT "tem angulos interior
es de ";ANGULO
  80 GO TO 30
```

Depreciação

Este programa é bastante explícito nas perguntas que faz ao utilizador. Note que a linha 280 (POKE 23692,-1) leva o computador a deixar de perguntar *Scroll?* quando o visor se encontra cheio. Vejamos como funciona o programa:

```
O preco de compra e' 48976.7$

A vida do produto e' 12 anos

Deprecia 4081.39$ por ano

Ano          Valor
1984        48976.7$
1985        44895.31$
1986        40813.92$
1987        36732.53$
1988        32651.14$
1989        28569.75$
1990        24488.36$
1991        20406.97$
1992        16325.58$
1993        12244.19$
1994        8162.8$
1995        4081.41$
```

E agora a listagem:

```
  10 REM DEPRECIACAO RECTA
  30 LET ANO=0
  50 INPUT "Indique preco de com
pra ";PRECO
  80 PRINT '"O preco de compra e
' ";PRECO;"$"
 100 INPUT "Indique duracao do p
roduto (em  anos) ";VIDA
 130 PRINT '"A vida do produto e
' ";VIDA;" anos"
 140 LET DEPREC=(INT (PRECO*100/
VIDA))/100
 150 PRINT '"Deprecia ";DEPREC;"
$ por ano"
 180 INPUT "Indique primeiro ano
 de uso ";ANO
 220 PRINT '"Ano";TAB 12;"Valor"
 240 PRINT ANO;TAB 11;PRECO;"$"
 250 LET PRECO=PRECO-DEPREC
 260 IF PRECO<1 THEN STOP : REM
ou COPY: STOP
 270 LET ANO=ANO+1
 280 POKE 23692,-1
 290 GO TO 240
```

Dia da semana

Esta simples rotina determina a que dia da semana corresponde uma determinada data, como se pode verificar no seguinte exemplo de execução:

```
19/12/84 - Qua

30/4/83 - Sab

25/12/83 - Dom
```

```
1/1/99 - Ter

14/6/37 - Seg

6/1/48 - Sab

13/7/44 - Qui
```

O programa é o seguinte:

```
  10 REM DIAS DA SEMANA
  20 LET A$="..SegTerQuaQuiSexSa
bDom"
  30 INPUT "Indique mes (p.ex.,
7): ";M
  40 IF M<1 OR M>12 THEN GO TO 3
0
  50 INPUT "Indique dia: ";D
  60 IF D<1 OR D>31 OR M=2 AND D
>29 THEN GO TO 50
  70 INPUT "Indique ano: ";A
  80 LET Q=A+(M<3)
  90 LET K=Q/100
 100 LET T=M-12*(M<3)
 110 LET R=INT (13*(T+1)/5)+INT
(5*Q/4)-INT (K)+INT (K/4)+D+5
 120 LET R=R-(7*INT (R/7))+1
 130 PRINT ''TAB 6;D;"/";M;"/";A
-1900;" - ";A$(R*3 TO R*3+2)
 140 GO TO 30
```

Relógio de segundos

De facto, o rigor deste relógio impede a sua utilização a não ser para efeitos de demonstração. O ponto que indica o tempo desloca-se no interior do círculo formado pelos números, levando aproximadamente um minuto para dar uma volta completa.


```
                60
        55          5

 50                     10

45                       15

 40                     20

        35          25
                30
```

O programa imprime os números, parando em seguida (PAUSE 0, na linha 40) à espera de que o utilizador carregue em qualquer tecla. Começa em seguida a contar os segundos. Se quiser obter um relógio mais rigoroso, reduza o 39 na linha 100 para 38, e acrescente uma linha 105 em que o computador realize qualquer cálculo (por exemplo determinar uma potência de um número), o que será suficiente para o tornar mais lento. Talvez os seus alunos se divirtam tentando calibrar convenientemente o relógio.

Vejamos agora a listagem do programa:

```
   5 REM RELOGIO DE SEGUNDOS
  10 FOR N=0 TO 60 STEP 5
  20 PRINT AT 10-10*COS (N/30*PI
),10+10*SIN (N/30*PI);N
  30 NEXT N
  40 PAUSE 0
  50 FOR T=0 TO 59
  60 PLOT 88+59*SIN (T/30*PI),90
+61*COS (T/30*PI)
 100 PAUSE 39
 110 PLOT OVER 1;88+59*SIN (T/30
*PI),90+61*COS (T/30*PI)
 120 NEXT T
 130 GO TO 50
```

Média, desvio padrão, variância

No final deste capítulo apresentamos um programa que determina a média, o desvio padrão e a variância. O programa faz perguntas óbvias ao utilizador, e a sua saída é simples de interpretar. Talvez o leitor deseje acrescentar uma rotina semelhante à usada no programa que analisa depreciações para limitar a saída a duas casas decimais: Vejamos um exemplo de execução:

```
24                4
25                3
26                2
23                3
22                2
21                1
27                1

A media e' 24
A variancia e' 2.5
O desvio padrao e' 1.5811388


230               7
228               9
235               3
236               1
225               2

A media e' 229.68182
A variancia e' 8.853302
O desvio padrao e' 2.9754499
```


E agora o programa:

```
  10 REM MEDIA, DESVIO PADRAO
  20 REM VARIANCIA
  35 INPUT "Quantos elementos? "
;N
  40 DIM A(N)
  50 DIM B(N)
  80 FOR Z=1 TO N
  90 POKE 23692,-1
 100 INPUT "Indique elemento ";(
Z);" ";A(Z)
 140 PRINT 'A(Z),
 180 INPUT "Indique frequencia "
;B(Z)
 190 PRINT B(Z)
 220 NEXT Z
 270 LET M=0
 280 LET A=0
 290 LET B=0
 300 FOR Z=1 TO N
 310 LET M=M+B(Z)
 320 LET A=A+B(Z)*A(Z)
 330 LET B=B+B(Z)*A(Z)*A(Z)
 340 NEXT Z
 380 PRINT '''A media e' ";A/M
 390 PRINT '"A variancia e' ";B/
M-A*A/(M*M)
 400 PRINT '"O desvio padrao e'
";SQR (B/M-A*A/(M*M))
 410 STOP
1000 LET P=0
1010 LET O=0
1020 LET L=0
```

12

SOFTWARE EDUCATIVO PARA O SPECTRUM

O Spectrum é uma máquina deliciosamente simples, mas potente, para uso e escrita de programas. Infelizmente, o mercado não parece suficientemente extenso para atrair as principais casas produtoras do software para a comercialização de programas educativos; limitam-se normalmente a programas de jogo ou de tipo utilitário. Existem em Inglaterra, no entanto, vários programas do tipo "escolha múltipla", de valor educativo normalmente duvidoso e de qualquer modo pensados para utilizadores de língua inglesa, e alguns simuladores de boa qualidade.

Torna-se portanto necessário que os próprios professores escrevam o software mais indicado para cada caso. O ideal seria que os professores passassem a programadores, e estes a professores! Entretanto, os professores tendem a adaptarem-se à nova tecnologia tão depressa quanto possível. Uma dificuldade é que a vasta maioria dos professores, como o leitor certamente compreende, são novos-vindos à arte de programar e começam apenas a desenvolver as suas capacidades neste campo. É demasiado esperar deles que produzam programas tão complexos e fiáveis que possam ter uso geral.

Por outro lado, é pena que muitas ideias excelentes se percam apenas porque o programa não tem sofisticação, não contendo validações das entradas (o que é discutido noutro ponto deste livro) ou por deficiências na concepção das saídas. Muitos outros professores não se interessam de modo algum pela programação, só lhes interessando o computador nos mesmos termos em que lhes interessaria um projector de slides ou um gravador vídeo. Outros ainda mantêm-se cépticos em relação à própria utilidade de um computador na sala de aula.

Nestas condições, de onde será de esperar que surjam programas educativos num futuro próximo? Existem já algumas firmas de software que produzem bons programas educativos para o Spectrum, muitos deles escritos por professores experientes, mas infelizmente todas essas casas são inglesas.

Como professor, compete-lhe a si decidir se o Spectrum é ou não útil em determinadas condições, e se existe algum programa comercial utilizável no campo que lhe interessa. Como auxiliar na sala de aula, o programa deve obviamente ser pelo menos tão bom como o método que vai substituir, quer se trate de exemplificar um aspecto definido ou de verificar os conhecimentos dos alunos em determinada área. E aqui surge um outro problema: poderá toda a aula participar simultaneamente no uso do programa?

No caso de muitas escolas que apenas podem possuir um pequeno número de microcomputadores, ou onde apenas um professor usa a máquina por se ter decidido a fazê-lo, apesar da falta de apoio de que dispõe, será difícil evitar a tentação de os usar para grandes grupos de alunos. Para permitir a uma classe inteira usar um mesmo computador simultaneamente, é necessário um visor que seja observado claramente por todos aqueles a quem o tema deve interessar.

É muito melhor usar um micro quando este pode fazer algo que nenhum outro método de ensino pode. Como já disse na introdução a este livro, as potencialidades do Spectrum no uso da cor, assim como na construção de gráficos de alta resolução, tornam-no ideal para produzir e armazenar todos os tipos de programas complexos. Isto pode ser muito útil em particular em temas como a Biologia, por exemplo. É muito útil dispor de diagramas animados, particularmente quando a animação é comandada pelo professor e pode ser parada em qualquer momento que justifique uma explicação. Neste livro é incluído um exemplo simples explicando o processo da fotossíntese, e dando uma ideia do tipo de programa que provavelmente mais interessará ao leitor.

Depois de ter decidido que é conveniente usar o computador na aula, quais as características do programa, comercial ou não, a utilizar?

1. O *objectivo* do programa deve ser claramente definido. A documentação deve conter informações quanto à gama de idades a que se aplica, ao nível de conhecimentos que os assistentes deverão ter, ao *tipo* de programa de que se trata, isto é, se é interactivo, requerendo dados ou respostas fornecidos pelos alunos, ou se é apenas demonstrativo. Pode igualmente tratar-se de um programa de teste para uso de um único aluno de cada vez.

2. A documentação deve indicar claramente qual a máquina a usar (16 K ou 48 K), e se é necessário dispor ou não de uma impressora.

3. Deve explicar claramente o modo como deverá ser usado, começando pela forma de o carregar no computador.

4. O programa propriamente dito deve estar completamente validado, não admitindo qualquer tipo de erros nas entradas. O facto de o programa aceitar um tipo errado de resposta — por exemplo, permitir a entrada de letras quando é necessário um dado numérico — pode tornar-se bastante desagradável numa aula, sendo semelhante a uma película que se rasgue durante a projecção.

5. Um programa deve, se possível, poder ser adaptado às necessidades concretas de uma determinada classe.

Uma fonte bastante útil de programas educativos são revistas, como a *ZX Computing* ou a *Sinclair User*. Estas contêm muitas vezes programas que já foram usados em salas de aula.

Perguntas que deve fazer a si mesmo

— Será este o momento apropriado, na fase de ensino em que se encontra, para usar o computador?

— O programa escolhido tem algum interesse do ponto de vista do esquema de ensino usado?

— O visor pode ser observado claramente por todos?

— O programa é acompanhado por uma documentação clara e suficiente?

— O programa é fácil de usar? Não permite erros nas respostas?

— Pode ser adaptado às suas necessidades?

13

USO DO SPECTRUM NA ESCOLA PRÉ-PRIMÁRIA ESTUDOS DE UM CASO

Christine S. Johnson é professora da Carlton Netherfield Infant and Nursery School em Netherfield, Nottingham. Tem usado o Spectrum (e o ZX81) com crianças entre os cinco e os sete anos e meio de idade. Neste capítulo explica-nos como usa o computador e qual é a aceitação deste entre os alunos.

Pergunta a um adulto o que recorda da sua primeira aula. Provavelmente descreverá uma sala grande, com paredes brancas e um tecto muito alto. Talvez houvesse degraus à entrada da sala, e é possível que as carteiras, em madeira, estivessem aparafusadas ao chão. Provavelmente a sala era fria no Inverno. Todas as crianças dispunham de pequenas lousas e pedaços de giz, e um pano para "apagar".

A situação era completamente diferente das salas de aula de hoje, coloridas, com imagens agradáveis e visores animados... Encorajam-se as crianças a fazerem perguntas e a descobrirem coisas sozinhas, a usarem livros atractivos e aparelhos excitantes.

Há mais de trinta anos os computadores eram extremamente raros; tratava-se de máquinas enormes, complicadas, caríssimas, e com muitas limitações de uso. Actualmente existem muitos computadores, são pequenos, baratos e extremamente versáteis. O seu impacto no futuro do ensino é impossível de prever. O progresso é tão rápido que quando as crianças de hoje chegarem a adultos quase tudo neste mundo se terá completamente alterado, e os computadores que conhecemos ter-se-ão transformado em peças de museu.

Hoje, no entanto, são fascinantes e extremamente úteis, e não é justo não dar às crianças uma oportunidade de aprenderem a utilizá-los.

A maior parte das pessoas concorda em que não é possível que todas as crianças sejam capazes de escrever programas complicados

aos sete anos de idade. No entanto, se forem ajudadas no uso do teclado, se lhes forem dadas instruções precisas, se forem encorajadas a aprender uma linguagem simples num computador fácil de usar e recorrendo a programas que tenham a ver com o programa a cumprir, podem interessar-se muito mais pelo estudo quando atingirem níveis de ensino mais avançados.

Os principais objectivos são aqui o encorajamento e a diversão. As crianças de hoje gostam dos computadores e estão dispostas a aceitarem toda a informação e auxílio que lhes possam ser oferecidos neste campo.

Imagine que tem seis anos de idade... Vê televisão todas as noites e gosta do que vê, mas na manhã seguinte já esqueceu tudo porque não está directamente envolvido naquilo que viu. Um dia, o professor mostra-lhe um teclado ligado a uma televisão, e diz-lhe como o pode usar.

Pede-lhe que toque em algumas teclas e que descubra as letras que formam o seu nome. Ajuda-o a tocar em mais algumas teclas de significado estranho e de súbito o seu nome aparece vinte vezes repetido no visor.

```
10 BORDER 4
20 PAPER 7
30 PRINT "Ana Maria"
40 PAUSE 80
50 GO TO 30
```

```
Ana Maria
Ana Maria
Ana Maria
Ana Maria
Ana Maria
Ana Maria
Ana Maria
Ana Maria
Ana Maria
Ana Maria
Ana Maria
Ana Maria
Ana Maria
Ana Maria
Ana Maria
Ana Maria
Ana Maria
Ana Maria
Ana Maria
Ana Maria
Ana Maria
Ana Maria
```


Outras crianças estão a ver, e querem também experimentar. Daí a pouco o professor não precisa de dar qualquer ajuda, porque toda a gente está a dar instruções: "Não te esqueças das aspas!", "Carrega aqui no fim da linha", etc.

Uma criança carrega numa tecla durante demasiado tempo, e o visor fica cheio de letras. Todos ficam preocupados, mas o professor diz: "Podemos sempre corrigir os erros" e ensina como carregar em duas teclas para apagar as letras a mais. Aproveita para ensinar como se pode obter uma imagem de fundo colorido.

Mais tarde o professor mostra um papel com instruções simples, e pede-lhe a si e aos seus colegas que ajudem outro grupo de crianças.

Quando o dia termina você corre para casa e diz à sua mãe: "Já sei mexer num computador!"

Que aprenderam as crianças na sua primeira lição?

1 — Que podem mexer num computador e que é divertido fazê-lo.
2 — Que existe um "código de segurança":
   — Só uma pessoa pode mexer no teclado de cada vez, se bem que as outras a possam ajudar a ler o programa ou a descobrir as letras e símbolos no teclado.
   — Ninguém pode desligar os cabos ou tocar nos interruptores (tudo é ligado pelo professor antes de a aula começar).
   — Quando surge um problema, é preciso dizer ao professor.
   — Quem usar incorrectamente o computador, ou incomodar alguém que o esteja a usar, será proibido de mexer na máquina durante alguns dias.
3 — Não é preciso ficar preocupado quando se faz um erro.
4 — As instruções são dadas ao computador em linhas, todas começando por um número.
5 — Pode-se mudar de letra maiúscula para minúscula ou vice-versa carregando numa tecla.
6 — Pode-se deixar espaço entre as palavras.
7 — É importante usar aspas depois da instrução que faz imprir as letras.
8 — No fim de cada linha deve-se carregar numa tecla.
8 — Podem-se alterar as instruções a fim de obter efeitos diferentes movendo um cursor sobre elas.
10 — Saber explicar a outros como se usa o computador.

Como tudo começou

Tudo começou em 1981. A propósito de um projecto de matemáticas enchemos uma caixa de "coisas interessantes". Um grupo de crianças mostrou grande interesse em coisas como ímans e lupas, e formámos uma espécie de clube. Dentro em pouco passámos de um telefone feito com caixas de iogurte e um pedaço de corda para algo mais sofisticado que ligava um corredor à sala da Directora.

O nosso simples automóvel para empurrar depressa se transformou num veículo a pilhas que mudava de direcção sempre que uma espécie de antena tocava numa parede.

Todas as crianças da escola podiam entrar para o clube. Assinavam os nomes num pequeno livro usavam pequenos distintivos.

Ao fim de algum tempo, a caixa já não chegava para conter a nossa colecção à medida que acrescentávamos coisas, como interruptores, pilhas, fios e pequenas lâmpadas, instrumentos musicais feitos em casa e até (muito importante) duas máquinas de escrever portáteis.

Desenvolvemos também uma livraria de obras científicas para crianças, e por isso tínhamos um armário grande. Retirámos as portas para que tudo fosse fácil de tirar.

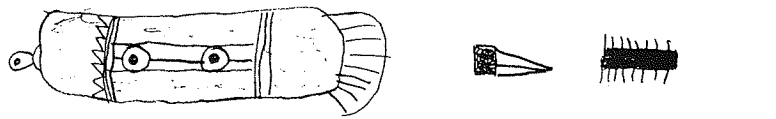
Adquirimos algumas válvulas e um reparador de televisões deu-nos um transistor e um circuito integrado, e daqui nasceu uma discussão interessante. Pedi a uma criança que acendesse e apagasse uma lâmpada, e afirmei que a válvula podia fazer o mesmo centenas de vezes mais depressa e um transistor ainda mais.

Olhámos para um televisor enorme e depois para outro de pequenas dimensões. As crianças gostaram de saber que quando os pais deles tinham a mesma idade os computadores tinham o tamanho de uma sala.

As crianças tinham igualmente acesso a um gravador. Depois de as máquinas de escrever terem sido usadas durante algum tempo montámos o televisor, um computador e uma impressora.

A única explicação que dei às crianças foi que o computador é incapaz de perceber a diferença entre um O maiúsculo e um zero, pelo que este devia ser escrito com um traço (0) para não se confundir com aquele. Fizemos uma rápida contagem decrescente para lançar um foguetão — 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, zero — e a explicação foi aceite.

Júlia Morais



Isto é uma válvula Isto é um transistor Isto é um circuito integrado

Estes são os tamanhos da válvula, do transistor e do circuito integrado.

Exemplificou-se o uso da tecla CAPS SHIFT e as crianças ocuparam-se alternadamente da escrita de frases curtas no teclado. Consideraram a experiência bastante divertida, e fácil na medida em que não custa corrigir os erros. Os seus esforços foram registados pelo professor na impressora. Uma cópia para a escola, e outra para levar para casa e mostrar aos pais.

```
Ana  Tenho 8 anos

Hoje brincamos com o computador.
O Joao sabe mexer nele e o Pedro
tambem. Eu escrevi nele e gostei
muito. A Teresa ficou a ver.
```

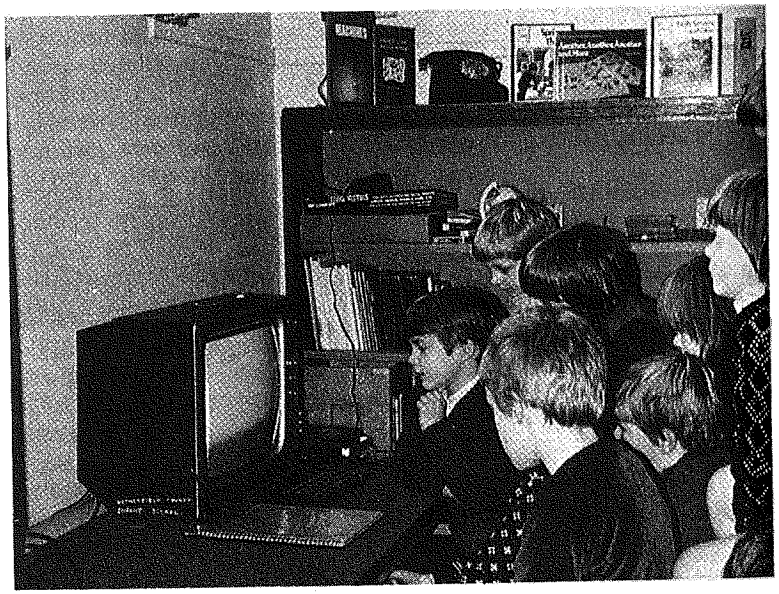
Além de usar o computador como uma interessante máquina de escrever, ensinou-se às crianças a "escreverem" programas simples na máquina (se bem que começassemos com um ZX81, passámos a um Spectrum assim que este ficou disponível).

Uso de caixas de arquivo

Temos duas grandes caixas de arquivo. Uma é chamada "Coisas

Interessantes" e a outra "O Computador". Todas as imagens ou texto escrito produzidos pelas crianças são guardados nestes caixas de arquivo.

```
   Alice    Tenho 7 anos
Temos um computador na nossa
sala de aula e fazemos muitas
coisas nele. Somamos numeros e
fazemos jogos. Escrevemos os
nossos nomes e alguns dos
rapazes mais crescidos fizeram
uma mensagem que se mexe na
televisao de um lado para o
outro.
```



Fichas de trabalho

Como se pode esperar que crianças muito jovens produzam algum trabalho quando enfrentam um teclado com 40 teclas e quase 200 letras, números, palavras e símbolos (com um mínimo de ajuda de um professor ocupado), e em particular quando a primeira coisa que querem é "mexer" nele?

É possível fazer um conjunto de fichas que ajudarão as crianças a escreverem programas desde início.

Como o leitor poderá verificar pelas fichas apresentadas em seguida, a maior parte das instruções são directas, exceptuando o facto de TO, = e outras serem escritas dentro de pequenas caixas, com a palavra SYMBOL na metade superior a fim de recordar à criança que deve carregar na tecla SYMBOL SHIFT e mantê-la premida enquanto carrega na tecla correspondente ao símbolo.

ESCREVA O SEU NOME

1Ø	BORDER 4	ENTER
2Ø	CAPS / SYMBOL + CAPS / PAPER 7	ENTER
3Ø	PRINT SYMBOL " Escreva o seu nome SYMBOL "	ENTER
4Ø	PAUSE 8Ø	ENTER
5Ø	GOTO 3Ø	ENTER
	RUN	ENTER

Para limpar o visor para a pessoa seguinte

CAPS / BREAK NEW ENTER

OUÇA

10 FOR X [SYMBOL =] 0 [SYMBOL TO] 24 ENTER

20 [CAPS BEEP] [SYMBOL SYMBOL] 1 [SYMBOL 9] X ENTER

30 NEXT X ENTER

RUN ENTER

Carregue outra vez em RUN ENTER.

Quantos sons consegue ouvir?

DELETE é indicada de modo semelhante, mas tendo a palavra CAPS na metade superior da caixa. Instruções mais difíceis como BEEP e PAPER são indicadas sob a forma de uma caixa dupla, recordando à criança que deve carregar primeiro em CAPS SHIFT e SYMBOL SHIFT, e depois em BEEP e SYMBOL SHIFT.

Os pequenos dedos das crianças conseguem normalmente dominar o teclado nas duas primeiras lições (David e Dale têm 6 anos de idade e escreveram os seus nomes — obtendo-os na impressora — apenas alguns minutos depois de terem visto pela primeira vez o Spectrum).

Nas primeiras fases escreve-se ENTER no final de cada linha, mas dentro em pouco é possível omitir esta indicação à medida que a criança ganha experiência. Ao fim de algumas semanas as crianças conseguem introduzir no computador alguns dos programas mais simples apresentados em livros e revistas.

VEJA AS CORES

10 FOR X [SYMBOL =] 0 [SYMBOL TO] 7 ENTER

20 BORDER X ENTER

30 [CAPS CAPS] [SYMBOL PAPER] 7 [SYMBOL —] X [SYMBOL :] [CAPS CLS] ENTER

40 PAUSE 150 ENTER

50 NEXT X ENTER

60 GOTO 10 ENTER

Para ver outra vez as cores carregue em RUN ENTER

```
10 FOR X=0 TO 7
20 BORDER X
30 PAPER 7-X: CLS
40 PAUSE 150
50 NEXT X
60 GO TO 10
```

Além das "fichas de trabalho", é útil dispor de um outro conjunto de fichas contendo informações para uso das crianças, como as apresentadas em seguida.

De início terão dificuldade em descobrir no teclado algumas das palavras e símbolos, apesar de não a terem na descoberta de algarismos e letras em particular no caso de terem tido alguma experiência com uma máquina de escrever. A ficha mais simples indica as palavras e símbolos mais vulgarmente usados. Muito mais tarde, a criança terá acesso a fichas indicando todas as palavras e símbolos que podem ser usados no computador.

PALAVRAS e SÍMBOLOS

AND	Y	INPUT	I	"	P
AT	I	INVERSE	M	.	M
BEEP	Z	LET	L	,	N
BORDER	B	LIST	K	;	O
BRIGHT	B	LLIST	V	:	Z
CAPS LOCK	2	LOAD	J	+	K
CIRCLE	H	L PRINT	C	-	J
CLS	V	NEW	A	*	B
CONT	C	NEXT	N	/	V
COPY	Z	PAPER	C	=	L
DELETE	Ø	PAUSE	M		
EDIT	1	PRINT	P	?	C
FLASH	V	RAND	T	£	X
FOR	F	REM	E		
GOTO	G	RETURN	Y		
GRAPHICS	9	RND	T		
IF	U	RUN	R		
INK	X	STOP	A		
INKEY$	N	THEN	G		
		TO	F		

As crianças gostam de guardar estas fichas consigo e de fornecerem indicações aos outros quando necessário.

Outras fichas que poderão ser úteis são:

— Instruções para introdução de um programa na máquina a partir de um gravador.

— Modo de apagar ou alterar as linhas de um programa.

— Utilização dos cursores.

— Lista das cores disponíveis.

— A prioridade das funções numéricas como (,),↑,*,/,+,—, para as crianças um pouco mais velhas.

JOGO PAREDE

LOAD [SYMBOL "] PAREDE [SYMBOL "] ENTER

Carrega em PLAY, no gravador, e observa o visor até veres PARAR O GRAVADOR.

Carrega em STOP no gravador.

Para jogar carrega em qualquer tecla do computador

Carrega em CAPS SHIFT e usa
O para a esquerda ←
P para a direita →

Escreve a tua pontuação

Carrega em S para jogares outra vez

Quando as crianças jogam com o computador, é útil dispor de uma ficha para cada jogo a fim de recordarem quais as teclas em que devem carregar — por exemplo O para fazer o vampiro andar para a esquerda, ou P para o deslocar para a direita.

É útil que todas as cassettes e fichas de trabalho estejam convenientemente identificadas com pequenas etiquetas coloridas, a fim de poderem ser agrupadas em conjuntos apropriados, por exemplo:

1. Amarelo — prática do computador
2. Verde — trabalho da escola.
3. Vermelho — jogos de computador.
4. Azul — Indicações do professor, indicações de gravação.

Pensa numa soma e verifica a resposta. +

PRINT carrega num número SYMBOL + carrega num número ENTER

A resposta surgirá no topo do visor. Acertaste? Experimenta outra vez.

Quando tiveres muitas respostas carrega em NEW e deixa um colega brincar também.

O grupo dois pode ser subdividido em temas escolares, como matemática, português, etc.

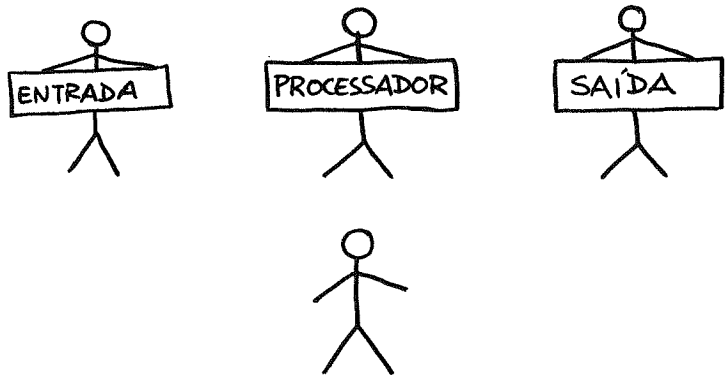
Podem-se escrever letras e algarismos nas etiquetas a fim de as crianças se encarregarem da sua própria evolução. Por exemplo, se a criança tiver terminado as fichas A1, A2, A3, etc., pode passar às fichas B1, B2, B3...

Quando adquirir o Spectrum, escrevi as instruções de uso do gravador de cassettes — tanto para SAVE como para LOAD — e coloquei uma lista completa de palavras-chave e símbolos junto ao computador. Senti-me assim muito mais confiante, tendo mais garantias de que não ficaria à frente de pais e alunos a olhar para os cabos sem saber o que fazer.

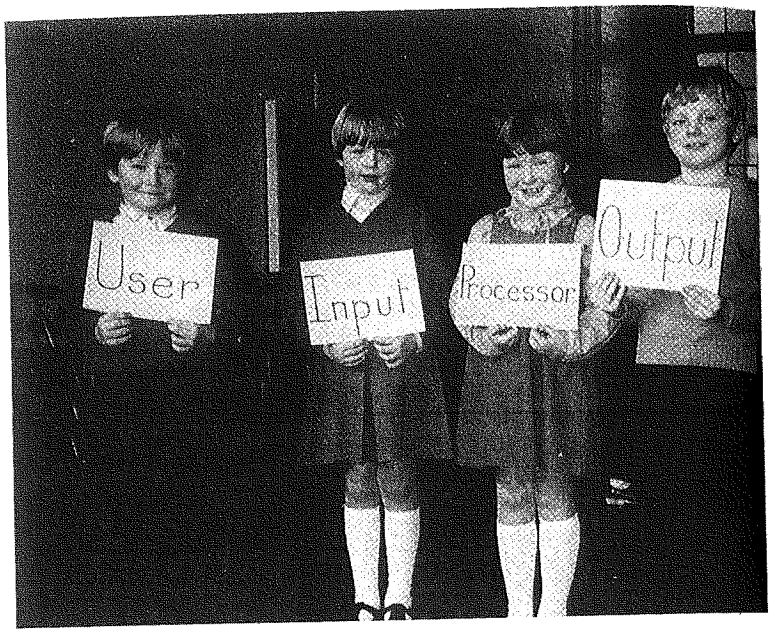
O jogo do computador

O "jogo do computador" constitui uma actividade de toda a turma, capaz de ensinar as crianças a usarem o computador com a maior facilidade.

Peça às crianças que se juntem em grupos de quatro. Uma delas será o "utilizador" e manter-se-á à frente das outras três, cada uma delas segurando num cartão:



O "utilizador" só pode falar para a "entrada", mas pode fazer a pergunta que quiser. Por exemplo: "Se eu tiver quatro doces e comer dois, com quantos fico?". A "entrada" pergunta ao "processador" o mesmo. Este deve descobrir a resposta enquanto salta na mesma posição. Depois passa a resposta a "saída", que por sua vez a diz ao "utilizador". As crianças gostam deste jogo, e podem praticá-lo com o ritmo que desejarem, mudando ocasionalmente as posições. É interessante ver os "processadores" fazendo as suas danças enquanto descobrem as respostas... O professor move-se de grupo para grupo verificando se tudo está bem.



O jogo do Byte

Um outro jogo bastante apreciado pelas crianças, tem a vantagem de vir a ser-lhes muito mais útil quando forem mais crescidos.

Cada criança passa a chamar-se um "bit" e desloca-se na sala ao som de música. Quando a música pára, todas tentam formar linhas de oito. Podem escolher ficar em pé, formando um byte "de alta tensão" (ou 1) ou sentar-se e formarem um byte "de baixa tensão" (ou 0). As crianças do primeiro grupo que conseguem formar o byte recebem uma ficha e a música recomeça para novo jogo.

No final da sessão a criança com mais fichas será o vencedor.

Até hoje não dei qualquer outra indicação sobre o sistema binário às crianças muito novas, excepto a menção de que um computador com uma memória de 1 K tem um pouco mais de 1000 bytes e que o nosso computador tem uma memória de 16 K.

Guardar o equipamento

Certos computadores devem ser deixados na mesma posição continuamente, mas o Spectrum pode ser transportado e usado com qualquer televisor (normalmente a cores, mas alguns programas dão bons resultados a preto e branco se o televisor a cores está a ser já usado).

Quando as pessoas recebem um computador pela primeira vez, guardam normalmente cada peça com todo o cuidado nas caixas originais. É certamente o método ideal, mas pode obrigar à perda de muito tempo numa escola.

Nem sempre é conveniente que o equipamento esteja montado durante longos períodos de tempo, e é positivo guardar tudo ao fim do dia.

Uma forma fácil de guardar o computador, a impressora, o gravador e algumas cassetes consiste em usar uma caixa de cartão forte, de dimensões médias, obtida num supermercado. Experimente com caixas de diferentes dimensões até descobrir uma onde possa guardar tudo o que usa normalmente. Coloque no fundo pedaços de espuma de modo a que cada peça fique convenientemente apoiada. Será provavelmente possível manter a impressora e alguns dos fios permanentemente ligados.

Um pedaço grande de cartão, também coberto com feltro, poderá ser usado como base onde colocar o equipamento em cima de uma mesa. As fichas de trabalho podem ser guardadas em caixas transparentes ao lado.

Medalhas

Todas as crianças gostam de usar medalhas, autocolantes, etc. Porque não dar-lhes esse prazer quando atingem um certo nível no uso do computador?

Uma criança de seis ou sete anos de idade sentir-se-á orgulhosa se puder usar uma medalha feita em casa que lhe tiver dado algum trabalho a obter — uma medalha como a da figura 20.

Para a ganhar poderá ter:

1. Copiado um programa de um ficha.
2. Carregado um programa da cassete para o computador.
3. Usado a impressora (esta e o gravador devem ser ligados ao computador no início de cada sessão).

Se existir um clube na escola, deve-se fazer uma medalha para cada membro.

Um professor não pode evidentemente estar todo o seu tempo junto a um computador. Não há porém qualquer problema nisso, dado que depois de as crianças terem ganho experiência no uso das fichas de programação e na forma de carregar um programa das cassetes podem ser deixadas sozinhas, seguindo as instruções escritas. Por vezes deixa-se um programa no computador durante toda uma sessão, podendo ser usado para dar instruções às crianças, como no exemplo que se segue:

```
Escreve sobre a tua visita aos
bombeiros.

 Palavras que te podem ajudar:

    carro dos bombeiros   alarme

escada          uniforme    agua

    botas     mangueira   machado

 capacete         perigo       fogo

        cuidado           seguro
```

Isto ajuda as crianças a aprenderem a ler e escrever, e é fácil mudar as palavras impressas no visor.

14

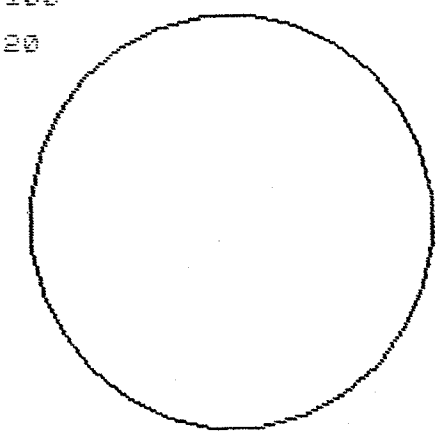
APLICAÇÕES ESPECÍFICAS NA ESCOLA PRÉ-PRIMÁRIA

Neste capítulo esboçamos algumas ideias concretas para uso do Spectrum nas escolas pré-primárias.

Escrita manual

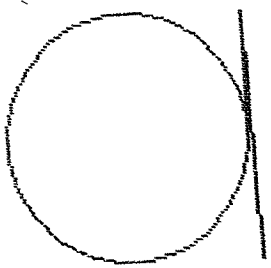
Algumas crianças têm dificuldade em formar as letras correctamente. Se lhes for pedido que introduzam no computador o programa bastante simples "UM Ó MÁGICO", podem depois seguir a formação da letra "O" quando surge no visor. Verá que o "O" é formado de um modo que mostra a direcção da linha (no sentido contrário aos ponteiros do relógio) com muita clareza. Uma criança mais velha ou um dos pais podem escrever este programa para a criança:

```
10 REM CIRCULO MAGICO
20 CIRCLE 100,100,75
30 PAUSE 100
40 OVER 1
50 GO TO 20
```

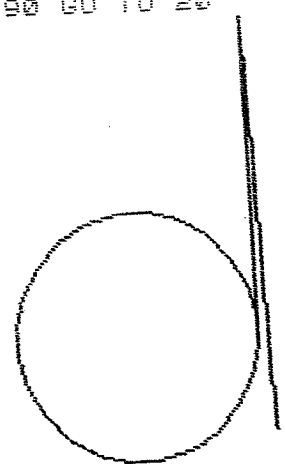


Podem-se apresentar depois deste os programas para a letra "a" ou "d":

```
10 REM LETRA A
20 CIRCLE 50,50,45
30 PAUSE 50
40 DRAW 0,50
50 PAUSE 80
60 DRAW 3,-90
70 PAUSE 200
80 CLS
90 GO TO 20
```



```
10 REM LETRA D
20 CIRCLE 50,50,45
30 PAUSE 30
40 DRAW 0,120
50 PAUSE 80
60 DRAW 5,-150
70 PAUSE 200
80 CLS
90 GO TO 20
```



Vejamos a ficha de trabalho para o programa da letra *d*:

FAZER A LETRA d

2Ø	CAPS / CAPS	SYMBOL / CIRCLE	5Ø	SYMBOL / ,	5Ø	SYMBOL / ,	45	ENTER
3Ø	PAUSE 3Ø							ENTER
4Ø	DRAW Ø, 12Ø							ENTER
5Ø	PAUSE 8Ø							ENTER
6Ø	DRAW 5	SYMBOL / ,	SYMBOL / —	15Ø				ENTER
7Ø	PAUSE 2ØØ							ENTER
8Ø	CLS							ENTER
9Ø	GOTO 2Ø							ENTER

Segue a forma com o teu dedo.

Auxílio à leitura

Experimente usar o visor de televisão como um auxiliar de leitura das crianças mais novas, ou para aquelas que sentem falta de confiança. Deixe um grupo de quatro ou cinco crianças observarem o professor enquanto imprime uma palavra ou duas usando algumas palavras dos livros mais simples usados na escola e incluindo o nome de cada criança se possível. Deixe as crianças ajudarem-se umas às outras, e quando algumas delas afirmam conseguir ler as palavras que se encontram no visor deixe-as carregar em COPY e ENTER a fim de disporem de cópias para guardarem e levarem para casa.

À medida que a criança se desenvolve, poderão ser usadas "páginas" maiores, com palavras mais difíceis. Dentro em pouco as crianças começarão a perguntar se podem escrever elas próprias...

```
O David tem uma bola nova.
Gostaria de ir jogar com ela
para a relva.
        A Ana recebeu uma boneca
        nova no dia de anos.
        Parece quase um bebe a
        serio.
O Joao gosta de andar na sua
bicicleta.
        O David, a Ana e o Joao
        gostam muito de brincar.
```


15

MANTER O INTERESSE

Não é difícil manter as crianças interessadas no uso do computador. No entanto, convém utilizar tantas potencialidades do Spectrum quanto possível, a fim de assegurar o interesse dos programas executados pelas crianças.

O som

Pode-se construir uma música divertida usando algumas notas aleatórias, repetidas. As crianças afirmaram num caso que parecia o ruído de alguém a correr, pelo que demos cores à margem e ao fundo, deixámos uma imagem de "pequenos pés" no visor, e escrevemos o seguinte no visor:

```
 Aqui estao os homens de um
dos planetas. Correm muito
depressa mas tem pes bastante
pequenos. Tem bracos finos
e pernas compridas.

 Acho que estao a fugir de um
horrivel monstro do espaco.

 Pensas que consegues fazer
um desenho com muitos homens
pequeninos vindos do espaco a
fugirem de um monstro grande
e mau?
```

Este caso concreto constituiu um bom exercício de leitura, e as imagens eram boas. Gravámos o programa a fim de o podermos usar de novo:

```
  10 REM  LEITURA 1
  20 PRINT "
                   Aqui estao os ho
mens de um         dos planetas. Cor
rem muito          depressa mas tem
pes bastante       pequenos. Tem bra
cos finos          e pernas comprida
s.

                   Acho que estao a
 fugir de um       horrivel monstro
do espaco.

                   Pensas que conse
gues fazer         um desenho com mu
itos homens        pequeninos vindos
 do espaco a       fugirem de um mon
stro grande        e mau?"

  30 BORDER 5
  40 PAPER 6
  50 LET a=0
  60 PAUSE 10
  70 BEEP .01,a
  80 LET a=a+8
  90 IF a>50 THEN GO TO 50
 100 GO TO 60
```

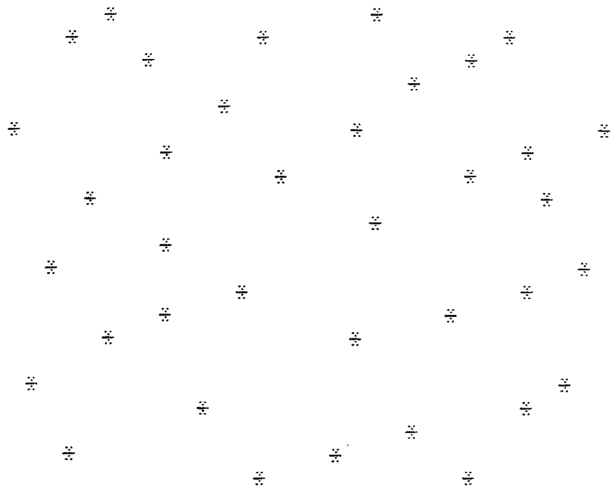
É possível gravar programas contendo melodias simples. Podem-se usar músicas familiares às crianças, ou pode-se deixar estas experimentarem músicas próprias. O grupo deve trabalhar num canto afastado, a fim de não incomodarem as outras crianças.

O programa que se segue contém uma música conhecida. O padrão de estrelas em fundo altera-se no final de cada linha. As cores e a distribuição das estrelas podem ser alteradas com bastante facilidade. deixámos o programa a correr enquanto as crianças faziam outras coisas. O som pode ser amplificado usando um gravador. As crianças gostam de acompanhar a música cantando elas próprias.

```
  10 REM "Estrela"
  20 BORDER 3
  30 PAPER 1
  40 INK 6
  50 GO SUB 1000
  60 CLS
  70 GO SUB 2000
  80 BEEP 1,5: BEEP 1,5: BEEP 1,
12: BEEP 1,12: BEEP 1,14: BEEP 1
,14: BEEP 2,12
  90 CLS
 100 GO SUB 1000
 110 BEEP 1,10: BEEP 1,10: BEEP
```



```
1,9: BEEP 1,9: BEEP 1,7: BEEP 1,
7: BEEP 2,5
 120 CLS
 130 GO SUB 2000
 140 BEEP 1,12: BEEP 1,12: BEEP
1,10: BEEP 1,10: BEEP 1,9: BEEP
1,9: BEEP 2,7
 150 CLS
 160 GO SUB 1000
 170 BEEP 1,12: BEEP 1,12: BEEP
1,10: BEEP 1,10: BEEP 1,9: BEEP
1,9: BEEP 2,7
 180 CLS
 190 GO SUB 2000
 200 BEEP 1,5: BEEP 1,5: BEEP 1,
12: BEEP 1,12: BEEP 1,14: BEEP 1
,14: BEEP 2,12
 210 CLS
 220 GO SUB 1000
 230 BEEP 1,10: BEEP 1,10: BEEP
1,9: BEEP 1,9: BEEP 1,7: BEEP 1,
7: BEEP 2,5
 240 CLS
 250 GO SUB 2000
 260 CLS
 270 GO SUB 1000
 280 CLS
 290 GO TO 50
1000 PRINT "             *
  *                    *
            *                *
                                  *
          *         *

       *                   *      *
                                 *

            *              *
   *                              *
         *
*                       *
           *
  *                            *
       *
                   *
               *
                               *
  *"
1001 RETURN
2000 PRINT "                *        *
```

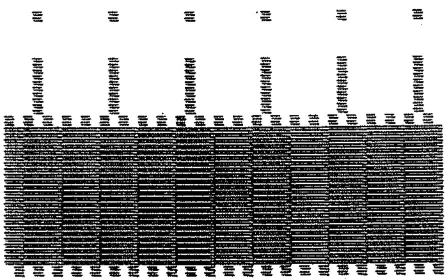


```
2001 RETURN
```

Feliz aniversário

Este programa pode ser usado para alegrar uma criança num dia especial. No visor aparece um bolo de aniversário, como se pode ver:

```
Sara faz hoje seis anos.
     Feliz aniversario!
```



A canção "Feliz Aniversário" é repetida enquanto a imagem é projectada. Neste exemplo, o bolo da criança tem seis velas, mas tanto este número como o nome das crianças podem ser alterados muito facilmente.


```
10 BORDER 4
20 PAPER 7
30 PRINT "          Sara faz hoje sei
s anos."
40 PRINT "          Feliz aniver
sario!
"
50 INK 6
60 PRINT "
"
70 PRINT "
"
80 INK 3
90 PRINT "
"
100 INK 5
110 PRINT "
"
120 INK 3
130 PRINT "
"
```

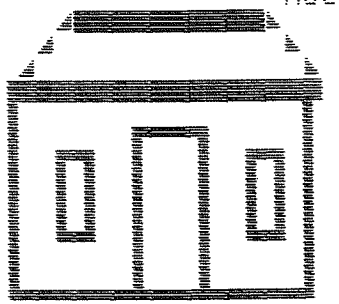
Se o programa tiver sido gravado em cassete previamente, usando várias idades diferentes e o número de velas correspondentes, será muito fácil carregar no computador a versão apropriada. É depois possível alterar rapidamente a linha 30 de modo a incluir o nome correcto.

Uso das cores do Spectrum

As crianças podem "construir" uma casa seguindo instruções simples e utilizando as teclas das cores, respondendo a perguntas como "De que cor queres o telhado?"

Depois de terem indicado a cor escolhida, surge a pergunta seguinte.

```
Obrigado por teres ajudado a co-
lorir a casa.
                        Adeus!
```



Quando a casa está terminada, pede-se às crianças que respondam "s" ou "n" (de "sim" ou "não") conforme querem ou não construir outra casa.

```
  10 REM "Casa"
  20 FOR n=0 TO 7
  30 READ linha: POKE USR "a"+n,
linha
  40 NEXT n
  50 FOR p=0 TO 7
  60 READ linha: POKE USR "b"+P,
linha
  70 NEXT p
  80 CLS : BORDER 7
  90 PRINT AT 0,0;"De que cor qu
eres o telhado?"
 100 INPUT a: INK a
 110 LET z=5: LET y=7
 120 PRINT AT z,y;"▗"
 130 IF z=3 THEN GO TO 160
 140 LET z=z-1: LET y=y+1
 150 GO TO 120
 160 LET y=y+1
 170 PRINT AT z,y;"▀"
 180 IF y=19 THEN GO TO 210
 190 LET y=y+1
 200 GO TO 170
 210 LET y=y+1
 220 PRINT AT z,y;"▙"
 230 IF z=5 THEN GO TO 260
 240 LET z=z+1: LET y=y+1
 250 GO TO 220
 260 INK 0
 270 PRINT AT 0,0;"De que cor qu
eres as paredes?"
```



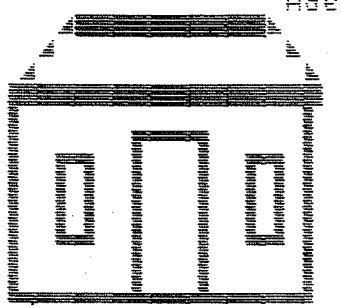
```
 280 INPUT b: INK b
 290 LET z=z+1
 300 PRINT AT z,y;"▐ "
 310 IF z=14 THEN GO TO 340
 320 LET z=z+1
 330 GO TO 300
 340 LET z=z+1
 350 PRINT AT z,y;"▀ "
 360 LET y=y-1
 370 PRINT AT z,y;"▀▀"
 380 IF y=7 THEN GO TO 410
 390 LET y=y-1
 400 GO TO 370
 410 LET y=y-1
 420 PRINT AT z,y;" ▀▀"
 430 LET z=z-1
 440 PRINT AT z,y;" ▐"
 450 IF z=6 THEN GO TO 480
 460 LET z=z-1
 470 GO TO 440
 480 LET y=y+1
 490 PRINT AT z,y;"▀▀"
 500 IF y=22 THEN GO TO 530
 510 LET y=y+1
 520 GO TO 490
 530 INK 0
 540 PRINT AT 0,0;"De que cor qu
eres as janelas"
 550 INPUT c: INK c
 560 LET z=9: LET y=9
 570 PRINT AT z,y;"▀▀"
 580 LET y=y+1
 590 PRINT AT z,y;"▀▀"
 600 LET z=z+1
 610 PRINT AT z,y;" ▐"
 620 IF z=11 THEN GO TO 650
 630 LET z=z+1
 640 GO TO 610
 650 LET z=z+1
 660 PRINT AT z,y;"▟"
 670 LET y=y-1
 680 PRINT AT z,y;"▙▙"
 690 LET z=z-1
 700 PRINT AT z,y;"▐ "
 710 IF z=10 THEN GO TO 740
 720 LET z=z-1
 730 GO TO 700
 740 LET z=9: LET y=19
 750 PRINT AT z,y;"▀▀"
 760 LET y=y+1
 770 PRINT AT z,y;"▀▀"
 780 LET z=z+1
```

```
 790 PRINT AT z,y;" "
 800 IF z=11 THEN GO TO 830
 810 LET z=z+1
 820 GO TO 790
 830 LET z=z+1
 840 PRINT AT z,y;""
 850 LET y=y-1
 860 PRINT AT z,y;""
 870 LET z=z-1
 880 PRINT AT z,y;""
 890 IF z=10 THEN GO TO 920
 900 LET z=z-1
 910 GO TO 880
 920 INK 0
 930 PRINT AT 0,0;"De que cor qu
eres a porta?
                "
 940 INPUT d: INK d
 950 LET z=14: LET y=13
 960 PRINT AT z,y;""
 970 IF z=9 THEN GO TO 1000
 980 LET z=z-1
 990 GO TO 960
1000 LET z=z-1
1010 PRINT AT z,y;""
1020 LET y=y+1
1030 PRINT AT z,y;""
1040 IF y=15 THEN GO TO 1070
1050 LET y=y+1
1060 GO TO 1030
1070 LET y=y+1
1080 PRINT AT z,y;""
1090 LET z=z+1
1100 PRINT AT z,y;" "
1110 IF z=14 THEN GO TO 1140
1120 LET z=z+1
1130 GO TO 1100
1140 INK 0
1150 PRINT AT 0,0;"De que cor qu
eres a margem?"
1160 INPUT e: BORDER e
1170 PRINT AT 0,0;"    A casa est
a' acabada. Queres fazer outra?
Responde s ou n."
1180 INPUT f$
1190 IF f$="s" THEN GO TO 80
1200 IF f$="n" THEN PRINT AT 0,0
;"Obrigado por teres ajudado a c
o-lorir a casa.
                              Adeus!":
STOP
1220 PRINT TAB 2;"Carrega outra
vez em s ou n."
```



```
1230 GO TO 1180
1240 STOP
1250 DATA BIN 00000000,BIN 00000
010,BIN 00000110,BIN 00001110,BI
N 00011110,BIN 00111110,BIN 0111
1110,BIN 00000000
1260 DATA BIN 00000000,BIN 01000
000,BIN 01100000,BIN 01110000,BI
N 01111000,BIN 01111100,BIN 0111
1110,BIN 00000000
```

```
Obrigado por teres ajudado a co-
lorir a casa.
                              Adeus!
```



CASA

LÊ A PERGUNTA

Escolhe a cor e carrega: —

1 para AZUL
2 para VERMELHO
3 para MAGENTA
4 para VERDE
5 para CIÃO
6 para AMARELO
Ø para PRETO

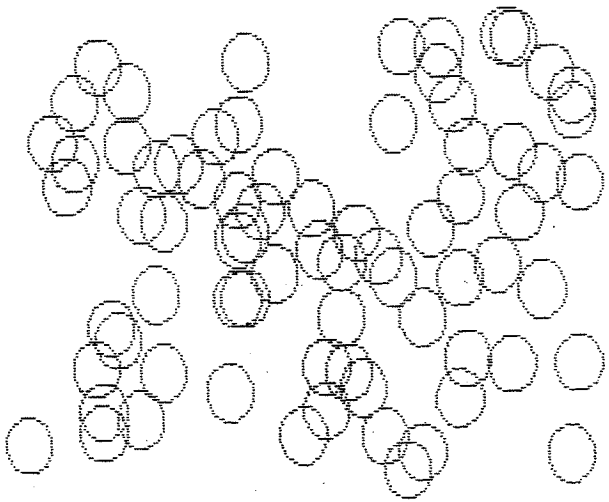
PARA FAZER OUTRA CASA CARREGA EM S PARA "SIM"

PARA PARAR O PROGRAMA CARREGA EM N PARA "NÃO"

Bolhas

O programa que se segue (adaptado de um outro publicado em *The ZX Spectrum Explored,* Sinclair Browne, 1982) é um dos favoritos entre as crianças. Chamamos-lhe "Bolhas", e tornámo-lo um pouco mais lento introduzindo uma linha de pausa:

```
10 REM  bolhas
20 PAPER 0
30 LET x=(234*RND)+10
40 LET y=(154*RND)+10
50 LET z=(5*RND)+1
60 CIRCLE INK z;x,y,10
70 PAUSE 30
80 GO TO 30
```



Os pais

Os pais são bem vindos às escolas, dado que podem ser úteis de muitas maneiras. São igualmente curiosos em relação aos computadores. Não tiveram oportunidade de trabalhar com eles quando andavam na escola, se bem que muitos os usem agora. Muitas vezes os pais e as crianças aprendem em conjunto.

Depois de as crianças terem mostrado que sabem escrever os seus nomes no computador, e que são capazes de seguir as instruções inscritas numa ficha, os pais gostam normalmente de mostrar as suas habilidades no teclado. É possível carregar o programa de treino do teclado contido na cassete de demonstração do Spectrum que acompanha este computador ao ser adquirido, deixando depois pais e filhos ajudando-se entre si. Depois de usar as fichas simples de instruções, os pais tornam-se rapidamente capazes de introduzir no computador programas encontrados em revistas e livros. Em dada ocasião um pai passou a tarde inteira na escola usando o Spectrum pela primeira vez, e escreveu o seguinte programa:

```
  10 REM "Somar"
  20 LET f=INT (RND*10)+1
  30 LET s=INT (RND*10)+1
  40 LET c=f+s
  50 LET t=0
  60 FOR t=1 TO 3
  70 PRINT f;s
  80 PRINT "Quanto e' a soma do
primeiro nu-mero com o segundo?"
  90 INPUT x
 100 IF x<>c THEN PRINT "E' pena
, esta' errado."
 110 PAUSE 100
 120 IF x=c THEN GO TO 170
 130 CLS
 140 IF t<>3 THEN PRINT "A respo
sta correcta e'...";c
 150 NEXT t
 160 PAUSE 100
 170 PRINT "Optimo! Esta' certo!
"
 180 PAUSE 100
```

Os pais e as crianças gostam de escrever mensagens uns aos outros, deixando-as no visor ou usando a impressora.

Recentemente observei um homem comprando um Spectrum para os seus três filhos. A loja estava cheia e o empregado olhava an-

siosamente para a bicha de clientes. "Quando eu o ligar", perguntava o pobre pai, "começa logo a jogar? Como é que ele sabe o que deve fazer? Não sei como lhes explicar o que devem fazer quando lhes der o computador..." Disse-lhe que senti exactamente o mesmo há dezoito meses atrás. Depois perguntei-lhe a idade das crianças, e ele disse-me que o mais velho tinha 14 anos e tinha usado um Spectrum na escola. Toda a gente na bicha mostrou sinais de alívio, afirmando que não haveria qualquer problema, e que ele teria sorte se conseguisse chegar perto do computador nos próximos dias. Alguém lhe perguntou onde vivia, e ofereceu-se para ir a sua casa dar-lhe uma ajuda. O homem fez perguntas sobre gravadores, e foi-lhe dito que poupavam muito tempo. Foram indicados títulos de livros e dados outros conselhos. De facto, quase se formou um clube de computadores imediatamente...

Verifiquei que a maior parte dos adultos pertence a uma de duas categorias: os peritos de programação, e os que afirmam fugir quando ouvem falar em computadores. As crianças pertencem todas à primeira categoria. São confiantes, e estão ansiosas por pôr as mãos num computador. Os pais "envergonhados" são dentro em pouco convertidos depois de verem as crianças usar o computador sem dificuldade.

História

As crianças gostam de ouvir contar como era o primeiro computador. Dou-lhes várias lições baseadas no material que se segue. A profundidade com que o tema é tratado depende da idade das crianças.

A máquina de Babbage (década de 1830)

Charles Babbage queixava-se da falta de rigor de certas tabelas astronómicas calculadas no início do século dezanove. Compreendeu que necessitava de um dispositivo sofisticado para realizar cálculos mais precisos.

Decidiu construir um dispositivo computador mecânico, capaz de tratar números com 20 algarismos. Um dos seus aparelhos, o seu "Motor da Diferença", foi parcialmente construído, e fez planos para outro, o "Motor Analítico".

As técnicas do seu tempo não permitiam pôr em execução ideias tão avançadas, e devido a isso e à quantidade de dinheiro necessário os seus projectos fracassaram.

Os cartões perfurados de Hollerith (1890)

Herman Hollerith trabalhou no American Census Bureau.

O censo de 1880 levou sete anos a completar, e Hollerith compreendeu que o censo de 1890 demoraria muito mais tempo devido ao rápido crescimento da população americana.

A menos que fossem utilizados dispositivos mecânicos, era possível que os resultados do novo censo não estivessem sequer prontos antes de se realizar o censo seguinte, de 1900. Hollerith recorreu a um código de cartões perfurados para armazenar informação, juntamente com dispositivos electromecânicos que ordenavam os cartões e imprimiam os dados e totais. O seu importante contributo permitiu terminar o estudo dos resultados do censo de 1890 em três anos.

O desenvolvimento da máquina de Babbage por Aiken (1940)

O "Automatic Sequence Controlled Calculator" foi desenvolvido por Howard Aiken na Universidade de Harvard e terminado em 1944. Tratava-se de um aparelho electromecânico, consistindo basicamente num grande número de máquinas de calcular controladas por instruções guardadas numa fita de papel. A máquina foi usada pela marinha americana durante quinze anos.

```
      Joao    Tenho sete anos

   Ha' muito tempo havia um homem
   chamado Sr. Babbage. Tentou
   construir uma maquina que foss
e  capaz de fazer depressa contas
   muito bem feitas. Mas a maquin
a  ficou tao grande que nao cabia
   numa sala. O Sr. Babbage ficou
   sem dinheiro e nao conseguiu
   acabar a maquina.
```

Os computadores, entre 1946 e 1959

Os primeiros computadores utilizavam válvulas, e eram relativamente lentos. Consistiam numa unidade processadora central e em dispositivos de entrada e saída. Utilizavam métodos de armazenamento de dados engenhosos mas primitivos, mais tarde substituídos por núcleos de ferrite.

Desenvolvimento entre 1960 e 1964

A invenção do transistor permitiu a construção de computadores muito mais pequenos. Estes utilizavam circuitos impressos com transistores, diodos, condensadores e resistências.

Os transistores, tal como as válvulas, são interruptores electrónicos através dos quais passa corrente em determinadas condições. Devido à sua "velocidade de comutação", os transistores permitiram ao computador funcionar a uma velocidade muito maior do que seria possível usando válvulas.

Nesta época começaram a ser usadas fitas e discos magnéticos juntamente com os computadores.

Os computadores depois de 1964

Os fabricantes de computadores começaram a investir no desenvolvimento de *software* tanto dinheiro como o que até então investiam na produção do *hardware*.

As dimensões do computador tornaram-se cada vez mais reduzidas devido à introdução no fabrico de circuitos integrados, dispositivos que contêm transistores, diodos, condensadores e resistências, sendo todos estes componentes fundidos simultaneamente num circuito único de pequenas dimensões.

16

JOGOS PARA O CÉREBRO

Como o leitor poderá ler no último capítulo deste livro, algumas escolas verificaram que o uso dado ao computador para além dos cursos depende até certo ponto da disponibilidade de *software* não directamente relacionado com cursos específicos.

Não seria apropriado fornecer aqui listas de programas do tipo "Invasores galácticos". Em vez disso, cito aqui alguns jogos que parecem ter algumas qualidades do ponto de vista educacional — e que principalmente são divertidos.

Verificará que estes programas são também úteis para "quebrar o gelo", ajudando os estudantes no seu primeiro contacto com o computador.

Leitor da mente

Este jogo pede ao estudante que indique o seu nome, e depois respeite um certo número de instruções que exigem alguma facilidade na realização de operações matemáticas. Tendo seguido as instruções, e carregado em ENTER depois de executar cada passo, o computador diz ao estudante a sua idade, e a quantidade de dinheiro trocado que tem no bolso... O programa não demora muito tempo a executar (apesar de, evidentemente, a execução ser mais demorada se o aluno é pouco competente na realização de cálculos e demora muito tempo a responder). É portanto possível que o programa seja usado por vários alunos em tempo aceitável.

Vejamos a listagem do programa:

```
 10 REM Leitor da mente
 20 INPUT "Como te chamas? ";a$
 30 PRINT ""Ola", ";a$
 40 PRINT ""E' o momento de te
star a tua     habilidade com os
```

```
numeros, e a  minha a ler a ment
e!"
  50 GO SUB 500
  60 PRINT "OK, ";a$;" comecemos
 por","multiplicar a tua idade (
em      anos) por dois."
  70 GO SUB 500
  80 PRINT "Agora soma cinco ao
total."
  90 GO SUB 500
 100 PRINT "E agora multiplica p
or 50."
 110 GO SUB 500
 120 PRINT "Subtrai agora 365 ao
 total."
 130 GO SUB 500
 140 PRINT "Soma agora o dinheir
o trocado    que tens no bolso."
 150 GO SUB 500
 160 PRINT "Diz-me agora qual fo
i o total."
 170 INPUT a
 180 GO SUB 500
 190 LET a=a+115
 200 LET b=INT (a/100)
 220 LET a=a-100*b
 230 PRINT "Tens ";a;" em dinhei
ro trocado"
 240 GO SUB 500
 250 PRINT '"E tens ";b;" anos
de idade"
 260 STOP
 500 FOR g=1 TO 300: NEXT g
 510 PRINT '"Carrega em [ ] q
uando quiserescontinuar."
 520 INPUT z$
 530 CLS
 540 RETURN
```

E agora um exemplo de execução:

```
Ola', Tim


E' o momento de testar a tua
habilidade com os numeros, e a
minha a ler a mente!


OK, Tim comecemos por
multiplicar a tua idade (em
anos) por dois.

Agora soma cinco ao total.

E agora multiplica por 50.

Subtrai agora 365 ao total.


Soma agora o dinheiro trocado
que tens no bolso.

Diz-me agora qual foi o total.

Carrega em [ ] quando quiseres
continuar.

Tens 27.5 em dinheiro trocado

E tens 38 anos de idade
```

Abrir códigos

O programa que se segue é uma forma simplificada do jogo Mastermind (copyright da Invicta Plastics). A Invicta comprou os direitos do jogo a um matemático amador — Mordechai Meirovitch — em 1971. O jogo já é no entanto popular há séculos em vários países, sob outros nomes.

Enquanto que o jogo comercial utiliza normalmente peças coloridas em plástico, e pede a um jogador que escolha um código de quatro cores que o outro jogador deve em seguida descobrir, esta versão utiliza algarismos em vez de cores. O código tem um comprimento de três algarismos e — para permitir uma solução mais fácil — não aceita a repetição de qualquer algarismo. Isto significa portanto que 123 e 973 serão códigos válidos, mas que 993 ou 272 já não o são. O zero não é usado.

Vejamos um exemplo de execução do programa:

```
Adivinha no. [ ]

[ ]  0 pretos 1 branco
     1 preto 0 brancos
     0 pretos 1 branco
     0 pretos 0 brancos
```

```
Conseguiste em 5 tentativas

O codigo era 375
```

Como o leitor poderá verificar neste exemplo, o computador atribui um "negro" a qualquer algarismo que se encontre na posição correcta, e um "branco" a qualquer algarismo que de facto pertença ao código mas não esteja na posição certa. O jogador tenta descobrir o código em menos de 10 tentativas, usando as respostas que o computador fornece após cada uma delas.

Vejamos agora a listagem do programa:

```
  10 REM Quebrar codigos
  20 BORDER 2: PAPER 2: INK 6: C
LS
  30 DIM a(3): DIM b(3)
  90 FOR z=1 TO 3
 100 LET a(z)=INT (RND*9)+1
 110 NEXT z
 120 IF a(1)=a(2) OR a(1)=a(3) O
R a(2)=a(3) THEN GO TO 90
 130 LET a=100*a(1)+10*a(2)+a(3)
 140 FOR c=1 TO 10
 150 PRINT AT 0,0;"Adivinha no.
"; INVERSE 1;c
 152 INPUT INVERSE 1;"Resposta?
";b
 155 IF b<100 OR b>999 THEN GO T
O 150
 157 PRINT AT c+1,1; INVERSE 1;b
; INVERSE 0;" ";
 160 LET b(1)=INT (b/100)
 170 LET b(2)=INT ((b-100*b(1))/
10)
 180 LET b(3)=INT (b-100*b(1)-10
*b(2))
 190 IF a=b THEN GO TO 430
 200 LET d=a
 210 LET n=0
 220 LET w=0
 230 FOR e=1 TO 3
 240 IF a(e)<>b(e) THEN GO TO 27
0
 250 LET n=n+1
 260 LET a(e)=0
 270 NEXT e
 280 FOR f=1 TO 3
 290 IF a(f)=0 THEN GO TO 340
 300 FOR e=1 TO 3
 310 IF b(f)<>a(e) THEN GO TO 33
0
 320 LET w=w+1
 330 NEXT e
 340 NEXT f
 350 LET a(1)=INT (d/100)
 360 LET a(2)=INT ((d-100*a(1))/
10)
 370 LET a(3)=INT (d-100*a(1)-10
*a(2))
 380 PRINT n;" preto";
 385 IF n<>1 THEN PRINT "s ";
 386 IF n=1 THEN PRINT " ";
 390 PRINT w;" branco";
 395 IF w<>1 THEN PRINT "s"
 400 NEXT c
 410 PRINT AT 12,0;'"Nao conseg
uiste"
 420 GO TO 440
 430 PRINT AT 12,0;"Conseguiste
em ";c;" tentativas"
 440 PRINT '"O codigo era ";a
```

A estrela mágica

Este programa produz uma estrela de cinco pontas, com um número nas intersecções das linhas imaginárias que a formam. Os números de cada linha dão sempre o mesmo resultado quando somados.

Quando se executa o programa, verifica-se que parte dos números se encontram substituídos por zeros. Cabe ao utilizador substituir novamente esses zeros pelos números correctos, em tão poucas tentativas quanto possível.

Vejamos alguns momentos de uma execução do programa em causa:

```
        1

    8   0   0   -6

      1       10

        6

      4     0
```

```
Esta e' a adivinha 1
Escreve um numero que penses fa-
zer parte da estrela
```

Adivinha no. 2

```
        1

  8   0   0   -6

    1     10

        6

    4     -1
```

```
Esta e' a adivinha 2
Escreve um numero que penses fa-
zer parte da estrela
```

Adivinha no. 7

```
        1

  8   0   4   -6

    1     10

        6

    4     -1
```

```
Esta e' a adivinha 7
Escreve um numero que penses fa-
zer parte da estrela


Resolveste em 7 vezes
```



```
        1

  8   8   4   -6

    1     10

        6

    4     -1


OK, obrigado por jogar
```

E finalmente a listagem:

```
  10 REM estrelas magicas
  20 GO SUB 480: REM definir est
rela
  30 GO SUB 340: REM imprimir es
trela
  40 GO SUB 70: REM pedir adivin
ha
  50 GO TO 30
  60 REM ********************
  70 REM pedir adivinha
  80 IF pontos>0 AND pontos<10 T
HEN PRINT AT 18,0;"Tens ";pontos
;" ate' agora "
  90 LET adivinha=adivinha+1
 100 PRINT AT 0,0; INVERSE 1;"Ad
ivinha no. "; FLASH 1; INK 2; PA
PER 6;adivinha
 110 INPUT FLASH 1; INK 2; PAPER
 6;"Esta e' a adivinha ";(adivin
ha),"Escreve um numero que pense
s fa-zer parte da estrela ";g
 130 LET pontos=0
 140 FOR j=1 TO 10
 150 IF g=a(j) THEN LET b(j)=a(j
)
 160 IF b(j)<>0 THEN LET pontos=
pontos+1
 170 NEXT j
 180 FOR p=1 TO 100: NEXT p
 190 IF pontos<10 THEN RETURN
 200 REM ******************
 210 GO SUB 340
 220 PRINT AT 16,0;"
```

```
                ";AT 0,0;"Reso
lveste em ";adivinha;" vezes"
 240 FOR p=1 TO 100: NEXT p
 250 PRINT : PRINT FLASH 1;"Carr
ega 's' para novo jogo ou    'n'
para parar"
 260 LET a$=INKEY$
 270 IF a$<>"s" AND a$<>"n" THEN
 GO TO 260
 300 IF a$="s" THEN RUN
 310 PRINT AT 18,0;"OK, obrigado
 por jogar": STOP
 330 REM **********************
 340 REM imprimir estrela
 350 PRINT AT 4,7;b(1)
 380 PRINT AT 6,2;b(2);"  ";b(3)
;"  ";b(4);"  ";b(5)
 400 PRINT AT 8,4;b(6);"    ";b(
7)
 420 PRINT AT 10,7;b(8)
 440 PRINT AT 12,4;b(9);"    ";b
(10)
 460 RETURN
 470 REM ********************
 480 REM definir estrela
 490 CLS
 500 RANDOMIZE
 510 DIM a(10): DIM b(10)
 520 LET adivinha=0
 530 LET pontos=0
 540 LET a=INT (RND*9)+1
 550 LET b=INT (RND*9)+1
 560 LET c=INT (RND*9)+1
 570 LET d=INT (RND*9)+1
 580 LET e=INT (RND*9)+1
 590 IF a=b OR a=c OR a=d OR a=e
 THEN GO TO 550
 600 IF b=c OR b=d OR b=e THEN G
O TO 550
 610 IF c=d OR c=e THEN GO TO 55
0
 620 IF d=e THEN GO TO 550
 630 LET x=INT (RND*3)+1
 640 LET a(1)=x
 650 LET a(2)=x-b+c+d
 660 LET a(3)=a+e
 670 LET a(4)=a+d
 680 LET a(5)=x-b-c+e
 690 LET a(6)=a
 700 LET a(7)=a+c
 710 LET a(8)=a+b
 720 LET a(9)=x-2*b+2*d+e
```



```
730 LET a(10)=x-2*b-c+d+2*e
740 FOR j=1 TO 10
750 LET b(j)=a(j)
760 IF a(j)=0 THEN RUN
770 NEXT j
780 FOR j=1 TO 3
790 LET b(INT (RND*10)+1)=0
800 NEXT j
810 RETURN
```

Palavras

Passamos agora a outro favorito, no qual o Spectrum "pensa" numa palavra, e o estudante tem de adivinhá-la. O número de tentativas de que dispõe depende do comprimento da palavra.

Como pode verificar nas linhas 500 a 540, o programa contém já um vocabulário de origem. No entanto, o leitor pode tornar o jogo muito mais útil substituindo estas palavras por outras que se adaptem melhor à situação dos alunos, ou a um dado tema concreto (por exemplo o nome de minerais estudados).

O jogo é fácil de jogar, e o *modus operandi* tornar-se-á claro se estudar o seguinte exemplo de execução:

```
Tens de adivinhar a minha pala-
vra em apenas 9 tentativas


------




Tens 0 letras correctas



Tens ainda 9 tentativas


Tens de adivinhar a minha pala-
vra em apenas 9 tentativas
 A E I O

-E-I-O
```

```
Tens 3 letras correctas



Tens ainda 8 tentativas


Tens de adivinhar a minha pala-
vra em apenas 9 tentativas
 A E I O B D C M T

ME-ITO




Tens 5 letras correctas



Tens ainda 5 tentativas


Tens de adivinhar a minha pala-
vra em apenas 9 tentativas
 A E I O B D C M T R

MERITO




Conseguiste em 10 tentativas

A palavra era MERITO



Ate' que enfim!
```


Finalmente, a listagem do programa:

```
  10 REM palavras
  20 PAPER 1: BORDER 1: INK 7: C
LS
  40 LET y=0
  50 RANDOMIZE
  80 FOR g=1 TO RND*22+1
  90 READ a$
 100 NEXT g
 110 GO SUB 480
 120 LET n=LEN a$
 125 DIM b(n): DIM d(n)
 130 FOR g=1 TO n
 140 LET b(g)=CODE a$(g)
 150 LET d(g)=b(g)
 160 NEXT g
 170 LET q=INT (n+n/2+.5)
 180 PRINT AT 0,0;"Tens de adivi
nhar a minha pala- vra em apenas
 ";q;" tentativas"
 200 GO SUB 480
 210 FOR j=1 TO q: LET y=y+1
 220 GO SUB 400
 230 IF h=n THEN GO TO 340
 240 PRINT AT 18,0;"Tens ainda "
;q+1-j;" tentativas
 250 INPUT "Indica a tua adivinh
a ";c$
 255 PRINT AT 2,2*y-1;c$
 260 LET f=CODE c$
 270 FOR g=1 TO n
 280 IF d(g)=f THEN LET d(g)=0
LET j=j-1
 290 NEXT g: NEXT j
 300 GO SUB 480
 310 GO SUB 400
 320 PRINT AT 12,0;"
                    ";AT 18,0;"
                                ";AT
 18,0;"E' pena mas nao conseguis
te"
 330 GO TO 370
 340 PRINT AT 12,0;"
                    ";AT 18,0;"
                                ";AT 1
8,0;"Ate' que enfim!"
 360 PRINT AT 10,0;"Conseguiste
em ";y-1;" tentativas"
 370 PRINT AT 12,0;"A palavra er
a ";a$
```

```
 390 STOP
 400 LET h=0: PRINT AT 4,0;
 410 FOR e=1 TO n
 420 IF b(e)=d(e) THEN PRINT "-"
;
 430 IF b(e)<>d(e) THEN PRINT FL
ASH 1; INK 2;CHR$ b(e); FLASH 0;
 INK 7;: LET h=h+1
 440 NEXT e
 450 PRINT AT 12,0;"Tens ";h;" l
etras correctas"
 470 RETURN
 480 FOR p=1 TO 30: BEEP .008,AN
D*40: NEXT p
 490 RETURN
 500 DATA "MERIDIANO","MERITO","
DIVERSAO","TELESCOPIO"
 510 DATA "OXIDANTE","OXIGENIO",
"PALPAVEL","ORTODOXO","INTEGRAL"
 520 DATA "PANDEMONIO","PANEGIRI
CO","PARADOXAL","BANDOLEIRO"
 530 DATA "SACRAMENTO","ESQUEMAT
ICO","SABRE","CALCIO","ANTONIMO"
 540 DATA "SEDIMENTO","SEXAGENAR
IO","TEMPERADO","LUNETA"
```

Formas

No final deste capítulo apresentamos um último jogo, escrito por Derek Cook, que afirma ter-se divertido bastante com ele juntamente com os irmãos (idades compreendidas entre 11 e 13 anos).

O programa é uma combinação de jogos clássicos. Pode auxiliar as crianças a aprenderem os rudimentos da geometria, além de reconhecerem formas geométricas. Como poderá verificar ao executá-lo, utiliza de forma bastante eficaz certas funções do Spectrum como a FLASH, a BEEP e as cores.

Vejamos a listagem:

```
Procuras:
  *
  * *
  **
Fixa bem a forma!
A forma esta' escondida num
quadrado:
```

```
com 100 portas. Para abrir uma
porta, escreve as suas coorde-
nadas (x, y). Se tiveres a cer-
teza de que conheces bem a forma
que procuras, carrega em ENTER
```

```
  10 PRINT "© D.L.Cook,1982"
  20 BORDER 2: INK 1: FLASH 1: P
RINT AT 11,12;"FORMAS"
  30 PAUSE 100: BORDER 1: INK 2:
 PRINT AT 11,0;"Carrega em ▓▓▓▓▓
 para instrucoes"
  40 INPUT q$: FLASH 0
  80 BORDER 1
  90 RANDOMIZE
  95 CLS
 100 DIM d$(3,3)
 110 FOR a=1 TO 5
 120 LET b=INT (RND*3+1)
 130 LET c=INT (RND*3+1)
 135 IF d$(b,c)="*" THEN GO TO 1
20
 140 LET d$(b,c)="*"
 150 NEXT a
 155 PRINT "Procuras: "
 160 FOR e=1 TO 3
 170 FOR f=1 TO 3
 180 IF d$(e,f)="*" THEN INK 2:
IF d$(e,f)="*" THEN FLASH 1: PRI
NT AT e,f;d$(e,f)
 190 NEXT f
 200 NEXT e
 210 FLASH 0
 230 PRINT "Fixa bem a forma!"
 250 PRINT "A forma esta' escond
ida num      quadrado:"
 260 INK 1: FOR a=1 TO 10
 270 PRINT "                ▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓"
 280 NEXT a
 300 PRINT "com 100 portas. Para
 abrir uma  porta, escreve as su
```

```
as coorde-  nadas (x, y). Se tiv
eres a cer- teza de que conheces
 bem a formaque procuras, carreg
a em ENTER"
 310 INPUT c$
 320 CLS : LET v=0
 330 LET p=INT (RND*8+1)
 340 LET q=INT (RND*8+1)
 350 FOR a=0 TO 9
 360 PRINT AT 6+a,10;9-a
 370 FOR b=0 TO 9
 380 PRINT AT 6+b,11+a;"▒"
 390 NEXT b
 400 PRINT AT 16,11+a;a
 410 NEXT a
 415 FOR n=1 TO 100
 420 INPUT "Escreve coordenadas
";x
 430 INPUT y
 432 PRINT AT 20,1;n;" tentativa
s"
 435 IF NOT ((x=p OR x=p+1 OR x=
p-1) AND (y=q OR y=q+1 OR y=q-1)
) THEN GO SUB 800
 440 IF d$(2+q-y,2-p+x)="*" THEN
 GO SUB 600
 450 IF f$(2+q-y,2-p+x)<>"*" THE
N GO SUB 700
 455 IF v=5 THEN GO TO 900
 460 NEXT n
 590 STOP
 600 FLASH 1: INK 2: PRINT AT 15
-y,11+x;"*"
 605 LET v=v+1
 607 BEEP .5,v
 610 RETURN
 690 STOP
 700 FLASH 0: INK 2: PRINT AT 15
-y,11+x;"▒"
 710 RETURN
 800 FLASH 0: INK 2: PRINT AT 15
-y,11+x;"▒"
 810 GO TO 460
 900 FLASH 1: PRINT AT 2,0;"Reso
lveste em ";n;" tentativas"
 910 BEEP .5,20: BEEP .5,25: BEE
P .5,30: BEEP .5,35
 920 FLASH 0: PRINT AT 4,0;"Outr
o jogo (s/n)? "
 930 INPUT m$
 940 IF m$="s" THEN GO TO 90
 950 IF m$="n" THEN STOP
```


17

SUGESTÕES FINAIS

Os computadores são particularmente úteis quando considerados como auxiliares do ensinamento oral dos professores, e não como seus substitutos. Em determinada área, a da realização de testes, o computador pode ser um óptimo auxiliar nas situações em que não é possível ao professor dar uma atenção cuidada a todos os alunos.

Um professor, entusiasta deste tipo de utilização dos computadores, afirma recorrer aos existentes na sua escola para revisões aos seus alunos de Francês. Todos os alunos cujos resultados não lhe parecem satisfatórios devem "treinar" com o computador depois das aulas. O professor afirma que o número de estudantes com maus resultados diminuiu significativamente (*).

Uma outra escola entrou na era dos computadores usando uma máquina emprestada por uma empresa local. Verificaram que o interesse dos alunos pelo uso da máquina aumentava ao mesmo tempo que o número de programas disponíveis. Os primeiros programas não eram particularmente sofisticados, e incluíam actividades como a procura de objectos em redes de 10 por 10 casas, ou definição de bio-ritmos; mas a escola verificou que o interesse dos alunos era já bastante grande antes de o computador ser oficialmente integrado no ensino (ver *The Micro in a Small School,* Interface Age, Outubro de 1979, pág. 64).

Se bem que isto acontecesse antes de o uso dos computadores se encontrar muito difundido, e quando portanto os computadores tinham maior valor como novidade do que hoje, o facto de o interesse ser proporcional à disponibilidade de programas mantém-se provavelmente verdadeira para o Spectrum.

(*) *Computer Assisted Instruction — Worth The Effort?,* F. Keplinger, revista "Compute!", Agosto de 1981, pág. 40.

Quando se escrevem programas para uso na escola, deve-se manter presente o facto de um estudante aprender (ou pelo menos tender mais a fazê-lo) quando é envolvido pessoalmente numa actividade. O professor deve fazer tudo o que lhe for possível para garantir que os programas provocam simultaneamente o interesse e o envolvimento dos alunos.

A necessidade de aprender qualquer coisa (por exemplo, a fim de passar num exame) pode, no entanto, ser tão forte que o estudante aprenderá, pelo menos até certo ponto, coisas que são apresentadas de um modo completamente falho de interesse ou que não os envolve de modo algum.

Uma das grandes vantagens do computador-como-professor relativamente ao professor propriamente dito é a paciência infinita da máquina. O Spectrum pode ser programado para reagir à velocidade de aprendizagem do estudante, assegurando que o aluno mais lento não se sinta tão frustrado que se torne incapaz de manter o interesse pelo material estudado.

Talvez o leitor encontre resistência na sua escola ao uso do Spectrum. Foram muitos os métodos que os professores até hoje advogaram como vantajosos, para mais tarde se verificar constituirem um fracasso. Uma experiência deste tipo, e a suspeita de que os alunos que "brincam" com computadores não estão a fazer aquilo que deviam, podem criar atitudes que impedem o uso generalizado do Spectrum na sua escola.

Os Spectrums usados nas escolas não devem ser tratados como objectos imensamente valiosos, para serem usados apenas em situações especiais. Entre outras coisas, os estudantes aprendem a familiarizarem-se com os computadores. Isto é provavelmente tão importante como o estudo de facto realizado através da máquina. Os computadores ocuparão um lugar muito importante no futuro dos estudantes, e tudo o que aumentar a capacidade para trabalhar com eles parece ter um enorme valor por si só.

Nestas condições, verifique se é sua intenção de impedir alguém de usar o Spectrum "porque está apenas a brincar". Sentir-se bem ao usar a máquina tornará mais fácil ao estudante dedicar-se a outros tipos de utilização, auxiliando portanto em última análise o professor no que toca a um emprego mais directo do computador no trabalho escolar.

Será evidentemente aconselhável um certo grau de vigilância (quanto mais não seja para garantir que os computadores não abandonam as escolas dentro de sacos...). No entanto, se o Spectrum ou os Spectrums existentes puderem estar disponíveis à hora de almoço e no final das aulas, o resultado será positivo. Deixe os seus alunos usarem à vontade as máquinas que foram compradas para eles. Encoraje-os a escreverem os seus próprios programas, e a trazerem programas originais de casa, a fim de os mostrarem aos seus colegas, Tente sempre fazer tudo o que estiver ao seu alcance para criar uma situação em que o acesso aos computadores seja tão fácil quanto possível, e em que simultaneamente as formalidades necessárias para o uso das máquinas sejam mantidas a um nível mínimo.

APÊNDICE A

SUGESTÃO DE UM TESTE SOBRE BASIC SPECTRUM

1. Que observa no canto inferior esquerdo do visor quando toca em alguma tecla depois de ligar o computador?
2. Qual o significado do que vê escrito?
3. Que escreve no computador para:

   a) somar 3 e 4
   b) dividir 8 por 4
   c) multiplicar 7 por 8

4. Que significa o cursor C?
5. Qual é a diferença entre o zero e a letra "o" maiúscula impressos pelo computador?
6. *a*) Para que serve a tecla CAPS SHIFT? Dê um exemplo.
   b) Para que serve a tecla SYMBOL SHIFT? Dê um exemplo.

7. Que observa quando imprime (PRINT) o seguinte (depois de carregar na tecla Enter):

   a) 1234
   b) 1;2;3;4
   c) 1,2,3,4
   d) 1.2
   e) 1,,2

8. Que nome dá a uma fiada de letras ou números encerrados entre aspas?
9. Se quisesse imprimir no visor a palavra SPECTRUM, como a escreveria?
10. Escreva o que se segue correctamente, a fim de ser executado no Spectrum:

    a) PRINT NADAR É DIVERTIDO
    b) PRINT "A JOVEM TEM" 17 ANOS DE IDADE"

11. Nomeie três *funções* disponíveis no teclado do Spectrum.
12. *a*) Como obtém uma função?
    b) Dê um exemplo.
13. Que faz a instrução PRINT AT 11,16?
14. Corrija a linha que se segue (para poder ser executada pelo Spectrum):

    PRINT AT 11,16 "O AUTOCARRO É VERMELHO

15. Imagine que tem o Spectrum ligado à sua frente. Você escreve o seguinte:

    LET A = 1 (Enter)
    LET B = 2 (Enter)
    LET C = 3 (Enter)

    Que verá se depois escrever o que se segue:

    a) PRINT A;B;C
    b) PRINT A + B,C
    c) PRINT A + C—B

16. Como obtém a constante PI no Spectrum?
17. Complete o programa seguinte:

    10 INPUT "raio ";r
    20 PRINT "A área do círculo e ";

18. Depois da linha 10 apresentada abaixo deveria encontrar-se uma linha 20 ordenando ao Spectrum que cumprisse a instrução da linha 10 repetidamente. Qual deverá ser a instrução da linha 20?

    10 PRINT "GATO";

19. Invente um programa simples com quatro linhas, e diga o que acontece quando o executa.
20. Como conta de 1 até 10 usando FOR e NEXT? Apresente um programa simples que o faça.

APÊNDICE B

IDEIAS DE EXERCÍCIOS E PROGRAMAS

Este apêndice contém um certo número de ideias que o leitor pode facilmente converter em material para os seus alunos a fim de os ajudar a desenvolverem a sua capacidade de escrita de programas que resolvam problemas específicos ou realizem um objectivo claro. Talvez outras ideias lhe pareçam mais apropriadas como ponto de partida dos programas que desenvolve para uso pelos seus alunos.

Desenvolva um programa ou uma série de programas que demonstrem a relação entre a velocidade, o tempo e a distância.

Ilustre as propriedades da luz (sombras, reflexões e os princípios da propagação da luz em linha recta).

Demonstre a teoria básica da corrente eléctrica de um modo gráfico.

O princípio de Arquimedes (flutuação, densidade) é um outro tema que se presta a uma apresentação animada.

A construção de frases, a sequência de palavras e outros problemas de linguagem podem ser facilmente testados usando um computador (e programas bastante fáceis de escrever).

Podem-se escrever programas de corridas, nos quais as crianças respondem por exemplo a perguntas de matemáticas avançando uma "casa" sempre que respondem certo. Estes programas obrigam a partilhar o Spectrum pelo menos entre duas crianças simultaneamente.

Os programas baseados em "puzzles" de quinze peças em que estas, contendo letras ou números, devem ser colocadas numa determinada sequência têm-se mostrado bastante populares.

Pode pedir aos alunos que escrevam programas capazes de executar os objectivos de muitos dos programas contidos neste livro, como os mais curtos do capítulo sobre matemáticas. Pode por exemplo pedir que escrevam um programa que, a partir de um número inteiro positivo, calcule o factorial (N!) deste.

Usando a fórmula Juro = Capital*Taxa de juro*tempo/100, peça aos alunos que escrevam um programa capaz de indicar o juro em diferentes períodos de tempo, partindo de diferentes capitais iniciais e de diferentes taxas de juro.

Construir um tabela de raízes quadradas e cúbicas de números numa determinada gama.

APÊNDICE C

CONVERSÃO BINÁRIA

Apresentamos em seguida a listagem completa dos números decimais entre zero e 255, juntamente com os seus equivalentes binários. Esta lista poderá ser-lhe muito útil quando estiver a produzir os seus caracteres gráficos, evitando a necessidade de converter de binário para decimal os valores a incluir em declarações DATA.

No caso de lhe interessar, o programa que faz a conversão é incluído a seguir à tabela.

0	00000000
1	00000001
2	00000010
3	00000011
4	00000100
5	00000101
6	00000110
7	00000111
8	00001000
9	00001001
10	00001010
11	00001011
12	00001100
13	00001101
14	00001110
15	00001111
16	00010000
17	00010001
18	00010010
19	00010011
20	00010100
21	00010101
22	00010110
23	00010111
24	00011000
25	00011001

26	00011010
27	00011011
28	00011100
29	00011101
30	00011110
31	00011111
32	00100000
33	00100001
34	00100010
35	00100011
36	00100100
37	00100101
38	00100110
39	00100111
40	00101000
41	00101001
42	00101010
43	00101011
44	00101100
45	00101101
46	00101110
47	00101111
48	00110000
49	00110001
50	00110010
51	00110011
52	00110100
53	00110101
54	00110110
55	00110111
56	00111000
57	00111001
58	00111010
59	00111011
60	00111100
61	00111101
62	00111110
63	00111111
64	01000000
65	01000001
66	01000010
67	01000011
68	01000100
69	01000101
70	01000110
71	01000111
72	01001000
73	01001001
74	01001010
75	01001011
76	01001100
77	01001101

78	01001110
79	01001111
80	01010000
81	01010001
82	01010010
83	01010011
84	01010100
85	01010101
86	01010110
87	01010111
88	01011000
89	01011001
90	01011010
91	01011011
92	01011100
93	01011101
94	01011110
95	01011111
96	01100000
97	01100001
98	01100010
99	01100011
100	01100100
101	01100101
102	01100110
103	01100111
104	01101000
105	01101001
106	01101010
107	01101011
108	01101100
109	01101101
110	01101110
111	01101111
112	01110000
113	01110001
114	01110010
115	01110011
116	01110100
117	01110101
118	01110110
119	01110111
120	01111000
121	01111001
122	01111010
123	01111011
124	01111100
125	01111101
126	01111110
127	01111111
128	10000000
129	10000001

130	10000010
131	10000011
132	10000100
133	10000101
134	10000110
135	10000111
136	10001000
137	10001001
138	10001010
139	10001011
140	10001100
141	10001101
142	10001110
143	10001111
144	10010000
145	10010001
146	10010010
147	10010011
148	10010100
149	10010101
150	10010110
151	10010111
152	10011000
153	10011001
154	10011010
155	10011011
156	10011100
157	10011101
158	10011110
159	10011111
160	10100000
161	10100001
162	10100010
163	10100011
164	10100100
165	10100101
166	10100110
167	10100111
168	10101000
169	10101001
170	10101010
171	10101011
172	10101100
173	10101101
174	10101110
175	10101111
176	10110000
177	10110001
178	10110010
179	10110011
180	10110100
181	10110101

```
182      10110110
183      10110111
184      10111000
185      10111001
186      10111010
187      10111011
188      10111100
189      10111101
190      10111110
191      10111111
192      11000000
193      11000001
194      11000010
195      11000011
196      11000100
197      11000101
198      11000110
199      11000111
200      11001000
201      11001001
202      11001010
203      11001011
204      11001100
205      11001101
206      11001110
207      11001111
208      11010000
209      11010001
210      11010010
211      11010011
212      11010100
213      11010101
214      11010110
215      11010111
216      11011000
217      11011001
218      11011010
219      11011011
220      11011100
221      11011101
222      11011110
223      11011111
224      11100000
225      11100001
226      11100010
227      11100011
228      11100100
229      11100101
230      11100110
231      11100111
232      11101000
233      11101001
```



```
234      11101010
235      11101011
236      11101100
237      11101101
238      11101110
239      11101111
240      11110000
241      11110001
242      11110010
243      11110011
244      11110100
245      11110101
246      11110110
247      11110111
248      11111000
249      11111001
250      11111010
251      11111011
252      11111100
253      11111101
254      11111110
255      11111111
```

```
  5 REM NUMEROS BINARIOS
  6 REM POR JEREMY RUSTON
  7 REM DE "COMO PROGRAMAR O
        ZX SPECTRUM"
 10 FOR B=0 TO 255
 20 LET J=B
 30 LET A$=""
 40 FOR N=0 TO 7
 50 LET T=J-INT (J/2)*2
 60 IF T=0 THEN LET A$="0"+A$
 70 IF T<>0 THEN LET A$="1"+A$
 80 LET J=INT (J/2)
 90 NEXT N
100 LPRINT B;TAB 7;A$
110 NEXT B
```

APÊNDICE D

A LINGUAGEM LOGO, E UMA INTRODUÇÃO AOS GRÁFICOS "TARTARUGA"

A Logo é bastante diferente da Basic. Foi concebida com o objectivo de constituir uma linguagem capaz de "ensinar a aprender" e, até certo ponto, conseguiu-o.

Apresentada pelo dr. Simon Papert quando era professor do Massachusetts Institute of Technology dos Estados Unidos (vive hoje em França, onde é um dos responsáveis pelo World Computer Centre), a linguagem LOGO foi pensada como a primeira linguagem de programação a ensinar aos interessados. A primeira linguagem que se aprende influencia inevitavelmente o modo como se programa, e o modo como se pensa na programação, até ao fim da vida. Os proponentes da Logo afirmam que a base fornecida pelo estudo inicial desta linguagem é muito mais apropriada a uma futura excelência em programação do que a produzida pela Basic.

Será válida esta afirmação? Papert considera que muitos professores apenas vêem os computadores como dispositivos que podem ampliar os modos tradicionais de fazer as coisas, e não como instrumentos de ensino completamente novos. A Logo, pelo contrário, seria uma linguagem "libertadora", que permitiria aos computadores serem usados para o ensino de novas capacidades, incluindo o "ensino de como aprender".

Seguindo observações de Jean Piaget no sentido de que as crianças são capazes de aprender coisas bastante complexas — como o próprio falar ou o caminhar — sem um treino formal, e o facto de esta informalidade bastante eficaz estar ausente do ensino tradicional, Papert lançou-se na criação de uma linguagem que remediasse esta deficiência. Papert afirma que grande parte da instrução dada em escolas a propósito da programação de computadores coloca a criança quase na situação de ser programada pelo computador. A Logo, pelo contrário, deixa a iniciativa à criança.

Isto é conseguido principalmente deixando ao programador a possibilidade de criar formas e acções novas — como o desenho de

um triângulo — e ordenando depois ao computador que execute essa acção escrevendo muito simplesmente TRIÂNGULO. A Basic não permite com efeito este modo de criação de novas ordens e funções.

Se enviarmos alguém para Inglaterra, e deixarmos a pessoa nesse país durante algum tempo, acabará por falar fluentemente o inglês, mesmo que anteriormente ela própria pensasse "não ser boa em línguas". O mesmo é verdadeiro para as matemáticas, afirma Papert. Parte da Logo é pensada no sentido de permitir ao utilizador "viver na terra das contas", onde não existe ninguém que não seja "bom em matemáticas". Num artigo ("Logo in the Schools, revista BYTE, Agosto de 1982, págs. 116 a 134), Daniel Watt afirma que "os professores têm notado que os alunos que frequentaram aulas de Logo estão mais predispostos a argumentar inteligentemente sobre questões matemáticas e a explicar as suas dificuldades matemáticas claramente".

Quando os computadores começaram a ser desenvolvidos, dispunham de uma quantidade de memória extremamente reduzida. Os programadores deviam adaptar o seu pensamento às exigências da máquina (como seja terem de designar as variáveis inteiras por nomes que começavam por letras bem definidas), independentemente do trabalho suplementar a que isso obrigava. A ideia de que os seres humanos devem estar sujeitos às exigências do computador manteve-se. Apesar de a Basic ser relativamente fácil de interpretar pelo computador e fácil de ensinar, não é uma linguagem muito flexível, obrigando por vezes à construção de programas que se assemelham a verdadeiros labirintos. Papert e a sua equipa decidiram, quando desenvolveram a Logo, não deixar que a tecnologia de computadores limitasse o seu trabalho. Em vez de orientarem o seu pensamento para os computadores baratos (na época) então existentes, quando iniciaram o seu trabalho em finais da década de 60, tentaram trabalhar com os maiores computadores que tiveram ao seu dispor.

O aspecto mais familiar da Logo são os gráficos "Tartaruga" (TURTLE), consistindo numa marca triangular que deixa um rasto no visor. Nestas condições, se a tartaruga se move na imagem um centímetro, roda 90 graus e avança mais um centímetro, roda e anda, roda e anda, acaba por representar um quadrado.

As tartarugas movem-se em "passos de tartaruga" tendo cada imagem aproximadamente 200 passos de altura). As ordens à tartaruga assumem muitas vezes a forma de uma direcção (como FORWARD, em frente) seguida de um número correspondente à quantidade de passos a percorrer; FORWARD 100 levará a tartaruga a percorrer metade do visor ("em frente" refere-se à direcção em que

o triângulo aponta).

No caso da Logo, é possível ensinar ao computador uma sequência de movimentos, como os necessários para descrever um quadrado, podendo essa sequência ser "recordada" pelo computador ao ser invocada por um determinado nome — por exemplo, QUADRADO. Depois, sempre que quisermos desenhar um quadrado, basta-nos escrever esta palavra.

Uma sequência de movimentos como esta é designada por *procedimento*. O processo de desenho de um quadrado pode ser ainda mais simples: Pensemos em FORWARD 100. O computador desenha uma linha no visor. A palavra DIREITA (RIGHT) seguida de um número obriga a tartaruga a virar-se para a direita, o número de graus especificado por esse número, pelo que RIGHT 90 levá-la-á a executar um ângulo recto. FORWARD 100 desenhará uma nova linha perpendicular à primeira. RIGHT 90 produzirá outro ângulo recto. Finalmente obter-se-á um quadrado.

Isto deve dar-nos uma ideia de como é possível criar o procedimento QUADRADO de um modo mais simples, recorrendo à palavra Logo REPEAT (repetir). REPEAT é seguida de um número, e o computador repete a instrução que se segue o número tantas vezes quantas a que ele indica. Assim, para criar um programa que desenhe um quadrado, necessitamos do seguinte:

```
TO QUADRADO
REPEAT 4 [FORWARD 100 RIGHT 90]
END
```

Note que a primeira linha deste programa, TO SQUARE, é a linha de título do procedimento. Depois de executar a rotina, o computador fica a saber o que é um quadrado, e torna-se capaz de produzi-lo sempre que encontrar a ordem SQUARE. Como o leitor certamente compreende, não é tão fácil em Basic criar uma ordem nova à nossa vontade.

A Logo possui outras ordens úteis, como CLEARSCREEN (limpar o visor), PENUP ("levantar a caneta", não desenhar) ou PENDOWN (baixar a caneta, desenhar). Podemos desenhar uma linha, levantar a caneta e deslocá-la para outra região do visor, baixá-la de novo e continuar a desenhar.

Voltemos à nossa definição do procedimento QUADRADO já referido. O leitor compreende que se dispusermos de um modo de indicar a dimensão (100 no nosso exemplo) de cada vez que executarmos o programa, é possível usar QUADRADO para desenhar quadrados de qualquer dimensão. Logo permite isto. Se incluir o

nome da variável na linha de título do procedimento, precedida por dois pontos, o computador esperará que o utilizador indique a dimensão necessária.

```
TO QUADRADO: COMPRIMENTO
REPEAT 4 [FORWARD: COMPRIMENTO RIGHT 90]
END
```

Para executar isto, escreve-se SQUARE 64 (dando este valor a COMPRIMENTO).

É fácil compreender que podemos fazer muito mais coisas além de alterar o comprimento do lado. Podemos definir facilmente um procedimento que permita especificar não só o comprimento do lado, mas também o número de repetições e o ângulo da rotação a executar pela tartaruga. Se tentar aperceber-se mentalmente dos efeitos de cada uma destas alterações, compreende certamente o poderoso instrumento que tem nas mãos.

```
TO FORMA: MUITOS: ÂNGULO: COMPRIMENTO
REPEAT: MUITOS [FORWARD: COMPRIMENTO
RIGHT: ÂNGULO]
END
```

Este simples procedimento permite obter um grande número de efeitos.

Para desenhar um triângulo com 35 passos de comprimento, basta escrever:

```
FORMA 3 120 35
```

Uma estrela com 55 passos em cada linha recta pode ser desenhada com a seguinte instrução:

```
FORMA 5 144 55
```

Terminaremos esta breve introdução aos gráficos Tartaruga sublinhando que depois de se ensinar ao computador uma palavra como QUADRADO o procedimento que a define pode ser usado no interior de outras definições. Isto é, se quiser que o computador imprima uma forma, se desloque um pouco para o lado, e a desenhe de novo, repetindo este processo várias vezes, pode-se definir o procedimento seguinte (considerando que o procedimento QUADRADO já foi definido anteriormente):

```
TO ASSOMBROSAMENTE: MUITOS
REPEAT: MUITOS QUADRADO 50 FORWARD 1
END
```

Existem quatro características importantes na Logo:

PROCEDIMENTOS: A linguagem funciona por definição de passos de programação designados por *procedimentos* que são depois invocados durante a execução. Os procedimentos podem incorporar outros procedimentos. O equivalente Basic mais próximo (e não é normalmente conveniente aprender Logo comparando as suas instruções com as equivalentes Basic) seria uma série de subrotinas chamadas por um nome (como por exemplo GO SUB QUADRADO, onde QUADRADO fosse uma variável à qual tivesse sido previamente atribuído um valor correspondente ao número da linha onde se inicia a subrotina).

INTERACÇÃO: Qualquer ordem, quer faça parte da linguagem original (como FORWARD ou PENUP) ou seja definida pelo utilizador como nome de procedimento, pode ser executada escrevendo-a simplesmente, como acontece com FORMA ou QUADRADO.

LISTAS: a linguagem suporta estruturas compostas designadas por *listas* que são muito mais fáceis de manipular do que as estruturas de dados do tipo dos quadros (arrays). O seu uso é mais flexível. Os procedimentos podem ser tratados como listas.

GEOMETRIA TURTLE: O "animal cibernético", a tartaruga, executará as instruções que lhe forem dadas para desenhar formas no visor. Os gráficos Turtle mostraram já ser um modo ideal de introduzir o conceito e prática da programação de computadores, assim como a base sobre a qual é possível construir um plano de instrução em matemáticas por computador.

APÊNDICE E

PROLOG — PROgramação em LÓGica

Em 1972, os trabalhos realizados por Kowalski, Colmerauer e Robinson culminaram no aparecimento de uma nova linguagem de computadores chamada PROLOG. A lógica existe desde o tempo dos antigos gregos, sendo considerada uma linguagem rigorosa para a enunciação de problemas. A PROLOG permite descrever problemas numa forma lógica susceptível de ser compreendida por um computador.

Desde 1972 foram escritos muitos produtos para adaptação da PROLOG a diferentes computadores. Encontra-se actualmente à venda uma versão dela, a micro-PROLOG (Clark, Ennals & McCabe, 1983) para uso no Spectrum. A Micro-PROLOG é uma versão que contém toda a estrutura da PROLOG, dispondo ainda, na versão para o Spectrum, de formas de utilizar as potencialidades sonoras e gráficas deste computador.

A linguagem pode ser usada de formas muito diferentes e para aplicações igualmente diversas. Uma das suas características mais poderosas é a facilidade com que pode ser adaptada a uma dada aplicação construindo aquilo a que se chama um "front-end". Este é um programa que permite o uso da linguagem de uma forma específica, definindo várias operações. Como exemplo de "front-end" poderíamos citar um que permitisse o uso de processos gráficos do tipo Logo (Steel 1983 *a*).

No Imperial College, em Londres, foi desenvolvido um *front-end* designado por "Simple" (Ennals 1982) para uso nas escolas. O trabalho nesta área não ficou de modo algum terminado, estando actualmente em desenvolvimento outras versões. "Simple" faz parte da Micro-PROLOG fornecida para o Spectrum. Tudo o que diremos neste capítulo utilizará a sintaxe do *front-end* "Simple".

Aprender a descrever os problemas

Para aprender a programar em PROLOG não são necessárias muitas ideias prévias sobre a programação "tradicional". O que é necessário é ser capaz de descrever claramente e rigorosamente as coisas. Como exemplo, examinaremos uma aplicação da PROLOG na exploração ecológica (Steel 1983 *b*). Este tema é suficientemente fácil de entender por pessoas comuns para não obrigar a um estudo prévio. Constituiu até hoje a base de vários cursos de introdução para professores que aprendem a PROLOG.

Um programa inicia-se normalmente por uma colecção de factos simples.

mocho é-um pássaro
pardal é-um pássaro
mosca é-um insecto
palmeira é-uma árvore
serralha é-uma erva
DDT é-um insecticida
toupeira é-um mamífero
toupeira come serralha
pardal como sementes

Estes factos podem ser comunicados ao computador exactamente como estão, usando a ordem Simple *add ( )*.

Por exemplo:

add (andorinha é-um pássaro)

É necessário fazer pequenas alterações nas frases em português que usaríamos normalmente a fim de serem aceites pela PROLOG Simple. Cada frase, neste caso, foi dividida em três partes: os nomes de dois objectos e uma relação entre eles. Os traços são usados do mesmo modo que em português se usaria normalmente um espaço.

Podem-se introduzir igualmente as regras que relacionam estes factos entre si.

x é-uma planta *if* x é-uma erva (ou seja, algo é uma planta *se* for uma erva)
x é-um animal *if* x é-um pássaro
x é-um animal *if* x é-um mamífero
x come y
    if x é-um pássaro
    and y é-um insecto

Nestas regras, *x* e *y* são variáveis. Representam valores desconhecidos. A última regra contém a informação de que todos os pássaros comem insectos. Completaremos o nosso exemplo ecológico com mais algumas regras.

x mau-para y *if* x come y
x mau-para y *if* x é-um insecticida
        and y é-um insecto
x mau-para y *if* y é-uma planta
        and x come sementes

Depois de se ter definido uma colecção de factos e regras, a PROLOG Simple permite fazer uma série de perguntas que utilizarão estas informações. O primeiro tipo de perguntas verifica se determinado facto é conhecido pelo computador. Saberá o computador, por exemplo, se uma toupeira é um animal? Podemos perguntar-lhe...

does (toupeira é-um animal)
YES

Só as informações conhecidas pelo computador podem ser usadas para responder às perguntas feitas. Por vezes isto conduzirá a respostas surpreendentes:

does (gato é-um animal)
NO

O computador responde NÃO porque ninguém lhe "falou" no gato...

Existe um tipo de pergunta muito mais potente para uso quando a resposta YES/NO é insuficiente. Permite-nos determinar o que come um mocho.

Which (x: mocho come x)

ou seja, em português, (indique um x tal que o mocho coma x):

answer is mosca
no more answers

A pergunta which pode dar origem a mais de uma resposta, e a sua forma pode ser mais genérica.

which (x, y: x mau-para y)

ou seja (indique x e y tal que x seja mau para y)

```
answer is (toupeira serralha)
answer is (pardal sementes)
answer is (mocho mosca)
answer is (pardal mosca)
answer is (DDT mosca)
answer is (pardal serralha)
```

É importante compreender que a PROLOG utiliza regras para determinar as respostas às perguntas. Ao contrário do que acontece em ficheiros normais, grande parte da informação pode ser generalizada. A escrita destas regras gerais é uma actividade muito útil para um grupo de alunos que estudem um dado tópico.

Uma terceira pergunta, particularmente útil no ensino, é *why,* (porquê). Esta pergunta permite ao computador explicar as regras que usa para obter a resposta a um *which* ou *does.*

```
why (pardal mau-para serralha)
(pardal mau-para serralha) shown by
   (serralha é-uma planta) and
   (pardal come sementes)
(serralha é-uma planta) shown by
   (serralha é-uma erva)
(serralha é-uma erva) is given
(pardal come sementes) is given
```

Programas específicos

É possível escrever programas semelhantes ao usado neste exemplo de ecologia para numerosas áreas de estudo em escolas. Numa escala mais vasta, estão a ser escritos programas semelhantes para auxílio a médicos, advogados e planeadores. Trata-se de sistemas que contêm conhecimentos especializados. Depois de construídos, estes programas podem ser consultados para obter respostas a perguntas, podem explicar as suas respostas, e utilizando um "front-end" diferente podem fazer eles próprios as perguntas, transformando o computador em professor.

Nos próximos dez anos é possível que vejamos sistemas especialistas como estes substituindo muitos dos programas usados em escolas. Recorrendo à PROLOG é possível explorar já estas ideias, construindo ficheiros ou consultando-os em áreas de história, geografia ou ciências em geral.

APÊNDICE F

SOFTWARE EDUCATIVO COMERCIAL

Apresentamos em seguida uma lista de algumas firmas produtoras de software educativo em Inglaterra. Se bem que muitos destes programas não sejam utilizáveis na sua forma original no nosso país, alguns deles — particularmente no caso de simulações — podem ser úteis. Notar que o número de empresas que se interessam por este campo aumenta continuamente, e que apenas são citadas aqui aquelas que já se dedicam a ele há algum tempo; por outro lado, a inclusão de uma firma não significa necessariamente uma aceitação da qualidade dos seus produtos, tal como a não inclusão de outra não implica um juízo negativo sobre o valor dos seus produtos.

Calpac Computer Software
108 Hermitage Woods Crescent,
St Johns, Woking, Surrey GU21 1UF

E.Z.U.G. (Educational ZX Users' Group)
Highgate School, Birmingham 12

Rose Cassettes,
148 Widney Lane, Solihull,
West Midlands, B91 3LH

SCISOFT,
5 Minster Gardens, Newthorpe,
Eastwood, Notts., NG16 2AT

Simon Software,
FREEPOST,
New End, Redditch, Worcestershire

APÊNDICE G

PUBLICAÇÕES QUE REGULARMENTE INCLUEM CRÍTICA DE SOFTWARE PARA O SPECTRUM

Note uma vez mais que esta lista não é de modo algum exaustiva, se bem que indique as publicações que parecem apresentar regularmente críticas de software. Como sabe, são praticamente lançadas revistas de informática todas as semanas (em inglês, claro...).

Educational Computing,
MAGSUB,
Oakfield House,
Perrymount Road
Haywards Heath,
West Sussex, RH16 3HD

Popular Computing Weekly,
Hobhouse Court,
19 Whitcombe Street,
London, WC2

Sinclair User,
ECC Publications
30-31 Islington Green,
London, N1 8BJ

ZX Computing,
145 Charing Cross Road,
London, WC2 OEE

Personal Computing Today
145 Charing Cross Road,
London, WC2H OEE

Your Computer.
IPC Electronic Press,
Quadrant House,
The Quadrant,
Sutton,
Surrey, SA2 5AS

APÊNDICE H

GLOSSÁRIO DE TERMOS DE COMPUTADOR

Acumulador — Parte da unidade lógica do computador que armazena resultados intermédios do processamento.

Alfanumérico — Formado por letras e números.

Álgebra de Boole — Corpo algébrico orientado para a tomada de decisões e a lógica, desenvolvido pelo matemático inglês George Boole, e na qual se fundamenta o funcionamento dos computadores.

Algoritmo — Sequência de passos necessários para resolver um problema.

APL — Significa *A Programming Language,* o que constitui uma linguagem de programação desenvolvida por Iverson no início da década de 60. Utiliza um conjunto de caracteres não standard.

ASCII — Significa American Standard Code for Information Interchange. Trata-se de um código praticamente universal para letras, números e símbolos, a cada um dos quais é atribuído um número entre 0 e 255 — por exemplo 65 para a letra A.

Assembler — Trata-se de um programa que converte outros programas, escritos numa linguagem "assembly" (nas quais uma instrução única, por exemplo ADD, é convertida também numa instrução única em linguagem-máquina) para a linguagem usada directamente pelo computador.

Base de dados — Um conjunto de dados, organizado de modo a permitir um acesso rápido por computador.

Basic — Significa Beginner's All Purpose Symbolic Instruction Code, sendo a linguagem mais usada em microcomputadores. É fácil de aprender, sendo muitas das instruções que usa semelhantes às palavras usadas em inglês normal.

Batch — Um grupo de transacções que devem ser processadas pelo computador de uma só vez, sem interrupção por qualquer operador.

Baud — Uma unidade de medida da velocidade a que é executada a transferência de dados. Indica geralmente o número de bits (unidades discretas de informação) por segundo.

Binário — Um sistema numérico que só utiliza dois símbolos, 0 e 1, ao contrário do sistema decimal normal onde se usam dez símbolos: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Os computadores "pensam" em binário.

Bit — Uma abreviatura das palavras inglesas "binary digit" (algarismo binário). Trata-se da mais pequena unidade de informação que pode ser reconhecida por um computador.

Buffer — Um mecanismo de armazenamento que guarda as entradas provenientes de um dispositivo como o teclado, libertando-as à velocidade indicada pelo computador.

Bug — Um erro do programa.

Bus — Um grupo de ligações eléctricas usadas para ligar um computador a um dispositivo auxiliar, ou a outro computador.

Bus S-100 — Trata-se de uma interface normalizada (ver RS-232) constituída por 100 linhas de comunicação paralelas utilizadas para ligar circuitos impressos no interior dos microcomputadores.

Byte — Um grupo de bits, normalmente 8, que são lidos ou nos quais se "escreve" em memória simultaneamente. A informação é sempre guardada em bytes completos.

Campo — Uma colecção de caracteres que constituem um grupo distinto, por exemplo um código de identificação, um nome ou uma data; os registos de um ficheiro são geralmente formados por campos.

COBOL — Significa Common Business Orientated Language, uma linguagem de programação principalmente usada em empresas.

Código de caracteres — O número em ASCII (ver ASCII) que corresponde ao símbolo considerado, por exemplo 32 para um espaço ou 65 para a letra "A".

Compilador — Um programa que traduz um outro, escrito numa linguagem de alto nível (próxima das linguagens de conversação normais) para a linguagem da máquina, que o computador compreende directamente.

Concatenar — Somar duas cadeias (strings).

CP/M — Significa Control Program/Monitor, constituindo um sistema de comando de disco praticamente universal desenvolvido e comercializado pela Digital Research, Pacific Grove, California.

Dados — Termo genérico aplicado à informação processada por um computador.

Debug — Eliminar os erros de um programa (bugs).

Disco — Um suporte de armazenamento magnético (também designado por "disco duro", "floppy", etc.) usado para guardar informação e programas de computador. Assemelham-se em certa medida a vulgares discos de reprodução sonora de 45 r.p.m., tendo no entanto dimensões variáveis. Para alguns sistemas existem também "micro-discos".

Documentação — Instruções e explicações escritas que acompanham um programa.

DOS — Significa Disk Operating System. Trata-se do programa que permite ao computador comandar um sistema de disco.

Dupla densidade — Designação dada aos discos quando são gravados usando uma técnica especial que, como o nome implica, duplica a quantidade de dados que o disco pode conter.

Endereço — Um número que identifica uma posição de memória onde se encontra informação.

EPROM — Significa Erasable Programmable Read Only Memory, um dispositivo que contém informação numa forma semi-permanente, e que obriga a uma exposição prolongada a radiação ultra-violeta para ser "limpo".

Ficheiro — Um grupo de registos relacionados entre si que são processados em conjunto, como um ficheiro de alunos, etc.

Firmware — Componentes sólidos de um sistema computador; são normalmente designados por "hardware", e os programas por "software"; mas os programas que são formados por ligações fixas dentro do computador são designados por "firmware". Em algumas circunstâncias o "firmware" pode ser alterado de uma forma limitada pelo "software".

Flag — Trata-se de um indicador de um programa, que dá a conhecer o estado de determinada condição.

Floxograma — Um esquema escrito da estrutura e modo de execução de um programa, usando símbolos gráficos especiais, como um losango para indicar uma decisão do computador, etc. Normalmente escrito antes de se começar a programar.

FORTRAN — Uma linguagem de alto nível, geralmente usada para trabalho de cálculo (significando FORmula TRANslation).

Gráficos — Um termo usado para qualquer saída do computador que não seja alfanumérica ou simbólica.

Hard Copy — Cópia da informação processada obtida numa impressora.

Hardware — As partes sólidas de um computador (ver Software e Firmware).

Hexadecimal — Um sistema numérico muito apreciado pelos programadores em código-máquina porque traduz bem os métodos

de armazenamento de números usados pelos computadores, e que se baseia no número 16 ao contrário do sistema decimal (que utiliza a base dez).

Impressora de linhas — Uma impressora que imprime de cada vez uma linha completa.

Impressora de matriz — Uma impressora que forma as letras e símbolos recorrendo a uma matriz de pontos, normalmente com as dimensões 8×8.

Input (Entrada) — Qualquer informação alimentada ao computador.

Instrução — Um elemento do código programado, que ordena ao computador que execute uma dada tarefa. Em linguagem-máquina, por exemplo, ADD ordena ao computador que execute uma soma.

Interpretador — Converte as linguagens de alto nível, mais próximas dos seres humanos, para a forma compreendida pelo computador.

I/O — Significa Input/Output (entrada/saída); um "port" I/O é um dispositivo usado pelo computador para comunicar com o mundo exterior.

Joystick — Um dispositivo analógico que alimenta um sinal ao computador em função da posição em que se encontra; usado geralmente em programas de jogos.

Linguagens de Alto Nível — Linguagens de programação semelhantes à língua inglesa. As linguagens de baixo nível estão mais próximas do computador. Como as de alto nível devem ser traduzidas para uma forma compreensível pela máquina antes de serem executadas, são normalmente mais lentas do que as de baixo nível.

Linguagem-máquina — Linguagem compreendida directamente pelo computador.

Mainframe — Palavra inglesa usada para designar os computadores de grandes dimensões. Outras categorias de computadores são os "minis" e os "micros".

Memória — Dispositivos usados pelo computador para guardar informação e programas, assim como o conjunto de instruções fixas guardadas na máquina que indica a esta como deve executar as ordens do programador. Existem basicamente dois tipos de memória (ver RAM e ROM).

Memória Dinâmica — Memória de computador que necessita de ser constantemente "recarregada" a fim de manter o seu conteúdo.

Memória Volátil — Um dispositivo de memória que perde o seu conteúdo quando o computador é desligado da rede de alimentação (ver memória, refrescar, ROM e RAM).

Microprocessador — Circuito integrado que se encontra no coração do computador. É ele que "pensa".

Modem — Significa MOdulator/DEModulator, sendo um dispositivo que permite a um computador comunicar com outro pelo telefone.

Monitor — *a*) Um visor de televisão usado pelo computador para comunicar com o exterior;

b) Um programa que permite ao computador compreender e executar certas instruções.

Network — Um grupo de computadores ligados entre si.

Octal — Um sistema numérico de base 8 (usando os algarismos 0, 1, 2, 3, 4, 5, 6, 7).

On-Line — Dispositivo que se encontra sob o comando directo do computador.

Output (saída) — Qualquer dado produzido pelo computador durante um processamento, quer seja apresentado no visor ou numa impressora, etc.

Pascal — Uma linguagem de alto nível, desenvolvida em finais da década de 1960 por Niklaus Wirth e que encoraja a programação estruturada.

PILOT — Uma linguagem de alto nível geralmente usada para desenvolver programas de educação.

Port — Um "buraco" de entrada ou saída no computador, através do qual é possível transferir dados.

Porta (Gate) — Um "componente" do computador capaz de tomar decisões, permitindo ao circuito funcionar de um modo ou de outro em função de certas condições.

Programa — Série de instruções destinadas a cumprir determinada tarefa.

Quilobyte — Uma unidade de medida de memória; um quilobyte (normalmente indicado pela letra K) é igual a 1024 bytes.

RAM — Significa Random Access Memory, designando a memória onde o computador guarda o programa que está a executar. O conteúdo da RAM pode ser alterado, enquanto que o da ROM não.

Refrescar — O conteúdo das memórias dinâmicas (ver Memória Dinâmica) deve receber periodicamente energia a fim de se poder manter. O sinal que "recorda" a memória do seu conteúdo é designado por sinal de "refrescar".

Registo — Uma posição da memória de um computador que armazena dados.

Reset — Um sinal que conduz o computador para o ponto em que se encontrava ao ser ligado.

ROM — Significa Read Only Memory. Ver RAM.

RS-232 — Uma interface standard (definida pela Electronic Industries Association) que liga um modem e equipamento associado a um computador.

Sintaxe — Tal como nas linguagens humanas, designa as regras estruturais que comandam o uso de uma linguagem de computador.

Sistema Operativo — É o programa ou série de programas fundamental para o computador, que controla o seu funcionamento básico.

SNOBOL — Uma linguagem de alto nível, desenvolvida pelos Bell Laboratories, que utiliza um sistema de reconhecimento de padrões e de manipulação de cadeias.

Software — Programa executado pelo computador.

Software de aplicações — Trata-se dos programas que são concebidos para realizar uma tarefa específica, como processamento de texto, etc.

Software de sistemas — Rotinas capazes de executar tarefas de administração dos recursos da máquina, ou auxiliam a escrever programas de aplicações, mas que não executam directamente tarefas de aplicação.

Stack — "Pilha" de posições de memória onde o computador guarda dados necessários ao seu funcionamento.

Subrotina — Um bloco de instruções, ou programa, que é chamado por outro.

Tempo Real — Quando o processamento acompanha o tempo do "mundo real", diz-se que é feito em "tempo real". Um exemplo será um programa que mostre o desenvolvimento de uma colónia de bactérias à velocidade a que esse desenvolvimento ocorre.

Time-sharing — Esta expressão é usada para indicar a presença num sistema de um vasto número de utilizadores, todos eles partilhando a mesma máquina.

Unidade Processadora Central (CPU) — O coração de um computador, onde são executadas funções aritméticas, lógicas e de comando.

ÍNDICE

AGRADECIMENTOS 7

1 — COMPUTADORES E EDUCAÇÃO 9

2 — IDEIAS BÁSICAS: O QUE É UM COMPUTADOR? 13

COMO FUNCIONAM OS COMPUTADORES? 13

3 — PROGRAMAÇÃO DO SPECTRUM EM BASIC 18

O teclado 18
Programa de testes 22
Adivinhar números 24
Lançamento de dados 28
Instruções adicionais 33

4 — TRATAMENTO DE CADEIAS E USO DESTAS EM MATEMÁTICAS 38

CHR$ 39
A utilidade de "TO" 40
LEN 41
VAL 44
Correcção para N algarismos significativos 46
Conversão para notação científica 48

5 — USO DO SPECTRUM PARA CÁLCULOS MAIS AVANÇADOS 50

Série de números 50
Série mistério 51

Números de Fibonnaci 52
Soma de séries 53
As raízes quadradas e Sir Isaac 55
Raízes quadradas por fracções contínuas 56
Raízes quadradas pela fórmula de Newton 57
Resolução de equações pelo método de Newton 59
Determinação de factores 64
Cálculo de N! (factorial) 65
Resolução de equações quadráticas 66
Série triangular 67
Triângulo de Pascal 68
Conversão da base decimal para a binária 69
Arredondar para o inteiro mais próximo 71
Exprimir números com N casas decimais 72
Geração de números primos 72
Testes de matemáticas 75
Testes de matemáticas: correlação/regressão 84
Séries temporais 91

6 — USO EFICAZ DOS GRÁFICOS DO SPECTRUM 93

I PARTE 93
Cruz de igreja 94
Floresta 95
Química 97
Pesos moleculares 99

7 — USO EFICAZ DOS GRÁFICOS DO SPECTRUM 104

II PARTE 104
O ciclo do Azoto 104
Desenho 106
Sinusóides 110
Bola em ressalto 110
Espirais 112
Formas 113
Manipulação de texto 117
Gerador de caracteres 119

8 — USO DO SPECTRUM NO ENSINO DE LÍNGUAS 125

Ortografia 125
Anagramas 127

Leitura rápida 129

9 — VALIDAÇÕES DE ENTRADAS 133

Fósforos 134

10 — PROGRAMAS DE ESCOLHA 137

11 — OUTROS PROGRAMAS COM INTERESSE 144

Histogramas e gráficos de barras 144
Gráficos de colunas 147
Rotina de ordenação 148
Comparação de qualidades diferentes 151
Desenho 152
Ângulo interior de um polígono regular 153
Depreciação 154
Dia da semana 155
Relógio de segundos 156
Média, desvio padrão, variância 158

12 — SOFTWARE EDUCATIVO PARA O SPECTRUM 160

Perguntas que deve fazer a si mesmo 162

13 — USO DO SPECTRUM NA ESCOLA PRÉ-PRIMÁRIA 163

ESTUDOS DE UM CASO 163
Como tudo começou 166
Uso de caixas de arquivo 167
Fichas de trabalho 168
O jogo do computador 176
O jogo do byte 177
Guardar o equipamento 177
Medalhas 177

14 — APLICAÇÕES ESPECÍFICAS NA ESCOLA PRÉ-PRIMÁRIA 179

Escrita manual 179
Auxílio à leitura 181

15 — MANTER O INTERESSE 183

O som 183

Feliz aniversário 186
Uso das cores do Spectrum 187
Bolhas 192
Pais 193
História 193

16 — JOGOS PARA O CÉREBRO 197

Leitor de mente 197
Abrir códigos 199
A estrela mágica 201
Palavras 205
Formas 208

17 — SUGESTÕES FINAIS 211

APÊNDICE A — SUGESTÃO DE UM TESTE SOBRE BASIC SPECTRUM 214

APÊNDICE B — IDEIAS DE EXERCÍCIOS E PROGRAMAS 216

APÊNDICE C — CONVERSÃO BINÁRIA 218

APÊNDICE D — SUGESTÃO DE UM TESTE SOBRE BASIC SPECTRUM 224

APÊNDICE E — PROLOG — PROgramação em LÓGica 229
Aprender a descrever os problemas 230
Programas específicos 232

APÊNDICE F — SOFTWARE EDUCATIVO COMERCIAL 233

APÊNDICE G — PUBLICAÇÕES QUE REGULARMENTE INCLUEM CRÍTICA DE SOFTWARE PARA O SPECTRUM 234

APÊNDICE H — GLOSSÁRIO DE TERMOS DE COMPUTADOR 235

Fotocomposto por
Navegráfica, Lda.
e impresso por
Scarpa, Lda.
em 1985 para a
EDITORIAL PRESENÇA