

ANO I N° 05

Cz\$ 1.150,00

CPU



MALA DIRETA

TECLADO INTELIGENTE

O MUNDO PERDIDO DA III DIMENSÃO

INSTRUÇÕES SECRETAS DO Z-80



A NEMESIS INFORMATICA está lançando mais uma série inédita de programas alucinantes e 100% NACIONAIS para o seu **MSX**!

MSX HELLO!

O SISTEMA OPERACIONAL **HELLO** chegou para suprir todas as necessidades dos usuários de drives com o **MSX**. O **HELLO** possui recursos inéditos como formatação personalizada por sistema de **LABELS**, recuperação de discos com **ERROS DE E/S**, testes de alinhamento radial de drives e testes de **HARDWARE**, além de todas as funções de sistema com maior rapidez e confiabilidade. Apenas em disco, com manual detalhado por Cz\$ 6.200,00.

MSX TURBO SPEED

Este utilitário, de extrema simplicidade de operação. Pode acelerar em até 60 vezes o seu programa em BASIC ou Linguagem de Máquina. Pode ser fornecido em cartucho por Cz\$ 12.000,00; ou na versão de disco ou fita por apenas Cz\$ 5.600,00.



JOE KOWALSKI N°2

Para quem não sabe **JOE KOWALSKI** é o protagonista dos mais fantásticos jogos já criados para os microcomputadores **MSX**.

Para os que já possuem os jogos da primeira série (**CHAUNTED HOUSE**, **PINBALL BLASTER**, **BLOW-UP** e **GUTI BLASTER**), a NEMESIS está lançando a segunda série de jogos: **VORTEX RAIDER**, um super duelo espacial; **HABILIT**, um jogo de muita estratégia e muita ação e **MAZE MASTER**, uma aventura fantástica no labirinto. Para quem ainda não possue, está e' a grande oportunidade:

Série JOE KOWALSKI número 1:
Apenas em disco - Cz\$ 7.000,00
Série JOE KOWALSKI número 2:
Apenas em disco - Cz\$ 8.000,00

PORTFOLIO MSX

Um sensacional programa de **AGENDA**, **DIARIO** e **LISTA TELEFÔNICA**, totalmente iconográfico e simples de se usar. Possui ainda um **CALENDARIO PERPETUO**, **CALCULADORA** e recursos de procura lógica entre dados e datas. O Programa perfeito para seu dia-a-dia de 1989.

Acompanha o Programa, um manual detalhado e um disco com a programação do ano que vem. Apenas em disco - Cz\$ 5.600,00

MSX PAGE MAKER N°1

Esta página da revista CPU INFORMATICA foi totalmente composta no sistema "Desktop-publishing", a ultima novidade em software de MSX.

O **MSX PAGE MAKER** é um software 100% nacional, desenvolvido por Alexandre Cruz e equipe da NEMESIS INFORMATICA.

Totalmente compativel com o **GRAPHOS III** de Renato Degiovanni, a nível de alfabetos, shapes e telas, além dos acessórios desenvolvidos pela NEMESIS :

PAGE MAKER FONTES N°1

PAGE MAKER FONTES N°2

Alfabetos inéditos para uso com o **MSX PAGE MAKER**, **GRAPHOS III** e outros softs compatíveis.

PAGE MAKER CARTOONS N°1

Uma coleção de shapes e figuras inéditas para compor suas páginas gráficas ou seus desenhos, incluindo molduras, figuras humanas, vinhetas, veículos e apliques decorativos. Compativel com o **MSX PAGE MAKER** e **GRAPHOS III**.

ACESSÓRIOS "MS-DESTAQUE"

Os acessórios da Linha MS DESTAQUE também são compatíveis com o **MSX PAGE MAKER**. Se você ainda não possui estes "best-sellers", aproveite a ocasião. São telas, alfabetos e shapes.

TABELA DE PREÇOS

MSX PAGE MAKER 1.0	Cz\$ 7.200,00
MSX PAGE MAKER FONTES 1	Cz\$ 6.200,00
MSX PAGE MAKER FONTES 2	Cz\$ 6.200,00
MSX PAGE MAKER CARTOONS 1	Cz\$ 6.200,00

MSX PAGE MAKER KIT: Incluindo o **MSX PAGE MAKER** os **FONTES 1 e 2** e o **CARTOONS 1** - Cz\$ 20.000,00

GRAPHOS III versão 1.3	2 OTB
ALFABETOS número 1	1 OTB
SHAPES número 1	1 OTB
TELAS número 1	1 OTB

MSX CHART

Um programa gerenciador de graficos comerciais e estatísticos com recursos inéditos sobre os programas do gênero existentes até então. Compatível com o **MSX PAGE MAKER** possibilitando a produção de relatórios impressos com altíssima qualidade. Em disco Por Cz\$ 5.600,00

TESTES INFORMATICA

Para obter informações mais detalhadas sobre nossos produtos entre em contato conosco:

NEMESIS INFORMATICA LTDA.
Caixa Postal 4583
Cep. 20.001 Rio de Janeiro
ou pelo telefone:

NEMESIS - (021) 222-4900
ou venha pessoalmente ao:

"SHOW-ROOM" **NEMESIS**
Rua Sete de Setembro n. 92
sala 1910 Centro - RJ/RJ.

CPU

ÁGUA INFORMÁTICA LTDA.
AV. N. SRA. DE COPACABANA 605/804
COPACABANA
RIO DE JANEIRO - RJ
CEP 22040
TEL: (021)235-3541
TELEX: 2138953

DIRETOR RESPONSÁVEL
GONÇALO MURTEIRA

DIRETORIA TÉCNICA
ANTONIO F. S. SHALDERS
CARLOS E. A. MOREIRA
ANDRÉ L. DE FREITAS
J. L. FONSECA

JORNALISTA RESPONSÁVEL
DOLAR TANUS
REGISTRO 430-RS

REVISÃO DE TEXTO
LAURA MARIA PINTO

CAPA
JOSÉ AGUILERA

ASSINATURAS
EDUARDO SIMPLÍCIO

ADMINISTRAÇÃO
JOSÉ A. NASCIMENTO

PROJETO GRÁFICO
LUCIANA MONTENEGRO

IMPRESSÃO
EDITORIA LUA NOVA

CPU é uma publicação da Águia Informática. Todos os direitos reservados. Proibida a reprodução parcial ou total do conteúdo desta revista por qualquer meio sem autorização expressa da editora.

Os artigos assinados são de total e única responsabilidade dos autores.

Os circuitos, dispositivos, componentes, etc., descritos na revista podem estar sob a proteção de patentes. Os circuitos publicados só poderão ser confeccionados sem qualquer fim lucrativo.

Os programas apresentados aos leitores, mesmo se fornecidos em disquete, são de propriedade dos autores, cabendo a eles todos os direitos previstos em lei.

EDITORIAL

O ano de 1988 está chegando ao final e, apesar de ter sido um ano cheio de surpresas econômicas e políticas, na área do MSX não podemos verificar lançamentos de hardware por parte dos fabricantes que lançaram o padrão no Brasil e que anunciaram uma série de periféricos para a linha. O lançamento de periféricos ficou a cargo de empresas que acreditam no desenvolvimento do MSX no Brasil.

Na área de software podemos dizer que o mercado está a pleno vapor, com Concursos de Software e lançamento de programas novos, desenvolvidos no Brasil, visando a atender as nossas necessidades e de acordo com as características técnicas dos equipamentos brasileiros.

A área editorial também vem apresentando um desenvolvimento significativo, com lançamento de livros de excelente qualidade e conteúdo praticamente todos os meses.

No número anterior de CPU, tivemos alguns problemas com a gráfica que efetuou a impressão da revista. Algumas falhas também foram constatadas nas listagens, isto porque, a partir do quarto número, toda a diagramação passou a ser feita através de processo eletrônico, inclusive as listagens que receberam tratamento idêntico ao texto. A partir deste número continuamos a utilizar processos eletrônicos para a diagramação da revista, sendo que as listagens dos programas utilizadas foram impressas em uma impressora comum. Neste número publicamos novamente as listagens do número anterior que podem ter ocasionado dificuldades na leitura.

Aproveito a oportunidade para, em nome de toda a equipe de CPU, desejar um Feliz Natal e Próspero Ano Novo.

Nos encontraremos novamente em janeiro, no sexto número de CPU, que continuará crescendo como sempre.

GONÇALO MURTEIRA

ÍNDICE

UM TECLADO INTELIGENTE PARA O MSX	5
AS INSTRUÇÕES SECRETAS DO Z-80	10
GERENCIANDO ARQUIVOS EM DISCO	14
INTELIGÊNCIA ARTIFICIAL EM PASCAL	18
INTERPRETADOR DE EXPRESSÕES	20
UTILIZANDO O DATA CORDER	26
CONECTIVIDADE MSX-PC-MAINFRAME	28
CÁLCULO DE CIRCUITOS RESSONANTES L-C	32
DBASEII PLUS MSX- D BEST	34
CURSO DE PASCAL II	36
THE TRAIN GAME SPRINTER	42
O MUNDO PERDIDO DA III DIMENSÃO	44
BATTLE FOR MIDWAY	49
BR-116 A RODOVIA DA MORTE	50
SOS FELINO	54
JOGO DA MEMÓRIA	55
BOLICHE	56

SEÇÕES

MSX NEWS	4
LIVROS	38
CARTAS	40
JOGOS & HIGH SCORES	51
JOGOS LANÇAMENTOS	52

MSX NEWS

NEWSOFT

A Newsoft, empresa pioneira no Rio de Janeiro em distribuição de software para a linha MSX, baseada em uma nova política empresarial, lança em todo o País o I Concurso Nacional de Software, para MSX, com o intuito de descobrir e revelar novos talentos no mercado de software.

O evento conta com o apoio do Cartão Nacional, que fará a distribuição dos prêmios aos ganhadores.

Todos os detalhes do concurso podem ser encontrados neste número de CPU.

A Newsoft também está vendendo diversos periféricos para a linha a MSX com a vantagem de o cliente poder pagar com o seu Cartão Nacional, tendo, portanto, até 30 dias para pagar.

Esta vantagem também é estendida para a compra de jogos e aplicativos.

Em breve, a Newsoft estará em novo endereço, com modernas instalações, aprimorando o atendimento e a qualidade que sempre dispensou à sua clientela.

EDITORIA ALEPH +50 DICAS

Será lançado, pela Editora Aleph, de São Paulo, o livro "+50 DICAS PARA MSX".

O livro vem a atender pedidos dos leitores do "CEM DICAS PARA MSX", que é um dos livros indispensáveis para quem tem MSX e programa, e apresenta 50 novas rotinas que poderão ser utilizadas pelos programadores.

Além de apresentar os programas, os autores fazem uma descrição completa de como chegaram aquele programa, tornando, assim, muito mais fácil a assimilação por parte do leitor da idéia do autor e facilitando o seu uso posterior.

NASHUA

A Nashua iniciará, em breve, a produção de disquetes de 5 1/4" no Brasil, visando a atender o mercado nacional.

TELCON - MODEM DE DISCAGEM AUTOMÁTICA

Foi lançado pela Telcon Telemática, empresa de Porto Alegre, que lançou, já há algum tempo atrás o Multimodem, um modem de discagem automática. O produto estará brevemente disponível nos revendedores autorizados da TELCON.

Os modems da Telcon permitem a conexão do MSX com o Videotexto, Cirandão e comunicação Micro a Micro, possuindo software, desenvolvido pela própria empresa, podendo ser utilizado tanto pelos usuários de unidade de disco ou fita.

NEMESIS

A Nemesis está lançando uma série de programas aplicativos novos, dos quais podemos destacar o Page Maker, o Portfolio, que vem a ser uma agenda computadorizada, com calendário perpétuo e o Hello, que vem a ser um sistema operacional que simplifica todas as funções de operação com disco, possuindo ainda testes de Hardware, tais como: velocidade de rotação de drive, alinhamento radial de cabeçote, teste de RAM e VRAM, mostrando o mapa de ocupação do disco e diversas outras funções.

O telefone da Nemesis é (021) 222-4900

CHAMPION SOFTWARE

A Champion Software, empresa Paulista da área de software, está de mudança marcada para breve.

O novo endereço da Champion será:

Rua Clélia 1837
Lapa - São Paulo - SP

MSX WORD

A Ciência Moderna acaba de lançar o livro "MSX WORD - das versões 1.6 à 3.0", de autoria de Sérgio Guy Pinheiro Elias e Paulo Roberto Pinheiro Elias, autores do Livro "dBase II Plus para MSX", também da Ciência Moderna e já analisado em CPU.

O MSX Word vem sendo desenvolvido pela Cibertron, que lançou há pouco tempo a versão 3.0 (veja CPU número 04) e, sem sombra de dúvida, é o processador de texto mais utilizado pelos usuários do MSX, sendo baseado no TASWORD, da Tasman Software Ltd.

Neste livro, os autores publicam as especificações técnicas mais importantes e uma documentação completa de como adaptar o MSX-WORD de qualquer versão às características da impressora do usuário.

De nada adianta ter um excelente software se não sabemos os seus comandos e de como proceder para tirar o melhor proveito do programa e ter um rendimento de 100%.

Da forma que o livro é apresentado substitui com inúmeras vantagens o manual que acompanha o software e, certamente, é um investimento que será rapidamente recuperado.

A Ciência Moderna fica situada à Av. Rio Branco 156 loja 127 Rio de Janeiro - RJ - 20043 - Telefones: 021-262.5723 e 021-240-9327 - atendendo a pedidos de todo o Brasil, possuindo, além deste, todos os demais livros disponíveis, atualmente, para MSX.

UM TECLADO INTELIGENTE PARA O MSX

PIERLUIGI PIAZZI,
MILTON MALDONADO

Uma grande quantidade dos atuais usuários de MSX se iniciou no mundo da informática com um pequeno Sinclair (TK 82, TK 85 ou CP 200).

Assim sendo, as comparações são inevitáveis. O MSX, obviamente, ganha de longe, por ser uma máquina de muito maior porte e recursos. Apesar disso, alguma saudadezinha do pequeno SINCLAIR ainda sobra, pois, com uns miseráveis 8K de ROM (comparados aos 32K da ROM do MSX!) fazia algumas coisas que o MSX não faz. Uma delas, por exemplo, era a poderosa função VAL.

No MSX, a string que serve de argumento ao VAL deve conter exclusivamente caracteres de algarismos (os outros são ignorados). No SINCLAIR ela pode conter qualquer coisa (algarismos, variáveis, fórmulas, etc), de maneira a substituir, com vantagem, o comando DEFFN.

O MSX, em compensação, é uma máquina versátil. Tão versátil a ponto de poder ser configurada para poder fazer coisas para as quais não foi projetada.

Ela é a concretização do sonho de qualquer programador (especialmente em Basic e Assembly). Só para exemplificar, desenvolvemos uma rotina que permite emular os recursos do VAL do SINCLAIR no MSX e que será brevemente publicada no livro "+50 dicas para MSX".

Esta versatilidade pode chegar às raias do absurdo: num ataque de extremado saudosismo, os programadores de uma softhouse, a XSM (até no nome fazem o MSX plantar bananeira!) desenvolveram um EMULADOR SINCLAIR, programa que simplesmente transforma o MSX nun SINCLAIR, lendo e rodando as fitas de toda uma biblioteca de programas que estava mofando em alguma gaveta esquecida.

Dentro desta onda de "saudosismo", resolvemos

escrever esta matéria: uma das queixas dos "sinclaristas" em relação ao MSX é referente ao tempo de digitação de longos programas em BASIC. Afinal de contas, no SINCLAIR bastava apertar uma tecla (ou uma combinação de teclas) para que na tela aparecesse toda a palavra reservada. Se, no começo de uma linha de BASIC, digitássemos Y, apareceria o RETURN, enquanto que no MSX é necessário digitar letra por letra: R-E-T-U-R-N.

O problema é parcialmente contornável se atribuirmos às teclas de função (F1 a F10) as palavras reservadas mais frequentes. Assim mesmo, estaremos limitados a apenas 10.

Porque não aproveitar a versatilidade do MSX para configurar seu teclado de maneira a emular o do SINCLAIR.

Aproveitando a tecla SELECT do Expert (SLCT do Hotbit), que não é usada quando se está no modo edição de um programa em BASIC, elaboramos uma rotina em Linguagem de Máquina que reconfigura o teclado de maneira a produzir uma palavra reservada toda a vez que a tecla Select é pressionada simultaneamente à de uma letra. As combinações que geram as palavras reservadas estão listadas na figura 1

Esta rotina foi inicialmente publicada no livro "100 DICAS PARA MSX", na página 15, tendo sido introduzidas, apenas, algumas pequenas modificações para reduzir as chances de erro na digitação.

Digite, portanto, o programa da figura 2, tomando bastante cuidado com os códigos hexadecimais das linhas DATA.

Como o programa não é muito curto, grave-o periodicamente, durante a digitação, para não perder todo o trabalho em caso de falta de energia elétrica (ou sobrinhos irrequietos tropeçando no fio da tomada!).

Quando o programa estiver pronto e gravado, digite RUN.

**MSX
CENTER**

GAMA SOFTWARE

GAMA SOFTNEWS. Aqui tem tudo o que lhe interessa sobre seu MSX. Notícias sobre o CURSO GAMA DE BASIC, o 1º curso de Basic por correspondência do Brasil. GAMA TELESOFT, saiba como receber pela GAMA SOFTWARE os seus programas gravados em disco, através do telefone. GAMA HARDWARE, como adquirir toda a linha de periféricos e suprimentos para o seu MSX, através do correio. E a linha pioneira de programas para MSX e para o MSX-2, que assim como o GAMA TELESOFT é uma inovação exclusiva da sua **GAMA SOFTWARE**.

GRÁTIS! *Solicite assinatura
do nosso catálogo!*

Preencha o cupom abaixo e remeta para:
Gama Software Ltda. - Caixa Postal 94368 - CEP 25800
Três Rios - RJ - Tel. (0242) 52-0687

NOME _____

ENDEREÇO _____

BAIRRO _____ CEP _____

CIDADE _____ ESTADO _____

DATA ____ / ____ / ____ ASSINATURA _____

```

1 L=12:SCREEN 0:WIDTH 40          61 DATA 41,52,50,54,52,28,00,43
2 FOR E=&HD000 TO &HD1D7 STEP 8    62 DATA 4F,53,28,00,45,58,50,28
3 S=0                            63 DATA 00,53,54,52,24,28,00,4C
4 FOR X=E TO E+7                 64 DATA 4E,28,00,0C,50,72,6F,67
5 READ C$:Y=VAL("&H"+C$)         65 DATA 72,61,6D,61,20,65,73,63
6 S=S+Y                          66 DATA 72,69,74,6F,20,70,6F,72
7 POKE X,Y                       67 DATA 3A,0D,0A,54,48,45,20,50
8 NEXT X                         68 DATA 49,4C,4F,54,20,65,6D,20
9 L=L+1                          69 DATA 4A,61,6E,65,69,72,6F,2F
10 PRINT USING##";L,:PRINT"=";   70 DATA 31,39,38,38,2E,00,7E,A7
:PRINT USING####";S,:PRINT" ";  71 DATA C8,CD,A2,00,23,18,F7,00
11 NEXT E                        72 DATA FIM
12 GOTO 73
13 DATA 21,9B,D1,CD,CE,D1,21,16
14 DATA D0,22,A5,FD,3E,C3,32,A4
15 DATA FD,AF,32,D7,D1,C9,4F,3A
16 DATA D7,D1,A7,79,20,05,FE,18
17 DATA 28,49,C9,FE,41,38,3D,FE
18 DATA 5B,30,08,D6,40,47,21,72
19 DATA D0,18,0E,FE,61,38,2D,FE
20 DATA 7B,30,29,D6,60,47,21,08
21 DATA D1,7E,A7,23,20,FB,10,F9
22 DATA 3E,C9,32,A4,FD,7E,A7,28
23 DATA 06,CD,A2,00,23,18,F6,3E
24 DATA C3,32,A4,FD,AF,32,D7,D1
25 DATA C1,C3,DA,08,4F,AF,32,D7
26 DATA D1,79,C9,3E,FF,32,D7,D1
27 DATA AF,C9,00,4E,45,57,00,42
28 DATA 45,45,50,00,43,4F,4E,54
29 DATA 00,44,49,4D,20,00,52,45
30 DATA 4D,20,00,46,4F,52,20,00
31 DATA 47,4F,54,4F,20,00,47,4F
32 DATA 53,55,42,20,00,49,4E,50
33 DATA 55,54,20,00,4C,4F,41,44
34 DATA 20,00,4C,49,53,54,20,00
35 DATA 4C,4C,49,53,54,20,00,4D
36 DATA 4F,54,4F,52,20,00,4E,45
37 DATA 58,54,20,00,50,4F,4B,45
38 DATA 20,00,50,52,49,4E,54,00
39 DATA 50,53,45,54,20,28,00,52
40 DATA 55,4E,00,53,41,56,45,20
41 DATA 00,54,52,4F,4E,00,49,46
42 DATA 20,00,43,4C,53,00,50,52
43 DATA 45,53,45,54,20,28,00,43
44 DATA 4C,45,41,52,00,52,45,54
45 DATA 55,52,4E,00,45,4E,44,00
46 DATA 00,46,52,45,28,00,49,4E
47 DATA 4B,45,59,24,00,44,53,4B
48 DATA 46,28,00,41,54,4E,28,00
49 DATA 54,41,4E,28,00,53,47,4E
50 DATA 28,00,41,42,53,28,00,53
51 DATA 51,52,28,00,41,53,43,28
52 DATA 00,56,41,4C,28,00,4C,45
53 DATA 4E,28,00,55,53,52,00,33
54 DATA 2E,31,34,31,35,39,32,37
55 DATA 21,00,4E,4F,54,00,50,45
56 DATA 45,4B,28,00,54,41,42,28
57 DATA 00,53,49,4E,28,00,49,4E
58 DATA 54,28,00,53,54,52,49,4E
59 DATA 47,24,28,00,52,4E,44,28
60 DATA 00,43,48,52,24,28,00,56
61 DATA 41,52,50,54,52,28,00,43
62 DATA 4F,53,28,00,45,58,50,28
63 DATA 00,53,54,52,24,28,00,4C
64 DATA 4E,28,00,0C,50,72,6F,67
65 DATA 72,61,6D,61,20,65,73,63
66 DATA 72,69,74,6F,20,70,6F,72
67 DATA 3A,0D,0A,54,48,45,20,50
68 DATA 49,4C,4F,54,20,65,6D,20
69 DATA 4A,61,6E,65,69,72,6F,2F
70 DATA 31,39,38,38,2E,00,7E,A7
71 DATA C8,CD,A2,00,23,18,F7,00
72 DATA FIM
73 PRINT:PRINT:PRINT"CONFIRMA?(S/N)"
74 A$=INPUT$(1)
75 IF A$="S" OR A$="s" THEN GOTO 79 ELSE
    INPUT"EM QUE LINHA TEM ERRO";L
76 LOCATE 0,20
77 PRINT "LIST";L,:REM-TECLE RETURN!
78 LOCATE 12,17:STOP
79 DEFUSR=&HD000:POKE 0,USR(0):PRINT:PRI
NT
80 FOR I=65 TO 90:PRINT "<SELECT>+"CHR$(I);;" = ";CHR$(24);CHR$(I):FOR T=0 TO 100
    :NEXT T:NEXT I
81 FOR I=97 TO 122:PRINT "<SELECT>+"CHR$(I);;" = ";CHR$(24);CHR$(I):FOR T=0 TO 10
    0:NEXT T:NEXT I
82 END

```

THUNDERSOFT THE NAME OF MSX

PEÇA O NOSSO SUPER
CATÁLOGO GRÁTIS!
TEMOS APPLE TAMBÉM!

JOGOS_200
APLICATIVOS_800
COPIADORES_1000
CP/M_1000
PREÇO DO DISCO_700
PREÇO DA FITA_800
CORREIO_200
ENTREGA EM 24 HORAS +
CORREIO



SUPER PACOTES

PACOTE Nº 1

10 jogos — apenas 2.200
(disco incluído)

PACOTE Nº 4

1 disco cheio — 2.200
2 disco cheios — 3.200

PACOTE Nº 2

10 jogos — apenas 2.700
(fita incluída)

PACOTE Nº 5

Aplicativo + copiador —
1.800(disco incluído)

PACOTE Nº 3

MSX TOOLS I e II apenas
3.200 disco incluído)

PACOTE Nº 6

10 Aplicativos + disco —
apenas 5.800

LANÇAMENTOS DO MES
TRANTOR • ELITE • BKT II • THUNDER BOOK • HARD
COPY • IMAGINE (EXCLUSIVO)

A cada 5 programas escolha 1 grátils

Para outros estados, os pedidos deverão ser feitos através de cheque
nominal e cruzado a MARCO ANTONIO TROV'A VAZ. R. Carvalho
Alvim 278/501, Rio de Janeiro, RJ, CEP 20510 Tel.: (021) 268-6360

FIGURA 2

<SELECT>+A = NEW	<SELECT>+a = FREC
<SELECT>+B = BEEP	<SELECT>+b = INKEY\$
<SELECT>+C = CONT	<SELECT>+c = DSKFC
<SELECT>+D = DIM	<SELECT>+d = ATNC
<SELECT>+E = REM	<SELECT>+e = TANC
<SELECT>+F = FOR	<SELECT>+f = SGNC
<SELECT>+G = GOTO	<SELECT>+g = ABSC
<SELECT>+H = GOSUB	<SELECT>+h = SQRC
<SELECT>+I = INPUT	<SELECT>+i = ASCC
<SELECT>+J = LOAD	<SELECT>+j = VALC
<SELECT>+K = LIST	<SELECT>+k = LENC
<SELECT>+L = LLIST	<SELECT>+l = USRC
<SELECT>+M = MOTOR	<SELECT>+m = 3.1415927!
<SELECT>+N = NEXT	<SELECT>+n = NOT
<SELECT>+O = POKE	<SELECT>+o = PEEKC
<SELECT>+P = PRINT	<SELECT>+p = TABC
<SELECT>+Q = PSET C	<SELECT>+q = SINC
<SELECT>+R = RUN	<SELECT>+r = INTC
<SELECT>+S = SAVE	<SELECT>+s = STRING\$C
<SELECT>+T = TRON	<SELECT>+t = RNDC
<SELECT>+U = IF	<SELECT>+u = CHR\$C
<SELECT>+V = CLS	<SELECT>+v = VARPTRC
<SELECT>+W = PRESET C	<SELECT>+w = COSC
<SELECT>+X = CLEAR	<SELECT>+x = EXPC
<SELECT>+Y = RETURN	<SELECT>+y = STR\$C
<SELECT>+Z = END	<SELECT>+z = LNC

Deverá aparecer na tela a figura 3 pedindo confirmação.

Confira cuidadosamente todos os números da sua tela com os da figura 3. Se algum deles não bater, você poderá achar rapidamente a linha na qual o erro foi cometido.

FIGURA 3

13=1072	14=1131	15=1240	16=1027	17=1004
18= 643	19= 952	20= 634	21=1085	22=1063
23= 740	24=1311	25=1133	26=1322	27= 676
28= 526	29= 401	30= 372	31= 495	32= 497
33= 489	34= 380	35= 501	36= 503	37= 507
38= 429	39= 470	40= 498	41= 466	42= 420
43= 444	44= 527	45= 460	46= 412	47= 495
48= 377	49= 499	50= 377	51= 458	52= 412
53= 419	54= 411	55= 423	56= 439	57= 425
58= 524	59= 415	60= 383	61= 500	62= 479
63= 401	64= 538	65= 764	66= 815	67= 418
68= 586	69= 759	70= 557	71= 873	

Digamos, por exemplo, que você obteve a tela da figura 4.

Checando esta tela com a da figura 3, você percebe que na sexta linha, ao invés de aparecer 39=470, está 39=473.

Consequentemente, você não confirmará a digitação, teclando N (de Não).

FIGURA 4

```
13=1072 14=1131 15=1240 16=1027 17=1004  
18= 643 19= 952 20= 634 21=1085 22=1063  
23= 740 24=1311 25=1133 26=1322 27= 676  
28= 526 29= 401 30= 372 31= 495 32= 497  
33= 489 34= 380 35= 501 36= 503 37= 507  
38= 429 39= 473 40= 498 41= 466 42= 420  
43= 444 44= 527 45= 468 46= 412 47= 495  
48= 377 49= 499 50= 377 51= 458 52= 412  
53= 419 54= 411 55= 423 56= 439 57= 425  
58= 524 59= 415 60= 383 61= 500 62= 479  
63= 401 64= 538 65= 764 66= 815 67= 418  
68= 586 69= 759 70= 557 71= 873
```

CONFIRMA?(S/N)

color auto goto list run

Basta, então, checar a linha listada com sua correspondente à listagem da figura 2. Leve o cursor até o erro e corrija-o. No nosso exemplo, um 8 foi confundido com um B (figura 6). Após a correção, tecle RETURN e rode o programa novamente. Repita o processo até ter certeza de que não há mais nenhum erro. Neste ponto, você confirma teclando S (de Sim).

FIGURA 6

```
18= 643 19= 952 20= 634 21=1085 22=1063  
23= 740 24=1311 25=1133 26=1322 27= 676  
28= 526 29= 401 30= 372 31= 495 32= 497  
33= 489 34= 380 35= 501 36= 503 37= 507  
38= 429 39= 473 40= 498 41= 466 42= 420  
43= 444 44= 527 45= 468 46= 412 47= 495  
48= 377 49= 499 50= 377 51= 458 52= 412  
53= 419 54= 411 55= 423 56= 439 57= 425  
58= 524 59= 415 60= 383 61= 500 62= 479  
63= 401 64= 538 65= 764 66= 815 67= 418  
68= 586 69= 759 70= 557 71= 873
```

CONFIRMA?(S/N)
EM QUE LINHA TEM ERRO? 39

```
Break in 78  
Ok  
LIST 39 :REM-TECLE RETURN!  
39 DATA 50,53,45,54,20,28,00,52  
Ok
```

color auto goto list run

Neste momento, aparecerá uma mensagem perguntando em que linha está o erro.

Você deverá digitar 39 (+ RETURN).

Feito isto, o MSX gerará um comando de listagem, colocando o cursor sobre o próprio comando (figura 5).

Aperte RETURN e a linha em questão será listada a seguir.

FIGURA 5

```
13=1072 14=1131 15=1240 16=1027 17=1004  
18= 643 19= 952 20= 634 21=1085 22=1063  
23= 740 24=1311 25=1133 26=1322 27= 676  
28= 526 29= 401 30= 372 31= 495 32= 497  
33= 489 34= 380 35= 501 36= 503 37= 507  
38= 429 39= 473 40= 498 41= 466 42= 420  
43= 444 44= 527 45= 468 46= 412 47= 495  
48= 377 49= 499 50= 377 51= 458 52= 412  
53= 419 54= 411 55= 423 56= 439 57= 425  
58= 524 59= 415 60= 383 61= 500 62= 479  
63= 401 64= 538 65= 764 66= 815 67= 418  
68= 586 69= 759 70= 557 71= 873
```

CONFIRMA?(S/N)
EM QUE LINHA TEM ERRO? 39

```
Break in 78  
Ok  
LIST 39 :REM-TECLE RETURN!
```

color auto goto list run

O programa salta para a fase seguinte, colocando na tela o pseudônimo do autor e listando todas as combinações de SELECT e letras que geram as palavras reservadas (figura 7).

FIGURA 7

Programa escrito por:
THE PILOT em Janeiro/1988.

```
<SELECT>+A = NEW  
<SELECT>+B = BEEP  
<SELECT>+C = CONT  
<SELECT>+D = DIM  
<SELECT>+E = REM  
<SELECT>+F = FOR  
<SELECT>+G = GOTO  
<SELECT>+H = GOSUB  
<SELECT>+I = INPUT  
<SELECT>+J = LOAD  
<SELECT>+K = LIST  
<SELECT>+L = LLIST  
<SELECT>+M = MOTOR  
<SELECT>+N = NEXT  
<SELECT>+O = POKE  
<SELECT>+P = PRINT  
<SELECT>+Q = PSET  
<SELECT>+R = RUN  
<SELECT>+S = SAVE
```

color auto goto list run

Você pode, agora, gravá-lo como um programa BASIC ou como uma rotina em Linguagem de Máquina. Neste último caso, os comandos para gravação e leitura estão relacionados na figura 8.

FIGURA 8

```
Para salvar:  
BSAVE "TECLADO.BIN", &HD000, &HD1D7 (drive)  
BSAVE "CAS:TECLAD", &HD000, &HD1D7 (grav.)  
  
Para carregar:  
BLOAD "TECLADO.BIN", R (drive)  
BLOAD "CAS:TECLAD", R (grav.)
```

Tome cuidado para não invadir, com o programa que você está digitando, a região de memória entre os endereços 53248 (&HD000) e 53719 (&HD1D7), pois é lá que se localiza a rotina de redefinição do teclado.

Seu MSX, agora, não tem nada a invejar a um Sinclair, com relação à digitação de palavras reservadas. Use esta rotina para digitar programas em BASIC. Inicialmente, você demorará até mais que o normal, mas vale a pena investir um pouco de seu tempo nisso. Depois de um certo tempo, você ficará tão familiarizado com esta técnica que sua velocidade de digitação de um programa chegará a dobrar.

Pierluigi Piazzzi é Diretor Editorial da Editora Aleph e co-autor dos livros:

Coleção de Programas para MSX vol.1 e 2, Usando a Planilha Eletrônica no MSX, Aprofundando-se no MSX, Como usar seu HOTBIT, Curso de BASIC MSX, Curso de Música para MSX, Drives Leopard de 3 1/2", 100 dicas para MSX e +50 dicas para MSX (em lançamento).

Milton Maldonado Jr. (The Pilot) é co-autor dos livros:

— Coleção de Programas para MSX Vol.1 e 2, Aprofundando-se no MSX, Programação Avançada em MSX, 100 dicas para MSX e +50 dicas para MSX.

— O programa fornecido neste artigo foi, originalmente, publicado no livro “100 DICAS PARA MSX”, da Editora Aleph, e foi testado na presente versão num Expert 1.1, gentilmente cedido pela Gradiente, e num HOTBIT.

Este programa pode ser usado por usuários de MSX, mas sua comercialização, seja na forma de programa isolado, seja na forma de sub-rotina de outro software, é vedada por lei sem o expresso consentimento, por escrito, da Aleph Publicações e Assessoria Pedagógica Ltda.

UM PROFISSIONAL ACIMA DE QUALQUER SUSPEITA

dBASE II PLUS MSX. Com ele você faz o melhor negócio.

O dBASE II Plus MSX é uma forma completamente nova de gerenciar seus dados.

E uma linguagem que permite criar, de forma fácil e rápida, um sistema completo de informações para seu negócio que faz exatamente o que você quer. Contabilidade, mala direta, controle de estoque, gerenciamento de produção, perfil de cliente, enfim, sistemas que irão manipular os problemas modernos que surgem a cada dia.

O dBASE II Plus MSX não é o único meio de manipular dados no seu microcomputador, mas é o melhor! Médicos e advogados, contadores e vendedores, corretores e imobiliárias, donas-de-casa, pequenas e grandes empresas, todos utilizarão melhor, seus dados, com o dBASE II Plus MSX.

Tudo o que você tem a fazer é trabalhar com as informações usando nomes e conceitos que você já conhece, e que seu micro passa a conhecer através do dBASE II. Você pode trabalhar com ele na forma Interativa ou Programada.

Portanto a comunicação é perfeita: clara, objetiva e rápida.

Produzido pela DATALOGICA-PRINCESSWARE sob licença da Ashton-Tate-U.S.A.

Produto em disco com seu respectivo nº de série, manual completo e garantia.

Conta também com suporte técnico e direito a atualização de nova versão gratuitos.

Software mundialmente aprovado.
PRÁCTICA E PRINCESSWARE

ATENÇÃO

Este produto você encontrará nos Revendedores Autorizados; exija sempre o original. A Lei de Software, traz benefícios mútuos. Vai proteger seu software.

PRINCESSWARE

O futuro desta Geração Já chegou.

Para maiores informações:
Av. Açoce, 579 - Indianópolis
CEP 04075 - São Paulo - SP
Tel.: (011) 549-0545
Caixa Postal 64635 - CEP 05497
- São Paulo - SP

AS INSTRUÇÕES SECRETAS DO Z-80

Linguagem de Máquina - MSX
Editora Aleph

Se você der uma olhada cuidadosa numa tabela de mnemônicos, perceberá que existem alguns "buracos" nas colunas "após CB" e "após ED". Talvez você já tenha se perguntado o que acontece se tentar utilizar essas "instruções". De fato, algo acontece e, por incrível que pareça, até com certa lógica! Temos, então, mais instruções disponíveis para utilizar o Z-80 que, no entanto, não são divulgadas em seus manuais e, portanto, não fazem parte das tabelas dos programas Assembler e Disassembler. Vamos, então, estudar essas "instruções secretas". O motivo pelo qual estas instruções não são divulgadas não é bem certo e preferimos não emitir opiniões tentando adivinhar o porquê. O fato é que elas existem.

A ESTRUTURA DAS INSTRUÇÕES

Se todas as combinações possíveis de um byte fossem utilizadas para indicar instruções, teríamos apenas 256 instruções disponíveis para o Z-80. Assim, quatro bytes foram reservados para permitir o uso de mais instruções:

CBH e EDH, que colocamos em "frente" a um dos 252 bytes restantes, produzindo outras instruções.

DDH e FDH que colocados em "frente" a quase todas as instruções que utilizam o par HL, permitem utilizar os pares IX e IY.

Desse modo, como oficialmente temos 248 instruções após CBH, 58 instruções após EDH e mais 140 instruções possibilidades por DDH e FDH, temos:

$$252 + 248 + 58 = 698 \text{ instruções oficiais.}$$

Na realidade, são "apenas" 696, pois os códigos 22H e ED 63H correspondem ambos a LD (KK), HL e os códigos 2AH e ED6BH correspondem, ambos, a LD HL, (KK).

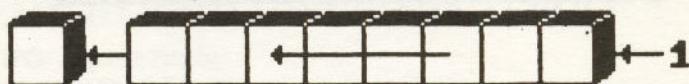
Vamos, agora, tentar preencher alguns malabarismos com os bytes para tentar "cavar" novas instruções. Num esforço de padronização, utilizaremos os mesmos mnemônicos apresentados na maioria da literatura a respeito dessas novas instruções.

AS INSTRUÇÕES NÃO OFICIAIS

Comecemos, então, a preencher os oito bytes restantes para as instruções após CBH, ou seja, de 30H a 37H. Teremos, então, uma nova instrução de rotação similar à SLA, que, em vez de resetar o bit 0, este é setado. Esta instrução é chamada de SLI (Shift Left Inverted) e corresponde a multiplicar o valor do registro, ou memória, por 2 e somar 1 ao resultado. OCARRY pode ser usado para detectar "estouros".

FIGURA 1 - A instrução SLI.

CARRY



Utilizando os bytes DDH e FDH, temos, então, dez novas instruções:

FIGURA 2 - Dez "novas" instruções de rotação (-128 <= Q >= 127).

INSTRUÇÃO	CÓDIGO
SLI A	CB37H
SLI B	CB30H
SLI C	CB31H
SLI D	CB32H
SLI E	CB33H
SLI H	CB34H
SLI L	CB35H
SLI (HL)	CB36H
SLI (IX+Q)	DDCBQ36H
SLI (IY+Q)	FDCBQ36H



- * DRIVE 5.1/4 SLIN COMPLETO
- * PLACA 80 COLUNAS
- * MODEM DE COMUNICAÇÃO

- * EXPANSOR DE SLOT (C/4 SLOTS)
- * GABINETE P/DRIVE COM FONTE FRIA
- * INTERFACE DUPLA P/DRIVE

* PACOTÃO EM DISCO: 100 JOGOS (ESCOLHER) + 5 APLICATIVOS + 10 DISCOS = 40.000,00
* PACOTÃO EM FITA: 100 JOGOS (ESCOLHER) + 5 APLICATIVOS + 7 FITAS = 40.000,00

SOLICITE NOSSO CATALOGO DE PROGRAMAS PARA FAZER A SUA ESCOLHA. ATENDEMOS TODOS OS ESTADOS EM 24 HORAS VIA SEDEX.
PARA FAZER SEU PEDIDO ENVIE CHEQUE NOMINAL COM CARTA DETALHADA PARA **A. NASSER**.

RIO DE JANEIRO: RUA GONZAGA BASTOS 411/203 — VILA ISABEL — RJ CEP 20541 — TEL (021) 234 0775

FILIAL CURITIBA: AV. 7 DE SETEMBRO, 3146 LOJA 20 — SHOPPING SETE — CURITIBA — PR — CEP 80010 — TEL 233 0046

Até agora, mencionamos que os bytes FDH e DDH poderiam ser colocados em frente de quase todas as instruções que envolvam o par HL para utilizar os registros IX e IY. De fato, as únicas "exceções" são as instruções EX DE,HL, EXX e as instruções que utilizam o par HL mas são precedidas pelo byte EDH. O que acontece se tentarmos utilizar o par HL? De fato, se colocarmos os bytes FDH e DDH em frente a qualquer instrução precedida por CBH, com exceção das instruções que utilizam o par HL e das instruções BIT, teremos 336 novas instruções, que trabalham de modo bastante peculiar: elas executam a instrução no endereço indicado por IX e IY mais Q (número entre -128 e 127) e, após isto, copiam o resultado no registro indicado pela instrução! Note que isto equivale a duas instruções oficiais. Por exemplo:

SET 7,A = CBFH DDCBQFFFH = SET 7,(IX+Q)
LD A,(IX+Q)

Fica difícil, entretanto, definir um mnemônico para este tipo de instrução. Poderíamos, por exemplo, utilizar o seguinte:

SET 7,A/(IX+Q)

Aqui estão, portanto, essas 336 novas instruções:

FIGURA 3 -
112 "novas" instruções usando SET (-128<=Q<=127).

reg/m bit	A/(IX+Q)	B/(IX+Q)	C/(IX+Q)	D/(IX+Q)	E/(IX+Q)	H/(IX+Q)	L/(IX+Q)
	A/(IY+Q)	B/(IY+Q)	C/(IY+Q)	D/(IY+Q)	E/(IY+Q)	H/(IY+Q)	L/(IY+Q)
0	DDCBQC7H	DDCBQC0H	DDCBQC1H	DDCBQC2H	DDCBQC3H	DDCBQC4H	DDCBQC5H
	FDCBQC7H	FDCBQC0H	FDCBQC1H	FDCBQC2H	FDCBQC3H	FDCBQC4H	FDCBQC5H
1	DDCBQCFH	DDCBQC8H	DDCBQC9H	DDCBQCAH	DDCBQCBH	DDCBQCCH	DDCBQCDH
	FDCBQCFH	FDCBQC8H	FDCBQC9H	FDCBQCAH	FDCBQCBH	FDCBQCCH	FDCBQCDH
2	DDCBQD7H	DDCBQD0H	DDCBQD1H	DDCBQD2H	DDCBQD3H	DDCBQD4H	DDCBQD5H
	FDCBQD7H	FDCBQD0H	FDCBQD1H	FDCBQD2H	FDCBQD3H	FDCBQD4H	FDCBQD5H
3	DDCBQDFH	DDCBQD8H	DDCBQD9H	DDCBQDAH	DDCBQDBH	DDCBQDCH	DDCBQDDH
	FDCBQDFH	FDCBQD8H	FDCBQD9H	FDCBQDAH	FDCBQDBH	FDCBQDCH	FDCBQDDH
4	DDCBQE7H	DDCBQE0H	DDCBQE1H	DDCBQE2H	DDCBQE3H	DDCBQE4H	DDCBQE5H
	FDCBQE7H	FDCBQE0H	FDCBQE1H	FDCBQE2H	FDCBQE3H	FDCBQE4H	FDCBQE5H
5	DDCBQEFH	DDCBQE8H	DDCBQE9H	DDCBQEAH	DDCBQEBH	DDCBQECH	DDCBQEDH
	FDCBQEFH	FDCBQE8H	FDCBQE9H	FDCBQEAH	FDCBQEBH	FDCBQECH	FDCBQEHD
6	DDCBQF7H	DDCBQF0H	DDCBQF1H	DDCBQF2H	DDCBQF3H	DDCBQF4H	DDCBQF5H
	FDCBQF7H	FDCBQF0H	FDCBQF1H	FDCBQF2H	FDCBQF3H	FDCBQF4H	FDCBQF5H
7	DDCBQFFFH	DDCBQF8H	DDCBQF9H	DDCBQFAH	DDCBQFBH	DDCBQFCH	DDCBQFDH
	FDCBQFFFH	FDCBQF8H	FDCBQF9H	FDCBQFAH	FDCBQFBH	FDCBQFCH	FDCBQFDH

OUTRO PROFISSIONAL ACIMA DE QUALQUER SUSPEITA

SuperCalc II MSX. Com ele você calcula as vantagens.

O SuperCalc 2 MSX é uma planilha de cálculo eletrônica, um programa de planejamento eletrônico.

Com o passar do tempo, a sua utilização se tornou cada vez mais frequente. Milhares de usuários no mundo todo acharam esta a melhor maneira de aproveitar toda a capacidade e eficiência de seus micros.

E um instrumento para planejamento e previsão financeira e numérica.

O Super Calc 2 MSX pode ser usado para desenvolver o orçamento inteiro de uma companhia, para organizar o orçamento doméstico de uma família ou para coletar dados numéricos/estatísticos.

Sem dúvida a ferramenta ideal para administradores, engenheiros, vendedores, pesquisadores, estudantes, etc.

Fácil de usar não requer grandes conhecimentos de computação; foi feito para ser usado logo no seu primeiro contato.

Nada mais de lápis, papel e calculadora, agora somente seu MSX e o SuperCalc 2 MSX. Produzido pela Compucenter-Princessware sob licença da Computer Associates- U.S.A.

Produto em disco com seu respectivo nº de série, manual completo e garantia. Conta também com suporte técnico e direito a atualização de nova versão gratuitos.

Software mundialmente aprovado.
PRÁCTICA E PRINCESSWARE

ATENÇÃO

Este produto você encontrará nos Revendedores Autorizados; exija sempre o original. A Lei de Software, traz benefícios mútuos. Vai proteger seu software.

PRINCESSWARE

O futuro desta Geração já chegou.

Para maiores informações:
Av. Açoce, 579 – Indianópolis
CEP 04075 – São Paulo - SP
Tel.: (011) 549-0545
Caixa Postal 64635 – CEP 05497
– São Paulo - SP

reg/m	A/(IX+Q)	B/(IX+Q)	C/(IX+Q)	D/(IX+Q)	E/(IX+Q)	H/(IX+Q)	L/(IX+Q)
bit	A/(IY+Q)	B/(IY+Q)	C/(IY+Q)	D/(IY+Q)	E/(IY+Q)	H/(IY+Q)	L/(IY+Q)
0	DDCBQ87H	DDCBQ80H	DDCBQ81H	DDCBQ82H	DDCBQ83H	DDCBQ84H	DDCBQ85H
	FDCBQ87H	FDCBQ80H	FDCBQ81H	FDCBQ82H	FDCBQ83H	FDCBQ84H	FDCBQ85H
1	DDCBQ8FH	DDCBQ88H	DDCBQ89H	DDCBQ8AH	DDCBQ8BH	DDCBQ8CH	DDCBQ8DH
	FDCBQ8FH	FDCBQ88H	FDCBQ89H	FDCBQ8AH	FDCBQ8BH	FDCBQ8CH	FDCBQ8DH
2	DDCBQ97H	DDCBQ90H	DDCBQ91H	DDCBQ92H	DDCBQ93H	DDCBQ94H	DDCBQ95H
	FDCBQ97H	FDCBQ90H	FDCBQ91H	FDCBQ92H	FDCBQ93H	FDCBQ94H	FDCBQ95H
3	DDCBQ9FH	DDCBQ98H	DDCBQ99H	DDCBQ9AH	DDCBQ9BH	DDCBQ9CH	DDCBQ9DH
	FDCBQ9FH	FDCBQ98H	FDCBQ99H	FDCBQ9AH	FDCBQ9BH	FDCBQ9CH	FDCBQ9DH
4	DDCBQA7H	DDCBQA0H	DDCBQA1H	DDCBQA2H	DDCBQA3H	DDCBQA4H	DDCBQA5H
	FDCBQA7H	FDCBQA0H	FDCBQA1H	FDCBQA2H	FDCBQA3H	FDCBQA4H	FDCBQA5H
5	DDCBQAFH	DDCBQA8H	DDCBQA9H	DDCBQAAH	DDCBQA8H	DDCBQAACH	DDCBQAADH
	FDCBQAFH	FDCBQA8H	FDCBQA9H	FDCBQAAH	FDCBQA8H	FDCBQAACH	FDCBQAADH
6	DDCBQB7H	DDCBQB0H	DDCBQB1H	DDCBQB2H	DDCBQB3H	DDCBQB4H	DDCBQB5H
	FDCBQB7H	FDCBQB0H	FDCBQB1H	FDCBQB2H	FDCBQB3H	FDCBQB4H	FDCBQB5H
7	DDCBQBFH	DDCBQB8H	DDCBQB9H	DDCBQB8H	DDCBQB8H	DDCBQBCH	DDCBQB8H
	FDCBQBFH	FDCBQB8H	FDCBQB9H	FDCBQB8H	FDCBQB8H	FDCBQBCH	FDCBQB8H

Figura 4 - 112 “novas” instruções usando RES (-128<=Q<=127).

reg/m	A/(IX+Q)	B/(IX+Q)	C/(IX+Q)	D/(IX+Q)	E/(IX+Q)	H/(IX+Q)	L/(IX+Q)
inst.	A/(IY+Q)	B/(IY+Q)	C/(IY+Q)	D/(IY+Q)	E/(IY+Q)	H/(IY+Q)	L/(IY+Q)
RLC	DDCBQ07H	DDCBQ00H	DDCBQ01H	DDCBQ02H	DDCBQ03H	DDCBQ04H	DDCBQ05H
	FDCBQ07H	FDCBQ00H	FDCBQ01H	FDCBQ02H	FDCBQ03H	FDCBQ04H	FDCBQ05H
RRC	DDCBQ0FH	DDCBQ08H	DDCBQ09H	DDCBQ0AH	DDCBQ0BH	DDCBQ0CH	DDCBQ0DH
	FDCBQ0FH	FDCBQ08H	FDCBQ09H	FDCBQ0AH	FDCBQ0BH	FDCBQ0CH	FDCBQ0DH
RL	DDCBQ17H	DDCBQ10H	DDCBQ11H	DDCBQ12H	DDCBQ13H	DDCBQ14H	DDCBQ15H
	FDCBQ17H	FDCBQ10H	FDCBQ11H	FDCBQ12H	FDCBQ13H	FDCBQ14H	FDCBQ15H
RR	DDCBQ1FH	DDCBQ18H	DDCBQ19H	DDCBQ1AH	DDCBQ1BH	DDCBQ1CH	DDCBQ1DH
	FDCBQ1FH	FDCBQ18H	FDCBQ19H	FDCBQ1AH	FDCBQ1BH	FDCBQ1CH	FDCBQ1DH
SLA	DDCBQ27H	DDCBQ20H	DDCBQ21H	DDCBQ22H	DDCBQ23H	DDCBQ24H	DDCBQ25H
	FDCBQ27H	FDCBQ20H	FDCBQ21H	FDCBQ22H	FDCBQ23H	FDCBQ24H	FDCBQ25H
SRA	DDCBQ2FH	DDCBQ28H	DDCBQ29H	DDCBQ2AH	DDCBQ2BH	DDCBQ2CH	DDCBQ2DH
	FDCBQ2FH	FDCBQ28H	FDCBQ29H	FDCBQ2AH	FDCBQ2BH	FDCBQ2CH	FDCBQ2DH
SLI	DDCBQ37H	DDCBQ30H	DDCBQ31H	DDCBQ32H	DDCBQ33H	DDCBQ34H	DDCBQ35H
	FDCBQ37H	FDCBQ30H	FDCBQ31H	FDCBQ32H	FDCBQ33H	FDCBQ34H	FDCBQ35H
SRL	DDCBQ3FH	DDCBQ38H	DDCBQ39H	DDCBQ3AH	DDCBQ3BH	DDCBQ3CH	DDCBQ3DH
	FDCBQ3FH	FDCBQ38H	FDCBQ39H	FDCBQ3AH	FDCBQ3BH	FDCBQ3CH	FDCBQ3DH

Figura 5 - 112 “novas” instruções de rotação.

Os registros IX e IY (de 16 bits) não podem ser divididos em 2 registros de 8 bits. Entretanto, se usarmos os bytes DDH e FDH em frente a qualquer instrução que utilize os registros H ou L separadamente, mas que não utilizem o par HL, executando as instruções precedidas por CBH ou EDH, teremos acesso às metades dos registros.

Se utilizarmos instruções com o registro H, estamos lidando com o byte mais significativo de IX ou IY, que chamaremos HX e HY respectivamente, e, se utilizarmos instruções com o registro L, estaremos lidando com o byte menos significativo de IX ou IY (que chamaremos LX e LY, respectivamente).

Temos, então, 92 novas instruções.

FIGURA 6 - 52 “novas” instruções.

LD	HX	LX	HY	LY
A	DD7CH	DD7DH	FD7CH	FD7DH
B	DD44H	DD45H	FD44H	FD45H
C	DD4CH	DD4DH	FD4CH	FD4DH
D	DD54H	DD55H	FD54H	FD55H
E	DD5CH	DD5DH	FD5CH	FD5DH
HX	DD64H	DD65H		
LX	DD6CH	DD6DH		
HY			FD64H	FD65H
LY			FD6CH	FD6DH

A	B	C	D	E	K
DD67H	DD60H	DD61H	DD62H	DD63H	DD26KH
DD6FH	DD68H	DD69H	DD6AH	DD6BH	DD2EKH
FD67H	FD60H	FD61H	FD62H	FD63H	FD26KH
FD6FH	FD68H	FD69H	FD6AH	FD6BH	FD2EKH

Apesar do aspecto “estranho” da figura 6, ela é simples de utilizar. Os seguintes exemplos devem esclarecer a questão:

LD C, HY = FD 4CH
 LD HX, LX = DD 65H
 LD LY, D = FD 6AH
 LD HY, 16 = FD 26 10H

FIGURA 7 - 24 “novas” instruções aritméticas e, finalmente, as instruções lógicas e de comparação.

	HX	LX	HY	LY
ADD A,	DD84H	DD85H	FD84H	FD85H
ADC A,	DD8CH	DD8DH	FD8CH	FD8DH
INC	DD24H	DD2CH	FD24H	FD2CH
SUB A,	DD94H	DD95H	FD94H	FD95H
SBC A,	DD9CH	DD9DH	FD9CH	FD9DH
DEC	DD25H	DD2DH	FD25H	FD2DH

FIGURA 8 - 12 “novas” instruções lógicas e 4 de comparação.

	HX	LX	HY	LY
AND	DDA4H	DDA5H	FDA4H	FDA5H
OR	DDB4H	DDB5H	FDB4H	FDB5H
XOR	DDA6H	DDA7H	FDA6H	FDA7H
CP	DDB6H	DDB7H	FDB6H	FDB7H

Você pode perceber que ainda existem alguns “buracos” nas instruções após EDH, e muitas outras combinações possíveis. No entanto, até agora nada foi publicado a respeito dessas possíveis novas instruções, possivelmente porque elas produzem algum resultado “ilógico” ou resultado nenhum. Se você dispuiser de tempo livre, divirta-se tentando descobrir o que estas instruções que faltam podem fazer. Entretanto, com estas 438 novas instruções, temos, agora, 1136 instruções! Já é o suficiente para se divertir.

Você poderia perguntar sobre o efeito dessas novas instruções nas flags. O efeito é equivalente às instruções “oficiais” e pode ser descoberto por analogia.

Novamente, salientamos que essas instruções não podem ser utilizadas com os Assemblers e Disassemblers. Entretanto, as pseudo-instruções (ou NOPs posteriormente preenchidos) podem facilitar as coisas.

DB 0DDH
 DEC L;equivalente à DEC LX
 DB 0FDH

SBC A,H ;equivalente à SBC A, HY

Alguns Assemblers utilizam duas pseudo instruções, usadas para definir dados de um byte (Byte), ou dois bytes (Word), respectivamente: DB e DW.

Com relação aos registros HX, LX, HY e LY, é conveniente salientar que sua utilização depende do seu programa. Se você precisar de muitos registros e puder deixar de lado as facilidades de endereçamento de memória através dos pares IX e IY, então pode utilizá-los sem problemas.

GERENCIANDO ARQUIVOS EM DISCO

BRUNO MARRUT

Aprenda os comandos gerenciadores de arquivos no MSX

É comum ouvirmos os usuários de microcomputadores, principalmente os iniciantes, dizerem que gostariam de guardar receitas de bold, por exemplo, em seu computador.

Os usuários de fita logo percebem que trabalhar com arquivos de dados em fita é inviável, pois a gravação/leitura de dados de uma fita cassete é uma tarefa que pode consumir alguns bons minutos do seu tempo.

Já os usuários de disco contam com uma enorme vantagem que é a rapidez com que os dados podem ser gravados, ou lidos, do disco, tornando, assim, o armazenamento de uma receita uma tarefa viável e, dependendo do programa, até com vantagens sobre o ultrapassado fichário.

Neste artigo irei comentar as funções de gravação e leitura em disco, apresentando como exemplo um programa de mala-direta.

Antes de iniciarmos, gostaria de apresentar as diferenças entre os dois tipos de arquivos que podem ser gerenciados no disco: arquivos sequenciais e arquivos aleatórios.

Arquivos sequenciais: se você estiver utilizando uma fita cassete, os seus arquivos estarão gravados sequencialmente, ou seja, um após o outro, sendo que você não poderá acessar o segundo sem passar pelo primeiro, mesmo que avance rapidamente a fita.

Suponhamos que o seu arquivo seja sequencial e que nele estejam gravadas 235 fichas. Caso você queira ler a ficha 234, terá que ler todas as 233 anteriores, mesmo sabendo que a ficha procurada é a 234.

Arquivos aleatórios: os arquivos aleatórios apresentam a vantagem de permitirem o acesso direto a um determinado registro, independente da sua posição, permitindo, assim, uma visualização de uma informação muito mais rapidamente.

Nos arquivos sequenciais a extensão de um registro pode ser variável, isto é, se você tiver programado a sua ficha para conter até 100 caracteres, mas, na digitação, tiver ocupado apenas 80 das 100 posições disponíveis, o espaço ocupado no disco será o correspondente às 80 posições utilizadas e não às 100 previamente previstas, resultando, portanto, em economia de espaço no disco.

Em se tratando de arquivo aleatório, a extensão é fixa, ou seja, o espaço ocupado por cada uma será igual, independentemente da quantidade de caracteres que tenhamos digitado.

Caso existam posições que não tenham sido utilizadas, as mesmas serão preenchidas com o código correspondente ao espaço em branco.

Em ambos os casos, a extensão máxima que podemos ter para cada registro é de 256 bytes ou caracteres.

Nos arquivos sequenciais a alteração de uma ficha já gravada, ou de um conjunto delas, é uma tarefa complicada e nada prática. Para alterar uma ficha de um arquivo sequencial, devemos transferir todos os registros para variáveis indexadas e, depois de efetuarmos as devidas correções, efetuar a regravação de todo o arquivo. Este processo está intimamente ligado ao fato do arquivo sequencial ter sua extensão variável para cada registro e limita a capacidade do arquivo à memória disponível do computador.

Um arquivo sequencial deve ser aberto e fechado cada vez que for acessado, o mesmo não ocorrendo com o aleatório, que deve ser aberto no início do programa e fechado no final do processamento.

Portanto, nosso arquivo de mala-direta será um arquivo aleatório. Os comandos que utilizaremos em nosso programa, e que pertencem ao Disk Basic, estão comentados abaixo:

OPEN

SINTAXE: OPEN "d:arquivo" AS # NRO LEN=NRO

FUNÇÃO: abrir um arquivo para a entrada e ou saída de dados.

Ao executarmos o comando OPEN, reservamos uma área da memória do computador, chamada de "BUFFER". É lá que os dados são armazenados, temporariamente, antes de serem gravados no disco, ou quando são lidos do mesmo.

NRO é utilizado para especificar o número do arquivo em processamento. É definido pela instrução MAXFILES.

LEN=NRO define a quantidade de bytes que cada registro irá ter. Caso nada seja especificado, será assumido pelo sistema o valor máximo, que é de 256 bytes.

Podemos abrir, simultaneamente, até 6 arquivos para a entrada de dados. Contudo, para a saída, apenas um poderá estar aberto.

CLOSE

SINTAXE: CLOSE # NRO

FUNÇÃO: Fechar um arquivo aberto anteriormente.

NRO - Especifica o arquivo a ser fechado. Caso nada seja especificado, todos os arquivos que se encontram abertos serão fechados.

O não fechamento de um arquivo após o mesmo ter sido aberto, tornará impossível a utilização das informações armazenadas. O comando CLOSE transfere todos os dados armazenados no BUFFER para o disco, o que só ocorre normalmente quando o BUFFER é totalmente preenchido.

LOF

SINTAXE: LOF NRO

FUNÇÃO: Utilizada para indicar a extensão de um arquivo.

FIELD

SINTAXE: FIELD #NRO, A AS X1\$,B AS X2\$

FUNÇÃO: organiza o espaço dentro de um registro, através de variáveis alfanuméricas, ou seja, informa como será dividido o registro.

As variáveis definidas através da instrução FIELD não podem receber os dados diretamente, através de LINE INPUT, por exemplo, pois elas são alocadas no BUFFER. Então, os dados entrados devem ser atribuídos inicialmente a outras variáveis que não sejam as definidas através do FIELD, através de LINE INPUT, por exemplo, e depois transferidas para as variáveis definidas por FIELD, através dos comandos LSET e RSET. Feito isto, os dados poderão, então, ser transferidos para o disco, através do comando PUT.

Para a leitura de dados do disco também utilizamos a instrução FIELD, sendo que, ao invés de utilizarmos PUT, iremos utilizar GET, que permite que seja lido um registro por vez, transferindo o seu conteúdo para o BUFFER, estando os dados disponíveis para manipulação.

LSET

SINTAXE: LSET variável 1 = variável 2

FUNÇÃO: Transferir os dados da memória principal para o BUFFER variável 1 = definida em FIELD
variável 2 = variável de trabalho

PUT

SINTAXE: PUT N,N'

FUNÇÃO: grava no disco as informações do registro.

N = número do arquivo aberto pela instrução OPEN

n' = número do registro que será gravado

GET

SINTAXE: GET N,N'

FUNÇÃO: lê os registros do disco, colocando-os no BUFFER.

n = número do arquivo aberto pela instrução OPEN

n' = número do registro que será lido. Caso não seja utilizado, será lido o registro após o último lido pelo comando GET, ou o primeiro.

MAXFILES

SINTAXE: MAXFILES = N

FUNÇÃO: Definir a quantidade de arquivos que podem ser abertos pela instrução OPEN

MKD\$ - MKS\$ E MKI\$

MKD\$ (nro. de dupla precisão)

MKS\$ (nro. de simples precisão)

MKI\$ (nro. inteiro)

FUNÇÃO: Converter números ou variáveis numéricas para variáveis alfanuméricas.

Em um arquivo de acesso direto só podemos trabalhar com STRINGS. Portanto, variáveis numéricas (dupla precisão, simples precisão ou inteiras) devem ser convertidas em STRINGS.

Quando formos definir o espaço ocupado pela variável em um registro, através da instrução FIELD, para dados numéricos, devemos usar a quantidade de "bytes" referentes aos números já convertidos, observando que:

MKD\$ - converterá o dado numérico de dupla precisão em uma variável alfanumérica de 8 "bytes".

MKS\$ - converterá o dado numérico de simples precisão em uma variável alfanumérica de 4 "bytes".

MKI\$ - converterá o dado numérico inteiro em uma variável alfanumérica de 2 "bytes".

A utilização de dados numéricos merece uma atenção especial na hora de definirmos a instrução FIELD, pois os campos não poderão ser menores e, caso sejam maiores, haverá desperdício de espaço.

Descrevemos acima os comandos do DISK BASIC que iremos utilizar para o nosso programa de Mala Direta.

Acreditamos que, analisando o programa e os conceitos transmitidos acima, você estará capacitado a desenvolver qualquer tipo de rotina para gravação e leitura de dados em disco, que poderá ser utilizada em seus próprios programas.

CHAMPION SOFTWARE

MSX 2
MSX

MSX : AMAUROTE, BANANAS, EL MUNDO PERDIDO,
CAR FIGHTER, OCEAN CONQUEROR, CAR JAMBOREE
MSX II: THUNDER, CHOPPER, CHESS, REDLIGHT, OF AMS
TERDAN, PIXEL 2, PHILLIPS DESIGNER'S
MEGARON MSX: PINGUIN ADVENTURE, F1 SPIRITS,
SALAMANDER, DRAGON QUEST, NEMESIS 2, CALL FORCE
MEGARON MSX 2: ZANAC, HINOTORI, KING KONG 2, 1942,
LABYRINT, METAL GEAR, VAMPIRE KILLER, LUPIN 3RD,
DRASLEY FAMILY, SUPER RAMBO



E MUITO MAIS!!!

Solicite nosso catálogo grátis pelo tel.(011) 212-8990 das 14 às 22 horas

CAIXA POSTAL 54243 - SÃO PAULO - SP-CEP 01296

MALA DIRETA

```

10 REM MALA DIRETA
20 REM REVISTA CPU NRO 5
30 REM BRUNO MARRUT
40 REM LIMPA TELA E DEFINE AREA DAS VAR
IAVEIS
50 CLS:SCREEN 0:WIDTH 40:KEYOFF
60 CLEAR 3000:MAXFILES=2
70 REM ABRE E DEFINE A ESTRUTURA DO ARG
UIVO
80 OPEN "A:ARQUIVO" AS #1 LEN=117
90 C=LOF(1)/117+1
100 FIELD #1,40 AS N$,40 AS EN$,15 AS
BA$,15 AS CI$,2 AS UF$,5 AS CE$
110 REM EXIBE MENU NA TELA
120 CLS:PRINT"REVISTA CPU":PRINT"ARQUIV
O DE MALA DIRETA"
130 PRINT:PRINT "1 - ENTRADA DE DADOS
140 PRINT "2 - BUSCA POR NOME
150 PRINT "3 - BUSCA POR REGISTRO
160 PRINT "4 - LISTA ARQUIVO NO VIDEO
170 PRINT "5 - FIM
190 Z$=INPUT$(1)
200 ON VAL (Z$) GOTO 220,500,640,1200,1
300
210 GOTO 190
220 REM ENTRADA DE DADOS
230 CLS:PRINT "REGISTRO ";C
240 PRINT:PRINT:LINE INPUT "NOME "; N$
250 LINE INPUT "ENDERECO "; E$
260 LINE INPUT "BAIRRO "; B$
270 LINE INPUT "CIDADE "; C$
280 LINE INPUT "ESTADO "; S$
290 LINE INPUT "CEP "; P$
300 CLS:PRINT "REGISTRO ";C
310 PRINT N$:PRINT E$:PRINT B$:PRINT C$:
:PRINT S$:PRINT P$
320 PRINT:PRINT"DESEJA EFETUAR ALGUMA A
LTERACAO? (S/N)"
330 Z$=INPUT$(1):IF Z$="S" OR Z$="s" TH
EN GOSUB 3400 ELSE IF Z$="N" OR Z$="n"
THEN 420
420 REM GRAVACAO DOS DADOS NO DISCO
430 LSET N$=N$:LSET EN$=E$:LSET BA$=B$:
:LSET CI$=C$: LSET ES$=S$:LSET CE$=P$:
435 PUT #1,C=C+1
440 CLS:PRINT "REGISTRO GRAVADO":PRINT:
PRINT"DESEJA EFETUAR NOVA ENTRADA? (S/N
)"
450 Z$=INPUT$(1):IF Z$="S" OR Z$="s" TH
EN 220 ELSE IF Z$="N" OR Z$="n" THEN 12
0
500 REM BUSCA POR NOME
510 CLS
520 IF C<2 THEN PRINT "ARQUIVO VAZIO":G
OTO 120
530 LINE INPUT "NOME A PROCURAR ";N$
540 FOR K=1 TO C-1
550 GET #1,K
560 IF N$=LEFT$(N$,LEN(N$)) THEN GOTO
585
570 NEXT K
580 PRINT "NOME NAO ENCONTRADO"
585 GOSUB 1000
590 PRINT:PRINT:LINE INPUT "PRESSIONE Q
UALQUER TECLA PARA VOLTAR"; Z$
600 GOTO 120
610 PRINT:PRINT "DESEJA (A)LITERAL (C)AN
CELAR OU (V)OLTAR AO MENU PRINCIPAL?"
620 Z$=INPUT$(1):IF Z$="A" OR Z$="a" TH
EN 800 ELSE IF Z$="V" OR Z$="v" THEN 12
0 ELSE IF Z$="C" OR Z$="c" THEN 750
630 GOTO 620
640 REM BUSCA POR REGISTRO
650 CLS:PRINT "QUAL O NUMERO DO REGISTR
O?"
660 K=0:INPUT K:IF K>C THEN PRINT "REG
ISTRO NAO EXISTE. ULTIMO REGISTRO EFETU
ADO FOI O DE NUMERO: "; C-1:PRINT:PRINT
"PRESSIONE UMA TECLA PARA VOLTAR"
670 GET #1,K:GOSUB 1000
680 GOTO 610
750 REM APAGAR REGISTRO
760 LSET N$="":LSET EN$="":LSET BA$="":
:LSET CI$="":LSET ES$="":LSET CE$=""
770 PUT #1,K
780 PRINT "REGISTRO DELETADO":PRINT"PRE
SSIONE QUALQUER TECLA PARA RETORNAR
785 Q$=INPUT$(1):IF Q$<>"" THEN 120
800 REM ALTERACAO DE REGISTRO
805 CLS
810 PRINT N$:PRINT EN$:PRINT BA$:PRINT C$:
:PRINT ES$:PRINT CE$:
820 GOSUB 5000
840 PRINT "PRESSIONE QUALQUER TECLA PAR
A VOLTAR AO MENU PRINCIPAL"
850 INPUT Z$:GOTO 120
1000 REM DISPLAY DE INFORMACOES NO VIDE
O
1005 PRINT:PRINT"REGISTRO ";K:PRINT
1010 PRINT N$:
1020 PRINT EN$:
1030 PRINT BA$;SPC(1);CI$;SPC(1);ES$;SP
C(1);CE$:
1070 RETURN
1200 REM RELATORIO NO V,DEO
1205 CLS:PRINT"PARA PARAR O ROLAMENTO D
A LISTAGEM PRESSIONE STOP"
1210 IF C<2 THEN PRINT "ARQUIVO VAZIO -
PRESSIONE QUALQUER TECLA PARA CONTINUA
R":INPUT Z$:IF Z$ <>"" THEN GOTO 120
1220 FOR K=1 TO C-1
1230 GET #1,K:GOSUB 1000
1240 NEXT K
1250 PRINT "FIM DA LISTAGEM. PRESSIONE
UMA TECLA PARA RETORNAR":INPUT Z$:GOTO
120
1300 CLOSE
1310 END
3400 PRINT:LINE INPUT "QUAL O CAMPO QUE
DESEJA ALTERAR?";Z$
3500 IF Z$="NOME" THEN LINE INPUT "NOME
"; N$
3600 IF Z$="ENDERECO" THEN LINE INPUT "
ENDERECO "; E$
3700 IF Z$="BAIRRO " THEN LINE INPUT "B
AIRRO "; B$
3800 IF Z$="CIDADE " THEN LINE INPUT "C
IDADE "; C$
3900 IF Z$="ESTADO " THEN LINE INPUT "E
STADO "; S$
4000 IF Z$="CEP " THEN LINE INPUT "CEP
"; P$
4020 RETURN
5000 PRINT:LINE INPUT "QUAL O CAMPO QUE
DESEJA ALTERAR?";Z$
5100 IF Z$="NOME" THEN LINE INPUT "NOME
"; N$
5200 IF Z$="ENDERECO" THEN LINE INPUT "
ENDERECO "; EN$:
5300 IF Z$="BAIRRO " THEN LINE INPUT "B
AIRRO "; BA$:
5400 IF Z$="CIDADE " THEN LINE INPUT "C
IDADE "; CI$:
5500 IF Z$="ESTADO " THEN LINE INPUT "E
STADO "; ES$:
5600 IF Z$="CEP " THEN LINE INPUT "CEP
"; CE$:
5720 RETURN

```

O MELHOR TAMBÉM É O MAIOR

TUDO PARA
MSX

ALÉM DE QUALIDADE • GARANTIA • SUPORTE

- mais de 20.000 clientes -
- o maior estoque do mercado -
- mais de 1.000 programas -
- a mais completa linha de periféricos -
- mais de 1.000 revendedores -

HARDWARE SOFTWARE PERIFÉRICOS ACESSÓRIOS CURSOS
ASSISTÊNCIA TÉCNICA PARA MICROS, MONITORES E DRIVES
INTERFACES DRIVES 80 COLUNAS MODEM IMPRESSORAS, ETC
REDE DE COMUNICAÇÃO PARA LIGAR SEU MSX A MICROS 16 BITS
CURSOS EM VIDEOCASSETE E MUITO MAIS...

Rua Apiacás,92 - São Paulo - CEP 05017 Fone 872.0730

ATENÇÃO

Preencha e remeta este
formulário o quanto antes

Ele garante as informações em primeira mão, que
você vai receber em casa, sobre todas as atualizações
e modificações do produto que você adquiriu, bem co-
mo dos novos lançamentos e de tudo que estiver rela-
cionado com o seu MSX.

Nome _____
Endereço _____
CEP _____ Cidade _____ Estado _____
Idade _____ Nacionalidade _____ Sexo _____
Equipamento _____ Periféricos _____



NOVO ENDEREÇO



Rua Apiacás,92

Fone 872.0730

São Paulo



O MAIOR SHOW ROOM DO PAÍS !!!

INTELIGÊNCIA ARTIFICIAL EM PASCAL

ANTONIO F.S.SHALDERS
VICTOR ELIASZ WELMAN

Programas que simulam o raciocínio humano já são uma realidade, deixando os cenários de filmes de ficção científica, como por exemplo os computadores HAL e SAL, do filme "2010". Tais computadores eram capazes de incríveis façanhas, tal como reconhecimento e síntese de voz, e um incrível poder de dedução e raciocínio.

É lógico que o que propomos neste artigo não chega nem aos pés disso, mas não deixa de ser muito interessante, principalmente se for levado em conta que não foi usada nenhuma linguagem específica para programação em IA, como o LISP e o PROLOG, sendo realizado totalmente em Pascal.

Um programa em IA simula de algum modo o pensamento humano (ou pelo menos tenta), fazendo que o próprio programa tome uma decisão ou então faça uma dedução sobre algum fato.

Você que sempre escuta de seus pais e colegas: "O que é que você tanto faz na frente daquela máquina burra ?!" terá chance de se defender, mostrando-lhes que o seu computador não é uma máquina assim tão burra, e que, se corretamente programado, pode até "aprender". Caso o programa cometa algum erro, este aprende a não fazê-lo novamente.

Um caso típico de programas que utilizam técnicas de IA são os jogos de tabuleiro como o xadrez e o gamão. As primeiras partidas são facilmente vencidas por você, mas, depois de um certo tempo, fica praticamente impossível vencer a máquina, pois o jogo "aprende" a não perder, com seus próprios erros.

O que podemos fazer em Pascal é simularmos matematicamente alguns procedimentos que facilitam a programação em IA.

O algoritmo utilizado é o da árvore (sem podas), por ser o de mais fácil implementação, mas nem por isso o programa deixa de ser interessante, pois o mesmo superou todas as expectativas em relação ao desempenho.

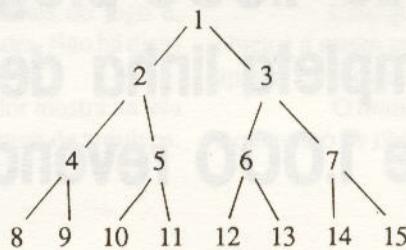
O método da árvore consiste em seguirmos um determinado caminho (galho) até chegarmos a uma resposta coerente (fruto).

O programa mostrado é conhecido tecnicamente por reconhecedor. No nosso caso, como o próprio nome indica, é capaz de reconhecer uma pessoa baseado em suas características ou em fatos a ela referentes.

A situação correspondente na vida humana é a seguinte: um amigo seu lhe pergunta se você conhece uma determinada pessoa. A priori, você não se lembra e vai, então, fazendo uma série de perguntas que o ajudarão a reconhecê-la. O reconhecimento, neste caso, dá-se quando uma imagem mental da pessoa em questão é formada no seu consciente.

Caso você tenha pensado em outra pessoa diferente, irá procurar alguma coisa que diferencie a pessoa em questão da que você pensou, de modo a não cometer o mesmo erro novamente, certo?

Analisemos, agora, como é constituída uma árvore. Para isso, observe a figura 1 atentamente.



O número 1 no topo da árvore é a semente (ou pergunta inicial). A partir desta pergunta, podemos distinguir até oito frutos (que estão na base da árvore). Os frutos são as terminações dos galhos.

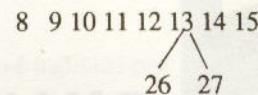
Note que há apenas um caminho possível para chegarmos a cada fruto e é isso que nos permite identificá-lo corretamente.

Suponha que nos "nós" da árvore (chamaremos de "nós" os pontos por onde passam dois ou mais galhos) hajam perguntas e cada vez que a resposta for "sim", você ande para a esquerda, e quando for "não", ande para a direita.

Por exemplo: para chegarmos ao fruto 13, devemos responder não à pergunta do nó 1, sim à do nó 3 e não à do nó 6. Chegaremos, então, ao fruto em questão.

Suponha, agora, que a resposta obtida em 13 esteja errada. Devemos, então, converter este fruto em um nó, com uma pergunta que nos permita distinguir a resposta certa da obtida, e criar dois frutos, um com a resposta certa e outro com a obtida.

No caso mencionado, as últimas linhas da árvore ficarão como as mostradas na figura 2.



Para construirmos uma árvore, basta escolhermos uma semente inicial (1) e ir multiplicando este valor por dois para uma resposta positiva, e multiplicando por dois e somando um para uma resposta negativa.

Concluímos, então, que um caminho é formado por números que são o dobro, ou o dobro mais um, do número anterior ao mesmo, no caminho.

Este é o procedimento matemático adotado pelo algoritmo da árvore (sem podas). São exemplos de caminhos:

- a) 1, 2, 4, 8, 17, 34, 69
- b) 1, 3, 6, 13, 26, 53, 107

O que o nosso programa faz é estruturar uma árvore deste tipo para então seguir os galhos até chegar a um fruto coerente.

No programa, a semente é a pergunta "É homem ?" e os dois frutos iniciais são "João", para uma resposta afirmativa, e "Maria" caso a resposta seja negativa.

Você deve substituir estes dois nomes por outros de sua conveniência.

Se o programa for utilizado com o Turbo Pascal em opção de compilação em memória, o array responsável pelo número de nomes não deverá ser superior a 220. Na opção de compilação em disco, não deverá ser superior a 840.

Este programa foi elaborado para reconhecer pessoas, mas nada impede que seja usado em áreas profissionais, como por exemplo na área médica.

Neste caso, se você é médico, poderá colocar em cada nó um sintoma e nos frutos um pré-diagnóstico.

É possível aplicarmos o programa em robótica.

Neste caso, suponha que você quer fazer com que um robô pegue um determinado objeto em sua casa e o leve até você. O robô teria um mapa de sua casa e iria lhe perguntando a respeito dos locais onde o objeto em questão estaria, selecionando, assim, o local exato para ir até lá e trazer-lhe o tal objeto.

As possibilidades de uso de IA são ilimitadas.

Há casos de programas para geoprospecção, estratégistas militares, diagnóstico de doenças, e muitos outros tipos.

PROGRAM INTART

```

program intart;
{otimização de arrays para
velocidade}
{$X-}

{área de definição de tipos}
type fi:string[40];
type pd=record
  k:integer;
  c:real;
  p:fi;
end;

{área de definição de variáveis}
var
  q      : array[0..250] of
pd; {array do número de pessoas}
  d,te,b   : pd;
  re,s    : string[3];
  ar,tr,aqv_arq : fi;
  co      : real;
  j,l,h,m,a : integer;
  aqv     : file of pd;

{le um caractere do teclado}
function gr:char;
var rp:char;
begin
  readln(rp);
  gr:=upcase(rp);
end;

function rx(x:integer):real;
var i :integer;
  ax:real;
begin
  ax:=1;
  for i:=1 to x do
    ax:=10*ax;
  rx:=ax;
end;

{dobrar}
function db(x:real):real;
begin
  a:=trunc(ln(x)/ln(10));
  if a<=9 then
    db:=2*x
  else
    db:=rx(a-9)*x
end;

{dobrar mais um}
function dm(x:real):real;
begin
  a:=trunc(ln(x)/ln(10));
  if a<=9 then
    dm:=2*x+1
  else
    dm:=rx(a-8)*x
end;

{achar nomes}
function achar(x:real):integer;
var ya:integer;
begin
  ya:=1;
  while q[ya].c<x do
    ya:=ya+1;
  achar:=ya;
end;

{ler dados do disco}
procedure lds;
begin
  j:=0;
  assign(aqv,aqv_arq);
  reset(aqv);
  while not eof(aqv) do
  begin
    read(aqv,d);
    q[j]:=d;
    j:=j+1;
  end;
  J:=J-1;
  q[0]:=b;
end;

{gravar no disco}
procedure gds;
begin
  assign(aqv,aqv_arq);
  rewrite(aqv);
  for h:=0 to J do
    write(aqv,q[h]);
  close(aqv);
end;

{inicialização do vetor}
procedure ir(ss:fi;rr:real);
var i,v :integer;
begin
  i:=0;
  while ((q[i].c<rr)and(i<=j)) do
    i:=i+1;
  for v:=j downto i do
    q[v+1]:=q[v];
  q[i].c:=rr;
  q[i].k:=2;
  q[i].p:=ss;
  j:=j+1;
end;

{controle principal de processos}
procedure ctr;
begin
  l:=0;
  while l<=j do
  begin
    te:=q[l];
    if te.k=1 then
    begin
      writeln(te.p,' ?');
      re:=gr;
      if re=s then co:=db(te.c)
      else co:=dm(te.c);
      l:=achar(co);
    end;
  end;
end;

{finalização do programa}
procedure final;
var v:string[11];
begin
  writeln('Salvar arquivo ?');
  v:=gr;
  if v=s then
  begin
    writeln('Nome do arquivo
?');
    readln(v);
    clrscr;
    aqv_arq:=v;
    lds;
    end;
  writeln;
end;

{apresentação}
procedure apresentação;
begin
  clrscr;
  writeln('_____
');
  writeln(' IntArt 1.00');
  writeln(' (C) 1988
by');
  writeln(' Victor E.
Welman');
  writeln('   e');
  writeln(' A. F.
Shalders');
  writeln('_____
');

{corpo do programa principal}
begin
  apresentação;
  gotoxy(1,10);
  q[0].k:=1;
  q[0].c:=1;
  q[0].p:='É HOMEM';
  b:=q[0];
  q[1].k:=2;
  q[1].c:=2;
  {*** semente da parte
masculina ***}
  q[1].p:='Joao';
  q[2].k:=2;
  q[2].c:=3;
  {*** semente da parte
feminina ***}
  q[2].p:='Maria';
  j:=2;
  s:='S';
  init;
  m:=1;
  while m>0 do
  begin
    writeln('Pense em alguém:');
    ctr;
    writeln('Mais alguém ? ');
    re:=gr;
    if re=s then m:=0;
  end;
  final;
end.

```

INTERPRETADOR DE EXPRESSÕES

ANTONIO F. SHALDERS

Muitos de vocês, leitores, já tiveram vontade de fazer um programa que pudesse gerar gráficos de funções de sua escolha.

Alguns, talvez, já até tenham começado, porém muitos de vocês esbarraram no problema de interpretar funções resolvendo (aliás, quebraram o galho) modificando-o para cada função que quisessem traçar. Outros apelaram para um menu fixo de funções com o máximo de funções que puderam imaginar.

Uma parte tentou fazer um interpretador de expressões e acabaram desistindo ou, se conseguiram, foi a muito custo.

E, finalmente, os raros que conseguiram sem grandes dificuldades.

Porém, estes já tinham um ponto de partida, uma idéia inicial.

Mas qual é a idéia inicial e qual é o ponto de partida?

Este artigo se destina a dar uma idéia inicial de como se faz um interpretador de expressões e, muito superficialmente, de como funciona um compilador, usando o método de descida recursiva, apresentando um exemplo de interpretador, cuja listagem está neste artigo.

Mas será este exclusivo a este artigo? Nada disso. Isto não implica que o nosso interpretador de comandos não possa ser usado juntamente com outros programas, pois ele é totalmente independente do programa de traçado de funções.

Sua operação é bastante simples: basta, apenas, seguir alguns pequenos cuidados.

Começarei definindo o que é "Token", "léxico", "sintática" e "semântica", com uma analogia com a língua portuguesa (que me perdoem os professores de português).

Digamos a sequência de palavras: formiga elefante a o matou.

Esta sequência não forma uma frase porque não está sintaticamente correta. Porém, lexicalmente, é correta: todas as palavras têm seus sentidos isoladamente.

Pode-se dizer que cada palavra é um Token por ser um elemento com sentido isoladamente.

Digamos, agora, uma nova sequência de palavras:
a formiga matou o elefante

Esta sequência apresenta-se sintaticamente certa, i.e., sem erros sintáticos, porém semanticamente está incorreta: uma formiga não pode matar um elefante!

Observe esta nova sequência: o elefante matou a formiga.

Esta sequência está correta tanto do ponto de vista lexical, como do sintático ou do semântico, pois todos os seus elementos existem na língua portuguesa, estão obedecendo às regras de sintaxe e possuem um sentido final.

Mas o que isto tudo tem a ver com o caso?

Uma linguagem de programação tem suas palavras reservadas, seus operandos, suas funções e suas constantes (seus elementos léxicos, que numa linguagem de programação é chamado de Token). Possui também uma sintaxe.

Um programa qualquer tem de ter um sentido, pelo menos o suficiente para gerar um código executável (ou para ser interpretado).

Agora já podemos definir a nossa "linguagem", i.e., a forma da expressão que deverá ser utilizada pelo interpretador.

Deverá haver:

- 1) Parênteses : Tokens "(" e ")" .
- 2) Multiplicação e divisão : Tokens "*" e "/" .
- 3) Soma e subtração : Tokens "+" e "-" .
- 4) Funções constantes e variáveis x e y : Tokens do tipo identificador, que o analisador léxico vai descobrir a que categoria pertence verificando em uma tabela de símbolos.
- 5) Já que algumas funções terão mais de um parâmetro, vai existir o Token "," como separador.
- 6) Os operadores unários "+" e "-" .
- 7) Constantes numéricas.

Com isto já temos os seguintes Tokens definidos : "(", ")", "*", "/", "+", "-", identificadores e constantes numéricas.

Mas como diferenciarmos os operadores "+" e "-" unários dos binários (com dois operandos) ?

Isto é simples: pelo seu contexto. Será mais fácil verificar isto vendo os diagramas sintáticos, que vão dos diagramas 1 ao 4 .

Um diagrama sintático é a representação gráfica de uma sintaxe.

Nos diagramas sintáticos, o que está em um círculo (ou em uma figura com extremidades arredondadas) é um Token. Este não pode ser mais expandido (apesar de se poder expandir mais nos casos de identificadores e constantes numéricas, porém esta expansão fica ao encargo do analisador léxico).

O que está em um retângulo pode ser expandido em um outro diagrama sintático.

O método de descida recursiva consiste em fazer com que cada diagrama seja uma rotina que será chamada para fazer a expansão do seu diagrama, chamando outras rotinas para fazer a expansão dos seus "sub-diagramas".

Para entendermos melhor como se implementa o diagrama no Pascal, usaremos como exemplo o analisador de expressões.

Para isto, será dada uma introdução de como utilizá-lo.

O ANALISADOR DE EXPRESSÕES.

O analisador de expressões está dividido em 3 arquivos que são "glbldef.p", "compila.p" e "interpr.p", que devem ser incluídos nesta ordem.

O primeiro arquivo contém as definições globais e uma rotina de inicialização que se chama "inicio".

O segundo contém a rotina de "compilação". Não entenda por compilação a geração de um código executável, mas sim uma codificação mais fácil de se interpretar e com isto faz-se a verificação sintática da expressão.

O nome da função é "compila" e recebe uma string e devolve a situação da "compilação", i.e., qual foi o erro e, se não houve, retorna "0".

Os erros são :

- 1) Estouro de constante
- 2) Token irreconhecível
- 3) Identificador desconhecido
- 4) Token inesperado
- 5) Parênteses não fechado
- 6) Parênteses não aberto
- 7) Número de parâmetros inválido

O terceiro arquivo contém o interpretador em si, que se chama "interpreta", tendo "x" e "y" como parâmetros, retornando o valor calculado e a situação na variável "status".

Esta variável vai nos indicar se o resultado é válido ou não, e se não, qual foi o erro.

Os códigos são (por máscara de bits) :

- \$01 divisão por zero.
- \$02 fora de faixa.
- \$04 estouro de exponenciação.
- \$08 erro com logarítmico.
- \$10 erro com potência.
- \$20 erro em raiz.

O exemplo de uso é o artigo mencionado anteriormente.

A IMPLEMENTAÇÃO

Na listagem do "compila.pas" vêem-se da linha 15 até a linha 41 a tabela de símbolos do interpretador, i.e., os identificadores que ele é capaz de reconhecer. Nela estão contidas as constantes, funções e os elementos "x" e "y".

As linhas que vão de 75 a 215 são do analisador léxico. Este identificará os Tokens, e se o Token for um identificador, será feita uma busca na tabela de símbolos para descobrir qual é o tipo e qual é o valor, ou o número de parâmetros (para funções).

Foi usada uma busca sequencial para facilitar a implementação de novas funções (um bom exercício).

O diagrama sintático 1 corresponde ao procedimento "expressão" declarado na linha 219 e tem seu corpo contido nas linhas que vão de 301 a 305 (função "expressão" declarada em 11 e com corpo em 252 a 266 no interpretador) que chama o procedure "mult" (função "mult" no interpretador) e vai repetir o loop se o Token atual é um "mais" ou um "menos", caso não haja erro.

O diagrama sintático 2 corresponde ao procedimento "mult" que está declarado na linha 228 com corpo nas linhas de 293 a 299 (função "mult" declarada em 16, com corpo em 228 a 250 no interpretador) que chama o procedimento "fator" (no interpretador), pedindo depois um novo Token e repetindo o loop, se este for um "vezes" ou um "divisão", caso não haja erro.

O diagrama 3 corresponde ao procedimento "fator" que está declarado na linha 230, com corpo nas linhas de 262 a 291 (função "fator" declarada em 21 e com corpo em 200 a 226 no interpretador) que faz a seguinte sequência:

- 1) Pede um Token ao analisador léxico.
- 2) Se este for das classes "mais" ou "menos", então pede um novo.

3) Caso este seja "idntf_fc", que indica uma função, então chama a "fc_trat".

Caso este seja da classe "abr_prt", que indica abertura de parênteses, então chama-se o procedimento "expressão" e testamos se houve um fechamento de parênteses.

Note que esta é uma chamada recursiva.

Caso seja o resto, i.e., os outros Tokens válidos para este contexto, aceitamo-os sem fazer mais nada.

O diagrama 4 corresponde ao procedimento "fc_trat" que está declarado na linha 232 com corpo nas linhas de 237 a 260 (função "fc_trat" declarada em 25 com corpo de 186 a 198 no interpretador) que faz a seguinte sequência :

- 1) Pede um Token ao analisador léxico.
- 2) Testa se este é um parênteses.
- 3) Faz um loop chamando o procedimento "expressão" e contando o número de parâmetros até que o Token testado seja diferente de "vírgula" ou haja um erro.
- 4) Testa o número de parâmetros.
- 5) Testa se o token atual é "fch_prt" que indica que o parênteses está fechado.

O interpretador, como já devem ter notado, não é muito diferente na implementação.

As principais diferenças são:

- 1) Nas fases "expressão" e "mult" existem variáveis temporárias para acumular o resultado parcial, o tipo da próxima operação e a chamada para o novo nível é feita diretamente em uma operação.
- 2) Existe a função "fc_calcula" para calcular as funções que fazem parte da expressão.
- 3) Não existem testes de sintaxe e semântica (a não ser testes de erros de execução), nem chamadas ao analisador léxico, o que nos faz ganhar tempo na interpretação.

A vantagem de fazer-se uma "compilação" prévia está no fato de se fazer a verificação sintática e semântica e a análise léxica uma vez só, mesmo precisando usar os Tokens muitas vezes.

Desta maneira economiza-se um tempo significante na interpretação.

Pode-se fazer uma comparação (que será um bom exercício), construindo um analisador de expressões que faça as análises léxica, sintática e semântica enquanto interpreta a expressão.

Uma observação: os interpretadores Basic fazem uma análise léxica prévia do programa que está sendo trabalhado, conhecida como forma compactada

de se salvar um programa, e quando se manda listar o programa em questão na tela, ele os transforma para formato ASCII, trocando os símbolos pelos seus identificadores correspondentes, porém não é feita uma análise sintática prévia. Esta é feita a tempo de interpretação, provocando perda de tempo durante os loops.

Nesta implementação a análise semântica ficou muito unida com a análise sintática, o que não acontece muito com os compiladores, pois é o analisador semântico que controla a geração de código, e sempre é o analisador sintático quem rege todas as ações (inclusive neste interpretador).

Uma coisa bem interessante é que se tomarmos algumas precauções, podemos reunir todos os includes apresentados em um único.

Note que o interpretador aceita funções de duas variáveis, estas chamando-se 'x' e 'y'.

Como se pode ver, apesar de não ser trivial, a leitura de expressões matemáticas pelo teclado é possível de ser feita (obviamente com muito bom senso). Em Basic, isto seria impossível, pelo menos no nível do interpretador apresentado, que é bastante sofisticado.

Espero que este artigo seja de seu agrado e que dê a idéia inicial de como funciona um interpretador (e um compilador) ao leitor, e espero que, em uma outra oportunidade, possa editar algo mais sofisticado como, por exemplo, um montador Assembly.

CPU

```

1 function compila( cadeia : tipo_cadeia ) : byte ;
2 type
3   tipo_fc_cte_xy = record
4     nome_idntf : string[ tam_idntf ] ;
5     case idntf_classe : tipo_token of
6       idntf_fc : ( n_param : byte ) ;
7       idntf_cte : ( cte_valor : real ) ;
8     end;
9   end;
10 const
11   num_idntf = 23 ;
12   tab_fc_cte_xy : array [ 1 .. num_idntf ] of tipo_fc_cte_xy =
13   (
14     { nome_idntf : 'abs' ; idntf_classe : idntf_fc ; n_param : 1 }, 96
15     { nome_idntf : 'cos' ; idntf_classe : idntf_fc ; n_param : 1 }, 97
16     { nome_idntf : 'cossec' ; idntf_classe : idntf_fc ; n_param : 1 }, 98
17     { nome_idntf : 'cotg' ; idntf_classe : idntf_fc ; n_param : 1 }, 99
18     { nome_idntf : 'e' ; idntf_classe : idntf_cte ; cte_valor : 2.7182818284 }, 100
19     { nome_idntf : 'exp' ; idntf_classe : idntf_fc ; n_param : 1 }, 102
20     { nome_idntf : 'ln' ; idntf_classe : idntf_fc ; n_param : 1 }, 103
21     { nome_idntf : 'log' ; idntf_classe : idntf_fc ; n_param : 1 }, 104
22     { nome_idntf : 'max' ; idntf_classe : idntf_fc ; n_param : 2 }, 105
23     { nome_idntf : 'min' ; idntf_classe : idntf_fc ; n_param : 2 }, 106
24     { nome_idntf : 'mod' ; idntf_classe : idntf_fc ; n_param : 2 }, 107
25     { nome_idntf : 'pi' ; idntf_classe : idntf_cte ; cte_valor : 3.14159265359 }, 108
26     { nome_idntf : 'pow' ; idntf_classe : idntf_fc ; n_param : 2 }, 110
27     { nome_idntf : 'rand' ; idntf_classe : idntf_fc ; n_param : 1 }, 111
28     { nome_idntf : 'round' ; idntf_classe : idntf_fc ; n_param : 1 }, 112
29     { nome_idntf : 'sec' ; idntf_classe : idntf_fc ; n_param : 1 }, 113
30     { nome_idntf : 'sign' ; idntf_classe : idntf_fc ; n_param : 1 }, 114
31     { nome_idntf : 'sin' ; idntf_classe : idntf_fc ; n_param : 1 }, 115
32     { nome_idntf : 'sqrt' ; idntf_classe : idntf_fc ; n_param : 1 }, 116
33     { nome_idntf : 'tg' ; idntf_classe : idntf_fc ; n_param : 1 }, 117
34     { nome_idntf : 'trunc' ; idntf_classe : idntf_fc ; n_param : 1 }, 118
35     { nome_idntf : 'x' ; idntf_classe : idntf_1 }, 119
36     { nome_idntf : 'y' ; idntf_classe : idntf_2 } ) 120
37 );
38
39 var
40   pos_cadeia : byte ;
41   tmp_cadeia : tipo_cadeia ;
42   pont_lst_atual : tipo_pont_lst_token ;
43   expr_erro : boolean ;
44
45 procedure inicializa;
46 var
47   aux : byte ;
48   p_aux : tipo_pont_lst_token ;
49 begin
50   while cabeca_lst () nil do
51   begin
52     aux := cabeca_lst ;
53     cabeca_lst := cabeca_lst^.proximo ;
54     dispose( p_aux )
55   end;
56   new( cabeca_lst );
57   pont_lst_atual := cabeca_lst ;
58   pont_lst_atual^.proximo := nil ;
59
60   pos_cadeia := 1 ;
61   expr_erro := false ;
62
63   cadeia := cadeia + #10
64 end;
65
66
67 function lexico : boolean ;
68 var
69   v_aux,base,valor : real ;
70   meio,i : byte ;
71   sinal : integer ;
72
73
74 function upcase( cadeia[ pos_cadeia ] ) : 'A'..'Z';
75 var
76   v_aux,base,valor : real ;
77   meio,i : byte ;
78   sinal : integer ;
79
80 begin
81   if upcase( cadeia[ pos_cadeia ] ) = 'E' then
82     begin
83       new( pont_lst_atual^.proximo );
84       pont_lst_atual := pont_lst_atual^.proximo ;
85       pont_lst_atual^.proximo := nil ;
86
87       lexico := true ;
88       while cadeia[ pos_cadeia ] in [ '#09,' ' ] do
89         pos_cadeia := succ( pos_cadeia ) ;
90
91       case cadeia[ pos_cadeia ] of
92         'a'..'z','A'..'Z':
93         begin
94           i := 0 ;
95
96           while cadeia[ pos_cadeia ] in [ 'a'..'z','A'..'Z' ] do
97             begin
98               i := succ( i ) ;
99               tmp_cadeia[ i ] := cadeia[ pos_cadeia ] ;
100              pos_cadeia := succ( pos_cadeia ) ;
101            end;
102
103            tmp_cadeia[ 0 ] := chr( i ) ;
104
105            meio := i ;
106            while ( tab_fc_cte_xy[ meio ].nome_idntf () tmp_cadeia ) and
107              ( meio ( num_idntf ) do
108                meio := succ( meio ) ;
109
110            if tab_fc_cte_xy[ meio ].nome_idntf = tmp_cadeia then
111              begin
112                pont_lst_atual^.classe := tab_fc_cte_xy[ meio ].idntf_classe ;
113                case pont_lst_atual^.classe of
114                  idntf_fc :
115                    pont_lst_atual^.idntf_fc.cod := meio ;
116                  idntf_cte :
117                    pont_lst_atual^.idntf_cte.valor :=
118                      tab_fc_cte_xy[ meio ].cte.valor
119                end
120              end
121            else
122              begin
123                pont_lst_atual^.classe := erro ;
124                pont_lst_atual^.n_erro := 3 ;
125                lexico := false
126              end;
127
128            pos_cadeia := pred( pos_cadeia ) ;
129
130          end;
131        end;
132      '0'..'9':
133      begin
134        pont_lst_atual^.classe := cte ;
135        valor := 0 ;
136
137        while cadeia[ pos_cadeia ] in [ '0'..'9' ] do
138        begin
139          valor := valor*10 + ord( cadeia[ pos_cadeia ] ) - 48 ;
140          pos_cadeia := succ( pos_cadeia ) ;
141        end;
142
143        if cadeia[ pos_cadeia ] = '.' then
144        begin
145          base := 0.1 ;
146          pos_cadeia := succ( pos_cadeia ) ;
147
148          while cadeia[ pos_cadeia ] in [ '0'..'9' ] do
149          begin
150            valor := valor + ( ord( cadeia[ pos_cadeia ] ) - 48 )/base ;
151            base := base/10.0 ;
152            pos_cadeia := succ( pos_cadeia ) ;
153          end;
154        end;
155
156        if upcase( cadeia[ pos_cadeia ] ) = 'E' then
157        begin
158          pos_cadeia := succ( pos_cadeia ) ;
159          v_aux := 0.0 ;
160        end;

```

```

161      sinal := 1 ;
162      if cadeia[ pos_cadeia ] in [ '>','-' ] then
163      begin
164          if cadeia[ pos_cadeia ] = '-' then
165              sinal := -1 ;
166          pos_cadeia := succ( pos_cadeia )
167      end;
168
169      while cadeia[ pos_cadeia ] in [ '0' .. '9' ] do
170      begin
171          v_aux := v_aux*10 + ord( cadeia[ pos_cadeia ] )- 48 ;
172          pos_cadeia := succ( pos_cadeia )
173      end;
174
175      if v_aux > 37.0 then
176      begin
177          pont_lst_atual^.classe := erro ;
178          pont_lst_atual^.n_erro := 1 ;
179          lexico := false
180      end
181      else
182          valor := valor*pot( abs( valor ),sinal*v_aux )
183      end;
184
185      if pont_lst_atual^.classe () erro then
186          pont_lst_atual^.cte_valor := valor;
187
188      pos_cadeia := pred( pos_cadeia )
189
190  end;
191 #10 :
192 '+' : pont_lst_atual^.classe := fim_lnh ;
193 '-' : pont_lst_atual^.classe := mais ;
194 '*' : pont_lst_atual^.classe := menos ;
195 '/' : pont_lst_atual^.classe := vezes ;
196 '(' : pont_lst_atual^.classe := divisao ;
197 ')' : pont_lst_atual^.classe := abr_prts ;
198 ',' : pont_lst_atual^.classe := fch_prts ;
199 ';' : pont_lst_atual^.classe := virgula ;
200
201 else
202 begin
203     pont_lst_atual^.classe := erro ;
204     pont_lst_atual^.n_erro := 2 ;
205     lexico := false
206 end;
207
208 end;
209
210 begin
211     pos_cadeia := succ( pos_cadeia )
212 end;
213
214 pos_cadeia := succ( pos_cadeia )
215 end;
216
217
218 {$A-}
219 procedure expressao ;
220
221 procedure expr_trat_erro( num_erro : byte );
222 begin
223     pont_lst_atual^.classe := erro ;
224     pont_lst_atual^.n_erro := num_erro ;
225     expr_erro := true
226 end;
227
228 procedure mult;
229
230 procedure fator;
231
232 procedure fc_trat;
233 var
234     p_aux : tipo_pont_lst_token ;
235     cont : byte ;
236
237 begin
238     p_aux := pont_lst_atual ;
239
240     if lexico then
241         if pont_lst_atual^.classe () abr_prts then
242             expr_trat_erro( 6 )
243         else
244             begin
245                 cont := 0 ;
246                 repeat
247                     cont := succ( cont ) ;
248                     expressao
249                 until ( pont_lst_atual^.classe () virgula ) or expr_erro ;
250
251                 if pont_lst_atual^.classe () fch_prts then
252                     expr_trat_erro( 5 )
253                 else if cont () tab_fc_cte_xy[ p_aux^.idntf_fc_cod ].n_param then
254                     expr_trat_erro( 7 )
255             end
256         else
257             expr_erro := true
258     end;
259
260 begin
261     if lexico then
262         begin
263             if pont_lst_atual^.classe in [ mais,menos ] then
264                 expr_erro := not lexico ;
265
266             if not expr_erro then
267                 case pont_lst_atual^.classe of
268                     abr_prts :
269                     begin
270                         expressao ;
271
272                         if not expr_erro then
273                             if pont_lst_atual^.classe () fch_prts then
274                                 expr_trat_erro( 5 )
275                             end;
276                         idntf_fc :
277                             fc_trat ;
278                         cte ,
279                         idntf_1 ,
280                         idntf_2 ,
281                         idntf_cte :
282                             ;
283                         else
284                             expr_trat_erro( 4 )
285                     end
286                 end
287             else
288                 expr_erro := true
289         end;
290
291 begin
292     repeat
293         fator;
294         if not expr_erro then
295             expr_erro := not lexico
296         until not ( pont_lst_atual^.classe in [ vezes,divisao ] ) or expr_erro
297     end;
298
299 begin
300     repeat
301         mult
302         until not ( pont_lst_atual^.classe in [ mais,menos ] ) or expr_erro
303     end;
304 {$A+}
305
306
307 begin
308     inicializa ;
309
310     expressao ;
311
312     if ( pont_lst_atual^.classe = fim_lnh ) and not expr_erro then
313         compil := 0
314     else if expr_erro then
315         compil := pont_lst_atual^.n_erro
316     else
317         compil := 8
318 end;

```

```

1
2
3 function interpreta( X,Y : real ; var status : byte ) : real ;
4 const
5     quase_zero = 0.000001 ;
6
7 var
8     pont_lst_atual : tipo_pont_lst_token ;
9
10 (SA- )
11 function expressao : real ;
12 var
13     tmp_valor : real ;
14     sinal      : integer ;
15
16 function mult : real ;
17 var
18     tmp_valor,aux : real ;
19     multiplicacao : boolean ;
20
21 function fator : real ;
22 var
23     sinal : real ;
24
25 function fc_trat : real ;
26 const
27     num_max_param = 2 ;
28
29 var
30     p_aux      : tipo_pont_lst_token ;
31     cont       : byte ;
32     tab_param : array [ 1 .. num_max_param ] of real ;
33
34 function fc_calcula : real ;
35 var
36     aux : real ;
37
38 begin
39     case p_aux^.idntf_fc_cod of
40         1 : { abs }
41             fc_calcula := abs( tab_param[ 1 ] );
42         2 : { cos }
43             fc_calcula := cos( tab_param[ 1 ] );
44         3 : { cossec }
45             begin
46                 aux := sin( tab_param[ 1 ] );
47                 if aux = 0 then
48                     begin
49                         aux := quase_zero ;
50                         status := status or $02
51                     end;
52
53                 fc_calcula := 1.0 / aux
54             end;
55         4 : { cotg }
56             begin
57                 aux := sin( tab_param[ 1 ] );
58                 if aux = 0 then
59                     begin
60                         aux := quase_zero ;
61                         status := status or $02
62                     end;
63
64                 fc_calcula := cos( tab_param[ 1 ] ) / aux
65             end;
66         6 : { exp }
67             if tab_param[ 1 ] > 87.0 then
68                 begin
69                     status := status or $04 ;
70                     fc_calcula := exp( 87.0 )
71                 end
72             else
73                 fc_calcula := exp( tab_param[ 1 ] );
74         7 : { ln }
75             if tab_param[ 1 ] <= 0 then
76                 begin
77                     status := status or $08 ;
78                     fc_calcula := 0
79                 end
80             else

```

```

81      fc_calcula := ln( tab_param[ 1 ] );
82 8 : { log }
83      if tab_param[ 1 ] <= 0 then
84          begin
85              status := status or $08 ;
86              fc_calcula := 0
87          end
88      else
89          fc_calcula := ln( tab_param[ 1 ] ) / 2.30250509299 ;
90 9 : { max }
91      if tab_param[ 1 ] > tab_param[ 2 ] then
92          fc_calcula := tab_param[ 1 ]
93      else
94          fc_calcula := tab_param[ 2 ];
95 10 : { min }
96      if tab_param[ 1 ] < tab_param[ 2 ] then
97          fc_calcula := tab_param[ 1 ]
98      else
99          fc_calcula := tab_param[ 2 ];
100 11 : { mod }
101     if tab_param[ 2 ] <> 0 then
102         fc_calcula := tab_param[ 1 ] -
103             tab_param[ 2 ]*trunc( tab_param[ 1 ]/tab_param[ 2 ] )
104     else
105         begin
106             fc_calcula := 0 ;
107             status := status or $01
108         end;
109 12 : { pow }
110     if tab_param[ 1 ] = 0 then
111         fc_calcula := 0
112     else if tab_param[ 1 ] > 0 then
113         begin
114             aux := tab_param[ 2 ]*ln( tab_param[ 1 ] );
115
116             if aux > 87.0 then
117                 begin
118                     status := status or $10 ;
119                     fc_calcula := exp( 87.0 )
120                 end
121             else
122                 fc_calcula := exp( aux )
123         end
124     else
125         begin
126             aux := tab_param[ 2 ]*ln( - tab_param[ 1 ] );
127             if ( aux < 87.0 ) and ( abs( tab_param[ 2 ] ) <= maxint ) then
128                 if odd( trunc( tab_param[ 2 ] ) ) then
129                     fc_calcula := - exp( aux )
130                 else
131                     fc_calcula := exp( aux )
132             else
133                 begin
134                     status := status or $10 ;
135                     fc_calcula := exp( 87.0 )
136                 end
137         end;
138 14 : { rand }
139     fc_calcula := random( trunc( tab_param[ 1 ] ) );
140 15 : { round }
141     fc_calcula := round( tab_param[ 1 ] );
142 16 : { sec }
143     begin
144         aux := cos( tab_param[ 1 ] );
145         if aux = 0 then
146             begin
147                 aux := quase_zero ;
148                 status := status or $02
149             end;
150
151             fc_calcula := 1.0 / aux
152         end;
153 17 : { sign }
154     if tab_param[ 1 ] = 0.0 then
155         fc_calcula := 0.0
156     else if tab_param[ 1 ] > 0 then
157         fc_calcula := +1.0
158     else
159         fc_calcula := -1.0 ;
160 18 : { sin }

```

```

161     fc_calcula := sin( tab_param[ 1 ] );
162 19 : ( sqrt )
163     if tab_param[ 1 ] < 0 then
164     begin
165         status := status or $20 ;
166         fc_calcula := 0
167     end
168     else
169         fc_calcula := sqrt( tab_param[ 1 ] );
170 20 : ( tg )
171     begin
172         aux := cos( tab_param[ 1 ] );
173         if aux = 0 then
174             begin
175                 aux := quase_zero ;
176                 status := status or $02
177             end;
178
179         fc_calcula := sin( tab_param[ 1 ] ) / aux
180     end;
181 21 : ( trunc )
182     fc_calcula := trunc( tab_param[ 1 ] )
183 end
184 end;
185
186 begin
187     p_aux := pont_lst_atual ;
188
189     pont_lst_atual := pont_lst_atual^.proximo ;
190
191     cont := 0 ;
192     repeat
193         cont := succ( cont ) ;
194         tab_param[ cont ] := expressao
195     until pont_lst_atual^.classe <> virgula ;
196
197     fc_trat := fc_calcula
198 end;
199
200 begin
201     pont_lst_atual := pont_lst_atual^.proximo ;
202
203     sinal := +1.0 ;
204     if pont_lst_atual^.classe in [ mais,menos ] then
205     begin
206         if pont_lst_atual^.classe = menos then
207             sinal := -1.0 ;
208
209         pont_lst_atual := pont_lst_atual^.proximo
210     end;
211
212     case pont_lst_atual^.classe of
213     abr_prt :
214         fator := sinal*expressao ;
215     idntf_fc :
216         fator := sinal*fc_trat ;
217     idntf_1 :
218         fator := sinal*X ;
219     idntf_2 :
220         fator := sinal*Y ;
221     idntf_cte :
222         fator := sinal*pont_lst_atual^.idntf_cte_valor ;
223     cte :
224         fator := sinal*pont_lst_atual^.cte_valor
225     end
226 end;
227
228 begin
229     tmp_valor := +1.0 ;
230     multiplicacao := true ;
231     repeat
232         if multiplicacao then
233             tmp_valor := tmp_valor*fator
234         else
235             begin
236                 aux := fator ;
237                 if aux = 0.0 then
238                     begin
239                         status := status or $01 ;
240                         aux := quase_zero ;
241
242                         end;
243                         tmp_valor := tmp_valor / aux
244
245                         end;
246                         pont_lst_atual := pont_lst_atual^.proximo ;
247                         multiplicacao := pont_lst_atual^.classe = vezes
248                         until not ( pont_lst_atual^.classe in [ vezes,divisao ] ) ;
249                         mult := tmp_valor
250 end;
251
252 begin
253     tmp_valor := 0 ;
254     sinal := +1 ;
255     repeat
256         tmp_valor := tmp_valor + sinal*mult ;
257
258         if pont_lst_atual^.classe = mais then
259             sinal := +1
260         else
261             sinal := -1
262
263         until not ( pont_lst_atual^.classe in [ mais,menos ] ) ;
264
265         expressao := tmp_valor
266 end;
267 ($At)
268
269 begin
270     status := 0 ;
271     pont_lst_atual := cabeca_lst ;
272
273     interpreta := expressao
274 end;
275
276

```

Arquivo : glbldef.pas pag. : 1

```

1
2 const
3     tam_idntf = 8 ;
4
5 type
6     tipo_cadeia = string[ 100 ] ;
7     tipo_token = ( mais,menos,vezes,divisao,abr_prt,fch_prt,virgula,
8                       cte,idntf_fc,idntf_cte,idntf_1,idntf_2,fim_lnh,erro ) ;
9     tipo_pont_lst_token = ^tipo_pont_lst_token ;
10    tipo_pont_lst_token = record
11        proximo : tipo_pont_lst_token ;
12        classe : tipo_token of
13            cte :( cte_valor : real ) ;
14            idntf_fc :( idntf_fc_cod : byte ) ;
15            idntf_cte :( idntf_cte_valor : real ) ;
16            erro :( n_erro : byte ) ;
17        end;
18
19
20 var
21    cabeca_lst : tipo_pont_lst_token ;
22
23
24 procedure inicio;
25 begin
26     cabeca_lst := nil
27 end;
28
29
30 function pot( a,b : real ) : real ;
31 begin
32     pot := exp( b*ln( a ) )
33 end;
34
35

```

UTILIZANDO O DATA CORDER

Bruno Marrut

A grande maioria dos usuários do MSX ainda trabalha com um gravador cassete, o Data Corder, como forma de armazenamento de dados, devido ao baixo custo que o mesmo apresenta.

Todos nós sabemos que existem meios melhores, mais confiáveis e, principalmente, mais rápidos de se gravar e recuperar dados, mas a intenção deste artigo é apresentar informações que poderão aumentar o desempenho deste periférico.

Pra gravarmos e lermos dados do cassete, o MSX possui os seguintes comandos:

CLOAD - Lê um programa em BASIC

CSAVE - Grava um programa em BASIC

MERGE - Junta o programa BASIC presente na memória com um programa lido do cassete gravado em ASCII

SAVE - Grava um programa no formato ASCII

LOAD - Lê um programa gravado no formato ASCII

BLOAD - Lê um programa em Assembler

BSAVE - Grava um programa em Assembler

Para verificarmos um programa gravado em BASIC temos o comando CLOAD?

Para acionarmos o motor do cassete, ou a sua parada, usamos o comando MOTOR.

GRAVANDO OS DADOS

Para gravar um programa em BASIC na fita cassete, podemos utilizar CSAVE ou SAVE.

Exemplo:

CSAVE "nomearq" ou SAVE "CAS:nomearq"

A gravação em formato ASCII é mais demorada mas apresenta a vantagem que o programa poderá vir a ser adicionado a um outro, ou seja, à medida que você vai desenvolvendo o seu programa, vai efetuando a sua gravação em formato ASCII e, ao final, efetua um MERGE de todos os blocos gravados, obtendo, assim, um único programa.

Ao utilizarmos o comando MERGE devemos tomar o cuidado para que as linhas do programa que vai ser lido não sejam as mesmas do programa que já se encontra na memória. Caso isto venha ocorrer, você perderá linhas de programação do programa que já se encontra na memória do micro.

Quando efetuamos a gravação de um programa podemos especificar, também, a velocidade de gravação, ou seja a taxa de transmissão de dados para o cassete, que poderá ser de 1200 ou 2400 bauds. Ao ligarmos o MSX, o mesmo já se encontra programado para assumir uma taxa de transferência de 1200 bauds, que é a velocidade recomendada pelos fabricantes, pois apresenta um grau de confiabilidade maior, tendo em vista que os dados transmitidos serão mais distribuídos ao longo da fita cassete. Quanto maior for a velocidade de transferência, menor será o espaço utilizado ao longo da fita.

Assim, para gravarmos um programa BASIC com uma taxa de 2400 bauds, utilizaremos o comando CSAVE do seguinte modo:

CSAVE"nomearq", 2 onde '2' especifica uma taxa de transferência de 2400 bauds.

Da mesma forma podemos operar com o comando

SAVE"CAS:nomearq" para alterarmos a taxa de transmissão.

Exemplo:

SAVE"CAS:nomearq",2

Ao utilizarmos os comandos SAVE e LOAD, para a gravação de arquivos no formato ASCII, devemos indicar qual o dispositivo de saída/entrada, tendo em vista que os mesmos são utilizados para a leitura e gravação de dados em disco.

LENDÔ DADOS

A leitura de dados do cassete pode ser feita com um dos comandos abaixo:

CLOAD - Lê o primeiro arquivo em BASIC encontrado

CLOAD "nomearq" - Lê o arquivo especificado

LOAD"CAS:" - Lê o primeiro arquivo em BASIC encontrado

LOAD"CAS:",R - Lê o primeiro arquivo em Basic encontrado e roda-o automaticamente após o carregamento

LOAD"CAS:nomearq" - Lê o arquivo especificado

LOAD"CAS:nomearq",R - Lê o arquivo especificado e roda-o ao final do carregamento.

RUN"CAS:" - Lê o primeiro arquivo encontrado e roda-o automaticamente

RUN"CAS:nomearq" - Lê o arquivo especificado e roda-o automaticamente

Como você já deve ter notado, para ler um programa com o comando CLOAD, o mesmo deverá ter sido gravado com CSAVE. Para ler um programa com LOAD, ou com RUN, o mesmo deverá ter sido gravado com SAVE.

O comando RUN"CAS:" é muito utilizado para a leitura de jogos que possuem mais de um bloco e que têm um pequeno programa em BASIC cuja função é efetuar o carregamento dos demais, evitando, assim, que se tenha que dar vários comandos de leitura. A função do comando RUN"CAS:", que é exatamente a mesma do LOAD"CAS:",R, tem a vantagem de não ser necessário dar o comando RUN após o carregamento, pois o programa será executado automaticamente.

Na leitura você não precisa se preocupar com a taxa de transmissão, pois isto é feito automaticamente pelo computador

JUNTANDO PROGRAMAS

Juntar dois programas pode vir a ser muito útil e poupar algum tempo de digitação de rotinas que já foram desenvolvidas e que se encontram armazenadas. Para tal existe o comando MERGE, que permite que um programa, gravado em formato ASCII, possa ser lido e intercalado com o programa que já se encontra na memória do micro.

A sintaxe do comando MERGE é a seguinte:

MERGE"CAS:nomearq"

PROGRAMAS EM LINGUAGEM ASSEMBLER

Para gravarmos um programa em ASSEMBLER usamos os comandos BSAVE e BLOAD. A gravação de programas em Assembler exige uma série de conhecimentos desta linguagem, sendo assunto de artigos que serão publicados nos próximos números de CPU.

I CONCURSO NACIONAL DE SOFTWARE PARA MSX

A NEWSOFT INFORMATICA LTDA. lança, para todo o país, o I Concurso Nacional de Software para MSX, com o objetivo de abrir espaço para a divulgação de novos talentos brasileiros.

Poderão participar do concurso programadores de qualquer idade, profissionais ou não da área de informática de todo o Brasil, sendo vedada a participação no concurso de funcionários, ou parentes, da empresa patrocinadora e/ou realizadora, bem como os membros da "Comissão Julgadora".

Os interessados poderão participar com um ou mais programas, programados em Basic, Linguagem de Máquina, ou qualquer outra, que versem sobre qualquer tema (jogos, aplicativos, utilitários, etc.), desde que sejam inéditos e de autoria do participante.

Os programas deverão ser remetidos em disco, ou fita, até o dia 31.12.88, data do encerramento das inscrições, acompanhados de uma declaração de autoria, contendo os dados pessoais, como nome, endereço e telefone para contato, do autor.

Os trabalhos de seleção e classificação serão realizados

por uma "Comissão Julgadora", designada pela empresa realizadora, sendo as suas decisões irrevogáveis e irretratáveis. Como critério de avaliação para apuração e classificação serão observados os seguintes ítems:

- a- criatividade;
- b- originalidade;
- c- adequação do software ao mercado de informática.

Todos os programas classificados, ou não, passarão a pertencer à Empresa realizadora, que se reserva ao direito de fazer o uso que lhe convier, podendo, ainda, utilizar o nome dos contemplados na divulgação do concurso, sem qualquer ônus para a mesma.

Os prêmios são:

- 1 lugar - 1 drive 3 1/2 polegadas
- 2 lugar - 1 microcomputador padrão MSX
- 3 lugar - 1 impressora

O resultado será divulgado através das revistas especializadas da área, contendo a relação dos ganhadores.

Os prêmios serão entregues pelo cartão nacional.

I CONCURSO NACIONAL DE SOFTWARE PARA MSX

REGULAMENTO

A NEWSOFT INFORMATICA LTDA.. lança em nível nacional o I CONCURSO NACIONAL DE SOFTWARE PARA MSX.

OBJETIVO DO CONCURSO: Abrir espaço para a divulgação de novos talentos brasileiros.

QUEM PODE PARTICIPAR: Podem participar programadores de qualquer idade, profissionais ou não da área de informática de todo o Brasil

QUEM NÃO PODE PARTICIPAR: Não poderão participar do concurso, funcionários ou parentes da empresa patrocinadora e/ou realizadora, bem como os membros da "Comissão Julgadora".

COMO PARTICIPAR: Os interessados poderão participar com um ou mais programas, programados em Basic, Linguagem de Máquina ou qualquer outra, que versem sobre qualquer tema (jogos, aplicativos, utilitários, etc.), desde que sejam inéditos e de autoria do participante. Os mesmos deverão ser remetidas em disco ou fita, até o dia **30.12.88**, data de encerramento das inscrições, acompanhados de uma "Declaração de Autoria" contendo seus dados pessoais (nome, endereço e telefone para contato).

DA SELEÇÃO E CLASSIFICAÇÃO: Os trabalhos de seleção e classificação serão realizados por uma "Comissão Julgadora" designada pela empresa realizadora, cuja decisão é irrevogável e irretratável. Como critério de avaliação para apuração e classificação, serão observados os seguintes ítems: **a) criatividade; b) originalidade; c) adequação do software ao mercado de informática.**

Todos os programas classificados ou não, passarão a pertencer à empresa realizadora, que se reserva o direito de fazer o uso que lhe convier, podendo ainda utilizar o nome dos contemplados na divulgação do concurso sem qualquer ônus para a mesma.

DA PREMIAÇÃO: 1º lugar — um drive 3 1/2 polegadas

2º lugar — uma impressora

3º lugar — um microcomputador padrão MSX

DO RESULTADO: O resultado será divulgado através da mesma revista onde está sendo publicado este regulamento, na edição de janeiro de 1989, contendo a relação dos ganhadores.

DA ENTREGA DOS PRÊMIOS: Os prêmios serão entregues através do CARTÃO NACIONAL

FICHA DE INSCRIÇÃO

Nome: _____

End.: _____

Cidade: _____ UF: _____

CEP: _____ Tel.: _____

Título do Programa: _____

DECLARAÇÃO DE AUTORIA

Declaro que o programa que estou enviando para participar do I CONCURSO NACIONAL DE SOFTWARE PARA MSX, é inédito e de minha autoria

Assinatura

Remeta para NEWSOFT INFORMATICA LTDA.
— Rua Senador Dantas 117, Sala 736 — Rio
de Janeiro — RJ — CEP 20031

FAÇA SUA
INSCRIÇÃO
HOJE MESMO!

APOIO:

Cartão Nacional

PARTICIPE!

O Cartão que está a seu lado

CONECTIVIDADE MSX-PC-MAINFRAME

Victor Grytz
MSX Informática

Quando há mais de três anos, ao mesmo tempo em que o padrão MSX era introduzido no Brasil, resolvemos criar a MSX Informática Ltda., especializada unicamente em MSX.

Muitos nos chamaram de loucos, pois achavam que o MSX não passava de um vídeo game de luxo.

A ausência de programas mais sérios, bem como a falta dos periféricos prometidos pelos fabricantes, faziam com que houvesse uma certa decepção quanto à consolidação do MSX no nosso país.

Hoje, passados estes anos iniciais, podemos comprovar que o nosso pioneirismo estava correto. Com mais de 2000 programas disponíveis nas áreas de lazer, profissional, educacional, aceitando diversas linguagens, como Cobol, Turbo Pascal, Forth, Fortran, Ada, Lisp, C, Prolog, etc., além de possuir revistas especializadas, como esta, em tão pouco tempo o MX se tornou o equipamento de 8 bits mais vendido no país.

A conversão de programas famosos, como o dBASE II, Supercalc e Wordstar, com a grande vantagem de possibilitar o intercâmbio de arquivos com micros de 16 bits do padrão IBM PC, consolidaram ainda mais a posição do MSX no Brasil.

A cada dia que passa é crescente o número de empresas que, sob diversas formas, se utilizam dos MSX para transferência de dados com micros de 16 bits e mainframes.

O serviço de Videotexto possui cerca de 10.000 micros para utilização como terminal, possibilitando o acesso às suas informações a uma infinidade de profissionais e empresas, que, além disso, o utilizam para os serviços de correspondência, finanças e controle nos seus próprios negócios.

Para as pessoas que possuem um MSX próprio, o acesso ao Videotexto, Cirandão e outras bases de dados, pode ser feita através de um modem, com um bom programa de comunicação, como o VTXMSX, por exemplo.

Nestes serviços podem ser encontradas informações para o uso diário, troca de mensagens e intercâmbio de programas. Assim, uma pessoa de São Paulo pode se conectar a outra do Rio de Janeiro, Brasília ou qualquer outra cidade do mundo.

Existem, ainda, outras formas de conexão para a troca de arquivos e programas com outros computadores. São as redes e os emuladores de terminal. No caso das redes, a Targus

permite a ligação de até 16 micros MSX em rede ligados a um IBM compatível, e os emuladores de terminal da DDX e da Cibertron permitem que até 25 MSX possam compartilhar os recursos de hardware software do PC ou mainframe.

A REDE TARGUS

A Rede Targus possui um multiplexador de canais projetado para gerenciar 16 portas seriais assíncronas, atuando como gerenciador da rede controlando todo e qualquer acesso aos arquivos de dados.

Seu sistema básico de transmissão e recepção é feito no formato "first in - first out", ou seja, o primeiro terminal que solicitar será o primeiro a receber prioridade de resposta, e assim sucessivamente.

O software, desenvolvido em Assembler, é capaz de gerenciar as solicitações de cada um dos 16 terminais, sem que haja degradação no tempo de processamento. Como o MSX não possui saída serial, foi desenvolvida, também, uma interface serial, que possui em sua estrutura interna uma eprom contendo rotinas de transmissão e recepção de dados necessárias para a comunicação dos MSX com o multiplexador e com o PC.

Na rede, as tarefas de atualização de arquivos, troca de informações, gravação de programas, etc. são processadas "online" em ambiente simulado de multiusuário. O sistema é extremamente fácil de ser operado, possibilitando que todos terminais solicitem a manipulação do mesmo programa, que cada terminal manipule um programa diferente e que o mesmo arquivo de dados possa ser manipulado por vários terminais e vários programas simultaneamente.

Os terminais da Rede Targus podem estar em distâncias superiores a 500 metros, sendo interligados por uma rede de cabos coaxiais. O multiplexador, ou MUX, custa cerca de 135 OTN's e cada interface serial custa, aproximadamente, 25 OTN's. Acompanha o multiplexador dois softwares gerenciadores da rede, um em Cobol e outro em dBASE III.

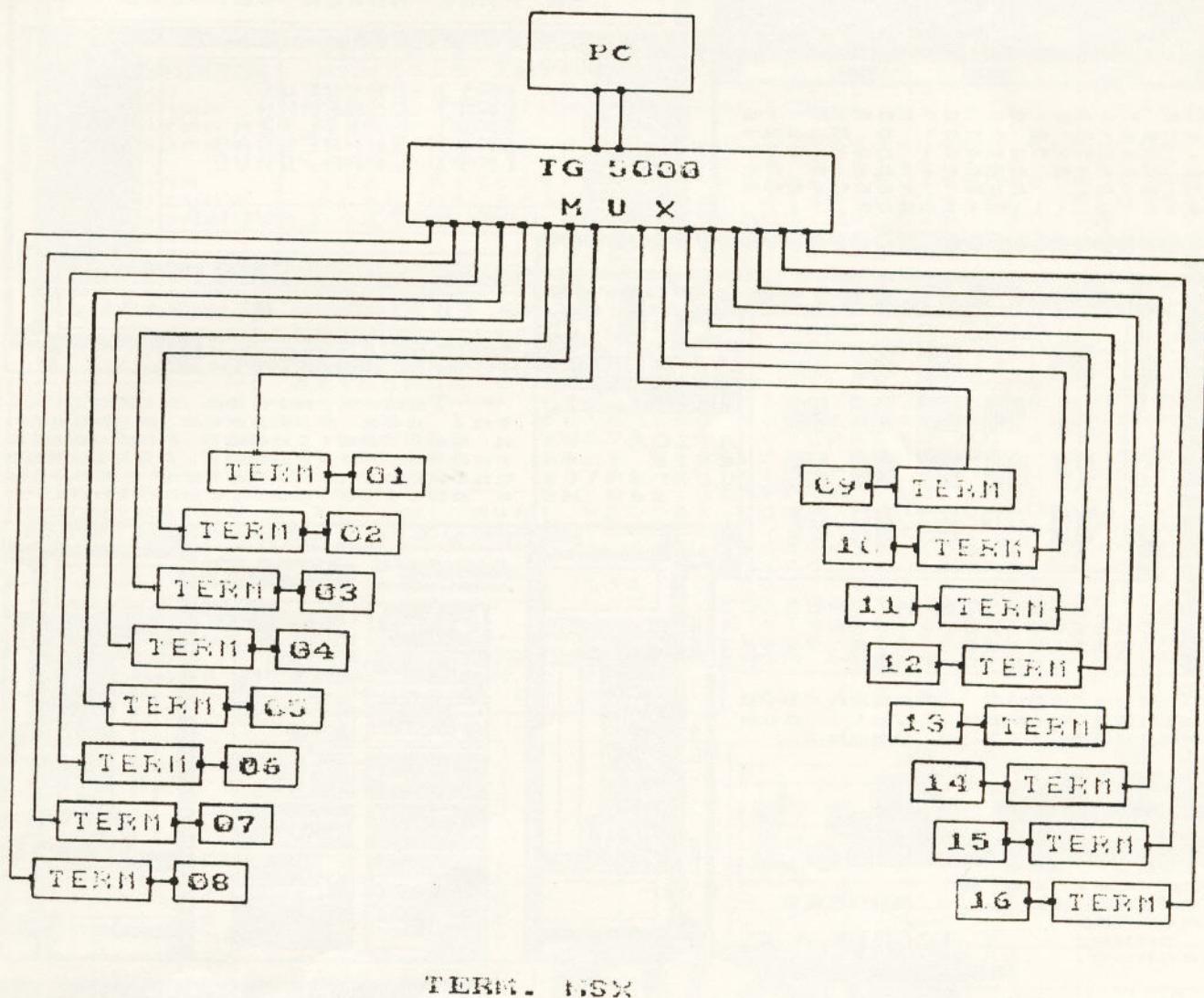
OS CARTUCHOS EMULADORES DDX E CIBERTRON

Os cartuchos emuladores de terminal DDX e Cibertron possuem um software residente que possibilita aos micros da linha MSX a emulação de terminais IBM PC (XT, AT ou 386), além da emulação de terminais mainframe que possuam conversor de protocolo.

O cartucho da Cibertron, por exemplo, possui uma série de características, como emulação de terminal do tipo VT52, saída para impressora local, hardcopy de tela, velocidade de até 19200 bps, caracteres gráficos do IBM disponíveis no MSX, para permitir que programas como o dBASE III, Lotus 123 e muitos outros possam ser utilizados no terminal MSX.

Estes cartuchos funcionam para distâncias mais curtas, até 50 metros, e o seu custo é de cerca de 40 OTN's cada um, mas não devemos nos esquecer que, nestes casos, há necessidade de um software do tipo Multilink, Pick, PCMOS, etc, na faixa de 150 OTN's.

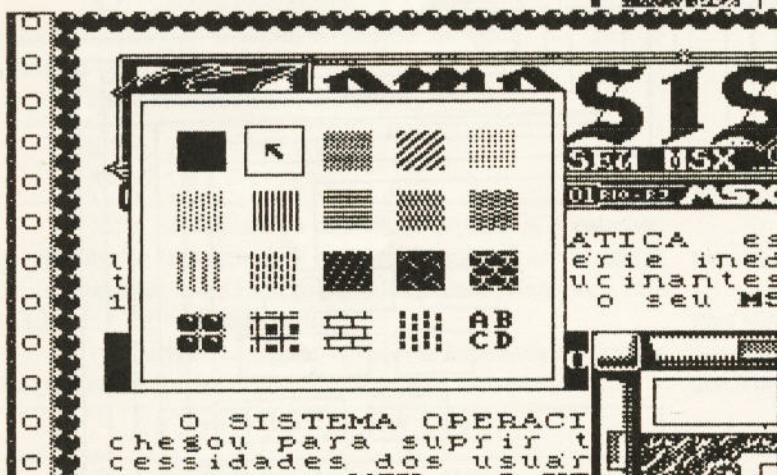
A grande vantagem das redes e emuladores de terminais MSX é que, além de terem um custo inferior a qualquer terminal dedicado existente no mercado, ainda possibilitam a utilização do micro para processamentos locais, possibilitando rapidamente o retorno do investimento realizado. Deste modo, com a entrada do MSX no restrito mercado das redes e terminais, fica consolidada a sua posição de líder no mercado de 8 bits em tão pouco tempo, mais uma vez comprovando a nossa crença no potencial deste equipamento.



MSX PAGE MAKER



Os módulos principais do programa são: o Redator, responsável por toda a parte escrita; o Desenhista, com recursos gráficos ilimitados...



O SISTEMA OPERACI
chegou para suprir t
cessidades dos usuár
veis com o MSX. O HE

O MSX PAGE MAKER 1.2
é ainda compatível com
o GRAPHOS III e seus:

SHAPES

Figuras deco
rativas que
podem ser co
locadas fa
cilmente
em seus
desenhos;



LETROS

ALFABETOS, ALFABETOS, ALFABETOS, e
diversos tipos de letras disponíveis,
inclusive COMPACTADA e INVERTIDA,
ou seja, centenas de combinações!

TELAS

Compatibilidade à nível de telas com todos os editores gráficos existentes no mercado para MSX. Telas no padrão ".SCR" e ".GRP" são carregadas sem problemas ("user-friendly").

A NEMESIS INFORMATICA está lançando um software revolucionário. O MSX terá duas épocas distintas: Antes e depois do:

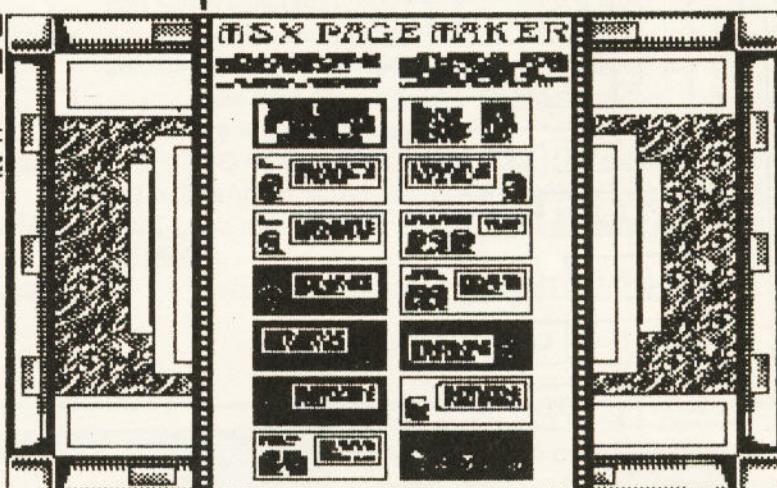
MSX PAGE MAKER 1.2

Mais que um processador, ele reúne textos e gráficos para o desenvolvimento de páginas super especiais para as mais diversas utilidades. De avas a sua imaginação:



1A EDITADA: NEMESIS1

Temos ainda o Estilista, que prepara a página a ser editada com diferentes bordas, títulos, cabecalhos, etc. Enfim, o estilo do trabalho.



MSX PAGE MAKER 1.2
SÓ EM DISCO - 3 DTN.

EN
TEMPO...



GRAPHOS III é marca registrada de Renato Degiovanni, sob "venda exclusiva" pela NEMESIS INFORMATICA LTDA.!

PROTEJA-SE CONTRA OS PIRATAS!

JANEIRO

SE	TE	RR	QU	SX	Sa	DO
1	9	8	7	6	5	4
10	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

D	T
P	G
C	E
BASIC	6

SEXTA 1

CTRL + STOP : MENU

TECLE A OPÇÃO: ■

ANOTACOES:

2:00hs - Dentista
2:45hs - Cabelereiro
4:30hs - Ir a Nemesis
9:20hs - TU Pirata
11:00hs - Video-Texto

PORTFOLIO

Entre em 1989 com o PORTFOLIO MSX, uma Agencia Computadorizada completa para voce nao esquecer os seus compromissos urgentes ou ate mesmo os telefones de seus amigos.

O PORTFOLIO MSX possui ainda: Calendario Perpetuo, Calculadora e Lista de Telefones para Emergencias. Apenas em disco - 2 OTN

DIRETORIO

EDITAR
EXAMINAR
FORMATAR
RESTAURAR
MAPA
ORDENAR
PESQUISAR
HARDWARE
SISTEMA
LIMPAR
VERSAO
BACK-UP
RETORNAR

EXECUTAR

RENOMEAR
APAGAR
EXIBIR
COPIAR
LABELS
RETORNAR

DRIVES LOG
DRIVES FISICOS: 2
MEMORIA RAM: 65535
MEMORIA VRAM: 16383

EXAMINANDO O SETOR: 345
ERRO NO SETOR: 234

VELOCIDADE
ALINHAMENTO
LOAD/SAVE
LIMPEZA
RETORNAR

RECUPERAR O SETOR
CONTINUAR O EXAME
RETORNAR AO MENU

MSX CHART

O MSX CHART é o primeiro gerenciador para graficos estatisticos e comerciais, realmente profissional a ser criado para os micros MSX.

Como voce pode notar os graficos gerados podem ser impressos e/ou utilizados juntamente com o MSX PAGE MAKER.

O programa calcula medias, gera graficos de barras, setoriais, lineares, de area, etc, juntos para comparacao, ou separados. Apenas em disco - 2 OTN

HELLO MSX

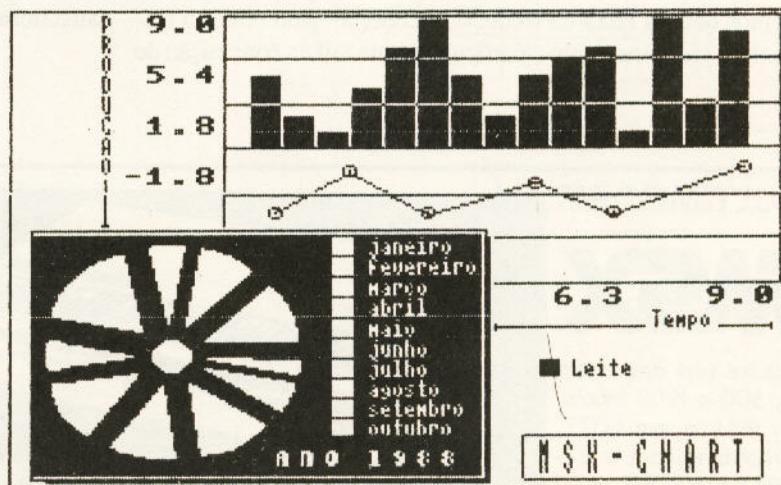
Agora, sempre que voce ligar o seu MSX, aparecera' na tela o seu nome e o nome da NEMESIS INFORMATICA! E' que voce nao pode deixar de adquirir o mais novo e revolucionario SISTEMA OPERACIONAL existente para a linha MSX.

O HELLO simplifica todas as funcoes de operacao com discos e disk-drives, formata ate discos defeituosos, avisando a "FAT" qual os setores nao disponiveis, conserta como milagre os "ERROS de E/S" automaticamente durante o exame de discos; possui testes de HARDWARE como:

- velocidade de rotacao de drive
- alinhamento radial de cabecote
- teste de memoria "RAM";
- teste de memoria "VRAM", etc...

O HELLO ainda ordena o diretorio de discos pelo nome ou pela extensao dos arquivos, mostra o "MAPA" de ocupacao do disco, edita e procura "STRINGS" em arquivos ou setores, possui um poderoso "ZAPPER" interno com inovacoes como funcao "UNDO", cria "LABELS", ou seja, personaliza o seu disquete com o seu nome ou o numero de cadastro de discos, ou ainda, o que voce imaginar. E aguarde por muitas outros segmentos opcionais e extensoes que estaremos lancando em breve. Prepare-se!

Apenas em disco 3 OTN



CÁLCULO DE CIRCUITOS RESSONANTES L-C

ANTONIO FERNANDO SHALDERS

Um grande problema para quem lida com rádio frequência é o correto dimensionamento dos circuitos ressonantes do tipo indutor capacitor, pois este é altamente crítico.

O programa apresentado dará uma imensa ajuda a rádio amadores, estudantes de engenharia eletrônica e afins, pois faz todos os cálculos relativos ao assunto com grande rapidez e confiabilidade.

Assim que o programa começar, aparecerá um menu de opções. Faça a sua escolha, pressionando o número em questão e, logo após, pressione [RETURN].

A frequência de ressonância é determinada quando as reatâncias capacitiva e indutiva dos componentes do circuito ressonante se igualam.

Estas reatâncias são dadas por:

$$X_C = 1 / (2 \times \pi \times F \times C)$$
$$X_L = 2 \times \pi \times F \times L$$

X_C e X_L indicam as reatâncias capacitiva e indutiva, respectivamente (em ohms). F indica a frequência em Hertz, C indica a capacidade em Farads, L a indutância em Henrys e π vale 3.141592654, aproximadamente.

Igualando-se as duas equações, obtemos a fórmula geral para ressonância:

$$F = 1 / (2 \times \pi \times \text{SQRT}(L \times C))$$

O programa apresentado usa esta fórmula. As unidades em questão são as mesmas.

No caso de operarmos com um capacitor variável, podemos obter a faixa de frequência coberta, pelo circuito ressonante, e vice-versa. Isto é particularmente útil na construção de

rádio transmissores ou receptores de FM, por exemplo.

Já o cálculo de indutores (ou bobinas) sempre foi muito problemático, mas o programa se sai muito bem em relação a isto.

Para isso, utilizam-se duas fórmulas específicas, uma para cada relação comprimento-diâmetro da bobina. Lembre-se que um pouco de bom senso é necessário quanto a isso. Não vá querer calcular um indutor com diâmetro de 1 mm e comprimento de 1 m!

As fórmulas utilizadas pelo programa são as seguintes (c é o comprimento e D o diâmetro):

$$\text{Se: } 0.3 < c/d < 1.0$$
$$N = \text{sqrt}((143 \times L \times c^{0.57}) / (d^{1.57}))$$
$$\text{Se: } 1.2 < c/d < 8.0:$$
$$N = \text{sqrt}((137 \times L \times c^{0.863}) / (d^{1.863}))$$

N indica o número de espiras, C o comprimento da bobina e D o diâmetro.

Se a relação c/d estiver fora dos intervalos acima mencionados, o resultado não será confiável.

Os resultados obtidos são extremamente confiáveis, pois vários indutores foram confeccionados segundo este método e conferidos em uma ponte de indutâncias da General Radio, de aproximadamente US\$ 6,000.00 e o erro foi menor que 5% !

Se você desejar saber o fator de qualidade do indutor, basta usar a seguinte fórmula:

$Q = (2 \times \pi \times F \times L) / R$, onde R é a resistência ohmica do fio. Esta opção não foi incluída no programa. Fica a seu critério incluí-la ou não, pois o Q das bobinas calculadas geralmente é satisfatório.

As bobinas são de seção circular e núcleo de ar.

MULTIMODEM

MSX

O único que opera em 75, 300 e 1200 bits/s, em BELL e em CCITT. Compatível com o EXPERT e o HOT-BIT.



Com o cartucho modem da TELCOM TELEMÁTICA você pode acessar o Criançário, o Aruanda, a Bireme, comunicar-se micro-à-micro, e contactar bancos de dados no Brasil e no Exterior.

O multimodem já incorpora a RS-232, e o software de comunicação é gratuito.

TELCOM TELEMÁTICA

Rua Anita Garibaldi, 1700 - f: (0512)419871
90430 - Porto Alegre - RS - Brasil

PROGRAM LC 30

Este programa calcula os valores de componentes para circuitos ressonantes L-C, bem como as faixas de operação dos mesmos. Calcula também as dimensões de indutores com seção circular e núcleo de ar.

{área de definição de variáveis}

VAR

```
C1,C2,L,F1,F2 :REAL;
A :INTEGER;
C :CHAR;
```

{definição da função potenciação}

```
FUNCTION POT(B,E:REAL):REAL;
BEGIN
  POT := EXP(E*LN(B));
END;
```

{cálculo da frequência de ressonância}

```
PROCEDURE FREQUENCIA;
BEGIN
  WRITELN('# CALCULO DA RESSONANCIA #');
  WRITELN;
  WRITE('L (H) : '); READLN(L);
  WRITE('C (F) : '); READLN(C1);
  WRITELN;
  F1:=1/(2*PI*SQRT(L*C1));
  WRITELN('RESSONANCIA EM : ',F1,' HZ');
END;
```

{cálculo da indutância}

```
PROCEDURE INDUTANCIA;
BEGIN
  WRITELN('# CALCULO DA INDUTANCIA #');
  WRITELN;
  WRITE('F (HZ) : '); READLN(F1);
  WRITE('C (F) : '); READLN(C1);
  WRITELN;
  L:=SQR(1/(2*PI*F1*SQRT(C1)));
  WRITELN('INDUTANCIA : ',L,' H');
END;
```

{cálculo da capacidade}

```
PROCEDURE CAPACITANCIA;
BEGIN
  WRITELN('# CALCULO DA CAPACITANCIA #');
  WRITELN;
  WRITE('F (HZ) : '); READLN(F1);
  WRITE('L (H) : '); READLN(L);
  C1:=SQR(1/(2*PI*F1*SQRT(L)));
  WRITELN;
  WRITELN('CAPACITANCIA : ',C1,' F');
END;
```

{cálculo da faixa de frequência em função de um capacitor variável}

```
PROCEDURE F_FREQ;
BEGIN
  WRITELN('# CALCULO DA FAIXA DE FREQUENCIA #');
  WRITELN;
  WRITE('CAPACITANCIA MINIMA (F) : ');
  READLN(C1);
  WRITE('CAPACITANCIA MAXIMA (F) : ');
  READLN(C2);
  WRITE('INDUTANCIA (H) : '); READLN(L);
  F1:=1/(2*PI*SQRT(L*C1));
  F2:=1/(2*PI*SQRT(L*C2));
  WRITELN;
  WRITELN('FREQ. MINIMA : ',F2,' HZ');
  WRITELN('FREQ. MAXIMA : ',F1,' HZ');
END;
```

{cálculo da faixa de capacidade em função de uma frequência variável}

```
PROCEDURE F_CAP;
BEGIN
  WRITELN('# CALCULO DA FAIXA DE CAPACITANCIA #');
  WRITELN;
  WRITE('FREQUENCIA MINIMA (HZ) : ');
  READLN(F1);
  WRITE('FREQUENCIA MAXIMA (HZ) : ');
  READLN(F2);
  WRITE('INDUTANCIA (H) : '); READLN(L);
  C1:=SQR(1/(2*PI*F1*SQRT(L)));
  C2:=SQR(1/(2*PI*F2*SQRT(L)));
  WRITELN;
  WRITELN('CAPACITANCIA MIN : ',C2,' F');
  WRITELN('CAPACITANCIA MAX : ',C1,' F');
END;
```

{cálculo da bobina}

```
PROCEDURE BOBINA;
BEGIN
  WRITELN('# CALCULO DA BOBINA #');
  WRITELN;
  WRITELN('# SEÇÃO CIRCULAR / NÚCLEO DE AR #');
  WRITELN;
  WRITE('INDUTANCIA (μH) : '); READLN(L);
  WRITE('DIAMETRO (cm) : '); READLN(C1);
  WRITE('COMPRIMENTO (cm) : '); READLN(C2);
```

{escolha da fórmula em função do diâmetro e do comprimento}

```
F1 := C2/C1 ;
IF ( (F1>0.29) AND (F1<1.01) ) THEN
  F2 := SQRT((143*L*POT(C2,0.57))/POT(C1,1.57))
ELSE
  IF ( (F1>1.19) AND (F1<8.01) ) THEN
    F2 := SQRT((137*L*POT(C2,0.863))/POT(C1,1.863))
  ELSE
    BEGIN
      WRITELN;
      WRITELN('*** RESULTADO NAO CONFIAVEL ***');
      WRITELN;
      WRITELN('TENTE NOVAMENTE');
      DELAY(1600);
      CLRSCR;
```

```
BOBINA;
END;
GOTOXY(1,10);
WRITE('A BOBINA DEVERÁ TER ',F2:4:1,' ESPIRAS.');
END;
```

{apresentação e menu principal}

```
PROCEDURE MENU;
BEGIN
  CLRSCR;
  GOTOXY(9,1); WRITE(' 0   1   2   3   4   5   6   7   8   9');
  GOTOXY(9,2); WRITE(' 0   1   2   3   4   5   6   7   8   9');
  GOTOXY(9,3); WRITE(' 0   1   2   3   4   5   6   7   8   9');
  GOTOXY(9,4); WRITE(' 0   1   2   3   4   5   6   7   8   9');
  GOTOXY(9,5); WRITE(' 0   1   2   3   4   5   6   7   8   9');
  GOTOXY(8,7); WRITE('(C) 1988 by A. F.
Shalders');
  GOTOXY(1,10); WRITE('-----');
  GOTOXY(8,14); WRITE('[1] - FREQUENCIA');
  GOTOXY(8,15); WRITE('[2] - INDUTANCIA');
  GOTOXY(8,16); WRITE('[3] - CAPACITANCIA');
  GOTOXY(8,17); WRITE('[4] - FAIXA DE FREQUENCIA');
  GOTOXY(8,18); WRITE('[5] - FAIXA DE CAPACITANCIA');
  GOTOXY(8,19); WRITE('[6] - CÁLCULO DA BOBINA');
  GOTOXY(8,24); WRITE('OPÇÃO : ');
  READ(A);
  IF ( (A<1) OR (A>6) ) THEN MENU;
  CLRSCR;
  CASE A OF 1: FREQUENCIA;
  2: INDUTANCIA;
  3: CAPACITANCIA;
  4: F_FREQ;
  5: F_CAP;
  6: BOBINA;
  END;
END;
```

{inicialização do programa}

```
PROCEDURE INIT;
BEGIN
  MENU;
  C1:=0;
  C2:=0;
  L:=0;
  F1:=0;
  F2:=0;
  GOTOXY(1,24);
  WRITE('DESEJA CONTINUAR (S/N)');
  READ(KBD,C);
  C:=UPCASE(C);
  IF (C='S') THEN INIT;
END;
```

{corpo do programa principal}

```
BEGIN
  INIT;
  CLRSCR;
END.
```

dBASE II PLUS MSX

dBEST

DINO H. POLETO
DIRETOR DA PRINCESSWARE

Não é preciso pensar muito para se perceber que quase todas as profissões ou empresas requerem algum sistema bem organizado para o armazenamento e recuperação de informações.

Mesmo a microempresa pode se beneficiar de um eficiente Sistema Gerenciador de Dados.

O gerenciamento de dados é, sem dúvida, um dos mais úteis empregos que se pode dar ao microcomputador.

O banco de dados é, simplesmente, um conjunto de dados organizados de forma a serem usados com um determinado objetivo. Um exemplo comum de banco de dados é a lista telefônica, que vem a ser um conjunto de informações organizadas de modo a possibilitar a fácil localização de números telefônicos.

Este banco de dados contém nomes, endereços e telefones de indivíduos, empresas ou instituições. Os endereços e telefones têm pequeno valor por si mesmos, a menos que estejam RELACIONADOS a um nome.

Quando paramos para pensar a respeito, ficamos realmente surpresos com o número de banco de dados que nos são familiares.

Alguns dos mais comuns são: um dicionário, um livro de receitas, um catálogo de uma loja, um relatório de estoques, de contas a receber, etc.

Empresas e outras organizações têm seus próprios sistemas de banco de dados: arquivos de clientes, arquivos com informações pessoais, inventários, registros de vendas, tabelas, etc.

As escolas têm currículos dos alunos, listas de chamadas, arquivos de funcionários, relatórios de frequência, etc.

Todos esses bancos de dados possibilitam o bom funcionamento de uma empresa ou organização.

Lidar com grandes quantidades de dados é uma necessidade constante em nossa sociedade moderna.

Poder acessar estes dados rapidamente, e deles extrair as informações desejadas, é bastante importante. Contudo, o mito que se criou em torno da figura do computador pode levar leigos a pensar que rapidez é uma característica inerente ao próprio computador.

Isto em parte é verdade. No entanto, se não dispusermos de um sistema eficiente e bem estruturado, no sentido de melhor manipular esta massa de dados, nossa rapidez ficará comprometida.

Eis aí um dos motivos pelos quais o dBase II surge como uma arma poderosíssima no tratamento de dados.

Antes de entrar propriamente no dBase II, vejamos um pouco de sua história.

A história do dBase II leva-nos até meados da década de 60 e a um sistema gerenciador de informações chamado RETRIEVE, que foi comercializado pela Tymshare Corporation.

O Jet Propulsion Laboratory, em Pasadena, Califórnia, usou o RETRIEVE até o final da década de 60, quando adquiriu os computadores UNIVAC 1108.

Long, um novo programador no JPL, recebeu a incumbência de escrever um programa que pudesse executar as mesmas funções que o RETRIEVE. O novo sistema gerenciador de arquivos, que Long desenvolveu, foi chamado de JPLDIS. Esse sistema continuou a evoluir nos vários anos seguintes e ainda é usado em muitos dos computadores do tipo UNIVAC 1100.

No final da década de 70, Wayne Ratliff, que trabalhava no JPL, sob contrato, com Martin Marietta, começou a se interessar por microcomputadores. Ratliff era um dos amigos de Jeb e começou a desenvolver um sistema que, para o usuário, era muito semelhante ao JPLDIS.

Conta a história que Wayne desenvolveu a versão inicial de seu sistema de banco de dados com o objetivo de ganhar apostas de futebol! Wayne comercializou essa versão JPLDIS sob o nome de VULCAN.

Embora tenha se desenvolvido rapidamente, ao ponto de adquirir seu próprio caráter e personalidade, o VULCAN não conseguiu muitas vendas. Em 1980 possuía somente sessenta clientes.

Então, um empresário, George Tate, entrou em cena, após ver um anúncio do VULCAN em uma publicação para computadores. Logo lhe deu um novo nome - dBase II - e uma nova companhia, a Ashton-Tate, que foi formada para comercializá-lo.

Desde o seu lançamento, diversas versões se sucederam, aprimorando o produto, implementando novas funções (2.3, A, B, C, D, 2.4, 2.41 e 2.42).

No Brasil, chegou oficialmente através da Datalógica, em 1982, sendo utilizado em computadores com processadores 8080, 8085 e Z-80, sob o sistema operacional CP/M.

No final de 1985, foi lançada uma nova versão, a 3.0, totalmente produzida no Brasil, o dBase II Plus.

Com a chegada do dBase III e III Plus, para micros de 16 bits, e o paulatino desuso de algumas linhas de micros de 8 bits, o dBase II estava em vias de esquecimento...

Mas, de repente, uma nova fase para os micros de 8 bits foi detectada e a nova arrancada se sucedeu com a chegada do padrão MSX ao Brasil.

Pensar no uso meramente doméstico para um computador é utopia, principalmente no Brasil, onde os custos são elevados.

Assim, o MSX foi se posicionando e, graças as suas qualidades, encontrou um lugar no meio profissional ou, ainda como alguns preferem, profissional leve.

Desta forma, surge, novamente, vitorioso, o dBase II, agora pelas mãos da Princessware e na sua mais nova versão, dirigida a este padrão: dBase II Plus MSX - v 1.0.

Denomina-se 1.0, pois é a primeira versão mundial do dBase II para MSX. Ela foi implementada a partir da versão 3.0 dos Apple's, PC's e TRS's, mas funcionando sob o sistema operacional MSX-DOS e, ainda, permitindo que se escreva em bom português, com todos os acentos.

O QUE É DBASE II?

Chamar o dBASE II de uma linguagem de programação e nada mais, é ter uma visão restrita, injusta, tanto para o dBASE como para seu novo usuário.

Na verdade, ele também é um instrumento para gerenciar banco de dados. Consistindo de uma bem elaborada linguagem de programação interpretada, dirigida especificamente para trabalhar com banco de dados com uma série de facilidades para se criar, manipular ou extrair informações concernentes a este banco de dados. Tudo isto cuidadosamente arrumado em um único pacote perfeitamente acessível, tanto para programadores como para os menos experientes em computação.

Há três diferentes maneiras de se encontrar e retirar informações de uma lista de dados. A primeira delas seria ir lendo os dados de uma forma sequencial, até encontrarmos o que queríamos.

Fica claro que, quanto maior for a lista de itens, mais demorada ficará esta busca, sendo, portanto, pouco prática. Outro jeito seria ordenar estes itens antes de iniciar a procura. A desvantagem está no fato de, toda a vez que acrescentarmos novos itens, a ordenação terá de ser refeita.

Muito mais eficiente é a utilização de arquivos indexados. Neste caso, além do banco de dados original, guardamos um outro com indicadores (pointers) que apontam para os dados.

Quando algum item é alterado, o arquivo com os "pointers" é corrigido, de modo que temos um ordenação efetiva sem, contudo, ter havido qualquer movimento no banco de dados original.

A busca e obtenção de informações poderá, então, ser feita por diversas técnicas, algumas até bastante complexas e que, felizmente, não precisam ser conhecidas pelo usuário do dBASE II, pois já estão embutidas no pacote, podendo ser utilizadas através de comandos e instruções facilmente assimiláveis.

Vale ainda ressaltar que no dBASE II estes índices não estão organizados de uma forma hierárquica, mas, como temos um banco de dados do tipo relacional, haverá necessidade de um único indicador para cada ítem, já que todas as informações relacionadas àquele ítem estão estreitamente ligadas num só registro. Inclusive, é para o número deste registro, que é único para cada item no dBASE II, que apontam os indicadores do arquivo de índices.

O usuário pode não apenas gerenciar um banco de dados com o dBASE II mas, se ele desejar, pode criar outros também. Para isto basta recorrer ao comando CREATE e o computador inicia com o usuário um diálogo a respeito de como ele deseja a estrutura de armazenamento e o tipo dos dados a serem guardados.

Há apenas três tipos de dados, a saber: caractere, numérico e lógico. Num tempo bastante curto você verá que pode aprender a descrever os campos de um novo banco de dados.

Aprendendo a utilizar as expressões adequadas, o novo usuário do dBASE II perceberá, facilmente, que ele pode acessar informações bastante específicas, sem necessidade de ter que descobri-las em meio a um amontoado de dados.

Assim, se, por exemplo, um banco de dados contiver os resultados das vendas anuais de um companhia, você poderá requerer o total de vendas de um único vendedor, num período de tempo específico, em uma determinada região. Mais do que conseguir retirar estas informações rapidamente com o dBASE II, você perceberá que não há a menor necessidade de conhecer programação para obtê-las, bastando para isto digitar uma ou duas linhas de comandos fáceis de se aprender.

Como podemos perceber, o dBASE II vem ao encontro das necessidades básicas de qualquer indústria ou organização, que é o serviço ou produto por ela gerado e as tarefas envolvidas na manipulação dos novos dados que surgem ao longo do ciclo operacional.

Uma vez criada a estrutura do banco de dados, novos

registros poderão ser acrescentados livremente, fazendo uso das incríveis capacidades de edição do dBASE II.

Tirar relatórios rápidos com o dBASE II é uma tranquilidade. Mais uma vez temos o próprio computador dialogando conosco e pedindo todas as informações necessárias para a confecção do relatório. É perguntado sobre o cabeçalho, sobre os subtotais, os títulos dos campos, etc. No caso do nosso primeiro exemplo, poderíamos tirar um relatório em que aparecessem, explicitamente, os totais de vendas por vendedor, por região, ou ainda pelos dois, conforme nosso interesse. Para isto, basta teclar 'REPORT' e o mais é se entender com a máquina.

Copiar arquivos por inteiro ou em parte é outra das facilidades permitidas pelo dBASE II. A maioria dos seus comandos permite uma filtragem de dados, de modo a operarem sobre ítems que satisfaçam certas características. Há, além disto, o "escape", escopo ou comando, que explica sobre que registros deve ser testada a condição de filtragem.

Vamos dar um exemplo mais específico. Imagine a lista telefônica da cidade de São Paulo sendo um banco de dados. É possível copiar para outro banco de dados apenas os assinantes que se situam na rua Augusta, cujas iniciais são "RO" e cujos telefones terminem em "32". Através do escopo, poderíamos procurar as condições acima em toda a rua Augusta, ou apenas nos primeiros registros, ou nos últimos, como quisermos.

Acho que já dá para avaliar o incrível potencial de nossa linguagem.

Até agora, falamos apenas, de características do dBASE II completamente independentes de qualquer conhecimento prévio de programação. Isto por si só já justificaria seu uso, mas vamos destacar outra de suas potencialidades que é a de criar procedimentos para serem utilizados por terceiros, através de programação.

Para isto, o dBASE II dispõe de um interpretador próprio capaz de entender as construções fundamentais em programação, como o comando interativo "do while", o condicional "if-else" e outros mais refinados como o "do-case", por exemplo.

Deste modo, o programador pode escrever "menus", procedimentos bastante explicativos para o usuário, ampliando o diálogo entre o computador e o homem, insistindo na correção dos dados digitados, fazendo observações sobre a manutenção de arquivos, enfim, tudo que um programador imaginativo pode "boalar" no sentido de reduzir ainda mais a possibilidade de erros na manipulação de dados.

Deste modo, é possível fazermos programas bastante inteligentes e eficientes para resolver problemas típicos, como folha de pagamento, sistema de contabilidade, controle financeiro, controle de estoque e outros, exclusivamente em dBASE II. Se sua empresa leva cerca de um mês para resolver estes problemas tão simples, é porque você está precisando de um dBASE II para, através da entrada de dados em poucas horas, rodar, em alguns minutos, os programas que lhe tirarão esta dor de cabeça.

SOFTWARE ORIGINAL

O usuário de um produto original, ao comprá-lo, não leva apenas um manual e o disco com o seu respectivo número de série. Leva, também, um serviço permanente e gratuito de suporte técnico e de atualização de versão, ou seja, um serviço profissional para profissionais.

O usuário do dBASE II, tanto da versão descrita neste artigo, como de qualquer outra, que não possua o seu número de série, deve saber que ela foi originada de um roubo e, como tal, é um crime, e quem a possui é um interceptador. Os responsáveis poderão ser indiciados criminalmente e responder a processo judicial, por utilização indevida do produto, de acordo com a Lei 7646 de 18/12/87 aprovada pelo Congresso Nacional e regulamentada sob o número 96036 em 12/05/88.

CURSO DE PASCAL -II

ANTONIO F. SHALDERS

Nesta segunda lição, iremos estudar as áreas de definição de procedures e funções e a do programa principal. Serão apresentadas as primeiras noções sobre loops em Pascal.

OS PROCEDIMENTOS

São pequenas subrotinas que podem ser acessadas através de seu nome, usado como uma palavra nova no Pascal. Um procedimento ou procedure tem como características estruturais a presença de um nome identificador, uma eventual área de variáveis (variáveis locais) e a área de definição do procedimento em si. Este tipo de estrutura é equivalente a um bloco do tipo GOSUB do BASIC. Abaixo, é mostrado um pequeno exemplo:

```
PROCEDURE ZERAR_MATRIZ ;
  VAR A : INTEGER ;
  BEGIN
    FOR A := 0 TO 100 DO
      S[ A ] := 0 ;
  END;
```

A função deste procedure é fazer com que a matriz S, composta de 101 elementos, seja zerada. Em Pascal é usual uma declaração terminar com um ponto e vírgula. Toda vez que digitarmos o nome do procedure dentro do programa principal, a subrotina a ele ligada será executada. Note que a subrotina propriamente dita é posta entre um BEGIN e um END;. Estes são os delimitadores de início e fim de bloco de operações em Pascal.

AS FUNÇÕES

Podem ser definidas novas funções (matemáticas ou não) em Pascal. Seu uso é muito parecido com o de procedures, porém pode ter apenas uma saída, ao passo que os procedures podem ter várias. Exemplos:

```
FUNCTION TAN(X:REAL):REAL;
BEGIN
  TAN:=SIN(X)/COS(X);
END;
```

Neste outro exemplo, definimos a função tangente, inexistente em Pascal. Feita esta definição, podemos usá-la como qualquer outra função residente no Pascal.

O REAL dentro dos parênteses é para que o argumento seja tratado como real, e o seguinte para que o resultado obtido seja real.

Os procedures e functions são uma das principais características de uma linguagem estruturada, pois se bem utilizados, evitam o uso desnecessário de GOTOS, o que é apelação em Pascal.

LOOPS SIMPLES (FOR)

O exemplo mais simples de loop em Pascal é o do tipo FOR. É muito semelhante ao FOR do BASIC, porém aceita somente variáveis inteiras no contador. Pode ser tanto crescente quanto decrescente.

Uma característica do FOR é que não existe o NEXT, como em BASIC, sendo o controle feito através de BEGINs e ENDs. Pode parecer complicado, mas na verdade é muito simples:

```
BASIC
FOR X=0 TO 255
PRINT CHR$(X);
NEXT X
PASCAL
FOR X:=0 TO 255 DO
  WRITE(CHR(X));
```

No caso anterior, apenas uma ação é executada dentro do loop.

Caso seja necessária a execução de mais de uma ação dentro do loop, estas deverão ser postas entre um BEGIN e um END; como no caso dos PROCEDURES:

```
BASIC
FOR X=0 TO 100
PRINT X
PRINT X^2
PRINT
NEXT X
PASCAL
FOR X:=0 TO 100 DO
  BEGIN
    WRITELN(X);
    WRITELN(SQR(X));
  END;
```

Note que a função SQR(X) é equivalente a X^2 do BASIC. A raiz quadrada em Pascal é definida por SQRT(X).

Não confunda! Algo MUITO IMPORTANTE em Pascal é em relação aos loops alinhados. A sintaxe para tal é mostrada no exemplo abaixo:

```
BASIC
FOR X=0 TO 100
PRINT X
FOR Y=75 TO 22 STEP -1
C:=C+X-Y
D:=D+C^2
NEXT Y
PRINT Y-D/C
PRINT
NEXT X
PRINT "FIM"
```

PASCAL

```
FOR X:=0 TO 100 DO
  BEGIN
    WRITELN(X);
    FOR Y:= 75 DOWNTO 22 DO
      BEGIN
        C:=C+X-Y;
        D:=D+SQR(C);
      END;
    WRITELN(Y-D/C);
    WRITELN;
  END;
  WRITELN('FIM');
```

Creio que, com este exemplo final, todas as eventuais dúvidas sobre este tipo de loop estejam sanadas. Convém lembrar que o ponto e vírgula no final de algumas linhas indica o terminador de comandos. Examine o programa que acompanha esta segunda lição do curso. Nele está contida toda esta lição, sendo sua análise um ótimo exercício.

Os comandos de impressão no vídeo são WRITE e WRITELN, equivalentes ao PRINT; e PRINT, respectivamente. O GOTOXY(a,b) é análogo ao LOCATE a,b do BASIC, e o CLRSCR ao CLS.

Chegamos ao final da segunda lição, e, em caso de alguma eventual dúvida, não hesitem em escrever-nos. Afinal, nós estamos aqui para isso!

Na próxima lição serão apresentados os loops dos tipos WHILE e REPEAT, muito poderosos por sinal.

program curso_2a ;

{este programa demonstra o uso de loops
desenhando uma moldura na tela e es-
crevendo uma mensagem em seu centro.}

{área de definição de variáveis}

```
var a,b : integer;           {define A e B como inteiros}
c : char;                   {define C como alfanumérica}
```

{área de definição
de procedimentos}

{imprime 40 caracteres
255 sequencialmente }

```
procedure linha_cheia;      {identificação do procedure}
begin                         {inicializa o procedure}
  for a := 1 to 40 do        {define o loop}
    write(c);                {imprime C dentro do loop}
  end;                        {finaliza o procedure}
```

{imprime as margens}

```
procedure margens;           {identificação do procedure}
begin                         {inicializa o procedure}
  for a := 1 to 21 do        {define o loop principal}
    begin                     {inicializa o loop principal}
      write(c);              {imprime C}
      for b := 2 to 39 do    {loop secundário para}
        write(' ');           {imprimir 38 espaços}
      write(c);              {imprime C fora do loop secundário}
    end;                      {finaliza o loop principal}
  end;                        {finaliza o procedure}
```

{início do programa principal}

```
begin                         {inicializa o bloco principal}
  c := chr(255);             {define a variável C como o
                           caractere 255}
  clrscr;                    {limpa a tela}
  linha_cheia;               {chama o procedure}
  margens;
  linha_cheia;
  gotoxy(14,10);             {põe o cursor em 14,10}
  write('CURSO DE PASCAL');  {imprime a mensagem
                           em 14,10}
  gotoxy(17,12);
  write('LIÇÃO II');
  gotoxy(1,1);
  delay(6000);               {aguarda alguns segundos}
  clrscr;
```

end. {finaliza o bloco principal}

program equação_do_segundo_grau ;

{declaração das variáveis}

var

```
a,b,c,          {coeficientes da equação}
discr,          {discriminante}
raiz_1,raiz_2,   {raízes}
vertice_x,vertice_y : real; {coordenadas do vértice da pa-  
rabola}
```

```
begin
  writeln('Raiz 1 = ',(-b/(2*a)):5:5,' ,sqrt(-
discr):5:5,' i');
  writeln('Raiz 2 = ',(-b/(2*a)):5:5,' +', sqrt(-
discr):5:5,' i');
  writeln
end;
```

{calculo das coordenadas do vértice da parábola}

{inicialização e entrada de dados}

```
procedure inicio;
begin
  clrscr;
  writeln('Equação do 2º grau');
  writeln('Curso de Pascal - II');
  writeln('Revista CPU');
  writeln('_____');
```

```
gotoxy(1,10);
writeln('Entre com os coeficientes da equação da forma
Ax²+Bx+C=0');
writeln;
write('A: ? '); readln(a);
write('B: ? '); readln(b);
write('C: ? '); readln(c);
end;
```

```
{programa principal}
begin
  inicio;
  discr := delta(a,b,c);
  if discr > 0 then raizes_reais
  else
    if discr = 0 then raizes_reais_iguais
    else
      if discr < 0 then raizes_complexas ;
  writeln;
  writeln('Δ = ',discr);
  writeln;
  vertice;
```

end.

{calculo das raízes reais e distintas}

```
procedure raizes_reais;
begin
  raiz_1 := (-b-sqrt(discr))/(2*a) ;
  raiz_2 := (-b+sqrt(discr))/(2*a) ;
  writeln('Raiz 1 = ',raiz_1:5:5);
  writeln('Raiz 2 = ',raiz_2:5:5);
end;
```

{calculo das raízes reais e iguais}

```
procedure raizes_reais_iguais;
begin
  raiz_1 := (-b+sqrt(discr))/(2*a) ;
  writeln('Raiz 1 = Raiz 2 = ',raiz_1:5:5);
end;z
```

{calculo das raízes complexas}

procedure raizes_complexas;

LIVROS

ASTROLOGIA NO MSX

LUIS TARCISIO DE CARVALHO JR
EDITORIA ALEPH - 1988

Como não podia deixar de acontecer, o MSX entrou na área mística! De fato, uma das grandes dificuldades dos astrólogos sempre foi um exaustivo trabalho de cálculos e consultas a tabelas de efeméides planetárias para levantamento do Mapa Astral.

Com a incrível capacidade e precisão de cálculos do MSX, esta tarefa pode ser relegada ao micro, ficando ao astrólogo a tarefa mais nobre da interpretação. Neste lançamento da Editora Aleph, o leitor passa a dispor de uma espécie de "kit" de Astrologia, dividido da seguinte forma:

Na parte I é fornecida uma listagem em Basic do programa MAPASTRAL, que se encarrega de fazer todos os cálculos astronômicos necessários à interpretação. Para os leitores "preguiçosos", a Aleph comercializa uma versão do livro associada a um disquete de 5 1/4", com todos os programas nele contidos.

Ao rodar o programa são solicitadas as informações pessoais do "consulente": nome, data e hora do nascimento. O programa pergunta se nesta data estava em vigor o horário de verão, sendo que no apêndice II do livro está uma tabela de todos os horários de verão que já vigoraram no Brasil, com espaço para complementações futuras.

A seguir devem ser colocadas as coordenadas geográficas do local do nascimento. No apêndice I são fornecidas as das principais cidades brasileiras. Se a sua cidade não estiver relacionada, uma boa fonte de consulta é o índice do Atlas da Encyclopaedia Britannica.

Após a entrada destes dados (figura 1), o micro efetua todos os cálculos e desenha o Mapa Astral na tela. No apêndice III é fornecido um programa que permite tirar uma cópia desta tela (e das seguintes) na impressora, para documentação posterior.

*** MAPA ASTRAL ***

FIGURA 1 Nome (máx=10 letras):?

Data de nascimento:
(após 15/10/1582) Dia ? 6
Mes ? 3
Ano ? 1967

Hora de nascimento: Hora ? 15
Minutos ? 30

Horário de Verão (S/N) ? n

Fuso Horário ? 3

Coordenadas do local de nascimento

Latitude: Graus ? 23
Minutos ? 33

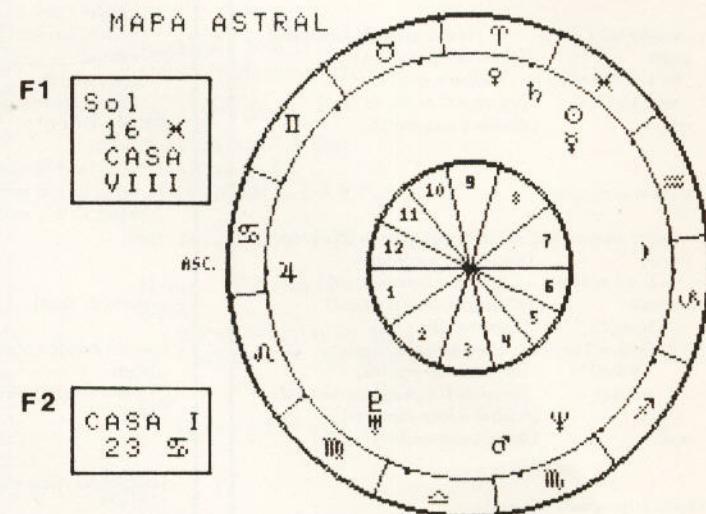
N ou S ? s

Longitude: Graus ? 46
Minutos ? 38

L ou 0 ? 0

Apertando-se a tecla F1, o programa lista a posição do Sol, Lua e Planetas e, com F2, a posição das casas (figura 2).

FIGURA 2



Na tela seguinte é fornecida a tabela dos aspectos planetários (figura 3) e, nas duas telas seguintes (figura 4 e 5), as posições dos planetas e casas é listada com toda precisão.

FIGURA 3

ASPECTOS PLANETARIOS

○	σ'							σ'			
)		□	σ'	*	Δ	*	Δ	σ'			*
♀											
	σ'										σ'
		4	Δ	*	Δ	*	σ'				

* = Sextil
Δ = Trígono
σ' = Oposição
σ = Conjunção
□ = Quadratura

FIGURA 4 *** MAPA ASTRAL ***

POSIÇÕES DOS PLANETAS

PLANETAS	LONGITUDE	SIGNO	CASA
Sol	15:30:23	Peixes	VIII
Lua	26:22:18	Capricórnio	VII
Mercúrio	10:31:18	Peixes	VIII
Vênus	13:21:17	Aries	IX
Marte	3:10:45	Escorpião	III
Júpiter	24:46: 3	Câncer	I
Saturno	0:21:21	Aries	VIII
Urano	22:41: 2	Virgem	II
Netuno	24:20: 1	Escorpião	IV
Plutão	19:26:43	Virgem	II

CASA DE CADA PLANETA:

- 140- Sol na casa VIII
- 151- Lua na casa VII
- 164- Mercúrio na casa VIII
- 177- Vênus na casa IX
- 183- Marte na casa III
- 193- Júpiter na casa I
- 212- Saturno na casa VIII
- 218- Urano na casa II
- 232- Netuno na casa IV
- 242- Plutão na casa II

ASPECTOS PLANETARIOS:

- 256- Sol/Mercúrio: Conjuncção
- 279- Sol/Plutão: Oposição
- 294- Lua/Marte: Quadratura
- 297- Lua/Júpiter: Oposição
- 299- Lua/Saturno: Séttil
- 302- Lua/Urano: Trígono
- 305- Lua/Netuno: Séttil
- 308- Lua/Plutão: Trígono
- 312- Lua/Ascendente: Oposição
- 341- Mercúrio/Meio do Céu: Séttil
- 387- Marte/Meio do Céu: Oposição
- 389- Júpiter/Saturno: Trígono
- 392- Júpiter/Urano: Séttil
- 395- Júpiter/Netuno: Trígono
- 398- Júpiter/Plutão: Séttil
- 400- Júpiter/Ascendente: Conjuncção
- 410- Saturno/Netuno: Trígono
- 422- Urano/Netuno: Séttil
- 424- Urano/Plutão: Conjuncção
- 428- Urano/Ascendente: Séttil
- 434- Netuno/Plutão: Séttil
- 437- Netuno/Ascendente: Trígono
- 443- Plutão/Ascendente: Séttil

FIGURA 5 *** MAPA ASTRAL ***

POSIÇÕES DAS CASAS

CASA COSSPIDE	LONGITUDE	SIGNO
I Ascendente	22:57:41	Câncer
II	29: 3: 5	Leão
III	5:25:36	Libra
IV I.C.	7: 0:36	Escorpião
V	3:30: 9	Sagitário
VI	27:37:32	Sagitário
VII Descendente	22:57:41	Capricórnio
VIII	29: 3: 5	Aquário
IX	5:25:36	Aries
X Meio do Céu	7: 0:36	Touro
XI	3:30: 9	Gêmeos
XII	27:37:32	Gêmeos

A seguir vem a parte mais interessante do livro. Em função do signo de cada planeta, do ascendente, das casas dos planetas e dos aspectos planetários, o programa lista uma série de números, correspondendo cada um a um verbete fornecido na parte III do livro (interpretação). Desta maneira, mesmo o leigo em astrologia pode fazer uma primeira interpretação do mapa. Obviamente, ela será superficial, pois é necessário um astrólogo experiente para solucionar eventuais informações conflitantes entre os aspectos (figura 6).

Trata-se, sem dúvida, de uma obra original e inovadora que, além de mostrar uma faceta insuspeitada do MSX, permite ampliar seu campo de uso, especialmente entre o público feminino que olha o micro com uma certa indiferença, ou até hostilidade.

FIGURA 6 *** MAPA ASTRAL ***

- 12- Signo Solar: Peixes
- 16- Ascendente em Câncer
- Combinação: 12 - 16

SIGNO DE CADA PLANETA:

- 34- Lua em Capricórnio
- 48- Mercúrio em Peixes
- 49- Vênus em Aries
- 68- Marte em Escorpião
- 76- Júpiter em Câncer
- 85- Saturno em Aries
- 102- Urano em Virgem
- 116- Netuno em Escorpião
- 126- Plutão em Virgem

Guia do Programador - Volume 1

Guia Técnico de Referência - Volume 2

Editora McGraw-Hill

A McGraw-Hill acaba de fazer um lançamento de dois livros que, ao nosso ver, em muito irão ajudar os programadores, iniciantes ou não, a usufruir ao máximo dos recursos dos computadores MSX.

A obra foi dividida em dois volumes, tendo a primeira recebido o título de "Programação Básico e Avançada", com 297 páginas e a segunda, com 346 páginas, recebeu o título de "Guia Técnico de Referência".

Tendo sido traduzida do livro The Complete MSX - Programmers Guide, que vem a ser uma das publicações mais conceituadas na Europa sobre o MSX, tendo sido traduzida, num trabalho muito bem feito, utilizando uma linguagem bastante acessível a todos os leitores.

Um dos fatores que tornam os livros de fácil utilização por parte dos leitores é a forma pela qual os comandos são apresentados, tendo sido agrupados por função. Assim, no capítulo 12, por exemplo, dedicado à leitura de dados em matrizes, encontramos as instruções DIM, READ, DATA e RESTORE e uma série de programas onde é exemplificado os seus usos.

Seguindo essa linha, temos 61 capítulos, que abrangem desde a organização do teclado (uso de RGRA, LGRA, ETC), até ao Sistema operacional e BIOS.

O Índice Analítico é outro ponto forte do livro e que permite uma consulta rápida e precisa.

Se quisermos, por exemplo, informações sobre o comando RUN, ao consultar o Índice Analítico, verificamos que ele foi mencionado nas páginas 15, 32 e 509.

CARTAS

Como já deve ser do conhecimento de V.Sas., a imagem gerada pelos computadores da linha MSX, particularmente no Hotbit, não é de boa qualidade. A fim de melhorar a qualidade da imagem obtida no meu Hotbit, solicito-lhes a seguinte informação:

Como obter uma saída RGB neste micro e como substituir o circuito modulador de RF por um de melhor qualidade? Se possível, gostaria de receber o desenho do esquema desta modificação.

Adquiri um monitor da marca Spectrum, do tipo usado em Apple e, quando o ligei nasaída de vídeo, não obtive nenhuma imagem. Depois de analisar e medir com osciloscópio o nível de sinal de vídeo, verifiquei que o mesmo era muito baixo.

Resolvi este problema montando um amplificador de vídeo, conseguindo, então uma imagem muito boa. Porém, quando rodo certos programas, que usam diversas cores, a imagem fica distorcida, como se perdesse o sincronismo. Ainda estou pesquisando como resolver este problema

Certo de que poderei contar com as informações solicitadas no menor prazo possível, subscrevo-mo.

Pedro Américo Sampaio Guimarães
Caixa Postal 98411
28500 - Cantagalo - RJ

No Hotbit existe uma pequena chave localizada perto da placa de identificação que serve para solicitar o tipo de monitor que se está utilizando.

Acreditamos que, ao efetuar a mudança da posição desta chave, você terá uma melhora sensível na qualidade de imagem do seu monitor.

Para podermos fornecer-lhe todos os dados necessários à solução do seu problema, encaminhamos a sua correspondência à Sharp, que, em tempo hábil, irá responder-lhe.

Encontrei em uma banca de jornal de minha cidade um exemplar da revista "CPU", editada pelos senhores e, como gostei muito da maneira como os assuntos foram tratados e percebendo ainda a possibilidade de me tornar assinante, solicito, se possível, que me enviem maiores esclarecimentos sobre o preço atual da assinatura e forma de pagamento.

A revista que encontrei é a de número 3 e gostaria que, junto com os esclarecimentos solicitados, fossem fornecidas instruções de como eu faria para adquirir também as de número 1 e 2, ou demais atrasadas, a fim de que eu não ficasse com matérias truncadas.

Possuo um micro Expert 1.1, já convertido pelo Ademir em 2.0, drive com interface DDX e impressora Lady-80, equipamento este que forneci ao meu filho Cesar de 12 anos para iniciação na área de informática, sendo que as matérias que abordam na revista de muito ajudam.

Otávio Alves Pereira
Rua Dr. Roque Barbosa Lima 108
Parque São Lucas
03264 - São Paulo - SP

As informações necessárias para efetuar a assinatura da revista podem ser encontradas no cupom próprio com esta finalidade, sendo o preço válido até a saída do novo número.

Com relação aos números 1 e 2 da revista, informamos que o número 2 está esgotado, devendo ser relançado no início do próximo ano. O número 1 foi relançado por nós este mês e em São Paulo poderá ser encontrado na Litec, ou solicitado diretamente a nós, ao custo de Cr\$ 660,00.

Os usuários de computadores pessoais são, por força do destino, uns pesquisadores e estão sempre à procura de fontes de informação. É com este espírito que sempre estou em bancas de jornais, ou livrarias, e, numa dessas investidas, encontrei a revista CPU que, até agora, está de parabéns pelo excelente nível dos artigos publicados e programas apresentados.

A idéia de apresentarem programas escritos em outras linguagens me agradou bastante pois é difícil encontrarmos este tipo de serviço em outras revistas da área.

Espero que continuem sempre assim, dando uma força aos programadores.

Antonio Pedro Rodrigues Silva
Caixa Postal 38080
22452 - Rio de Janeiro - RJ

Sou programador e encontro dificuldades em entrar em contato com outros programadores para troca de idéias e programas em qualquer linguagem.

Gostaria de corresponder-me com programadores de qualquer linguagem, a fim de juntos possamos iniciar um clube de programadores e que, assim, possamos nos auxiliar em problemas do "dia a dia".

Sandro Daniel Minetto de Carvalho
Caixa Postal 4117290 - Macatuba - SP

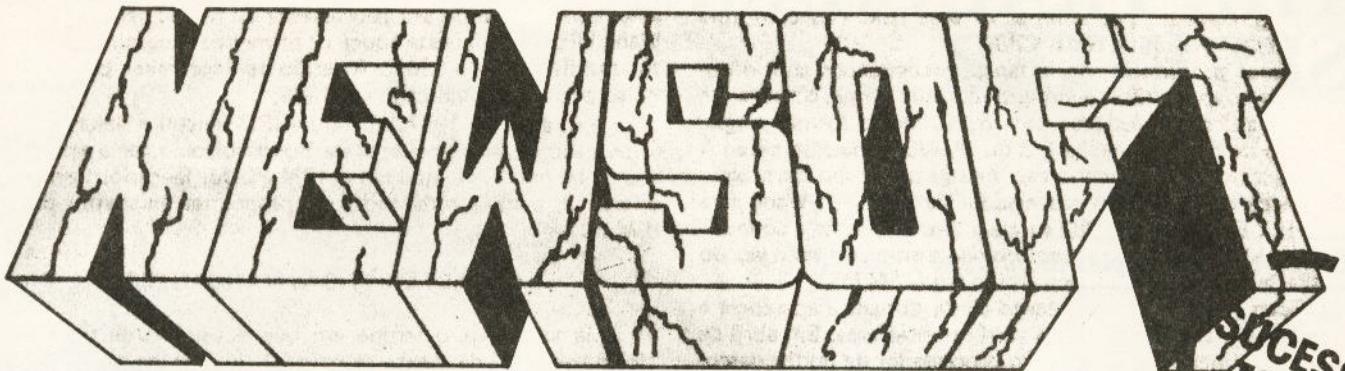
Quero parabenizá-los pela excelente qualidade da revista.

A meu ver, deveriam aumentar a parte relativa aos jogos, incrementando a seção de dicas de mil vidas e High Scores.

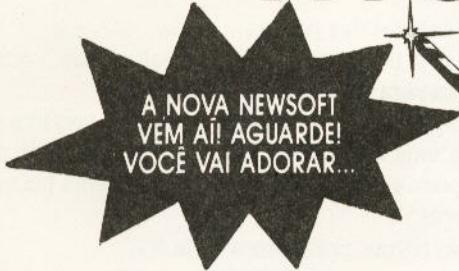
Achei muito bom e oportuno o artigo do MSX 2 por transformação. Espero que logo seja lançado também no Brasil. Também, com a situação econômica atual do país, não me admiro que os fabricantes ligados à área do MSX não estejam pensando em não fazer lançamentos no momento.

Se possível, gostaria de ver publicados na revista vários mapas de jogos que nos auxiliem a achar a saída.

Ana Maria Santiago



★ A FORÇA DO MSX ★



EXPANSOR PARA 4 SLOTS

Adote um expanson e multiplique a capacidade do seu micro com ele você pode conectar:
INTERFACE - DRIVE - PLACA DE 80 COLUNAS - EXPANSÕES DE MEMÓRIA - MODEMS - SOFTWARE - RESIDENTES E ATÉ JOGOS
Apenas 22 OTN'S

VIDEO STATION

Uma mini central para video cassette capaz de transmitir imagens num raio de 50 metros
Por apenas 10 OTN'S



MODEM COMPLETO

Permite acesso ao REN PAC - CIRANDÃO - VIDEO TEXTO E ARUANDA
Além de comunicação micro a micro
Só 30 OTN'S

DRIVE DDX - 5 1/4 FACE DUPLA

360 K FORMATADOS
O único com garantia mesmo.
Acompanha - INTERFACE - MANUAL E DISCO MESTRE
Apenas 60 OTN'S

PLACA PARA 80 COLUNAS

Com ela você sentirá que é bem mais fácil programar.
Explore ao máximo seu MSX.
Apenas 22 OTN'S

PAPAI NOEL EXISTE MESMO

6 jogos por apenas Cz\$ 4.000 (Disco ou Fita incluído)

ROMA - A CONQUISTA DO IMPÉRIO • SKATE DRAGON • CATCH THAT GIRL • KIMPO FIGHTER • MINDER • GRADIUS • DOBSHOW • STRIPGIRL II • DIZZY DICE • TAI-PAN • CONDE DE MONTE CRISTO • DOM QUIXOTE 1 • DOM QUIXOTE 2 • GROTTEN VON OBERON • BOGY'84 • MOUSER • CAPTAIN SEVILHA 1 • CAPTAIN SEVILHA 2 • BLACK BEARD • MAD MIX • CRAZY CARB • HUNDRA • ARKANOID REVENGE • PINBALL MAKER • ARKOS 1 • ARKOS 2 • SUPER STAR SOCCER • VENON STRIKES BACK • REX HARD • STREAKER • INDIANA JONES • CAR JAMBREE • SASA • OCEAN CONQUEROR • ANARROUTE • SQUASH 2 • EL MUNDO PERDIDO • MANES • JAST • EAGLE • BANANA JONES • ALBATROSS PHOENIX GOLF • AFTERIORDS • TURBO GIRL • ALEHOP • MATCH DAY 2 • NEW 21 • GAME OVER 1 • GAME OVER 2 • CAR FIGHTER • LEGEND OF KAGE • ZONE OUT • TONIGHT AT THE PUB • DINO SOURCERS • GENIUS • MOVIE PAC MAN • GLASS • DROIDS WHITE WITCH • CRIBBAGE • SIMON • TRIANGULANDO • 3D SQUASH • NUCLEAR BOWLS • SWING MAN • JUNGLE JIM • MOBILE SUIT GUN DAM • BOUNCE • TEMPTACION • TIME BOMB • HARD BOILED • MISSION RESGATE • HE MAN • ROMAN NO BOUKEN • POLICE ACADEMY 2 • EL CID • STAR DUST • PAPAI NOEL • COMBLOT • BOLDER DASH 2 • ULTRAMAN • TRIAL SKY • RAPIER MAN 2 • PANEL PANIC • CETUS • ANGLE BALL • CASTILO DE DRACULA • TEDOKU • FREDDY HARDEST 1 • FREDDY HARDEST 2 • ZAIDER OF PEGUS • SMALL JONES • UFO AZ • TT FACE • PEGASUS • KNIGHT LEON • HYPE • BATTLE CHOPPER • WONDER BOY • SPACE CAMP • KENDO • INDY 500 • ICE HOCKEY • GULKAVE • ALPINE SKY • JACK THE NIPPER 2 • GOODY • STAR BLAZER • SKY GALDO • NIGHT FLIGHT • SCARLETT 7 • SUPER SNACK • YAYAMARU • SPLASH • THE POLICE STORE • ACUSO • RAMBO 2 • MEMORY GAME • LAPTICK 2 • JETALF STRIKES BACK • EXTERMINATOR • APEMAN STRICK AGAIN • CHOPPER • MOONSWEeper • BOMULUS • THE LOST CROWN • BOUNCING BLOCK • BALLBLAZER • MILK RACE • ALIEN O RESGATE • QUINELA HIPICA • TARO • PENTAGRAM • MINI • LODE

RUNNER 1 • ICE WORLD • ICE KING • FIRST • INCA 1 • LA ABEJA SABIA • WORDS GAME • TRIDIMAN • STARBYTE • SEA KING • CABBAGE PATCH KIDS • NICK NEAKER • KNIGHT GHOST JUMP COASTER • EXCHANGER • COMET TAIL • AQUAPOLIS SOS • OTHETO 2 • SKYGAWK • O'MAC FARMER • DOMINOES • MOLE MOLE 2 • HOPPER • GODZILA • BMX REKENCROSS • ANTARES • SPY STORE • SAFARI X • PACHINCO • STRANGE LOOP • COSMIC ABSORBER • ROTORS • FINAL JUSTICE • MERLIN • SAILORES DELIGHT • RASTER SCAN • TANK BATALION • MARTIANOIDS • NONAMED • ACE OF ACES • CAN OF WORMZ • CUBERTH • WRANGLER • WOID RUNNER • EL MISTERIO DEL NILO • TRAFFIC • SLOT MACHINE 2 • STAR SEEKER • MONSTER FAIR • JUMP LAND • HIGH WAY ENCOUNTER • LEONARD • BOING BOING • INFERNAL MINER • HOWARD THE DUCK • COASTER RACE • BUBBLER • MIDNIGHT BROTHERS • PHANTIS 1 • PHANTIS 2 • EL

MAGO VOADOR 1 • EL MAGO VOADOR 2 • AMIDA • EWOB AND DANDELION • FRED AND BUBLOIDS • ROCKY O LUTADOR • CHICK FIGHTER • INVASION USA • DONKEY KONG NITENDO • ARQUIMEDES • SPARKIE • RISE OUT • POPCUM • DIG DUG • PINK CHASE • JET SET WILLIE 1 • ACROBATA • HELITANK • AUTOROUTE • GOLFE 5 • COBRA • DEFUSE • DEMAND ZAXXON • VESTRON • ALCAZAR • FIGHTING RIDER • COLT 36 • ONE ONE 1 • MOBILE PLANET • LAS TRES LUCES DE GLAUROING • BREAK OUT • THE WALL • OUTROYD • PAY LOAD • XETRA • BRIAN JACKS 2 • SPACE MAZE ATTACK • HANG ON • REAL TIME POLE • DIAMOND MINE 2 • SKOOTER • DEATH WISH 3 • TZR GRAND PRIX RIDER • RALLY X • LORICIELS RUNNER • EXODE 2 • ZEXXAS 2 • DUSTIN • ATTACK OF TOMATOES • THE PROTECTOR • ASTRO PLUMBER • FLY BOAT • WAR CHESS • M47 • TENSAI RABBIA • ZANAC 2 • MAHJONG • PICOT • PAIRS • BOXE KONAMI • COAST PINBALL • VIDEO DROME • ALPHAROID • ICICLE WORKS • BREAK IN • COLLOR BALL • INSPECTEUR Z • HEAD OVER HEELS • KRAKOUT • MR DO WILDRIDE • KING BALLOON • GRID TRAP • THE LIVING DAYLIGHT • KILLER STATION • EUROPEAN GAMES • NUTS & MILK • LEUCOCYTE • MINI GOLF • PUZZ BALL • HORROR EM AMITYVILLE • COLONY • COBA NOSTRA • SHOOTER • MSX BASEBALL • 10 TH FRAME • BMX SIMULATOR • WEST BANG BANG • LAST MISSION • PYROMANIA • BEACH HEAD • QBERT KONAMI • SCIENTIPEDE • SMASH OUT • DEFENDER CASTLE • EXPLODING ATOMS • GENOGRAMS • PANIC JUNCTION • ROBO FROG • ANTY • SCIENCE FICTION JONY • SPACE SHUTTLE • POLICE ACADEMY 1 • SHINWAR • JOHNNY COMOMOLO • THING BOUNCES BACK • COMMAND 2 • THE MEANING OF LIFE • TOP ROLLER • SURVIVOR • INDIANA NO BOUKEN • THE SPRINTER • BOOM • BATALHA NAVAL 2 • PEETAN • TERMINUS • JAKLE & WILDE • VENGANZA • BOL OF A ROBOT • POKER REAL • NABUM • WINTER GAMES 2 • MOON RIDER • LIVING STONES • ICE • CRUZADER • SPECIAL OPERATIONS • KICK IT • WINTER GAMES 1 • STAR WARS

PEDIDOS PARA OUTROS ESTADOS

ATRAVÉS DE CHEQUE NOMINAL A NEWSOFT INFORMÁTICA LTDA. - RUA SENADOR DANTAS, 117 SALA 736 - RIO DE JANEIRO - RJ CEP 20000 OU VALE POSTAL AGENCIA "ARCOS" COD. 522317

Seja qual for o seu pedido acresça a quantia de Cz\$ 1.000,00 Valor correspondente a disco ou fita de excelente qualidade comportam em média 6 programas.

PEDIDO MÍNIMO: 2.000,00

PEÇA SUPER LISTÃO - GRÁTIS

THE TRAIN GAME SPRINTER

Descubra os macetes deste trem

EQUIPE SILVA SOFT
RODRIGO
ZÉ MÁRIO
ROBERTO PINHÃO

Este jogo mostra a cabine de comando do trem, bem como a vista frontal para a via férrea. Depende de você viajar o máximo de milhas com este trem.

O registrador de milhas não funcionará caso você escolha uma velocidade menor que a permitida.

Você pode começar o jogo com a velocidade de 120 mph.

O Sprinter tem certas particularidades que o distinguem de todos os outros trens, sendo as mais importantes as seguintes:

- 1- Acionamento e parada com uma única alavanca;
- 2- Controle automático de velocidade por meio de botões;
- 3- Operação de abertura e fechamento das portas pelo maquinista.

Alavanca de açãoamento e parada

Você pode mover esta alavanca com as teclas do cursor (para a frente e para trás), em quatro posições:

1 - Alavanca toda para a frente: se foi previamente escolhida a velocidade, o trem começa a mover-se. Os freios estão livres e o indicador de freio está na oposição "off". O amperímetro mostra uma corrente elevada passando pelos motores. Se a alavanca está nesta posição e o trem está em movimento, você pode escolher outra velocidade.

2 - Posição neutra: os freios estão livres e os motores parados. Quando o trem está em movimento a velocidade cai lentamente. Nesta posição você pode escolher uma velocidade.

3 - Freios: os motores estão parados. O ar aciona o cilindro do freio e o manômetro indicará alta pressão.

4 - Alavanca toda para trás: como na posição 3, mas a pressão nos freios é maior.

Seletores de velocidade

Próximo do centro do painel de controle, você verá oito botões para controle da velocidade. Estão disponíveis as seguintes velocidades:

- 4 - 40 mph
- 6 - 60 mph
- 7 - 70 mph
- 8 - 80 mph
- 9 - 90 mph
- 0 - 100 mph
- 1 - 110 mph
- 2 - 120 mph

Para selecionar a velocidade, a alavanca de açãoamento e parada não devem estar na posição de parada.

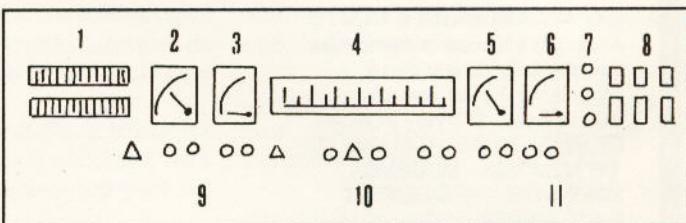
Operação central das portas

Acima dos oito botões de seleção de velocidade, você encontrará dois botões para operação das portas.

Antes de partir, você deverá fechar as portas, pressionando a tecla 'D'. Soará uma campainha para indicar que você pode partir. Quando as portas estão fechadas, acende-se uma luz verde no painel de controle.

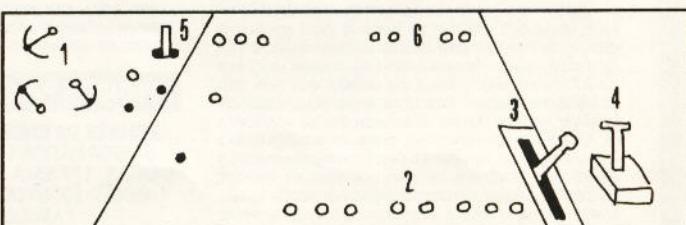
Para abrir as portas, pressione a tecla 'O'.

Painel indicador



- 1 - Controles de luz - aquecimento
- 2 - Voltímetro (1800 V)
- 3 - Amperímetro (300 - 500 A)
- 4 - Velocímetro
- 5 - Pressão no reservatório de ar
- 6 - Pressão nos cilindros de freio
- 7 - Chaves de inclinação
- 8 - Chaves de alta voltagem
- 9 - Luzes indicadoras (portas)
- 10 - Botões do CAT (Controle Automático do Trem)
- 11 - Luzes indicadoras dos freios

Painel de controle



- 1 - Chaves de barra
- 2 - Botões seletores de velocidade
- 3 - Alavanca de açãoamento e parada
- 4 - Pino de parafuso
- 5 - Chave de ignição
- 6 - Portas - limpadores de pára-brisa

Controle Automático do Trem

O equipamento de controle do trem é interligado aos sinais localizados ao lado da via férrea, os quais enviam um código ao trem, que, traduzido, informa o limite de velocidade para aquela parte do trajeto.

Esta informação é transmitida por sinais sonoros e luminosos.

verde - máximo 120 mph
verde + 8 - máximo 80 mph
verde + 6 - máximo 60 mph
amarelo + 8 - reduzir para 80 mph
amarelo + 6 - reduzir para 60 mph
amarelo - reduzir para 40 mph

Quando você passa um sinal que permite outra velocidade, soa uma campainha dentro da cabine. Se a sua velocidade é muito alta, soa outra campainha e você deverá acionar os freios imediatamente, caso contrário, o CAT parará o trem.

Quando a velocidade for reduzida suficientemente, a

campainha tocará de novo. Isto significa que você pode parar de frear, escolhendo novamente uma velocidade.

O CAT não reagirá a sinais vermelhos, prevenindo, apenas, a proximidade de um deles, por meio de um sinal amarelo - 4.

Segurança Extra

Quando a alavanca de acionamento está na posição neutra e o trem atinge uma velocidade abaixo de 40 mph, soa uma cigarra. Se o maquinista não pressionar o botão de saída (tecla 'Q'), o trem parará.

As estações

O Sprinter é um trem paradouro e, assim sendo, deve parar em todas as estações.

Você deve parar o trem o mais próximo possível do fim da plataforma, perto do sinal de parada.

Joystick

Frente/Trás - Alarme de acionamento/parada

Esquerda/Direita - Escolha de velocidade



SILVASOFT MSX PETROPOLIS MSX

Aqui você encontra:

Educativos

- Abelha Sábia
- Mago Voador
- Trigonometria
- Corpo Humano

Aplicativos

- Controle de Estoque
- Controle Bancário
- MSX Tools
- Editores de Gráfico/Som/Voz



E os melhores jogos:

Madmix—Black Beard—World Games—Taipan—Labadia del Creimen—Turbo Girl—Match Day II—Super Star Soccer—Alehop—Indiana Jones

Para adquirir estes programas basta escrever uma carta com nome e endereço legíveis pedindo os programas desejados e um cheque nominal e cruzado a Silvasoft Ltda. ou vale postal. Peça nosso listão grátis, ou venha ao nosso Show Room:

Rua do Imperador 518/302 CP 91.321 CEP 25600.

Peça nosso Jornal

O MUNDO PERDIDO DA III DIMENSÃO

GUILHERME A. L. DA SILVA

Numa sombria e assustadora noite de sexta-feira, 13 de agosto de 2004 — ano bissexto, para o seu azar — houve um acidente geodimensional, devido ao fortalecimento das forças malignas que se intensificam neste dia maldito e você foi “arrancado” da paz do seu lar para ser jogado num mundo de morte e crueldade.

Mas ainda há uma chance. Reprogramar o único teleportador dimensional existente no universo que se encontra, por coincidência, em algum lugar nessa dimensão inóspita. Reprograma-o, pois esta é a única chance para que você possa retornar ao nosso querido planeta.

DICAS

O jogo oferece as seguintes funções de ajuda:

Verbo ajudar: fornece todos os verbos existentes no jogo e, dependendo da situação, dá dicas de como proceder. Não há dicas falsas.

Arquivos: escrevendo esta palavra, o computador mostra na tela um menu contendo opções para a gravação e leitura de arquivos.

O mundo perdido da III dimensão é habitado por:

MUTANTES DO APOCALIPSE:

Eles aparecem para saquear ou, até, sanguinariamente, matar. São perigosíssimos.

FANÁTICOS RELIGIOSOS:

Oferecem perigos maiores que os mutantes. Todo cuidado com eles é pouco.

O jogo apresenta um nível de dificuldade razoável, que poderá ser alterado. Para que o jogo fique ainda mais difícil, delete do programa as linhas 331 a 337.

Caso queira dificultar um pouco mais, mude o limite de objetos a serem carregados por vez pelo aventureiro. Para isso, apague os comandos REM das linhas 843, 844 e 845.

O limite de objetos é dado pela variável NJ da linha 845 e o número de objetos é fornecido pela variável DO.

MSX PROFISSIONAL É NO MISC

O MISC lança o PACOTÃO GLOBAL PROFISSIONAL que consiste de: seleção e venda de equipamentos adequados, softs especificamente desenvolvidos para cada profissão, com as adaptações necessárias para cada caso particular, treinamento e assistência técnica. Estes são os Softs já disponíveis: ADVOCACIA, CONTABILIDADE, ENGENHARIA CIVIL. Conheça as atividades do MISC: 1^a TRANSFORMAÇÃO PARA VERSÃO 2.0: agora você pode transformar seu Hot-Bit ou Expert na segunda geração de MSX, rigorosamente dos padrões internacionais do MSX 2.0. 2^a CARTUCHO DE 256 Kb. PARA JOGOS MEGAROM: com ele e usando seu drive você tem acesso aos mais avançados jogos existentes, com mais ação, maior resolução e melhor música. 3^a SOFT-HOUSE: diariamente o MISC está ampliando seus títulos. Extensa coleção de jogos e aplicativos para 2.0 jogos megaram. 4^a SERVIÇOS: orientação na utilização e seleção de equipamentos e softs. 5^a COMPRA E VENDA de Expert e Hot-Bit. 6^a PRODUTOS: drives, impressoras, modems, interface para Praxis 20, capas para micros, drives e impressoras, livros especializados etc.

ASSOCIE-SE AO MISC. Usufrua das vantagens que proporcionamos aos nossos associados na aquisição de softs, produtos e serviços. O Jornal do MISC é uma exclusividade dos nossos associados. Pague uma taxa única de CzS 4.000,00 (não temos mensalidades) e receba **grátis** uma coleção de jogos em fita ou disco. Enviar cheque nominal à Embass Editora Ltda. ou depositar no BRADESCO na agência 0108, conta nº 141.184-5. **MSX INTERNATIONAL SERVICE CLUB** - A solução definitiva para o usuário MSX.



Peça o catálogo detalhado sobre o MISC

Rua Xavier de Toledo, 210 - Cj. 23 - CEP 01048 - São Paulo - SP - Fone: (011) 34-8391 e 36-3226.

```

10 REM ---- ADVENTURE
20 REM ---- O MUNDO PERDIDO
30 REM ---- PARA A LINHA MSX
40 REM ---- GUILHERME A.L. DA SILVA
50 REM ---- 22/06/88
60 REM ---- GUARARAPES
61 CLEAR 1000
62 KEY ON:DEFUSR=63E
63 STOPON:ON STOP GOSUB1040
64 POKE&HFCAB,1
70 KEY1,"PEGAR":KEY2,"DEIXAR":KEY3,"L
ISTAR":CHR$(13):KEY4,"AJUDAR":CHR$(13):
KEY5,"DESISTIR":CHR$(13):KEY7,"ARQUIVOS
":CHR$(13):KEY8,"PUXAR":KEY9,"JOGAR":
KEY10,"DEFENDER-SE":KEY6,"VESTIR"
80 COLOR 7,1:CLS:LOCATE0,3:PRINT#0 MU
NDO PERDIDO DA TERCEIRA DIMENSAO #":LOC
ATE 7,10:INPUT" CASSETE OU DISQUETE "
;C$:LOCATE 7,10:PRINT" Quer instrucoes
(S/N) ?";SPC(16):LOCATE0,17:PRINT"DESEN
VOLVIDO POR GUILHERME A.L.DA SILVA"
90 A$=INKEY$:IF A$="" THEN 98
100 IF A$="S" THEN GOSUB 2420
110 'PREPARAR MATRIZES DAS RESPOSTAS
120 DIM R$(24),R(24)
130 RESTORE140:FOR K=1TO24:READR$(K),R(
K):NEXT
140 DATA JOGAR,4,LIMPAR,5,LIGAR,6,PUXAR
,7,DESISTIR,8,LISTAR,9,DEFENDER-SE,10,A
JUDAR,11,REPROGRAMAR,12,VESTIR,13,VERIF
ICAR,14,ASSUSTAR,15
150 DATA PEGAR,2,DEIXAR,3,CHAMAR,16,INS
ERIR,17,JANELA,18,ARQUIVOS,19,LIVRAR-SE
,20,BEBER,21
160 DATA NORTE,1,SUL,1,LESTE,1,ESTE,1
170 REM PREPARA MATRIZES DE OBJETOS
180 READ NB
190 DIM OB(NB),OB$(NB),S$(NB)
200 FOR I=1 TO NB:READ OB(I),OB$(I),S$(I):
NEXTI
210 DATA 13,7,ELIXIR,Ha um vidro com um
elixir no chao
220 DATA 9,CORDA,Tem uma corda 5 polega
das perto de voce
230 DATA 3,BATERIA,Tem uma bateria de 1
2 voltos no sofa
240 DATA 15,ROUPA ANTI-RADIACAO,Ha uma
roupa anti-radiacao no chao
250 DATA 20,CONTADOR GEIGER,Ha um conta
dor geiger perto de voce
260 DATA 16,COMPUTADOR MSX - DESLIGADO,
Tem um Super computador nos seus pes
270 DATA 11,RUBI,Um RUBI enorme esta no
chao
280 DATA 0,URANIO,Ha uma magnifica jazi
da de uranio 238
290 DATA 0,ANEL DO MEDO,Nos seus pes e
sta o lendario anel do medo cravejado d
e brilhantes
300 DATA 0,ESPADA DE OURO,uma linda esp
ada de ouro que foi deixada pelos segui
dores!!!
310 DATA 19,TELEPORTADOR,Tem um telepor
tador dimensional no cantoda sala
320 DATA 0,MUTANTES,Um grupo de mutante
s aparece numa fumacavermelha
330 DATA 0,DISQUETE,Ha um disquete de c
omputador no bolso
331 RESTORE335:DIML$(24):FORI=1TO24:REA
DL$(I):NEXT
335 DATA DESFILADEIRO DO NUNCA,,CASA DO
ESPANTO,,,SALA DE FENIX,FLORESTA PETRI
FICADA,SAARA DIMENSIONAL
336 DATA ENCRUZILHADA,GRUTA DO TERROR,T
EMPO DA BRUTA,MONTANHAS ROCHOSAS,,,USI
NA NUCLEAR
337 DATA SAIDA LESTE,PONTE DE CORDA,CAC
HOEIRA DE ACIDO,SALA DO TELEPORTADOR,LA
BORATORIO,REATOR DA USINA,,,IDOLO DE PE
DRA
340 '** POSICAO INICIAL **
350 L=1
360 ' ACHAR LOCAL
370 IF FM>1 AND MU=2 THEN GOTO 2620
371 CLS:PRINT" O mundo perdido da 3
a dimensao "
380 COLOR 15,4
390 NT=NT+1:C=INT(RND(-TIME)*5):IF FM<
1 AND NT>25 AND C=4 THEN MU=1:OB(12)=L
391 IF FM>1 AND MU=2 THEN GOTO 2620
392 IF FM>1 AND MU=1 THEN MU=2
394 IF L<11 THEN ON L GOSUB180,,1840,,,
1880,1910,1940,1970,2000:GOTO 428
400 IF L<21 THEN ON L-10 GOSUB2040,2080
,,2120,2150,2180,2220,2260,2290:GOTO 4
28
410 IF L>26 THEN ON L-20 GOSUB2320,,,
23
60
420 'COLOCA CADA OBJETO EM LUGAR
430 FOR I=1TONB:IF OB(I)=L THEN PRINT S
$(I)
440 NEXT
450 ' APONTAR DIRECOES
460 '
470 PRINT"Podre seguir"
475 PRINT
480 IF N>8 THEN PRINT TAB(5);";NORTE: ";
L$(L-6)
490 IF S>0 THEN PRINT TAB(5);";SUL : ";
L$(L+6)
500 IF E>0 THEN PRINT TAB(5);";LESTE: ";
L$(L+1)
510 IF D>0 THEN PRINT TAB(5);";ESTE: ";
L$(L-1)
520 ' INSTRUICOES
530 PRINT:INPUT"E agora ";I$
540 GOSUB 1298
550 ' SELECCIONA OPCAO
560 IF I=0 THEN GOTO 580
570 ON I GOTO 600,770,880,950,1050,1110
,,1220,1000,700,2700,1300,1400,1500,1620
,,1670,1730,2500,370,3000,2900,3500
580 PRINT:PRINT"Eu nao sei como ";V$:GO
TO 430
590 ' ROTINA DE MOVIMENTO
600 IF I$="0" AND N>0 THEN L=L-1:GOTO 3

```

```

1020 A$=INKEY$:IF A$="" THEN 1020
1030 IF A$="S" THEN RUN
1040 KEYON:COLOR 15,1,1:CLS:A=USR(0):EN
D
1050 ' LIMPAR
1060 IN=INSTR("VEGETACAO",N$):IF IN<1 T
HEN PRINT"ISTO NAO DA'.";GOTO 390
1070 IF OB(8)=24 OR OB(8)=-1 THEN PRINT
"JA FOI FEITO .":GOTO 390
1080 IF L>24 THEN PRINT"IMPOSSIVEL !!!
?":GOTO 390
1081 IF OB(4)<-1 THEN PRINT"VOCE FOI C
ONTAMINADO PELA ALTISSIMA RADIACAO
DO URANIO 238.";N$=0:E=0:S=0:M=0:GOTO 39
0
1090 LV=1:OB(8)=24:GOTO 390
1100 ' LIGAR
1110 FOR G=1TO1B
1120 IF N$=LEFT$(OB$(G),LEN(N$)) THEN 1
150
1130 NEXT
1140 PRINT"?????":GOTO 390
1150 IF G<6 AND G>11 THEN PRINT"NAO D
A'.";GOTO 390
1160 IF OB(6)<-1 THEN PRINT"FALTA COMP
UTADOR.":GOTO 390
1165 IF G=6 AND OB(3)<-1 THEN PRINT"FA
LTA BATERIA":GOTO 390
1170 IF G=11 AND L>19 THEN PRINT"LUCA
R ERRADO.":GOTO 390
1180 IF G=11 AND RR=0 THEN PRINT"COMO V
OCE NAO REPROGRAMOU SE DANOU, POIS O TE
LEPORTADOR SOLTOU UMA TAMANHA DESCARGA
DE LASER EM VOCE..ESTAS MORTO!":GOTO 10
00
1185 IF G=11 AND L<>1 THEN PRINT"LIVRE
-SE DOS OBJETOS DESTE MUNDO.":GOTO 390
1190 IF G=11 AND RR=1 THEN GOTO 2390
1200 IF G=6 AND IS=0 THEN PRINT"FALTA I
NSERIR DISQUETE NO COMPUTADOR.":GOTO 39
0
1210 IF G=6 AND OB(3)=-1 AND IS=1 THEN
PRINT"OK.":LR=1:OB(3)=0:OB(13)=0:OB$(6)
=LEFT$(17,OB$(6))+";LIGADO    ";GOTO 390
1220 ' PUXAR
1230 IN=INSTR("CORDA",N$):IF IN<1 THEN
PRINT"IMPOSSIVEL - ";N$;"???":GOTO 390
1240 IF OB(2)<L THEN PRINT"AQUI NAO TE
M CORDA.":GOTO 390
1250 IF L>18 THEN PRINT"VOCE NAO PRECI
SA FAZER ISSO!":GOTO 390
1260 IF CJ=0 THEN PRINT"JA FOI FEITO"
1270 PRINT"OK.":CJ=0:OB(2)=-1:GOTO 390
1280 ' INSTRUCAO DE CHECAGEM
1290 I=0:N$="":I=INSTR(I$, " ")
1300 IF I=0 THEN N$="?????":V$=I$:GOTO
1330
1310 V$=LEFT$(I$,I-1)
1320 N$=MID$(I$,I+1)
1330 I=0
1340 FOR K=1TO24
1350 IF INSTR(R$(K),V$)=1 THEN I=R(K):I$=LEFT$(V$,1)
1360 NEXT
1370 RETURN
1380 ' AJUDAR
1390 PRINT:PRINT"SEUS VERBOS.":FOR U=1
TO20:IF U>20 THEN PRINTR$(U):PRINT:NEXT
:GOTO 1400 ELSE PRINTR$(U);", ";:NEXT
1400 IF L=9 AND BR>1 THEN PRINT" Beba
o elixir da coragem e da vida.":GOTO 3
90
1410 IF L=16 THEN PRINT" Nao passe pel
a ponte com o computador..":GOTO 390
1420 IF L=10 OR L=12 THEN PRINT"Assuste
os seguidores impiedosos.":PRINT:GOTO
390
1430 IF L=24 AND OB(5)<-1 THEN PRINT"P
egue o contador geiger.":PRINT:GOTO390
1440 IF L=6 THEN PRINT"Chame Fenix...":P
RINT:GOTO 390
1450 IF L=19 THEN PRINT"Para ligar o te
leportador procure: disquete,computador
,bateria,rubi,uranio 238.":PRINT:GOTO 3
90
1455 IF OB(4)=-1 AND INT(RND(-TIME)$2)=
1 THEN PRINT"Verifique bolso da roupa a
nti-radiacao.":GOTO 390
1456 IF INT(RND(-TIME)$5)+1=3 THEN PRIN
T"Defenda-se com a espada ou assuste co
m o anel os mutantes.":GOTO 390
1457 IF INT(RND(-TIME)$2)=1 THEN PRINT"
Seu objetivo e reprogramar o teleportad
or para isso precisa de:p/ calculos, u
m COMPUTADOR ligado(c/ BATERIA E DISQUE
TE), um RUBI estabilizador de fotons, u
ma jazida de URANIO 238 para combustive
l.":PRINT:GOTO 390
1460 GOTO 390
1470 ' REPROGRAMAR
1480 IN=INSTR("TELEPORTADOR COM COMPUTA
DOR",N$):IF IN<1 THEN PRINT"NAO DA'.";G
OTO 390
1490 IF L>19 THEN PRINT"LUCHAR ERRADO.":G
OTO 390
1500 IF OB(4)<-1 THEN PRINT"FALTA COMP
UTADOR."
1510 IF OB(4)=-1 AND LR=0 THEN PRINT"CO
MPUTADOR DESLIGADO."
1520 IF OB(7)<-1 THEN PRINT"FALTA RUBI
ENERGETICO."
1530 IF OB(8)<-1 THEN PRINT"FALTA URAN
IO 238"
1535 IF IS=0 THEN PRINT"O DISQUETE NAO
ESTA INSERIDO NO COMPUTADOR..."
1540 IF IS=1 AND LR=1 AND OB(7)=-1 AND
OB(8)=-1 THEN PRINT"OK. Pronto para fun
cionar.":RR=1:S$(11)="Ha o teleportador
reprogramado para as cordenadas terraq
ueas na sala":GOTO 390
1550 GOTO 390
1560 ' VESTIR
1570 IF OB(4)<L THEN PRINT"NAO ESTA AB
UI.":GOTO 390

```

1790 ' DESCRIÇÃO DOS LOCAIS
 1800 ' LOCAL 1
 1810 PRINT" Voce esta num desfiladeiro sem fundo, o DESFILADEIRO DO NUNCA..."
 1820 N=0:E=0:S=1:W=0:RETURN
 1830 ' LOCAL 3
 1840 PRINT" Voce esta na Mansao Mal-As sombrada."
 1850 IF BR<1 THEN PRINT" Os espiritos da Mansao lhe deram um susto e voce se arreou de medo...":GOTO 1000
 1860 N=0:E=0:S=1:W=0:RETURN
 1870 ' LOCAL 6
 1880 PRINT" Aqui e a Sala Encantada de Fenix - a ave sagrada."
 1890 N=0:E=0:S=1:W=0:RETURN
 1900 ' LOCAL 7
 1910 PRINT" Voce esta na Floresta Petrificada."
 1920 N=1:E=1:S=0:W=0:RETURN
 1930 ' LOCAL 8
 1940 PRINT" Voce esta no maligno Deserto do Saara Dimensional."
 1950 N=0:E=1:S=0:W=1:RETURN
 1960 ' LOCAL 9
 1970 PRINT" Aqui e a encruzilhada dos Indecisos."
 1980 N=1:E=1:S=1:W=1:RETURN
 1990 ' LOCAL 10
 2000 PRINT" Tu estas na fantasmagorica e lendaria, GRUTA DO HORROR!!!"
 2010 IF (OB(10)=-1 AND AU<1) OR (OB(10)<-1 AND AU>1) THEN PRINT" Os seguidores de THANHANDACUCA, estao vindo te pegar. Faca alguma coisa!"
 2020 N=0:E=1:S=0:W=1:RETURN
 2030 ' LOCAL 11
 2040 PRINT" Tu estas no templo dos seguidores de THANHANDACUCA, os impiedos os..."
 2050 IF AU>1 THEN PRINT" Os impiedosos e carniceiros seguidores de THANHANDACUCA te embalsamaram num sarcofago para sempre!!!!":GOTO 1000
 2060 N=0:E=1:S=0:W=1:RETURN
 2070 ' LOCAL 12
 2080 PRINT" Voce saiu da Gruta e agora esta numa montanha rochosa."
 2090 IF (OB(10)=-1 AND AU<1) OR (OB(10)<-1 AND AU>1) THEN PRINT" Mas os seguidores de THANHANDACUCA, estao vindo te pegar. Faca alguma coisa!"
 2100 N=1:E=0:S=1:W=1:RETURN
 2110 ' LOCAL 15
 2120 PRINT" Voce acaba de entrar na grande e potente USINA NUCLEAR DE TANGA DOS REIS..."
 2130 N=1:E=1:W=0:S=1:RETURN
 2140 ' LOCAL 16
 2150 PRINT" Voce esta na saida leste da usina."
 2160 N=0:E=1:W=1:S=0:RETURN
 2170 ' LOCAL 17
 2180 PRINT" Voce esta na margem de um g

rande vale, que e atravessado por uma ponte de corda."
 2190 IF OB(6)=-1 THEN PRINT" A ponte se partiu por excesso de peso do computador...":GOTO 1000
 2200 N=0:E=1:S=0:W=1:RETURN
 2210 ' LOCAL 18
 2220 PRINT" A sua frente ha uma cachoeira de acido sulfurico."
 2230 IF CJ<1 THEN PRINT" Voce nao pode atravessar a nado!!!":N=0:E=0:S=0:W=1:RETURN
 2240 N=1:E=0:S=1:W=1:RETURN
 2250 ' LOCAL 19
 2260 PRINT" Voce esta na sala do teleportador dimensional."
 2270 N=0:E=1:S=0:W=0:RETURN
 2280 ' LOCAL 20
 2290 PRINT" Voce esta no laboratorio da usina."
 2300 N=0:E=1:S=0:W=1:RETURN
 2310 ' LOCAL 21
 2320 PRINT" Tu estas no centro do reator termo-nuclear da usina."
 2330 IF OB(4)<-1 THEN PRINT" Como a radiatividade e muito grande e voce nao esta com a roupa certa, voce morreu contaminado.":GOTO 1000
 2340 N=1:E=0:S=0:W=1:RETURN
 2350 ' LOCAL 24
 2360 PRINT" Ha uma vegetacao muito estranha sobre um idolo."
 2370 IF OB(5)<-1 THEN N=1:E=0:S=0:W=0:RETURN
 2380 IF LV<1 THEN PRINT" O contador de raios esta denunciando alguma coisa na vegetacao!!!"
 2381 N=1:E=0:S=0:W=0:RETURN
 2390 ' VITORIA
 2400 CLS:COLOR 5,1:PRINTTAB(13);A Teleportacao"
 2401 PRINT:PRINT" O ";CHR\$(&H22);"T" foi acionado. . .":PRINT" Raios de laser de varias cores e espessuras percorrem a sala, no painel de controle as luzes piscam freneticamente, de repente um alarme toca!"
 2402 PRINT" O que sera ???"
 2403 PRINT:PRINT" E' a teleportacao que esta se processando. Um raio o atrevessa, voce nem sente e..."
 2404 PRINT" Quando voce abre os olhos, esta de volta ao seu mundo, a sua casa.":PRINT" Agora esta no seu quarto, ao redor, nao tem ninguem. Olha o relógio e:Sexta, 13 de Agosto de 2004 21:11:31 da noite!"
 2405 PRINT" Foi um sonho?!"
 2406 LOCATE9,21:PRINT"Pressione Qualquer Tecla":IF INKEY\$="" THEN 2486
 2407 CLS:PRINTTAB(13);A Teleportacao"
 2408 PRINT:PRINT:PRINT" Voce põe sua mao no bolso e sente alguma coisa de

estranho."
 2409 PRINT" Mas como, voce abandonou tudo ?:PRINT" Voce pega, olha."
 2410 PRINT" Um papel! Voce o abre. Uma Carta! Voce le esta."
 2412 PRINT:PRINT" Isto nao foi um sonho.":PRINT" Foi uma aventura e tanto!!!"
 2415 PRINTSPC(5);Eu voltarei, bravo aventureiro.":PRINTSPC(5);Eu voltarei...":PRINTSPC(5);Voltarei para novas e emocionantes":SPC(6);aventuras.":PRINTTAB(17);ASS: LORD TREVAS":PRINT
 2417 LOCATE0,9:FORK=107016:LOCATE0,K:PRINT" :LOCATE39,K:PRINT" :NEXT
 2418 LOCATE0,17
 2419 PRINT:GOTO 1000
 2420 LOCATE 0,22:COLOR 15,13:PRINT:PRINT
 2430 PRINT:PRINT" O mundo perdido da 3a dimensao":PRINT
 2440 PRINT" Houve um acidente geo-dimensioninal numa Sexta-feira, 13 de Agosto de 2004(bissexto) As 11:30 da noite."
 2441 PRINT" Devido ao fortalecimento da massa terrestre neste detestado dia, voce foi puxado para ";CHR\$(&H22);O Mundo Perdido da 3a Dimensao";CHR\$(&H22);".
 2450 PRINT" Voce deve reprogramar o unico tele portador existente no universo, para voltar a terra rever sua familia."
 2460 PRINT" Como esse mundo e muito hostil, cuidado! Voce enfrentara cachoeiras de acido, mansoes mal-assombradas, fanaticos religiosos e muito mais.. ."
 2470 PRINT:PRINT" Quando nao souber o que fazer escreva AJUDAR (em maiusculos)."
 2480 PRINT:PRINT:PRINT:PRINTTAB(9);Pressione Qualquer Tecla"
 2481 L\$=INKEY\$:IF L\$="" THEN 2481
 2482 LOCATE13,22:PRINT:PRINT"
 'CONTINUACAO':PRINT
 2483 PRINT" Como desgraca pouca e bobagem ha tambem um grupo maligno de mutantes do apocalipse, quando estiver na presencia deles digite DEFENDER-SE DOS MUTANTES (qdo estiver com a espada) ou ASSUSTAR MUTANTES(qdo estiver c/ o anel)."
 2485 PRINT" Se nao fizer isto voce sofrera na carne um ataque mutante devastador.":PRINT:PRINT" A opcao ARQUIVOS lhe da um menu com opcoes de LEITURA, GRAVAÇÃO, ETC para as-sim gravar o jogo em qualquer parte."
 2486 PRINT
 2487 PRINT" Esse adventure foi desenvolvido por: ";PRINT" GUILHERME ARAUJO LIMA DA SILVA (C)1987 ";
 2488 PRINT
 2489 PRINT:PRINT" Qualquer tecla pa

ra continuar"
 2498 L\$=INKEY\$:IF L\$<>"" THEN RETURN ELSE 2498
 2500 ' INSERIR
 2510 IN\$=INSTR("DISQUETE NO COMPUTADOR",
 N\$):IF IN\$>1 THEN PRINT"IMPOSSIVEL??":
 GOTO 398
 2520 IF OB(6)<-1 THEN PRINT"QUE COMPUTAD
 OR??"
 2530 IF OB(13)<-1 THEN PRINT"QUE DISQU
 ETE?"
 2535 IF IS=1 THEN PRINT"JA ESTA INSERID
 O...":GOTO 398
 2540 IF OB(13)=-1 AND OB(8)=-1 THEN PRI
 NT"OK...ESTA INSERIDO!":IS=1
 2550 GOTO 398
 2600 ' MUTANTES
 2620 IF MU\$1 THEN MU\$1=GOTO 398
 2622 IN\$=0:FOR I=1TONB:IF OB(I)=-1 THEN I
 N\$=IN\$+1
 2623 NEXT I
 2625 IF IN\$>0 THEN GOTO 2800
 2630 IF IN\$=0 THEN PRINT"OS MUTANTES TE
 MATARAM POIS VOCE NAOTINHA NADA PAR
 A SER ROUBADO.":GOTO 1000
 2680 GOTO 398
 2700 ' DEFENDER
 2701 IN\$=INSTR("DOS MUTANTES",N\$):IF IN\$
 >1 THEN PRINT"IMPOSSIVEL - ";N\$;"???":G
 OTO 398
 2710 IF OB(12)<0 THEN PRINT"DO QUE?":G
 OTO 398
 2720 IF OB(10)<-1 THEN PRINT"COM O QUE
 ?":GOTO 398
 2740 FM=1
 2741 PRINT"Voce e atacado por um mutant
 e...":PRINT"Voce o corta...ele geme e c
 ai!":FORV=1TO1000:NEXT
 2742 PRINT"Vem outro...Voce da um soco.
 .":PRINT"Ele rodopia e cai!":FORV=1TO1
 000:NEXT
 2743 PRINT"Vem o chefe com uma espada..
 .voce o ataca...ele defende e trope
 ca...":FORV=1TO2000:NEXT
 2745 PRINT"Voce o mata friamente,ele so
 ita um horripilante hurro e abraca
 a morte!!!":FORV=1TO500:NEXT
 2750 IF M=8 THEN OB(12)=8:MU=0:GOTO 398
 ELSE PRINT"Nos seus pes estao os obje
 os saqueados.":BEEP:BEEP:BEEP:BEEP
 2770 FOR I=1TON
 2771 OB(M(I))=L:NEXT
 2780 OB(12)=8:GOTO 398
 2800 ' ROTINA DOS MUTANTES
 2810 K=INT(RND(1)*NB)+1:IF OB(K)<-1 TH
 EN GOTO 2810
 2820 PRINT"OS MALDITOS MUTANTES":PRINTT
 AB(9); "ROUBARAM-LHE";OB\$ (K):M=M+1:OB(K
)=8:M(M)=K:MU=0:MT=0:OB(12)=0:GOTO 398
 2900 ' LIVRAR-SE
 2910 IN\$=INSTR("DOS OBJETOS",N\$):IF IN\$1
 THEN PRINT"LIVRAR-SE DO QUE????!!":GOT
 O 398

2915 BJ=0
 2916 IF LD=1 THEN PRINT"OUTRA VEZ?":GOT
 O 398
 2920 FOR I=1TONB:IF OB(I)=-1 THEN BJ=BJ+
 1
 2921 NEXT
 2925 IF RR>1 THEN PRINT"AGORA?":GOTO 3
 98
 2930 IF BJ=0 THEN PRINT "QUE OBJETOS??"
 :GOTO 398
 2940 FOR I=1TONB-1:IF I=6 OR I=7 OR I=8
 OR THEN NEXT ELSE OB(I)=0:NEXT:LD=1
 2950 PRINT"OK...ALGUNS DESAPARECERAM,MA
 S:COMPUTADOR LIGADO,URANIO E RUBI NAO."
 :GOTO 398
 3000 ' ARQUIVOS
 3010 CLS:COLOR 1,11
 3011 IF LEFT\$(C\$,1)="C" OR LEFT\$(C\$,1)=
 "c" THEN F\$="CAS":F=1
 3015 ON ERROR GOTO 3600
 3020 PRINTTAB(5)"LOAD / SAVE / KILL / F
 ILES"
 3030 LOCATE14,6:PRINT" 1 SAVE"
 3040 LOCATE14,7:PRINT" 2 LOAD"
 3050 LOCATE14,8:PRINT" 3 KILL"
 3060 LOCATE14,9:PRINT" 4 FILES"
 3065 LOCATE14,10:PRINT" 5 VOLTA"
 3070 LOCATE10,17:PRINT"QUAL A SUA ECOLH
 A :"
 3080 S\$=INKEY\$:IF S\$="" OR S\$<"1" OR S\$
 >"5" THEN 3080
 3090 ON VAL(S\$) GOTO 3100,3200,3300,340
 0,370
 3100 ' SAVE
 3110 CLS:PRINTTAB(14)"# SAVE #"
 3120 LOCATE10,10:INPUT "QUAL O NOME";NA
 \$
 3130 NA\$=F\$+NA\$+".ADV":OPEN NA\$ FOR OUT
 PUT AS #1
 3135 PRINT#1,LV:PRINT#1,LR:PRINT#1,LO:P
 RINT#1,A:PRINT#1,AU:PRINT#1,CJ:PRINT#1,
 RR:PRINT#1,IS:PRINT#1,BR
 3140 PRINT#1,NB
 3150 FOR I=1TONB
 3160 PRINT#1,OB(I)
 3170 NEXT
 3180 PRINT#1,M:PRINT#1,MT:PRINT#1,MU:PR
 INT#1,FM:PRINT#1,L
 3181 IF M=0 THEN 3185 ELSE FOR I=1TON:PR
 INT#1,M(I):NEXT
 3185 CLOSE #1
 3190 LOCATE17,20:PRINT"OK."
 3199 IF INKEY\$="" THEN 3199 ELSE GOTO 3
 000
 3200 ' LOAD
 3210 CLS:PRINTTAB(14)"# LOAD #"
 3220 LOCATE10,10:INPUT "QUAL O NOME";NA
 \$
 3230 NA\$=F\$+NA\$+".ADV":OPEN NA\$ FOR INP
 UT AS #1
 3235 INPUT#1,LV:INPUT#1,LR:INPUT#1,LO:I
 NPUT#1,A:INPUT#1,AU:INPUT#1,CJ:INPUT#1,
 RR:INPUT#1,IS:INPUT#1,BR

3240 INPUT#1,NB
 3250 FOR I=1TONB
 3260 INPUT#1,OB(I)
 3270 NEXT
 3271 INPUT#1,M:INPUT#1,MT:INPUT#1,MU:IN
 PUT#1,FM:INPUT#1,L
 3272 IF M=0 THEN CLOSE#1:GOTO 3290:ELSE
 FOR I=1TON:INPUT#1,M(I):NEXT:CLOSE#1
 3274 FOR I=1TON
 3275 OB(M(U))=0
 3276 NEXT:GOTO 3290
 3290 LOCATE17,20:PRINT"OK."
 3299 IF INKEY\$="" THEN 3299 ELSE GOTO 3
 000
 3300 ' KILL
 3301 IF F=1 THEN 3000
 3310 CLS:PRINTTAB(14)"# KILL #"
 3320 LOCATE10,10:INPUT "QUAL O NOME";NA
 \$
 3325 LOCATE17,20:PRINT"OK."
 3330 NA\$=NA\$+".ADV":KILL NA\$
 3335 LOCATE17,20:PRINT"OK."
 3340 IF INKEY\$="" THEN 3340 ELSE GOTO 3
 70
 3400 ' FILES
 3401 IF F=1 THEN 3000
 3410 CLS:PRINTTAB(13)"# FILES #"
 3420 LOCATE0,9:FILES">#.ADV"
 3425 LOCATE17,20:PRINT"OK."
 3430 IF INKEY\$="" THEN 3430 ELSE GOTO 3
 000
 3500 ' BEBER
 3510 IN\$=0:IN\$=INSTR("ELIXIR",N\$):IF IN\$1
 THEN PRINT"NAO FAZA ME RIR !":GOTO 398
 3520 IF OB(1)<-1 THEN PRINT"VOCE NAO P
 ODE BEBER O QUE NAO TEM...":GOTO 398
 3530 IF BR>0 THEN PRINT:PRINT" VOCE T
 ONOU UMA 'OVERDOSE' DE ELIXIR":PRINT"
 SUA GULA D MATOU !!!":GOTO 1000
 3540 PRINT"OK! VOCE ESTA FORTE COMO UM
 TOURO !!!":BR=1:GOTO 398
 3600 ' ERROS
 3610 IF ERR>53 THEN 3620 ELSE PLAY"V15
 07C8":LOCATE8,16:PRINT"ARQUIVO INEXISTE
 NTE !!!":R4\$=INPUT\$(1):RESUME3000
 3620 IF ERR>56 THEN 3630 ELSE PLAY"V15
 07C8":LOCATE10,16:PRINT" NOME INCORRETO
 !!!":R4\$=INPUT\$(1):RESUME3000
 3630 IF ERR>67 THEN 3640 ELSE PLAY"V15
 07C8":LOCATE10,16:PRINT" DIRETORIO CHEIO
 !!!":R4\$=INPUT\$(1):RESUME3000
 3640 IF ERR>66 THEN 3650 ELSE PLAY"V15
 07C8":LOCATE13,16:PRINT" DISCO CHEIO !!!
 ":R4\$=INPUT\$(1):RESUME3000
 3650 IF ERR>68 THEN 3660 ELSE PLAY"V15
 07C8":LOCATE11,16:PRINT" DISCO PROTEGIDO
 !!!":R4\$=INPUT\$(1):RESUME3000
 3660 IF ERR>69 THEN 3670 ELSE PLAY"V15
 07C8":LOCATE13,16:PRINT" ERRO DE E/S !!!
 ":R4\$=INPUT\$(1):RESUME3000
 3670 IF ERR>70 THEN 3680 ELSE PLAY"V15
 07C8":LOCATE13,16:PRINT" FALTA DISCO !!!
 ":R4\$=INPUT\$(1):RESUME3000
 3680 LOCATE 14,16:PRINT" ERRO !!!":R4\$=I
 NPUT\$(1):RESUME3000

BATTLE FOR MIDWAY

TANIA ALVES
MSX INFORMÁTICA

APRENDA A JOGAR UM DOS MELHORES
JOGOS DE ESTRATÉGIA PARA O MSX.

Battle for Midway é um dos melhores jogos de estratégia feitos para o MSX. Nele se equilibram a veracidade do fato histórico real com o desafio de um excelente jogo. Tendo por base a verdadeira batalha travada entre americanos e japoneses, este jogo consegue aliar de forma supreendente estratégia e muita ação.

Assim, consegue agradar tanto os aficionados em jogos de guerra como os de qualquer outro tipo de jogo. Muito mais realista do que um jogo de tabuleiro, onde o jogador "todo-poderoso" tem o total controle da situação. Em Battle for Midway, o computador permite que o elemento surpresa seja aproveitado, de modo que você possa provar que é um bom estrategista.

Por outro lado, com a utilização dos recursos do seu MSX, o jogo também apresenta cenas de combate nas quais você participa e tem a chance de mostrar as suas habilidades ao se defrontar com o inimigo.

Para conseguir jogá-lo até o fim, há a necessidade de se estudar muito bem o manual, sem o qual poucas coisas poderão ser feitas para se vencer a batalha. Nele estão explicados os diversos níveis de jogo, bem como as velocidades disponíveis do jogo e sua relação com o tempo real. Além disso, ele explica as diversas unidades aéreas e navais, com seus códigos, cores e relatórios, sem deixar de lado as explicações da movimentação da unidade, condições de vôo, reconhecimento, ataques aéreos e de superfície, etc.

TÁTICAS DE JOGO

Battle for Midway foi programado visando reconstruir, da forma mais real possível, as tensões e os imprevistos de uma batalha, além de desafiar a capacidade estratégica do jogador. Nele foram incluídos um relógio e limitações de combustível que tiveram por objetivo recrivar as pressões que sofre um comandante durante um combate. As sequências de ação em tempo real são utilizadas para propiciar uma visão de batalha. Mas o jogo tem "armadilhas". Por exemplo: o jogador nunca pode ter certeza de que um porta-aviões afundou mesmo que possavê-lo em chamas.

De outro lado, porta-aviões, que aparentemente saíram ilesos do combate, poderão afundar posteriormente.

Os aviões de reconhecimento são usados para representar os hidro-aviões que estavam baseados em Midway. Como são hidro-aviões, permanecem operacionais mesmo que a base aérea de Midway seja destruída.

Algumas vezes são notadas falhas na área de reconhecimento. A intenção é de "punir" o jogador pelo uso de táticas inadequadas, mas também representa a confusão que resultaria de uma observação mal feita em uma situação real.

As frotas japonesas têm a capacidade de sumir por algum tempo, depois de um combate de superfície. Este recurso foi

utilizado para simular a dificuldade em se diferenciar um navio do outro, quando um grande número deles se encontram em uma mesma área.

Algumas pessoas podem achar que os combates de superfície são injustos para com os americanos. Isto visa compensar a verdadeira situação dos americanos durante a guerra, quando as unidades de superfície japonesas apresentavam um desempenho bom na guerra. Mais tarde, com a utilização do radar pelos americanos é que a situação mudou.

Para sair vitorioso, você deverá, assim como os americanos, afundar mais de dois porta-aviões para cada um dos seus que for afundado. O principal objetivo dos japoneses era o de destruir a frota de porta-aviões americana. Neste estágio da guerra, os japoneses estavam preparados para perder dois porta-aviões para cada um dos americanos que conseguissem destruir.

Muito mais poderia ser dito, mas a real dimensão deste jogo você vai perceber após jogar algumas partidas com o auxílio do seu manual. Boa sorte.

LAZAROFT MSX

HORAS

Os Magos do software para MSX

ENIGMA • STREAKER • GROTTEN • BLACK PIRATE •
HUNDRA • CAPTAIN SEVILLA VII • ARKOS VII • FUNKY PANKY •
ELITE • DESPERADO • ABADIA • TRANTOR • DUAD • MADMIX • BOP •
NEMESIS • LINKER'S SUPER STAR SOCCER • SIR FRED • INDIANA
JONES • CARFIGHTER • VENOM • OCEAN "SUB CONQUERER" •
EL MAGO (EDUCAT.) • GAME OVER 1 & 2 • ZAIDER • THE PUB •
STAR FIGHTER • REX HARDEST • MUITOS OUTROS

Você avalia um serviço pela sua eficiência:

• Nossa qualidade indiscutível
• Nossa experiência inigualável
• Nosso prazo entrega 24 horas + correio
• Nosso acervo + 2000 títulos
• Nossa documentação farta
• Nosso super catálogo grátis (peça o seu!)
• Nossos lançamentos semanalmente
• Nossas promoções diversas

Como vé, ninguém tem mais a oferecer do que o Mago da Lazarof.
Escreva-nos e receba um brinde! Você só tem a ganhar!

• JOGOS Cz\$ 300,00
• APPLICATIVOS/UTILITÁRIOS Cz\$ 1.000,00 +
• LINGUAGENS/COPIADORES Cz\$ 1.500,00
• DISCO/FITA Cz\$ 800,00 Cz\$ 900,00
• DESPESAS POSTAIS Cz\$ 300,00

Enviar cheque nominal cruzado a Carlos Henrique B. Magalhães, fornecendo o máximo de informações sobre seu equipamento e telefone para um eventual contato.

Caixa Postal 1965 - CEP 20001 - Rio de Janeiro - RJ
Tel.: (021) 248-1575

BR 116 - A RODOVIA DA MORTE

GUILHERME A. L. DA SILVA

Navalhada Jones é um grande aventureiro que busca emoções e perigo incansavelmente. Pensando nisso, o nosso herói foi assistir ao Grande Prêmio de Fórmula 1 do Brasil e teve uma idéia repentina: roubar um carro e sair "rodando" pela BR 116 !!!

Mas como? Navalhada Jones não sabe pilotar um fórmula 1 !

Então, você deverá ajudá-lo a conduzir o carro. Lembre-se que Jones é "barbeiro". Portanto, conduza com nervos de aço e bata recordes de permanência na rodovia.

O PROGRAMA

O programa é feito totalmente em basic e SCREEN 1; mistura textos, sprites e caracteres gráficos.

A tela, ou melhor, a pista, se move de cima para baixo, obtido por um scroll

invertido, dando a impressão de movimento.

O ruídos do motor e da explosão são produzidos pelos comandos SOUND.

COMO USAR

As setas para esquerda e para direita controlam a direção do carro e as setas para cima e para baixo a aceleração.

Os carros estão em ordem de dificuldade:

NIVEL 1: AIRTON SENA
NIVEL 2: NELSON PIQUET
NIVEL 3: MAURICIO GUGELMIN

VARIÁVEIS

CP\$() = nomes dos recordistas
CD() = distâncias recordes
DD() = auxiliar de CD()
PP\$() = auxiliar de CP\$()

KM = quilômetros rodados

TS = tempo

B1\$ = bloco gráfico 1

B2\$ = bloco gráfico 2

B3\$ = bloco gráfico 3

SO = intensidade do sound

X = X do carro

Y = Y do carro

VL = sem verificação de trombada

ET() = tabulação da estrada

P = cor do carro

PT = tempo de espera

DF = dificuldade

ES = sorteio

PL = número do piloto

I = contador

C = stick

AC = aceleração

CX = X do carro em colunas

EX = X do começo do acostamento

```
10 REM--BR - 116
20 REM--PARA MSX
30 REM-----
40 REM--BY GUILHERME
50 REM--14/10/1988
60 REM--GUARARAPES
70 REM-----
80 DATASENA,2000,PROST,1000,BERGER,450,
PIQUET,350,GUGELMIN,300
90 RESTORE80:FORJ=1TO5:READCP$(J),CD(J)
:NEXT
100 DIMET(50)
110 KEYOFF:SCREEN1,,0:KM=0:TS=2:B1$=CHR
$(219):B2$=CHR$(215):B3$=CHR$(220):GOSU
B430
120 DATA00011000,01111110,01111110,0001
1000,00011000,11011011,11111111,1101101
1
130 RESTORE120:FORH=0TO7:READP$:S$=$+C
HR$(VAL("B"+P$)):NEXT
140 SPRITE$(1)=S$
160 SO=13:I=8
170 X=120:Y=80:VL=1:FORK=1TO22:ET(K)=10
:NEXT:GOSUB250:ET=10:GOSUB510
180 COLOR 1,15,PC
190 ES=INT(RND(-TIME)*(3-DF))+DF
200 IFES=1ANDET<2THENET=ET+1
210 IFES=2ANDET>0THENET=ET-1
220 N=N+1:ET(N)=ET:IFN=11THENGOSUB250
230 IFNN=1THENFORI=11TO1STEP-1:ET(I)=ET
(I-1):NEXT:ET(1)=ET:GOSUB250
240 GOTO190
250 LOCATE,0:FORKJ=1TO5:NEXT
260 I=I+1:LOCATEET(I):PRINTB2$;B2$;B2$;
B1$;" ";B1$;B2$;B2$;NN=1:N=0:C=STI
```

```
CK(0):IFC=7THENX=X-8
270 IFC=3THENX=X+8
280 IFC=1ANDY>8THENY=Y-8:AC=AC+1:SO=SO-
1:GOSUB510
290 IFC=5ANDY<8THENY=Y+8:AC=AC-1:SO=SO
+1:GOSUB510
300 PUTSPRITE1,(X,Y),PC:IFCSRLIN=10THEN
VL=0
310 CX=INT(X/8-2):EX=ET(INT(Y/8)+1):IFC
X<EX+40CX>EX+5THENGOSUB350
320 IFI<11THEN340
330 KM=KM+((10-PT)-DF)+AC:VX=POS(0):VY=
CSRLIN:LOCATE12,16:PRINTUSING" #####";K
M;:PRINT" Km ":TS=TS-.05:LOCATE9,18:PRI
NT"CRONOMETRAGEM":LOCATE14,20:PRINTUSIN
G"##":TS:LOCATEVX,VY
331 IF TS>0 THEN RETURN ELSE GOTO 341
340 GOTO260
341 SOUND7,7:SOUND8,15:FORI=1TO1000:NEX
T:SOUND8,0:PUTSPRITE1,(-8,-8),0
342 CLS:LOCATE5,8:PRINT"Acabou o seu te
mpo...":LOCATE5,10:PRINT"Percorreu:";KM
;" km.":LOCATE7,21:PRINT"Pressione Espa
ço":K$=INPUT$(1):IFK$<>" " THEN342ELSE39
0
350 REMBAU
360 IFV>0THENRETURN
370 PUTSPRITE1,(-8,-8),0:LOCATECX,INT(Y
/8):PRINT" " :SOUND7,7:SOUND8,15:FORI=1T
O1000:NEXT:
380 SOUND8,0:CLS:LOCATE7,8:PRINT"Voce c
olidiu...":LOCATE5,10:PRINT"Percorreu:";
KM;" km.":LOCATE7,21:PRINT"Pressione E
spaco":K$=INPUT$(1):IFK$<>" " THEN380
390 GOSUB420:FORI=1TO5:IFKM>CD(I)THEN
```

```
0T0400:ELSENEXT:GOT0410
400 FORN=1TO6:PP$(N)=CP$(N):DD(N)=CD(N)
:NEXT:FORN=1TO5:CP$(N+1)=PP$(N):CD(N+1)
:DD(N):NEXT:LOCATE0,17:PRINTI;"$ NOME "
;:INPUTCP$(I):CD(I)=KM:GOSUB420
410 LOCATE8,21:PRINT"JOGA DE NOVO ?":K$=
INPUT$(1):IFK$<>" " NTHENGOTO 110ELSEEND
420 CLS:PRINTTAB(9);"CAMPEONATO":LOCATE
5,6:PRINT"NONE";SPC(8);"DISTANCIA";:FOR
I=1TO5:LOCATE5,10+I:PRINTCP$(I);:LOCATE
18,10+I:PRINTUSING"#####";CD(I):NEXT:R
ETURN
430 REMPAINEL
440 FORI=0TO11:LOCATE0,I:PRINTSTRING$(2
9,B2$):NEXT:FORI=12TO21:LOCATE0,I:PRINT
STRING$(29,B1$):NEXT:LOCATE0,11:PRINTST
RING$(29,B3$)
450 LOCATE0,19:PRINTB1$;"1-SENA 2-PIQUE
T 3-GUGELMIN";:P$=INPUT$(1):PL=VAL(P$)
460 IFPL<10RPL>3THEN450
470 IFPL=1THENP$=" AIRTON SENA - MACLA
REN ";PC=9:PT=1:DF=1
480 IFPL=2THENP$=" NELSON PIQUET - LO
TUS ";PC=10:PT=2:DF=0
490 IFPL=3THENP$=" MAURICIO GUGELMIN - M
ARCH";PC=5:PT=7:DF=0
500 LOCATE0,19:PRINTSTRING$(58,B1$):LOC
ATE2,12:PRINTP$:LOCATE7,14:PRINT" KM PE
RCORRIDOS ":LOCATE11,16:PRINTUSING" #####
";KM;:PRINT" Km "
501 LOCATE9,18:PRINT"CRONOMETRAGEM":LOC
ATE14,20:PRINTUSING" ##";TS:RETURN
510 SOUND7,254:SOUND8,1:SOUND1,SO:SOUND
6,20:SOUND8,16:SOUND11,255:SOUND12,1:S
OUND13,13:RETURN
```

JOGOS & HIGH SCORES

JOGO	SCORE	RECORDISTA	JOGO	SCORE	RECORDISTA
ALIEN 8	82%		LAZY JONES	200.250	
ALPHA BLASTER	89.235		LES FICLES	100.200	
BARNSTORMER	279.955		LE MANS	42.530	
BATTLESHIP CLAPTON	97.300		MANIC MINER	117.321	
BEAMRIDER	207.520		MAXIMA	211.120	PEDRO MARIANI
BLAGGER	231.520		MONKEY ACADEMY	461.200	ROBERTO T F MORAES
BOOM	99.240		MONPIRANGER	840.100	
BOULDERDASH	59.848		MUTANT	737	
BOUNDER	321.624	BRUNO MURRAT	NIGHTSHADE	137.000	
BOXING	10		NINJA	42.750	
BUCK ROGERS	310.900		OH MUMMY	5.030	
CENTIPEDE	53.795	ALEXANDRE C GREIG	OH NO	76.250	
CHILLER	42.201		OILS WELL	215.700	
CHORO Q	42.380		PANIC JUNCTION	14.919	ROBERTO T F MORAES
CIRCUS CHARLIE	1.198.460		PASTFINDER	24.205	
DISK WARRIOR	1.400.000		PILLBOX	2.800	
DOGFIGHTER	10.100		PINBALL	1.240.680	
ELIDON	94%		PITFALL II	199.000	
ERIC AND FLOATERS	1.844.160		POLAR STAR	289.990	ALBERTO G SANTOS
FINDERS KEEPERS	18.323		PUNCHY	8.434.070	
FIRE RESCUE	29.540		PRICE MAGIK	12%	
FLIGHT DECK	7.210	MARCOS A LACERDA	PYRAMID WARP	820.758	
FRUIT FRANK	21.000		RIVER RAID	73.450	
GALAGA	452.200	ALEXANDRE C GREIG	ROAD FIGHTER	986.675	
HOSTBUSTERS	\$999.900		ROLLER BALL	4.580.120	BRUNO MURRAT
GOLF	28		SASA	200.195	ALBERTO G SANTOS
GRIDTRAP	558.120		SCION	95.300	
GUNFRIGHT	\$150.000		SOCCER	40-0	
HEIST	384.201		SPACE WALK	1.846.200	
HERO	692.120		SPOOKS AND LADDERS	189.930	
HIGHWAY	339.360	MARCOS A LACERDA	STEP UP	60.250	
HOOPER	100.050		STQP THE EXPRESS	7.360	
HOTSHOE	187.575	PEDRO M FRACT	SUPER COBRA	501.100	
HUNCHBACK	2.700.000		SWEET ACORN	6.438.460	
HUSTLER	8	ROBERTO T F MORAES	TENNIS	6-0 6-0	
HYPER RALLY	310.100		THE SNOWMAN	36.510	
HYPER SPORTS I	2.050.800		THE WRECK	23.975	
HYPER SPORTS II	500.500		TIME BANDITS	9.990	
HYPER SPORTS III	65.532		TIME CURB	274.040	
HYPER VIPER	127.500		TIME PILOT	689.000	
INTER. KARATE	999.999		TRACK AND FIELD I	266.540	
JET FIGHTER	214.950		TRACK AND FIELD II	500.300	
JET SET WILLY	120		TURMOIL	12.520	
KINGS VALLEY	5.642.600		VACUMANIA	22.340	
KNIGHTMARE	478.200		VALKYR	47.205	

JOGOS

NEMESIS

KNIGHT NINJA

TIPO- Aventura oriental
APRESENTAÇÃO- 10
GRÁFICOS- 9
SOM- 9
INTERESSE-10
NÚMERO DE BLOCOS- 3
TOTAL GERAL- 9.5



SKY VISION

TIPO- Espacial de ação
APRESENTAÇÃO- 7
GRÁFICOS- 8
SOM- 7
INTERESSE- 9
NÚMERO DE BLOCOS- 3
TOTAL GERAL- 7

BOP

TIPO- Aventura infantil
APRESENTAÇÃO- 7
GRÁFICOS- 7
SOM- 8
INTERESSE- 8
NÚMERO DE BLOCOS- 3
TOTAL GERAL- 7.5

MAD FOX

TIPO- Espacial de ação
APRESENTAÇÃO- 7
GRÁFICOS- 7
SOM- 7
INTERESSE- 7
NUMERO DE BLOCOS- 3
TOTAL GERAL- 7

KIMPO FIGHTER

TIPO- Aventura oriental
APRESENTAÇÃO- 8
GRÁFICOS - 9
SOM- 8
INTERESSE- 8
NÚMERO DE BLOCOS- 1
TOTAL GERAL- 8

EMILIO BUTRAGUENO (FUTEBOL)

TIPO- Jogo de Futebol
APRESENTAÇÃO- 10
GRÁFICOS- 9
SOM- 7
INTERESSE- 9
NÚMERO DE BLOCOS- 5
TOTAL GERAL- 9



LADY SAFARY

TIPO- Aventura na Selva
APRESENTAÇÃO- 8
GRÁFICOS- 8
SOM- 6
INTERESSE- 7
NÚMERO DE BLOCOS-4
TOTAL GERAL- 7

CAVERN OF DEATH

TIPO- Aventura e ação

APRESENTAÇÃO- 7

GRÁFICOS- 7

SOM- 7

INTERESSE- 7

NÚMERO DE BLOCOS- 3

TOTAL GERAL- 7



HUMPREY

TIPO- Jogo de estratégia

APRESENTAÇÃO- 10

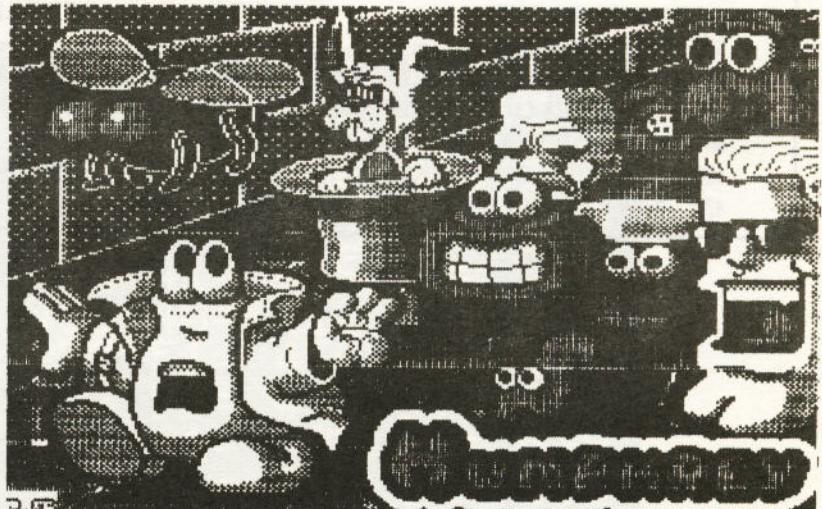
GRÁFICOS- 9

SOM- 8

INTERESSE- 9

NÚMERO DE BLOCOS- 6

TOTAL GERAL- 9



MINDER

TIPO- Adventure animado

APRESENTAÇÃO- 10

GRÁFICOS- 8

SOM- 8

INTERESSE- 8

NÚMERO DE BLOCOS- 7

TOTAL GERAL- 8

SOS FELINO

```

20  SOS FELINO -- (c) Esfera
40  Copyright by SChan. 1988
70
80  I N I C I A L I Z A R
90
100 CLEAR500:COLOR15,4,4:KEYOFF:SCREEN1
,2,0:WIDTH32:C=RND(-TIME)
110
120  DEFINIR BLOCOS GRAFICOS
130
140 FORI=384T0983:A=VPEEK(I):B=AORA/2:V
POKEI,B:NEXT
150 DATA170,85,170,85,170,85,170,85
160 DATA255,239,219,189,189,255,103,2
170 DATA36,98,189,24,255,24,126,153
180 DATA66,189,129,165,129,153,129,126
190 FORI=1536T01543:READA:VPOKEI,A:NEXT
:FORI=1600T01607:READA:VPOKEI,A:NEXT:FO
RI=1664T01679:READA:VPOKEI,A:NEXT
200 VPOKE8216,97:VPOKE8217,196:VPOKE821
9,160
210
220  B O M B E I R O
230
240 DATA0000000110000000
250 DATA0000000100100000
260 DATA0000111111110000
270 DATA0000000100100000
280 DATA0000000100100000
290 DATA0000000110000000
300 DATA0000111001110000
310 DATA0001010000101000
320 DATA0001010000101000
330 DATA0001010000101000
340 DATA0001011111101000
350 DATA0001010000100000
360 DATA0000011111100000
370 DATA0000011001100000
380 DATA0000011001100000
390 DATA0001111001111000
400 FORI=0T015:READA$:B$=B$+CHR$(VAL("&
8*LEFT$(A$,8))):C$=C$+CHR$(VAL("&B"+RI
GHT$(A$,8))):NEXT:SPRITE$(0)=B$+C$
410
420  DEFINIR VALORES INICIAIS
430
440 A$="04cecdd":B$="05cecdd":S=10:T=0:
N=1:E=.75:F=1:L(1)=1:L(2)=6:L(3)=11:L(4)
)=16:L(5)=16:ONINTERVAL=900GOSUB730
450
460  O  C E N A R I O
470
480 PLAYA$+B$:PLAYA$,B$:IFF>15THENF=15:
E=10
490 CLS:X=32:Y=159:LOCATE12,10:PRINT"NI
VEL";N:FORI=0T02000:NEXT:FORI=0T015STEP
5:LOCATE2,I:PRINTCHR$(200);SPC(26);CHR$(
200):PRINT;" ";STRING$(3,200);SPC(24);
STRING$(3,200)
500 PRINT" ";STRING$(2,200);STRING$(12,
192);";";STRING$(12,192);STRING$(2,200
):PRINT" ";STRING$(13,200);";";STRING
$(13,200):PRINTTAB(13);STRING$(2,200);
";";STRING$(2,200):NEXT
510 FORI=640T0767:VPOKEBASE(5)+I,219:NE
XT
520 FORI=2T018:LOCATE15,I:PRINTSTRING$(
2,192):NEXT:LOCATE13,19:PRINTSTRING$(6,
192)
530
540  P R O C E S S A M E N T O
550
560 INTERVALON:GOTO680
570 A=STICK(0):B=(Y+1)*4+X/8:IFA=3ANDX<
224ANDVPEEK(BASE(5)+B+2)<>192ANDVPEEK(B
ASE(5)+B+34)<>192THENX=X+B:GOTO590
580 IF(A=7ANDX>16)ANDVPEEK(BASE(5)+B-1)<
192ANDVPEEK(BASE(5)+B+31)<>192THENX=X-8
590 IFX<-16ANDY>-1THENX=32:Y=Y-40
600 IFX>224ANDY<159THENX=200:Y=Y+40
610 INTERVALSTOP:D=RND(1)*15:IFC(.75THE
NLOCATEINT(RND(1)*11+4),L(INT(RND(1)*4+
1))):PRINTCHR$(209);ELSEIFC<1THENLOCATE]
NT(RND(1)*11+17),L(INT(RND(1)*4+1)):PRI
NTCHR$(209);
620 C=RND(1)*15:IFC<ETHENLOCATEINT(RND(
1)*11+4),L(INT(RND(1)*4+1)):PRINTCHR$(2
08);ELSEIFC<FTHENLOCATEINT(RND(1)*11+17
),L(INT(RND(1)*4+1)):PRINTCHR$(208);
630 INTERVALON
640 IFVPEEK(BASE(5)+B+33)=209THENVPOKEB
ASE(5)+B+33,32:BEEP:S=S+1:GOTO680
650 IFVPEEK(BASE(5)+B+32)=209THENVPOKEB
ASE(5)+B+32,32:BEEP:S=S+1:GOTO680
660 IFVPEEK(BASE(5)+B+33)=208ANDS>0THEN
VPOKEBASE(5)+B+33,32:S=S-1:PLAY"c12"ELS
EIFVPEEK(BASE(5)+B+33)=208ANDS=0THEN690
:GOTO680
670 IFVPEEK(BASE(5)+B+32)=208ANDS>0THEN
VPOKEBASE(5)+B+32,32:S=S-1:PLAY"c12"ELS
EIFVPEEK(BASE(5)+B+32)=208ANDS=0THEN690
680 PUTSPRITE0,(X,Y),1,0:GOSUB780:GOT05
70
690 INTERVALOFF:LOCATE11,10:PRINT"Fin d
e Jogo"
700 PLAYA$+B$:PLAYA$,B$:FORI=0T02000:NE
XT:LOCATE9,10:PRINT"Novamente (S/N)"
710 IFINKEY$<>""THEN710
720 A$=INKEY$:$IFA$="S"ORA$="s"THENPUTSP
RITE0,(-32,-32):GOTO440ELSEIFA$="N"ORA$=
"n"THENSCREEN0:ENDELSE720
730 INTERVALSTOP:S=S-1:PLAY"a12"
740 IFS<0THEN690ELSESET=T+1:GOSUB780:INTE
RVALON:RETURN
750
760  L I F E & T I M E
770
780 INTERVALSTOP:LOCATE0,22:PRINT"Life"
;S;" Time";T;" Nivel";N
790
800  P A S S A G E M D E N I V E I S
810
820 IFT=10THENPUTSPRITE0,(-32,-32):T=0:
N=N+1:E=E+.375:F=F+.5:FORI=0T01000:NEXT
:GOT0480
830 INTERVALON:RETURN
840  A$,B$ - MUSICA
850  S - LIFE  T - TIME
860  E,F - NUMERO DE INSETOS
870  N - NIVEL DE JOGO
880  L(1-5) - VERTICAL
890  A - VALOR DAS SETAS
900  B - VALOR NA VRAM DA PO-
SICAO DO BOMBEIRO

```

JOGO DA MEMÓRIA

```

10 REM ---- Jogo da Memoria
20 REM ---- Guilherme A. L. da Silva
30 REM ---- 22/06/88
40 REM ---- GUARARAPES - S.P.
50 REM ---- Para a linha MSX
60 CLEAR1000:JO=1
70 SCREEN1,,0:KEY OFF:COLOR 15,12,10
80 STOP ON
90 ON STOP GOSUB 1400
100 REM JOGO DA MEMORIA
110 GOTO 390
120 DIM C$(16),A$(16),B$(16),F$(8)
130 F$(1)=CHR$(1)+CHR$(76):F$(2)=CHR$(1)
  +CHR$(66):F$(3)=CHR$(1)+CHR$(69):F$(4)
  =CHR$(1)+CHR$(70)
140 F$(5)=CHR$(1)+CHR$(67):F$(6)=CHR$(1)
  +CHR$(68):F$(7)=CHR$(1)+CHR$(79):F$(8)
  =CHR$(1)+CHR$(75)
150 SCREEN1
170 PRINTTAB(5)" Jogo da memoria "
180 LOCATE 0,5
200 PRINTTAB(8)"! ! ! ! !"
210 FOR0=1TO3
220 PRINTTAB(8)"! ! ! ! !"
230 PRINTTAB(8)"! ! ! ! !"
240 NEXT
250 PRINTTAB(8)"! ! ! ! !"
260 PRINTTAB(8)"!+!+!+!+!+"
270 LOCATE0,15:PRINT"Qualquer tecla par
a comefar."
280 A$=INKEY$:IF A$="" THEN GOTO 280
290 LOCATE 0,18:PRINTTAB(5);:"Estou calc
ulando":GOSUB 1240
300 FOR I=1 TO 8
310 A=INT(16*RND(-TIME)+1)
320 IF A>B THEN 330 ELSE 310
330 IF B$(A)<>" " THEN 310 ELSE B$(A)=F$
(I)
340 B=INT(16*RND(10)+1)
350 IF B>A THEN 340 ELSE 340
360 IF B$(B)<>" " THEN 340 ELSE B$(B)=F$
(I)
370 NEXT
380 Z=0:FORI=1 TO 16:A$(Z)=B$(I):Z=Z+1:
NEXT:RETURN
390 REM
400 GOSUB 530:REM N.JOG.
410 GOSUB 710:GOSUB 800:REM TELA
420 GOSUB 860:REM INICIO
425 GOTO 420
430 IF P(1)>P(2) THEN JO=2
440 IF P(2)>P(1) THEN JO=2
450 FORU=1TO1000:NEXT:CLS:LOCATE0,21:PR
INTTAB(5)" Jogo da memoria ":PRINT:PRIN
T:PRINT:PRINT:PRINT
460 IF P(2)=P(1) THEN PRINTTAB(10)"EMPA
TE!!!!":GOTO 480
470 PRINT"Vencedor(a): ";N$(JO)
480 PRINT:PRINT"COM O PLACAR DE";P(JO);
"PARES."
490 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT
:PRINT:PRINT:PRINT
500 GOSUB1240:LOCATE0,19:PRINT"JOGAR OU
TRA VEZ?(S/N)":A$=INKEY$:IF A$="" THEN
500
510 IF A$<>"S" THEN END
520 CLS:RUN
530 REM N€ JOGADORES
540 CLS
560 PRINTTAB(5)" Jogo da memoria "
580 PRINT:PRINT"Quantos Jogadores?":INP
UTJ
590 PRINT:PRINT:PRINT"Qual o nome do jo
gador 1":INPUT N$(1)
600 IF J=1 THEN PRINT:N$(2)="Computador
-MSX":GOTO 620
610 PRINT"Qual o nome do jogador 2":INP
UT N$(2):GOTO 640
620 PRINT"Qual o nLev1?":PRINT:PRINT:PR
INTTAB(5)"0.Principiante":PRINTTAB(5)"3
.Aprendiz":PRINTTAB(5)"6.Mestre":PRINTT
AB(5)"9.Profissional":PRINTTAB(5)"12.Co
bra":PRINTTAB(5)"15.Expert"
630 PRINT:INPUT"Escolha?":BN
640 GOSUB 120
650 LOCATE0,15:PRINTSPC(32)
660 LOCATE 0,15
670 PRINT N$1,N$2
680 PRINTUSING"Placar:##";P1;
690 PRINTUSING" Placar:##";P2
700 RETURN
710 REM TELA
720 COLOR 15,4,10
730 LOCATE0,18:PRINTSPC(64):LOCATE0,18:
PRINT" Preste atencao nos pares."
740 IF BN>15 THEN RETURN
750 N=0:FOR LX=9 TO 15 STEP 2
760 FOR LY=6 TO 12 STEP 2
770 LOCATE LX,LY:PRINTA$(N)
780 N=N+1
790 NEXT:NEXT:FOR TP=1TO3000:NEXT:RETUR
N
800 N=0:FOR LX=9 TO 15 STEP 2
810 FOR LY=6 TO 12 STEP 2
820 N$=HEX$(N)
830 LOCATE LX,LY:PRINTN$"
840 N=N+1
850 NEXT:NEXT:RETURN
860 REM JOGO
870 LOCATE0,18:PRINTSPC(64):LOCATE0,18
880 PRINT"Jogador";JO;"(2 numeros)":;IN
PUTH$:W$=MID$(H$,2,1):D$=MID$(H$,1,1)
890 IF H$="" THEN 870
900 TI=VAL("8H"+D$)
910 TP=VAL("8H"+W$)
920 ON TI+1 GOSUB 1080,1090,1100,1110,1
120,1130,1140,1150,1160,1170,1180,1190,
1200,1210,1220,1230
930 PX=X:PY=Y
940 ON TP+1 GOSUB 1080,1090,1100,1110,1
120,1130,1140,1150,1160,1170,1180,1190,
1200,1210,1220,1230
950 LX=X:LY=Y
960 PLAY"V15L16EFG":LOCATE PX,PY:PRINTA
($TI)
970 LOCATE LX,LY:PRINTA$(TP)
980 IF TI=TP THEN LOCATE0,18:GOTO 880
990 IF A$(TI)=C$(TI) AND A$(TP)=C$(TP)
GOTO 1010
1000 IF A$(TI)=A$(TP)THENLOCATE0,19:PRIN
T"0 jogador";JO;"conseguiu":P(JO)=P(JO)
+1:LOCATE0,16:PRINTUSING"PLACAR:##";P(1
);:PRINTUSING" PLACAR:##";P(2);A$(T
I)=" ";A$(TP)=" ";C$(TI)=C$(TP)=
A$(TP):IFP(1)+P(2)=8THEN GOTO 1030 ELSE PLA
Y"V15L8CDL16EGLBAB"
1010 FORI=1TO1000:NEXT:GOSUB 800
1020 IF JO=2 AND J=1 THEN JO=1:GOSUB 87
0
1030 IF JO=1 AND J=1 THEN JO=2:GOSUB 12
70:GOSUB920
1040 IF J=1 THEN RETURN
1050 IF JO=1 THEN JO=2:GOTO 870
1060 IF JO=2 THEN JO=1
1070 RETURN
1080 X=9:Y=6:RETURN
1090 X=9:Y=8:RETURN
1100 X=9:Y=10:RETURN
1110 X=9:Y=12:RETURN
1120 X=11:Y=6:RETURN
1130 X=11:Y=8:RETURN
1140 X=11:Y=10:RETURN
1150 X=11:Y=12:RETURN
1160 X=13:Y=6:RETURN
1170 X=13:Y=8:RETURN
1180 X=13:Y=10:RETURN
1190 X=13:Y=12:RETURN
1200 X=15:Y=6:RETURN
1210 X=15:Y=8:RETURN
1220 X=15:Y=10:RETURN
1230 X=15:Y=12:RETURN
1240 REM MUSICA
1250 PLAY"V15T128L6M5000$1105EEFGGFEDCC
DEEDDEEFGGFEDCCDEDCCLBDECDEFECDEFEDCGE
EFGGFEDOL5CCDEDCC"
1260 RETURN
1270 F=0:REM JOGADA COMP.
1280 F=0:FOR K=1 TO BN
1290 LOCATE0,18:PRINTSPC(64)
1300 RI=INT(15*RND(1))
1310 IF BN>2 AND C$(RI)=A$(RI) THEN 138
0
1320 RP=INT(15*RND(-TIME)):IF RP=RI THE
N 1320
1330 IF F>BN>2 THEN 1380
1340 IF BN>2 AND C$(RP)=A$(RP) THEN F=F
+1:GOTO 1320
1350 IF A$(RI)=A$(RP) THEN 1380
1360 IF BN=0 THEN 1370
1370 NEXT K
1380 LOCATE0,18:PRINT"Eu joguei os nMne
ros: ";HEX$(RI);HEX$(RP)
1390 TI=RI:TP=RP:J=1:JO=2:RETURN
1400 SCREEN0,,1:PRINT"DESISTIU. E' O FI
M!":KEY DN:COLOR 15,1,1:END

```

BOLICHE

```

28 BOWLING - (c) by Esfera
30 Copyright 1988 by SChan
50
60 I N I C I A L I Z A
70
80 CLEAR500:COLOR1,10,10:KEYOFF:SCREEN1
,,0:WIDTH32:DEFUSR=&H41:DEFUSR1=&H42
90
100 D E F I N E B L O C O S
110
120 BEEP:PLAY"7V1504DCDDCC03BB04FEFF6F
DDEFFFDFGDDFGEECCFF"
130 FORI=384T0983:VPOKE1,VPEEK(I)ORVPEE
K(I)/2:NEXT
140 DATA170,85,170,85,170,85,170,85
150 DATA1,3,6,12,25,51,103,207
160 DATA128,192,96,48,152,204,230,243
170 DATA255,195,153,165,165,153,195,255
180 DATA255,255,255,255,195,185,165
190 DATA165,185,195,255,255,195,185,165
200 DATA165,185,195,255,255,255,255,255
210 DATA255,255,255,255,255,255,255,255
220 DATA255,153,153,153,153,153,153,153
230 DATA255,249,249,249,249,249,249,249
240 DATA60,126,251,249,249,251,118,60
250 DATA0,16,16,16,16,56,124,254
260 FORI=1472T01479:READA:VPOKE1,A:NEXT
270 FORI=1536T01591:READA:VPOKE1,A:NEXT
280 RESTORE210:FORI=1600T01623:READA:VP
OKE1,A:NEXT
290 FORN=0T01:A$="":FORI=0T07:READA:A$=
A$+CHR$(A):NEXT:SPRITE$(N)=A$:NEXT
300 VPOKE8215,113:VPOKE8216,111:VPOKE82
17,31:VPOKE8218,22:VPOKE8219,241
310
320 V A L O R E S I N I C I A I S
330
340 B=B:Q=20:P=0:S=0
350
360 D E S E N H A T E L A
370
380 A=USR(0):LOCATE0,0:PRINTSTRING$(21,
184):FORI=0T08:PRINTCHR$(184);SPC(19);C
HR$(184):NEXT:PRINTSTRING$(21,184)
390 A$=STRING$(7,219):LOCATE1,1:PRINTA$;
STRING$(5,200);A$:LOCATE1,2:PRINTA$;CH
R$(202);STRING$(4,201);A$
400 FORI=5T017STEP2:LOCATE1,A=(20-(I+2)
)/2:PRINTSTRING$(A,219);CHR$(192);STRIN
G$(I,198);CHR$(193);STRING$(A,219):NEXT
410 LOCATE,11:PRINTCHR$(184);STRING$(19
,219);STRING$(12,184);course p
ower";CHR$(184);SPC(10);CHR$(184);
420 PRINTSTRING$(32,184);A$=CHR$(184)+
STRING$(30,219)+CHR$(184);B$=CHR$(184)+
STRING$(30,198)+CHR$(184)
430 FORI=0T01:PRINTA$;;NEXT:FORI=0T03:P
RINTB$;;NEXT:FORI=0T01:PRINTA$;;NEXT:FO
RI=704T0767:VPOKEBASE(5)+I,184:NEXT:LOC
ATE0,0
440 LOCATE28,16:PRINTCHR$(195);CHR$(194
):LOCATE27:PRINTCHR$(194);CHR$(196);CHR
$(194)
450 LOCATE27:PRINTCHR$(194);CHR$(196);C
HR$(194):LOCATE28:PRINTCHR$(197);CHR$(1
94):GOSUB1050:A=USR1(0)
460 FORI=0T02:PUTSPRITE1,(-32,-2):NEXT:
LOCATE21,12:PRINTSPC(10);
470 IFINKEY$<>"THEN470
480
490 E S C O L H A R U M O
500
510 C=0:LOCATE14,12:PRINT" ":LOCATE6,12
:PRINT"<":FORX=8T0152STEP2:PUTSPRITE1,(X
,87),1,1
520 IFINKEY$=" THEN560
530 NEXT:FORX=152T08STEP-2:PUTSPRITE1,(X
,87),1,1
540 IFINKEY$=" THEN568
550 NEXT:GOT0510
560 FORI=8T01000:NEXT:IFX<84ANDX>76THEN
C=4
570 IF(X<=76ANDX>68)OR(X)>84ANDX<92)THE
NC=3
580 IF(X<=68ANDX>60)OR(X)>92ANDX<100)TH
ENC=2
590 IF(X<=60ANDX>52)OR(X)>100ANDX<108
)THENC=1
600 IFX>84THEND=158:Z=.5
610 IFX<76THEND=120:Z=-.5
620 LOCATE6,12:PRINT" ":LOCATE14,12:PRI
NT">"
630 IFINKEY$<>"THEN630
640
650 E S C O L H A P O W E R
660
670 F=0:FORX=8T0152STEP2:PUTSPRITE1,(X
,87),1,1:IFINKEY$=" THEN700
680 NEXT:FORX=152T08STEP-2:PUTSPRITE1,(X
,87),1,1:IFINKEY$=" THEN700
690 NEXT:GOT0670
700 IFX>140THENF=6
710 IFX>130ANDX<140THENF=5
720 IFX>120ANDX<130THENF=4
730 IFX>110ANDX<120THENF=3
740 IFX>100ANDX<110THENF=2
750 IFX>90ANDX<100THENF=1
760
770 P R O C E S S A M E N T O
780
790 IFC=4AND(F=60RF=5)THENY=138:Y1=80:E
=4:M$=" Strike":N=10:GOT0920
800 IFC=4AND(F=40RF=3)THENY=138:Y1=80:E
=2:M$=" 8 Bottles":N=8:GOT0920
810 IFC=3AND(F=60RF=5)THENY=138+Z*I4:Y1=
80+Z*I4:E=4:M$=" 7 Bottles":N=7:GOT0920
820 IF(C=3AND(F=40RF=3))OR(C=2AND(F=60R
F=5))THENY=138+Z*I8:Y1=80+Z*I8:E=3:M$=" 6
Bottles":N=6:GOT0920
830 IFC=4AND(F=20RF=1)THENY=138:Y1=80:Z
=1:E=1:M$=" 5 Bottles":N=5:GOT0920
840 IFC=2AND(F=30RF=4)THENY=138+Z*I8:Y1=
80+Z*I8:E=2:M$=" 4 Bottles":N=4:GOT0920
850 IF(C=3AND(F=20RF=1))OR(C=1AND(F=50R
F=6))THENY=138+Z*I10:Y1=80+Z*I10:E=3:M$=" 3
Bottles":N=3:GOT0920
860 IF(C=2AND(F=10RF=2))OR(C=1AND(F=30R
F=4))THENY=138+Z*I8:Y1=80+Z*I8:E=1.5:M$=" 2
Bottles":N=2:GOT0920
870 IFC=1AND(F=10RF=2)THENY=138+Z*I10:Y1
=80+Z*I10:E=1:M$=" 1 Bottle":N=1:GOT0920
880 Y=138:FORX=8T0239STEPF/2+1:PUTSPRIT
E0,(X,Y),1,0:PUTSPRITE2,(80-(138-Y),70-
X/5),1,0
890 Y=Y+Z:IFZ=.5THENIFY=DTHENY=D
900 IFZ=-.5THENIFY<=DTHENY=D
910 NEXT:GOT0970
920 FORX=8T0220STEPE:PUTSPRITE0,(X,Y),1
,0:PUTSPRITE2,(Y1,70-X/4.5),1,0:NEXT:PL
AY"s10@00t140112cecfedcc":P=P+N:S=S+N
930 LOCATE21,12:PRINTM$'
940
950 N O V O N I V E L
960
970 IFB=1THENB=0:GOT0980ELSEB=1:GOSUB
1850:FORI=8T01000:NEXT:GOT0460
980 BEEP:PLAY"V12L16CDE05CDE03CDED02BB
03CC04CDED0C03BB04CC"
990 GOSUB1050:IFP>QTHENLOCATE6,20:PRIN
T"QUALIFICATION REACHED":FORI=8T02000:N
EXT:LOCATE6,20:PRINTSTRING$(21,219):Q=Q
+10:P=0:B=0ELSEI1100
1000 IFQ>50THENB=B+((Q-50)/10)
1010 GOSUB1050:GOT0460
1020
1030 P L A C A R
1040
1050 LOCATE22,0:PRINT"Bowling I":LOCATE
21,2:PRINT"Quali":Q:LOCATE21,4:PRINT"Po
int":P:LOCATE21,6:PRINT"Balls":B
1060 LOCATE22,8:PRINT"Score":LOCATE21
,9:PRINTSPC(11)::LOCATE21,9:PRINTS::RET
URN
1070
1080 G A M E O V E R
1090
1100 LOCATE11,20:PRINT"GAME OVER
1110 BEEP:PLAY"V15CDE0C03B04C"
1120 IFINKEY$<>"THEN1120
1130 IFINKEY$=""THEN1130ELSE340
1140
1150 S - S C O R E B - B A L L S
1160 Q - Q U A L I F I C A T I O N
1170 P - P O I N T S C - R U M P

```

ASSINE CPU POR 6 MESES

PREÇO ESPECIAL ATÉ 15. 12. 88

**DESEJO EFETUAR A ASSINATURA
DA REVISTA CPU PELO PERÍODO
DE SEIS MESES.**

PARA TAL, ESTOU ENVIANDO CHEQUE NOMINAL
A AGUIA INFORMÁTICA, OU VALE POSTAL
(PAGÁVEL NA AGENCIA COPACABANA), NO
VALOR DE Cz\$ 4.000,00 (QUATRO MIL CRUZADOS).

CPU
LEIA E ASSINE

AGUIA INFORMÁTICA

AGUIA INFORMÁTICA

ASSINE CPU POR 6 MESES

PREÇO ESPECIAL ATÉ 15. 12. 88

**DESEJO EFETUAR A ASSINATURA
DA REVISTA CPU PELO PERÍODO
DE SEIS MESES.**

PARA TAL, ESTOU ENVIANDO CHEQUE NOMINAL
A AGUIA INFORMÁTICA, OU VALE POSTAL
(PAGÁVEL NA AGENCIA COPACABANA), NO
VALOR DE Cz\$ 4.000,00 (QUATRO MIL CRUZADOS).

CPU
LEIA E ASSINE

AGUIA INFORMÁTICA

AGUIA INFORMÁTICA

NOME:

ENDEREÇO :

BAIRRO : CEP :

CIDADE : ESTADO :

DADOS DO EQUIPAMENTO :

.....

NOME:

ENDEREÇO :

BAIRRO : CEP :

CIDADE : ESTADO :

DADOS DO EQUIPAMENTO :

.....

CPU

LEIA
PARTICIPE
ASSINE

CPU

LEIA
PARTICIPE
ASSINE

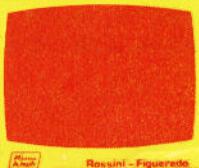
100 DICAS PARA MSX



NEditora
Aleph

**TÉCNICAS E
TRUQUES DE
PROGRAMAÇÃO**

**linguagem
de máquina**
ASSEMBLY
Z-80 **MSX**



Rossini - Figueiredo

Nossos livros podem ser encontrados em livrarias e lojas de computação. Se o seu livreiro ou fornecedor habitual não os tiver disponíveis, entre em contato conosco pelo telefone (011) 843-3202.

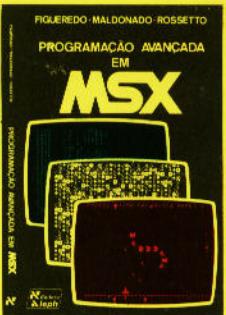
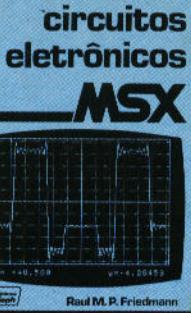
Se você não está recebendo seu boletim gratuitamente pelo correio, ou tem algum amigo que gostaria de recebê-lo, não deixe de enviar o cupom abaixo à EDITORA ALEPH - C.P. 20707 - CEP: 01498 - SÃO PAULO-SP.

NOME:

END.:

CEP: CIDADE: UF:

TEL: (.....) MICRO(S) QUE POSSUI:



Com a palavra, um Expert:



Se você ainda não me conhece, tenho certeza de que já ouviu falar muito a meu respeito.

Sou Expert MSX, o micro projetado e construído pela máquina mais perfeita do mundo: o homem. Com toda a tecnologia e vanguarda de quem sempre pesquisou e evoluiu para tornar a vida do homem muito melhor: a Gradiente.

A imagem e semelhança da Gradiente, sou um pioneiro. Meu design, moderno e profissional, inaugurou um estilo. E até hoje eu sou o único a lhe oferecer teclado separado do console. Tenho 3 processadores, processo informações 3,5 vezes mais rápido que meus concorrentes e meus arquivos são compatíveis com IBM-PC*.

Claro! Todo homem quer crescer nos negócios e na família. E quando isso acontece eu continuo lá, útil e prático, ao lado dele. O melhor testemunho de minha qualidade é o tempo de garantia que me acompanha: o maior que você pode encontrar. E para sua comodidade, tenho também a maior rede de assistência técnica do país, dez vezes superior a qualquer outra marca.

Entre softwares, tudo que você imaginar em aplicativos e jogos eu aceito, entendo e decifro.

E como se tudo isso não bastasse, existem vários periféricos e livros disponíveis no mercado feitos especialmente para mim.

Expert MSX da Gradiente.

Conte comigo.

EXPERT

SISTEMA
MSX

gradiente