

ANO 1 · JUNHO 84 · N.º 2

DATA



SERPRO SERVIÇO FEDERAL DE PROCESSAMENTO DE DADOS

As figuras das capas do **BYTE PAPO** serão os gráficos de programas *Two Liners*. Estes programas caracterizam-se por realizarem impressionantes atividades no microcomputador em apenas duas linhas de programa.

O desenho da capa foi produzido por processo de digitalização de imagens, constituído por uma interface que liga uma câmera ao micro. Este desenho pode ser encontrado no *Software* intitulado "*Zoom Grafix*" da *Phoenix Software*.



Tente, desde já, construir seu programa *Two Liners* e remeta-o para "SDI/DSI/DEGD/DIREC, 1.º andar — ala A — sala 112 — ramal 2426", para que possamos utilizar o gráfico criado publicando-o na capa com suas referências e a listagem. Aguardamos sua participação !!!

BYTE PAPO é uma publicação mensal do SERPRO — Serviço Federal de Processamento de Dados.

Edição

DIREC

Operacionalização

Superintendência de Recursos
Administrativos

Correspondência e informação

DSI/DEGD/DIREC — Ala "A" — sala
110 — 1.º andar — ramais: 2424,
2425, 2426 ou 2429 — Rogério Faezy
ou Ricardo Ribeiro

Responsáveis pelas seções

Novidades — Ricardo Ribeiro
Softpapo — Rogério Faezy
Escovando Bits — Ricardo Ribeiro
Macetes & Mutretas — Gustavo
Benigno
Desafio — Rogério Faezy
Conversão — Gustavo Benigno
Cursos — Luciane Shütte
Artigos — Luciane Shütte

Produção e Editoração

Carlos Seabra

Ilustração

Rogério Borges

Revisão

Simone Biehler Mateos

Paste-up

João Perianez Ruiz

Composição: LYMAC. *Fotolito:* BI-
NHOS. *Impressão:* Editora PARMA.
Tiragem: 4.000 exemplares

A revista não se responsabiliza por
matérias assinadas. As matérias po-
dem ser reproduzidas, se mencionada
a fonte. Venda proibida.

EDITORIAL

Uma máquina de datilografia sem fita, um carro sem combustível, um balão sem ar... Assim definiríamos um micro sem programas: um "brinquedo caro" sem utilidade.

É crescente a conscientização dos consumidores em relação à importância do software no Brasil. Algumas lojas afirmam ter alcançado um aumento de 70% das vendas de programas nos últimos meses — estes são comprados, em sua maioria, por empresas — ficando os amadores com uma percentagem menor, porém significativa, de "jogos", principalmente na linha SINCLAIR (TKs e similares).

Podemos dizer, grosso modo, que existem entre nós três tipos de programas: os importados (mais difundidos e em boa parte copiados), os adaptados e os aqui desenvolvidos.

Neste panorama, vivemos duas necessidades crescentes. Uma delas é que se valorize mais a potencialidade dos "pacotes prontos" à venda no mercado, na sua maioria produzidos no exterior. A outra é que se crie um software nacional. É claro que isto não significa que devamos tentar substituir todos os programas desenvolvidos no exterior pelos nossos! Afinal, o que seria de seu Imposto de Renda sem um Visicalc, da sua secretária sem um Magic Window, ou de seus filhos sem um Olympic Decathlon? O que queremos dizer é que ainda há muito o que fazer, pois temos problemas próprios, e para estes resta-nos encontrar soluções específicas. Afinal, possuímos o dom do "jeitinho" — e o que é um software senão uma forma de "dar um jeitinho"?

Este número da BYTE PAPO é dedicado à criatividade. Sim, à criatividade, que dá fruto à obra do artista, à Teoria da Relatividade, a uma solução simples para um problema complexo, enfim, ao desenvolvimento de um software "verde e amarelo", um software brasileiro.

LUCIANE SCHÜTTE
SDI/DSI/DEGD/DIREC

NOVIDADES

LEGENDA POR COMPUTADOR

A turma dos vídeo-clubes não dorme no ponto: estão usando a tecnologia dos micros para fazer legendas nos filmes. Mais uma utilidade para os micros. Parabéns.

VIDEO & MICRO

Estamos pesquisando a interface de vídeo interativo, uma placa que liga o vídeo-cassete ao microcomputador, e este passa a controlar todas as suas funções, FAST, FORWARD, REWIND, STOP, PLAY etc. (veja seção *Novidades* do BYTE PAPO n.º 1). É um dos melhores meios de promover instrução programada por microcomputador. Você poderá montar um banco de imagens no vídeo-cassete e exibir os quadros que desejar, bastando para isto que o aluno aperte uma tecla do micro. Fiquem de olho nesta!

MICRO TAGARELA

Quanto de nós ao observar essas máquinas loucas, fruto da tecnologia de nosso tempo, não exclamam admirados:

— Só falta falar!...

O micro é uma dessas que causam

este tipo de reação. "Só falta falar!" deixou de ser um mito para tornar-se uma realidade. Isto foi possível através do sintetizador de voz, as cordas vocais do computador.

O sintetizador é composto por uma placa ligada a um pequeno alto-falante acompanhada por um *software*. Ele é capaz de emitir até 57 fonemas que, combinados, produzem sons e palavras.

Os módulos que acompanham o *software* são o "teclado falante", "teste de compreensão", "tradutor de fonemas" e um módulo de ajuda a pessoas com deficiência visual.

O micro pode falar grosso, fino, normal, rápido ou lento, porém, só em inglês. Se você combinar os fonemas corretos poderá fazê-lo falar em português.

Enfim, você escolhe a voz que ele irá ter. Usufrua esta novidade e dê asas à sua imaginação na criação das mais diversas aplicações.

OLHA O PASSARINHO!...

Outra novidade interessante nos EUA é a placa da *Micro Works* que quando conectada a um APPLE permite que uma câmera fotográfica ou uma câmera de vídeo-cassete envie imagens para a tela do micro. A resolução é de 256×256 com 64 níveis de cinza.

As imagens podem ser varridas da tela por um programa em BASIC e impressas em uma impressora gráfica (vide capa desta edição).

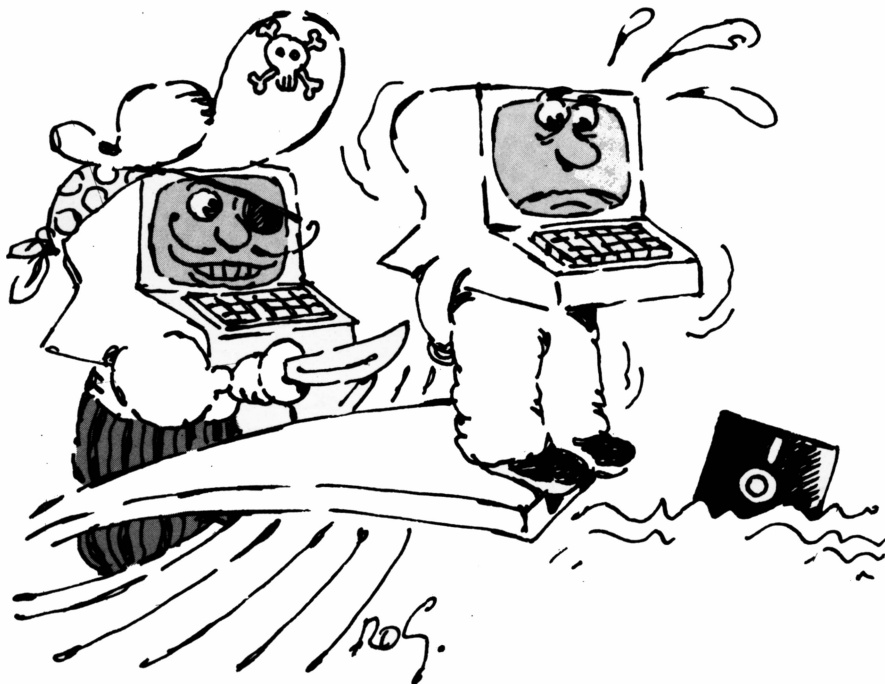
INCREMENTANDO O SEU MICRO

Foi lançado nos EUA, para a turma do APPLE, o *hardware* chamado de *Saybrook* — a terceira alternativa. Trata-se de uma placa que transforma o APPLE II ou APPLE IIe em um super-micro equipado com o co-processador 68000 de 16/32 *bits*, com capacidade de memória de até 512K em módulos de 128K, expansível até 8 *megabytes*, além de ser de 10 a 20 vezes mais rápido que o APPLE original. Está previsto para breve o sistema operacional UNIX. O sistema custa aproximadamente 1.150 dólares e acompanha os seguintes compiladores: BASIC, FORTRAN, USCD PASCAL e CP/M em 68K opcional.

WILD CARD

Falando de piratas & corsários, o número de copiadores de programas lacrados está aumentando. Parece que a turma está cada vez mais motivada a descobrir como copiar estes programas aparentemente "invioláveis". Já possuímos no Brasil o maior pirata dos piratas: *The Wild Card*.

Este "pirata eletrônico" é uma placa que copia qualquer tipo de *software* protegido transformando-os em programas deslacrados compatíveis com o DOS 3.3 do APPLE. Esta placa copia 48K de memória em cerca de 30 segundos e é fabricada pela *East Side Software Co. NY*. Mas os nossos piratas tropicais já a possuem em uma versão nacional que pode ser encontrada na *Livraria Ciência Moderna Computação Ltda.* no Rio de Janeiro.



MICRO-PROGRAMADOR

Um dos programas mais cogitados hoje em dia são aqueles considerados "programadores" ou seja, programas que escrevem programas para você. Com simples toques de teclas selecionáveis por "menus", você escolhe o que fazer, e o micro-computador em poucos minutos (cerca de 10) cria um programa pronto para ser executado, sem erros e sem a necessidade de testes.

Alguns deles chegam ao ponto de compilarem o programa-fonte, testar

os programas fonte e compilado quanto ao ideal de espaço e velocidade e optar pelo melhor. É quase que um pacote pronto para elaboração de programas não muito complexos. As listagens dos programas podem ser listadas para que o usuário as estude.

Como exemplo destes "programadores" temos, para a linha APPLE, *The Last One*. Para a linha TRS-80, *The Last One*, *Criador/Relator*, *Quickpro* e *Producer*.

Estamos de olho neles!!!

ESCREVENDO NA TELA

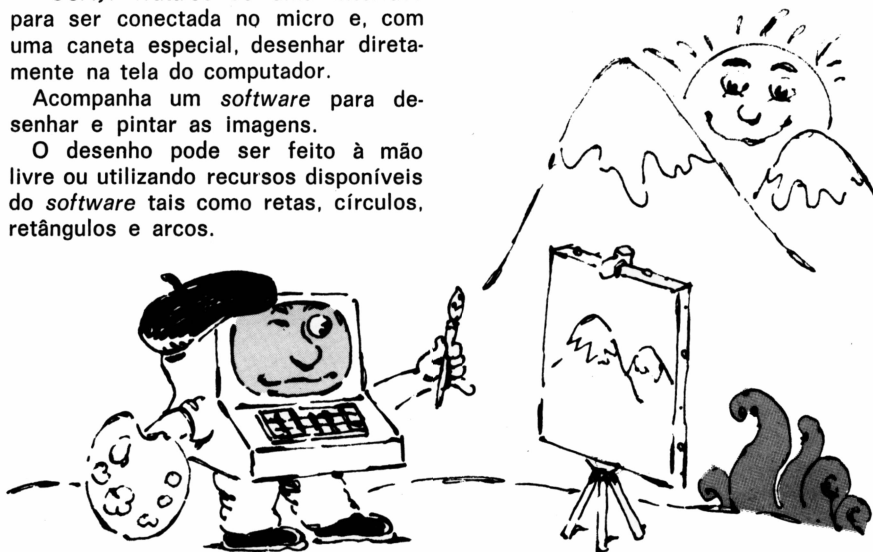
Para os fascinados em gráficos e desenhos nos microcomputadores, preparem-se para a chegada no mercado da *Light Pen* (Gibson Laboratories — USA). Trata-se de uma interface para ser conectada no micro e, com uma caneta especial, desenhar diretamente na tela do computador.

Acompanha um *software* para desenhar e pintar as imagens.

O desenho pode ser feito à mão livre ou utilizando recursos disponíveis do *software* tais como retas, círculos, retângulos e arcos.

A parte mais interessante é colorir o desenho, bastando para isto selecionar uma cor com a ponta da caneta (como se a enchesse com tinta) e apontar para a área do desenho a ser colorida. Esta interface já está também disponível para o CP-500 e colorido somente para o APPLE.

ROGÉRIO FAEZY
SDI/DEGD/DIREC
Ramal 2426



VISILOT-VISITREND

Para aquelas pessoas que têm uma atividade que envolve a elaboração de trabalhos estatísticos ou que têm que construir gráficos demonstrativos, e que ainda se utilizam de lápis e papel, quando muito de uma calculadora, mostraremos neste artigo um aplicativo de muita valia.

VISILOT/VISITREND é um *software* da linha APPLE, que roda em qualquer similar brasileiro (Microengenho, Unitron, Maxxi e outros) criado pela Personal Software Inc. — USA.

Na verdade são dois aplicativos distintos, cada qual desempenhando uma tarefa específica, mas que permitem uma comunicação dos dados entre si.

VISITREND se encarrega da elaboração de séries e tabelas, cálculos e estatísticas, cabendo ao VISILOT a confecção de gráficos demonstrativos dos mais diversos.

CARACTERÍSTICAS

Vamos agora descrever algumas facilidades oferecidas por cada um destes dois aplicativos.

VISITREND

A sua característica principal está voltada para a análise de séries temporais sob o ponto de vista dos dados.

Permite a entrada de qualquer série, bem como a sua alteração. Permite a visualização da série sob diversos formatos; inclusão e exclusão de dados, mudança nas especificações da série, geração de séries Aritméticas e

Geométricas, bem como Interpolações.

Pode-se colocar no vídeo mais de uma série, permitindo uma análise comparativa.

Cálculos estatísticos, como VALOR MÍNIMO, VALOR MÁXIMO, MÉDIA ARITMÉTICA, VARIÂNCIA, DESVIO PADRÃO e COEFICIENTE DE CORRELAÇÃO, são obtidos a partir de qualquer série na memória do micro.

Realiza REGRESSÃO LINEAR MÚLTIPLA (Método dos Mínimos Quadrados), podendo-se definir até 5 variáveis independentes.

Também calcula: ESTATÍSTICA-T, ESTATÍSTICA-R, CORREÇÃO DE R, ERRO PADRÃO DA REGRESSÃO E DOS COEFICIENTES, SOMA DO RESIDUAL AO QUADRADO, ESTATÍSTICA-F e ESTATÍSTICA DE DURBIN-WATSON.

Sobre qualquer série, poderão ser geradas duas outras de muita importância: SÉRIE AJUSTADA (FITTED SERIES) e SÉRIE RESIDUAL.

Permite que você possa fazer um estudo de TENDÊNCIAS, gerando uma série LINEAR AJUSTADA para uma variável dependente qualquer.

O pacote dispõe de algumas funções de grande utilidade e fácil manipulação:

- Cálculo da MÉDIA MÓVEL (MOVING AVERAGE), utilizando o métono $N+1$.
- UNIFORMIZAÇÃO DE DADOS (SMOOTING): Executa uma uniformização exponencial da série. Exige um fator de uniformização que é calculado segundo a fórmula:

$Y(i) = X(i-1) + (1-x) * Y(i-1)$
onde $X(i)$ são os pontos da nova série e $Y(i)$ os pontos da antiga.

- **% CHANGE:** Calcula o percentual de variação entre os dados n e $n+1$ e armazena o resultado no dado $n+1$ da nova série. Tem como objetivo verificar os picos de divergência na série original.
- **LAG:** Faz uma transferência dos valores da série tantos períodos acima quanto se desejar.
- **LEAD:** Transfere os valores da série tantos períodos abaixo quanto se desejar.
- **TOTAL:** Gera uma série a partir de outra, onde os valores gerados são acumulativos.
- **XFORM:** Gera novas séries a partir de qualquer série ou grupo de séries já existentes, utilizando operações matemáticas. Utiliza os operadores aritméticos usuais, os operadores lógicos AND, OR, NOT, os operadores relacionais $<$, $<=$, $=$, $>$, $>=$, as funções SGN, INT, EXP, RND, ABS e as funções trigonométricas SENO, COSSENO, TANGENTE, e ARCO-TANGENTE.

VISILOT

Da mesma forma que o VISITREND, o VISILOT também permite fazer uma análise das séries, agora oferecendo uma visão gráfica graças a diversas funções.

As funções gráficas permitem a elaboração dos seguintes gráficos:

LINHA.

BARRAS: Que podem ser

- barra sobre a ordenada (eixo x);
- barra à direita da ordenada;
- barra à esquerda da ordenada.

ÁREA.

SETOR: Gráfico no formato de uma torta, onde serão distribuídas percentualmente as séries em questão.

HI-LO: É um gráfico de linha onde serão visualizadas duas séries, uma superior e outra inferior. Permite ter uma visão comparativa das duas séries.

SCATTER: Este gráfico será composto por duas séries, onde os valores da primeira aparecerão na coordenada X e os valores da segunda aparecerão na coordenada Y. Ou seja, uma série será função da outra.

Algumas facilidades na confecção destes gráficos também são oferecidas:

- **WINDOW:** Permite que você defina janelas em seu vídeo, possibilitando que até dois gráficos sejam visualizados ao mesmo tempo. Pode-se ter duas janelas horizontais (partes superior e inferior da tela) ou duas janelas verticais (partes direita e esquerda da tela).
- **PRINT:** Saída em impressora de seu gráfico. Neste caso existe limitação, onde se exige a existência de alguma Impressora gráfica (tipo GRAFIX, EPSON, PAPER TIGER etc.).
- **PIXSAVE:** Permite que você armazene seu gráfico em disco, podendo recuperá-lo posteriormente, listando-o na sua impressora.
- Você poderá colocar títulos em seu gráfico, na parte superior, inferior e esquerda.
- Se lhe interessa uma melhor visualização de sua série, você poderá definir uma nova escala, obtendo um gráfico em maior ou menor dimensão.
- Uma visualização setorial do gráfico de sua(s) série(s) lhe interessa? Então use a função de gradeamento, onde seu gráfico será subdividido em linhas verticais e horizontais, favorecendo um estudo de diferentes períodos.
- Você deseja fazer um gráfico de apenas um intervalo de sua série. Então mude os valores inicial e final da ordenada X.
- Cores? Mude a cor de fundo para laranja e faça o gráfico em verde. Que tal? Existem outras cores se você desejar.
- Se você deseja plotar mais de uma série sobre o mesmo gráfico, faça

Isto usando a função OVERPLAY. Use cores diferentes para cada caso. Se a visualização não ficou bem definida, mude também a escala.

CONCLUINDO

Se o seu trabalho envolve este tipo de cálculo, ou se você é um inte-

ressado nas teorias estatísticas, e possui um micro da linha APPLE à sua disposição, experimente este *software* e veja o quanto ele o poderá ajudar.

RICARDO RIBEIRO
SDI/DEGD/DIREC
Ramal 2426

Olympic Decathlon

Para todos os atletas, de todas as idades, aqui vai uma dica de um *software* que vai proporcionar horas de lazer para você, seus amigos e filhos.

OLYMPIC DECATHLON é um programa muito interessante com bons efeitos gráficos.

Você participará de dez competições, e quem fizer o maior número de pontos será o ganhador da medalha de ouro.

Lembramos que o recordista do decatlo é Bruce Jernner, que conseguiu um total de 8.618 pontos nas Olimpíadas de 1976, em Montreal, Canadá.

Podem participar até oito "atletas" nas competições.

Aqui vai a lista das 10 modalidades:

- 1) 100 metros rasos
- 2) Salto em distância
- 3) Arremesso de peso
- 4) Salto em altura
- 5) 400 metros rasos
- 6) 110 metros com barreira
- 7) Arremesso de disco
- 8) Salto com vara
- 9) Lançamento de dardo
- 10) 1.500 metros

Agora daremos uma rápida descrição de como jogar o decatlo, tanto na linha APPLE como na linha TRS-80.

Modalidade 1

TRS-80: usa as teclas 1 e 2 e as setas para esquerda e direita.

APPLE: idem.

Modalidade 2

TRS-80: usa a barra de espaço, seta para baixo e <ENTER>.

APPLE: usa a barra de espaço, a tecla × e <CR>.

Modalidade 3

TRS-80: usa a seta para esquerda e seta para cima.

APPLE: usa o *joystick* (aperte ESC para sair se não possuir o *joystick*).

Modalidade 4

TRS-80: usa a barra de espaço, a seta para baixo e <ENTER>.

APPLE: usa a barra de espaço, a tecla × e o <CR>.

Modalidade 5

TRS-80: usa as teclas 1 e 2 e as setas para direita e esquerda.

APPLE: idem.

Modalidade 6

TRS-80: usa as teclas 1 e 2 e as setas para esquerda e direita.

APPLE: usa os botões do *joystick* (aperte <ESC> para sair se não possuir o *joystick*).

Modalidade 7

TRS-80: usa a barra de espaço e <ENTER>.

APPLE: usa a barra de espaço e <CR>.

Modalidade 8

TRS-80: usa as setas para esquerda e direita, seta para baixo e <CLEAR>.

APPLE: usa as setas para esquerda e direita, a tecla T e <CR>.

Modalidade 9

TRS-80: usa as setas para esquerda e direita, seta para cima e <ENTER>.

APPLE: usa as setas para esquerda e direita, seta para cima e <CR>.

Modalidade 10

TRS-80: usa as teclas:

A, W, S, Z para o jogador 1

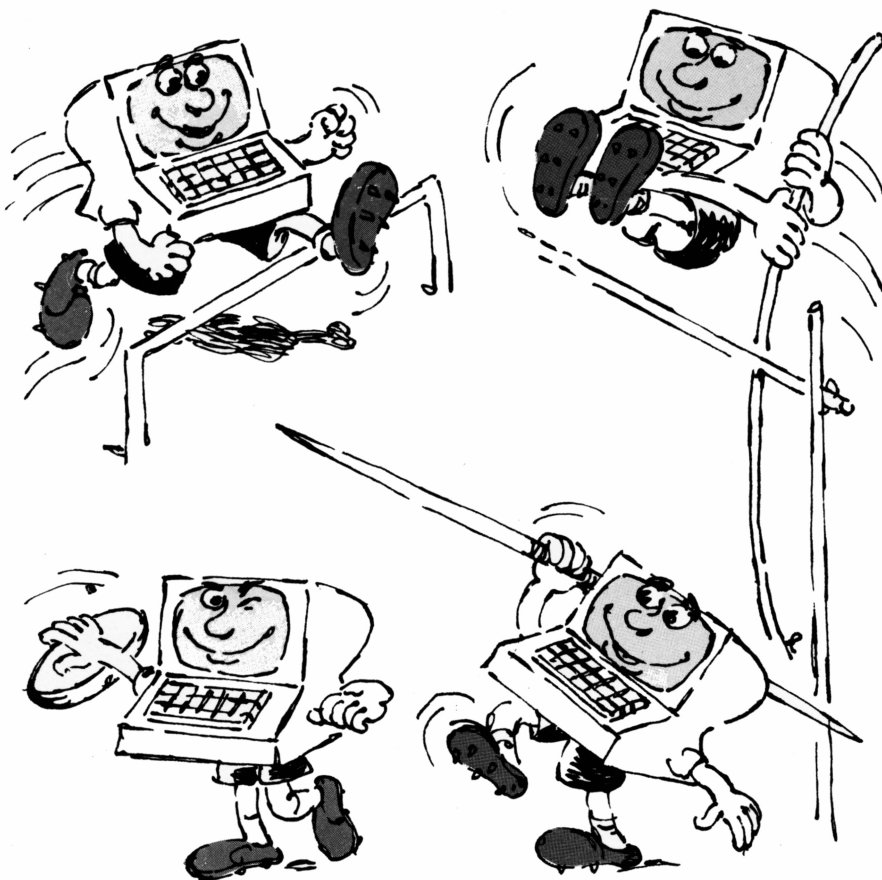
P, L, ;, . para o jogador 2

APPLE: idem.

Esperamos que você se distraia bastante com este soft.

Vá treinando para as Olimpíadas de Los Angeles!!!

GUSTAVO BENIGNO
SDI/DEGD/DIREC
Ramal 2426



CONVERSÃO

Dando continuação ao número passado, estamos novamente lhe oferecendo mais uma série de instruções para a conversão de programas.

Statement

SET

S.

O *statement* SET é usado pelo TRS-80 para acender um bloco gráfico no vídeo.

Para acender o bloco desejado precisamos fornecer as coordenadas entre parênteses. Exemplo: SET (5,8) instrui o computador a acender o bloco que está na quinta linha e oitava coluna.

PROGRAMA TESTE

```
10 REM PROGRAMA TESTE DE SET
20 PRINT "ENTRE C/ A COORDENADA X";
30 INPUT X
40 PRINT "ENTRE C/ A COORDENADA Y";
50 INPUT Y
60 SET (X,Y)
70 PRINT "SET PASSOU NO TESTE"
80 END
```

EXECUÇÃO

```
ENTRE C/ A COORDENADA X ? 65
ENTRE C/ A COORDENADA Y ? 40
SET PASSOU NO TESTE
```

ABREVIações

Alguns computadores aceitam S. como abreviação.

Statement

PRINT AT

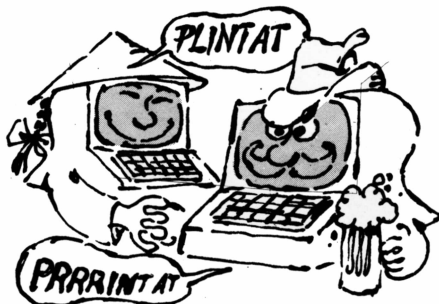
PRINT AT é usado pelo TRS-80 Level I para indicar a impressão de alguma coisa a partir de uma determinada posição. O valor de AT pode ser um número, uma variável numérica ou uma operação matemática. Uma vírgula ou ponto e vírgula podem ser inseridos entre o valor de AT e a *string*. São 1.024 posições de vídeo: 16 linhas e 64 colunas.

PROGRAMA TESTE

```
10 REM TESTE PARA O PRINT AT
20 PRINT AT 128,"2. ESTA LINHA E'
   IMPRESSA DEPOIS DA 1a"
30 PRINT AT 0,"1. O STATEMENT AT PASSOU
   PELO TESTE"
40 END
```

EXECUÇÃO

1. O STATEMENT PRINT AT PASSOU PELO TESTE
2. ESTA LINHA E' IMPRESSA DEPOIS DA LINHA 1



Função

SIN

A função SIN calcula o seno do ângulo onde este é expresso em radianos (não em graus). Um radiano é aproximadamente 57 graus.

PROGRAMA TESTE

```
10 REM TESTE PARA O SIN
20 PRINT "ENTRE COM O ANGULO (RADIANOS)"
30 INPUT R
40 Y = SIN (R)
50 PRINT "O SENO DE " ; R ; "E' " ; Y
60 END
```

EXECUÇÃO

ENTRE COM O ANGULO (RADIANOS) ? 1
O SENO DE 1 E' .841471

SUBROTINA DE CONVERSÃO

```
30364 X=X*57.29578
30366 IF X=0 THEN 30408
30368 Z = ABS(X)/X
30370 C=X:X=Z*X:IF X<360 THEN 30378
30376 X=X-INT(X/360)*360
30378 IF X<=90 THEN 30398
30380 X=X/90:Y=INT(X):X=(X-Y)*90
30386 ON Y GOTO 30388,30392,30396
30388 X=90-X:GOTO 30398
30392 X=X-X:GOTO 30398
30396 X=X-90
30398 X=Z*X/57.29578
30400 IF ABS(X)<2.48616E-4 THEN 30408
30402 Y=X*X
30404 Y=((((Y/72-1)*Y/42+1)*Y/20-1)*Y/6+
1)*X
30406 GOTO 30410
30408 Y=X
30410 X=C/57.29578
30412 RETURN
```

GUSTAVO BENIGNO
SDI/DEGD/DIREC
Ramal 2426



A sociedade da informação

Se fizermos uma retrospectiva no tempo, notaremos que cada período da História da Humanidade se caracterizou por um determinado padrão social. Atualmente, ao caracterizarmos nossa época, fatalmente teremos que citar o fenômeno da INFORMÁTICA como notadamente marcante.

O microcomputador é um veículo das transformações advindas deste período. Contudo, mais importante que encará-lo como um fruto em si, é notá-lo como peça de uma engrenagem maior, a engrenagem da teleinformática. Esta, por sua vez, é um ponto chave para o que convencionamos chamar "Sociedade da Informação".

Será esta uma perspectiva longínqua? Não: mesmo no Brasil, muita coisa vem sendo feita, como, por exemplo, o *Projeto Ciranda*, primeira rede nacional de computadores pessoais, desenvolvida pela Embratel, que colocou mais de 2000 funcionários interligados por microcomputadores pessoais.

As informações trafegam pelas linhas telefônicas convencionais trazendo respostas imediatas. O *Ciranda* estende sua malha por todos os Estados e Territórios, com exceção de Fernando de Noronha.

Há duas espécies de contatos que podem ser estabelecidos entre os usuários, um com o outro, ou do usuário com a Central do Rio de Janeiro,

onde existe um minicomputador (COBRA 530), dedicado ao Projeto.

Os "cirandeiros", como se auto-nominam os usuários do sistema, já realizaram, dentre outras "travessuras", ajuda mútua quanto a um problema de computação; troca de programas dos mais diversos tipos (de jogos a utilitários importantes); informações quanto ao preenchimento do formulário do Imposto de Renda, congelamento de alimentos, aulas de matemática para ajudar as crianças em seus deveres escolares, receitas de bolo; explicações de como reconhecer os sintomas de doenças mais comuns, além de uma lista com nomes, endereços e telefones de médicos capazes de prestar assistência; resposta a questões de História do Brasil, Geografia, Leis, Ciência, Tecnologia etc. Porém, o campeão de chamadas é o "correio eletrônico", onde é estabelecido um intercâmbio de todo tipo de informações e pedidos.

No momento, a utilização da Rede está restrita aos funcionários da Embratel. Contudo, a experiência do *Ciranda* tende a expandir-se pela comunidade, seja pela abertura à participação externa, ou pela adoção de outras formas de operacionalizar a mesma idéia básica.

Dentro deste enfoque, estamos lançando a coluna "BYTENDO PAPO" que tem como finalidade facilitar a troca

de idéias e conhecimentos entre pessoas com áreas de interesse comum em micros. Envie-nos seu NOME, LOCALIZAÇÃO, RAMAL e ÁREA DE INTERESSE para publicação na BYTE PAPO, permitindo que pessoas que se interessem pelo mesmo assunto possam contactá-lo diretamente. Ensaíamos assim nosso primeiro passo a caminho desta nova sociedade.

Enfim, o significado desta tecnologia é mais profundo do que possa parecer de início, e sua influência há de ser percebida em pouco tempo, tornando a "Sociedade da Informação" mais que um mero exercício de futurologia.

LUCIANE SCHÜTTE
SDI/DSI/DEGD/DIREC
Ramal 2426

O microcomputador no ensino

DOS SONS GUTURAIS AOS BITS E BYTES

A velocidade das mudanças tecnológicas tem aumentado surpreendentemente nos últimos tempos. O que influencia a rapidez destas transformações? Que "coisa" tão importante liga os homens e os faz colocarem foguetes na lua, água no deserto, carros velozes nas ruas, computadores no trabalho? Provavelmente, nada disso teria sido obtido sem a existência de um fator intimamente ligado ao desenvolvimento e sobrevivência da humanidade. Este fator é a "comunicação", presente em todos os níveis, da escrita à fala, dos sons guturais aos *bits* e *bytes* de um computador.

Uma nova era surge, fruto da comunicação, a era da informática. Tantos se negam a compreendê-la e aceitá-la, cegos que estão pelo "mito dos poderosos cérebros eletrônicos", e este não passa, infelizmente, de um produto da atual ignorância sobre a simplicidade desta caixa preta, fruto apenas do sistema binário e da imaginação humana.

Dentro desta perspectiva era natural portanto que a informática lanças-

se sua rede de influências sobre uma área do conhecimento intimamente ligada à comunicação, a EDUCAÇÃO. A novidade é recente e controversa... Mas as controvérsias estão tremendamente ligadas a mitos e estereótipos da computação.

Muitos pais temem que seus filhos se tornem crianças "robotizadas", mas "robotizar" uma criança independe do instrumento que se usa para ensiná-la, pode-se "robotizá-la" com uma simples aula de memorização dada por um professor humano qualquer... O que robotiza é o método de ensino, e o fato de o computador ser uma máquina não implica na conseqüente mecanização de seus ensinamentos.

O MICROCOMPUTADOR SUBSTITUIRÁ O PROFESSOR?

Outro estereótipo vigente nesta área é a "fatal substituição do homem pela máquina". A professora da Universidade Federal do Rio Grande do Sul (UFRGS), Lucila Santarosa, ataca bem esta problemática em seu depoimento à revista *Informática*: "Antes de ser ameaça como substituto do professor, seja o computador percebido e utiliza-

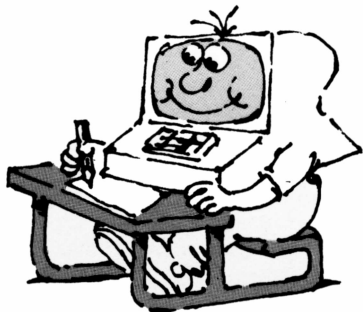
do como recurso que libere o professor das tarefas rotineiras que absorvem o seu tempo, que ele poderá ocupar na integração com o aluno... Embora pareça paradoxal, o computador, que freqüentemente é acusado de desumanizar o ensino, talvez venha a contribuir para o aspecto humanístico da avaliação".

O ensino através da microcomputação vem, então, como uma arma poderosa, munida de agentes motivacionais e esquemas não punitivos de aprendizagem. "Aprender brincando", eis o grande instrumento que desperta a mente e o coração, não só de crianças, mas de adolescentes e adultos. Este é o segredo que todos aqueles que se aventuram nesta área devem usar sempre.

O resultado da soma destes fatores é que o ensino, através da microcomputação faz com que o aprendizado tenda cada vez mais para a personalização, fator de extrema importância no aperfeiçoamento da Educação. Prevê-se, inclusive, para o futuro, o momento em que cada indivíduo possa seguir seu próprio ritmo de estudo, livre da atual escravização a um tempo padrão para início e término dos cursos.

APRENDENDO A APRENDER

Bem, mas não só através da instrução programada que ocorre aprendizado pelo microcomputador. Projetos como o LOGO de Seymour Papert, assim o demonstram. Seguindo um mo-



delo de aprendizagem espontânea, citado em seu livro *Mindstorms: Children, Computers and Powerful Ideas*, Papert busca utilizar o LOGO partindo do princípio que "nenhuma pessoa consegue dar instruções a um computador para realizar determinada tarefa sem que ela própria consiga fazê-la".

Trabalhos sobre este projeto realizaram-se na Universidade de Campinas, sob orientação da professora Maria Cecília Calani, recebendo crianças com idade em torno de 10 anos. Numa entrevista à revista *Veja* a professora Calani declara: "As crianças aprendem a aprender".

Através de uma tartaruguinha (um triângulo na tela) elas deduzem, por tentativa e erro, a importância da geometria e a combinação de vários ângulos na formação de figuras. Para isto, utilizam o LOGO, que contém instruções como "GIRE 90", "VÁ PARA CIMA", que a criança utiliza para mover a tartaruga e formar suas figuras. Mas não só à geometria se resume o LOGO, nem só ao ensino de crianças, outras potencialidades podem ser desenvolvidas, porém não são tão exploradas.

MICROINFORMÁTICA: UMA NOVA ETAPA NA HISTÓRIA DA EDUCAÇÃO

Em suma, com o crescente barateamento dos custos, os micros tendem a se popularizar. Com isto, um novo mundo se abre. Não se espera que os erros da atualidade se repitam neste universo, mas sim que ele traga, junto às mudanças de "instrumento" de ensino, a inovação de "métodos" de ensino, quando, conforme palavras do professor Wilson José Tucci: "...Na ânsia de esgotar as possibilidades da máquina o aluno acaba se superando". Cabe aos professores esgotar as possibilidades desta máquina na busca de superar sua didática.

LUCIANE SCHÜTTE
SDI/DSI/DEGD/DIREC
Ramal 2426

Tabelas de formas-1

Você provavelmente já utilizou gráficos em baixa e alta resolução no microcomputador. Neste caso você também já deve ter sentido a dificuldade que há em manipular as figuras na tela. Por exemplo, pode-se desejar rodar uma figura sobre um eixo, ou fazê-la aparecer maior ou menor na tela. As formas em alta resolução têm essas possibilidades, basta utilizar para isto o conceito de **tabelas de formas** que nos propomos a ensinar neste curso voltado para micros da linha APPLE.

Para que o processo de construir uma tabela de formas se torne mais claro, vamos construir um exemplo onde, por indução, se definem os passos-padrão para a utilização de qualquer forma. Para isto, dividimos as ações em 8 passos e o curso em 4 aulas com 2 passos cada. Ao final da última aula, você deverá estar capacitado a desenhar e manipular qualquer forma imaginada.

PASSO 1

As formas em alta resolução requerem planejamento. Em essência, faz-se mais que dizer ao computador para desenhar uma linha do ponto A ao ponto B. Usando formas, descreve-se inteiramente a figura antes de orientar o computador para desenhá-la. Definem-se formas de alta resolução em uma TABELA DE FORMAS, chamada assim por conter as características codificadas da figura a ser desenhada.

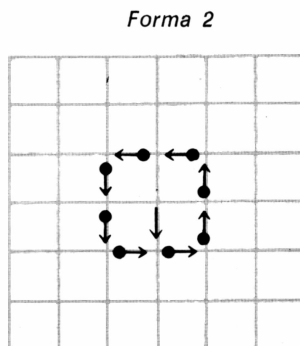
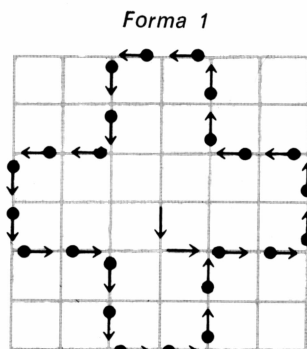
Este primeiro passo constitui-se na

definição da figura desejada. Para isto deve-se desenhar a(s) forma(s) com vetores (de preferência use papel quadriculado).

Adotaremos aqui:

- movimento sem desenhar
- movimento desenhando

No nosso caso construiremos a tabela de duas formas em conjunto:



PASSO 2

Codificar os vetores em binário conforme a seguinte tabela:

VETORES DE DESENHO E SEUS CÓDIGOS BINÁRIOS

SIMB.	AÇÃO	C.B.	C.D.
↑	Move para cima sem desenhar	000	0
→	Move para direita sem desenhar	001	1
↓	Move para baixo sem desenhar	010	2
←	Move para esquerda sem desenhar	011	3
↑	Move para cima desenhando	100	4
→	Move para direita desenhando	101	5
↓	Move para baixo desenhando	110	6
←	Move para esquerda desenhando	111	7

Abrindo os vetores das nossas formas como uma "lingüiça" e fornecendo a cada vetor um código, temos:

VETORES	CODIGOS
↓	010
↓	001
→	101
→	101
→	100
→	100
→	100
→	100
→	111
→	111
→	100
→	100
→	111
→	111
→	110
→	110
→	111
→	111
→	110
→	110
→	101
→	101
→	110
→	110
→	101
→	101
→	100
→	100

Forma 1

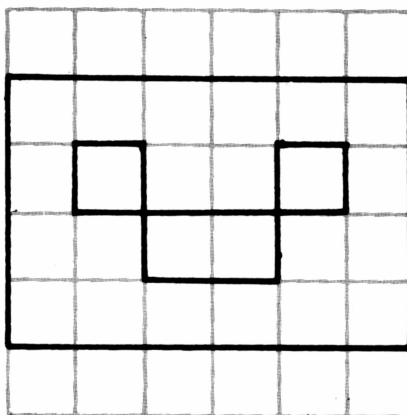
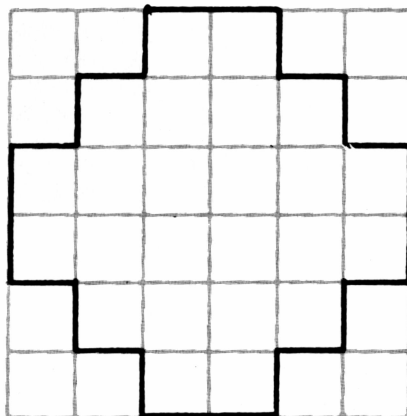
VETORES	CODIGOS
↓	010
↓	101
↓	100
↓	100
→	111
→	111
→	110
→	110
→	101

Forma 2

Acompanhe-nos até a quarta aula quando estas figuras estarão "pintando o sete" no seu vídeo!!

Aproveitamos para sugerir que você crie agora sua(s) figura(s) e pratique os **Passos 1 e 2** com ela(s), aproveitando para utilizá(s) até o final do curso como exercício, pois só a prática irá torná-lo realmente um profundo conhecedor de TABELAS DE FORMAS.

Para aqueles que não pensaram em nada especial sugerimos tentarem praticar durante o curso com as figuras que se seguem:



Observe o desenho do **Passo 1**. Você há de notar que na verdade não existe nenhum segredo especial, concorda?

LUCIANE SCHÜTTE
SDI/DSI/DEGD/DIREC
Ramal 2426

DESAFIO

SOLUÇÃO DO DESAFIO 1

Esperamos que você tenha tentado resolver o problema proposto. Aqui vai a explicação:

O programa fornecido está correto e, ao pé da letra, o micro consegue armazenar os 70000 caracteres na memória de apenas 64000. É apenas uma questão de interpretação do que seja "armazenar" a informação. O BASIC do APPLE, do TRS-80 ou do CP/M armazenam as variáveis da seguinte forma:

endereço	DOS
mais	
baixo	
	PROGRAMA
	EM
	BASIC
	VARIAVEIS
	SIMPLES
	MATRIZES
	E
	VETORES
	ESPAÇO
	LIVRE
	STRINGS
endereço	
mais alto	

OBS.: Repare que o programa cresce de baixo para cima e as *strings* de cima para baixo.

Conforme o esquema, podemos ver que cada parte ocupa um pedaço da memória do micro. Logo após o seu programa vem a área reservada para as variáveis simples e depois os vetores e matrizes de seu programa. Começando do fim da memória (endereço mais alto) para o início da memória (endereço mais baixo), está a área de armazenamento das *strings* (cadeia de caracteres). Nesta área só é armazenada a informação propriamente dita. A área de variáveis simples tem o seguinte formato:

VARIAVEIS NUMÉRICAS	VARIAVEIS STRINGS
TIPO DA VARIÁVEL	TIPO DA VARIÁVEL
2a LETRA	2a LETRA
1a LETRA	1a LETRA
V A	TAM.
L O	E N D
R	E R E
	M O

As variáveis numéricas são armazenadas de uma forma diferente das variáveis *strings*. Seus valores estão logo após as informações pertinentes às variáveis, tais como: nome da variável (1.ª e 2.ª letra) e o seu tipo.

Já as variáveis *strings* possuem após a informação básica (tipo e nome) o tamanho da *string* e o endereço de memória onde está a cadeia de caracteres pertencentes a esta variável. Ha! Ha! Aí está o macete! Quando escrevemos programas do tipo dado no desafio, vejamos:

```
40 A$(I) = "1234567890"
```

A *string* é a própria cadeia mostrada no exemplo da linha 40.

Resumindo, como não fazemos nenhuma operação com a *string* (exemplo: concatenação, funções do tipo LEFT\$, RIGHT\$, MID\$ etc.), todas as variáveis usadas A\$(i) possuem no parâmetro endereço da *string*, uma referência para dentro do programa em BASIC, ou melhor, apontam para a *string* da linha 40.

Resultado, a área de *strings* no topo da memória não é utilizada. Na verdade você só gastou os 3 bytes (tamanho, e os 2 bytes para o endereço) para os 7000 elementos, totalizando 21000. Eis o motivo pelo qual coube na memória. A *string* destes elementos propriamente dita, já está armazenada na memória, só que no meio dos comandos do seu programa.

Para você notar a diferença, digite o seguinte trecho de programa:

```
10 DIM A$(1000)
20 FOR I = 1 TO 1000
30 PRINT "ELEMENTO = "; I
40 INPUT "ENTRE COM UM STRING "; A$(I)
50 NEXT I
60 END
```

Agora execute-o e observe até quantas *strings* você consegue armazenar antes de esgotar a memória.

É bem menos não é? As *strings* agora não estão mais fixas no corpo do programa, logo são transferidas para sua área especial no fim da memória.

A seguir mostraremos o esquema de armazenamento dos dois programas.

PROGRAMA 1 — O DESAFIO

endereço	-----	
de	10 DIM A\$(7000)	!
início	20 ...	!
do	30 ...	!
programa	40 A\$(I) =	!
	"1234567890" <-----	!
	:GOTO ...	!
	!
	-----	!
início da	TIPO = 03	!
area das	2a LETRA = " "	!
variáveis	1a LETRA = "A"	!
	-----	!
1R0	TAMANHO = 10	!
ELEMENTO	ENDEREÇO	!----->!
	DO STRING	!
	-----	!
2D0	TAMANHO = 10	!
ELEMENTO	ENDEREÇO	!----->!
	DO STRING	!
	-----	!
	!
	-----	!
ELEMENTO	TAMANHO = 10	!
7000	ENDEREÇO	!----->!
	DO STRING	!
	-----	!
	!
	-----	!
	AREA	!
	DOS	!
	STRINGS	!
	-----	!



PROGRAMA 2

	AREA DO PROGRAMA	
	
	10 DIM A\$(1000)	
	
AREA DAS VARIÁVEIS SIMPLES	TIPO = 03	
	2DA LETRA = "	
	1RA LETRA = "A"	
	TAMANHO = 10	
ELEMENTO 1	ENDEREÇO	
	DO STRING	
	
ELEMENTO 1000	TAMANHO = 10	
	ENDEREÇO	
	DO STRING	
	
AREA DOS STRINGS	JOAQUIM	
	1234567890	
	ABCDEFGHIJ	
	
TOPO DA MEMORIA		

ROGÉRIO LEMOS DE FAEZY
SDI/DSI/DEGD/DIREC
Ramal 2426



UM PROGRAMA QUE VOCÊ NÃO UTILIZA TODOS OS DIAS MAS É INTERESSANTE

Aqui está um pequeno programa que irá converter os números arábicos em romanos (fantástico, hein?). Esta é uma sugestão. Tente reescrevê-lo em menos linhas.

Aceita o desafio?

```
10 TEXT:
    HOME:
    PRINT"CONVERSOR DE NUMEROS ARABICOS
EM ROMANOS (1-9999)"
20 X$="IVXCDMLXCIUX":
    GOTO 130
30 PRINT:
    INPUT"NUMERO ARABICO: ";N$
40 N=VAL(N$):
    L=LEN(N$):
    IF N<1 OR N>9999 THEN 30
50 IF L<4 THEN FOR I=1 TO 4-L:
    N$="0"+N$:
    NEXT
60 FOR I=1 TO 4:
    N(I)=VAL(MID$(N$,I,1)):
    NEXT
70 PRINT:
    PRINT"NUMEROS ROMANOS: ";
80 FOR I=1 TO 4:
    RESTORE:
    FOR J=1 TO N(I)+1:
    READ A$:
    NEXT:
    IF A$="0" THEN 120
85 DATA 0,1,11,111,12,2,21,211,2111,13
90 IF I=1 AND N(I)<4 THEN FOR X=1 TO
N(I):
    PRINT"M",:
    NEXT:
    GOTO 120
100 FOR X=1 TO LEN(A$):
    PRINT
    MID$(X$,3*(I-1)+VAL(MID$(A$,X,1)),1):
    NEXT
110 IF I=1 AND N(I)>3 THEN L=LEN(A$):
    CALL -998:
    HTAB PEEK(36)+1-L:
    FOR X=1 TO L:
    PRINT CHR$(95):
    NEXT:
    CALL -922
120 NEXT
130 PRINT:
    PRINT:
    FOR I=1 TO 40:
    PRINT"--":
    NEXT:
    GOTO 30
```

ROGÉRIO FAEZY
SDI/DEGD/DIREC
Ramal 2426

MACETES & MUTRETAS

CUIDADOS COM O DISQUETE



"Seu disquete flexível,
Cuide dele, por favor,
Só o tire do envelope
Pra ir pro acionador!
Não lhe toque a superfície,
Nem de leve, com a mão.
Evite qualquer lugar onde haja
Imantação,
Aliás — e especialmente —
A sua televisão.
Não o dobre, nem o vergue
Porque ele não atura.
De 10 a 50 graus
É sua temperatura,
Não exceda no calor,
Nem lhe refresque a memória
Cuidado c'o cafezinho
E não venha com estória,
Que foi um azar danado,
Que não hove a intenção...
Afinal, o seu disquete
Guarda a sua informação!!!"

LUIZ EUGÊNIO BARBOZA

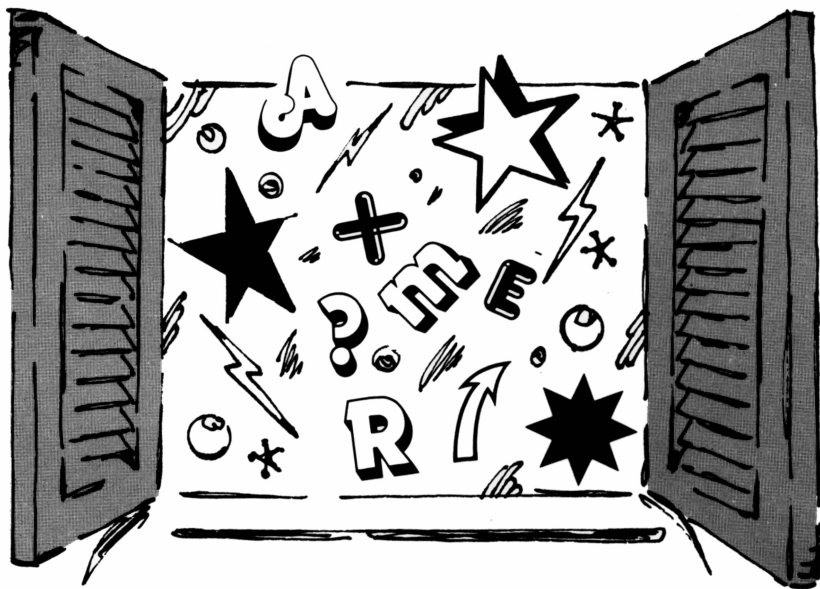
Você sabia que um disquete virgem guardado por muito tempo em sua caixa está mais sujeito a dar erros de gravação após formatado (inicializado) do que um inicializado e guardado na mesma caixa?

Você sabia que para reduzir a pro-

babilidade de erros de gravação e leitura de disquetes devemos centralizar o mesmo em sua jaqueta antes de inseri-lo no acionador de discos?

*ROGÉRIO FAEZY
SDI/DEGD/DIREC
Ramal 2426*

DICAS DO MAGIC WINDOW



"Diz-me com quem andas e te direi quem és!"

Magic Window é um dos processadores de texto mais simples e eficientes da linha APPLE. E quem ainda não ouviu falar de um Processador de Texto?

Para aqueles que o desconhecem, um P.T. nada mais é que um *software* que torna todas as rotinas que envolvem a datilografia muito mais fáceis de serem executadas.

Eis algumas facilidades que o *Magic Window* oferece:

- Excluir um caractere de uma palavra.
- Excluir uma palavra da linha.
- Abrir espaço entre duas palavras para a inclusão de uma terceira.
- Excluir e incluir linhas.
- Ajustar o texto sobre linha.

E muito mais, muito mais ...

Se você ainda não teve a oportunidade de trabalhar com algum P.T., não perca tempo. Eles são de muita utilidade.

Mas o objetivo desta matéria não é discutir as qualidades do *Magic Window*, mas sim descrever alguns comandos de controle que, em geral, não estão descritos nos manuais.

Qualquer que seja o texto a ser digitado, com certeza iremos utilizar caracteres como ç, Ç, ã, Ä, õ, Ö, etc. Por mais simples que sua impressora possa ser, certamente ela possuirá tais caracteres. Basta colocá-la em modo TESTE para verificar o conjunto de caracteres que ela possui. Mas como acionar estes caracteres a partir do *Magic Window*? Uma saída seria você testar todos os caracteres possíveis do teclado (todos os normais, SHIFT's, CONTROL's e ESC's, e

combinações destes). Com certeza você encontrará aqueles que lhe interessam, e muitos outros que você não conseguirá entender o significado, pois são controles internos ao sistema.

Mas o nosso objetivo é tentar evitar que você funcione como um "machado sem cabo no escuro", descobrindo por tentativa e erro. Com a **Tabela de Caracteres Especiais** que estamos publicando, você poderá melhorar sensivelmente a qualidade dos seus textos gerados pelo *Magic Window*.

CARACTERES ESPECIAIS "MAGIC WINDOW"

TECLADO	IMPRESSORA	SIGNIFICADO
SHIFT-M	ç	CEDILHA MINÚSCULA
SHIFT-L	ã	a MINÚSCULO COM TIL
SHIFT-K	õ	O MAIÚSCULO COM TIL
SHIFT-P		CRASE
SHIFT-7		ACENTO AGUDO (*)
ESC-SHIFT-M	Ç	CEDILHA MAIÚSCULA
ES-SHIFT-L	Ã	A MAIÚSCULO COM TIL
ESC-SHIFT-K	Õ	O MAIÚSCULO COM TIL
ESC-SHIFT-N	ˆ	ACENTO CIRCUNFLEXO
CTRL-B 2		LIGA CARACTER EXPANDIDO
CTRL-B 4		LIGA CARACTER MORMAL

OBS.: esta tabela foi feita para a impressora EMILIA.

(*) O acento agudo só poderá ser utilizado quando a vogal a ser acentuada for a última da palavra, pois este sinal ocupa uma posição.

Foram incluídos dois controles adicionais que desempenham um papel importante na estética de seu texto. São CTRL-B 2 e CTRL-B 4. A função do primeiro é acionar os caracteres expandidos da impressora. Após este

controle, todos os caracteres digitados serão expandidos, até que o segundo caractere de controle (CTRL-B 4) seja recebido pela impressora, passando a imprimir caracteres normais.

Estes dois controles são similares ao CHR\$(18) e CHR\$(20) do BASIC.

Para complementar este artigo, oferecemos uma última tabela, onde o usuário do *Magic Window* poderá encontrar um pequeno resumo dos comandos mais utilizados por este PROCESSADOR evitando constantes consultas ao manual, facilitando sua utilização.

CONTROLES DO MAGIC WINDOW

Controles Gerais

ESC	capitaliza o próximo toque
ESC ESC	capitaliza os próximos toques
<CR>	muda de linha

Controle do Cursor

CTRL Z	cursor desce uma linha
CTRL Q	cursor sobe uma linha
CTRL X	cursor desce 12 linhas
CTRL W	cursor sobe 12 linhas
CTRL C	cursor para a última linha do texto
CTRL E	cursor para a primeira linha do texto
CTRL A	cursor tabula à esquerda
CTRL S	cursor tabula à direita
SETAS	movem o cursor na linha

Controle de Toques

CTRL D	apaga o toque sob o cursor
CTRL F	insere um espaço sob o cursor

Controle de Linhas

CTRL O	apaga a linha contendo o cursor, guardando as 16 últimas linhas apagadas
CTRL I	insere uma linha sob o cursor
CTRL R	reconstitui a última linha apagada ou elimi-

CTRL K	nada na linha do cursor elimina a linha sob o cursor, guardando as 16 últimas linhas eliminadas
CTRL T	corta a linha a partir do cursor, posicionando-a na linha imediatamente abaixo
CTRL G	preenche a linha do cursor com o texto da linha de baixo
Justificação	
CTRL J	chama a função de justificação
CTRL C	justifica a linha no centro
CTRL R	justifica a linha à direita
CTRL L	justifica a linha à esquerda
CTRL E	expande a linha na página
CTRL S	pula a linha sem alterá-la

Funções Especiais

CTRL Y	chama as funções especiais
CTRL N	passa para a próxima cadeia pesquisada
CTRL V	copia o toque imediatamente acima do cursor

Outras

SHIFT-CTRL-P	volta para o MENU principal
CTRL-P	permite deslocar para a página desejada
CTRL-B	permite inserção de caracteres de controle
CTRL-L	deleta do cursor até o final da linha.

Esperamos que este artigo lhe proporcione maiores facilidades na utilização deste excelente PROCESSADOR da linha APPLE.

*RICARDO RIBEIRO
SDI/DEGD/DIREC
Ramal 2426*

COMO REGULAR A VELOCIDADE DE SEU ACIONADORES DE DISQUETE

Muitas vezes o nosso programa favorito não carrega, ou melhor, o acionador de discos se recusa a lê-lo.

Será problema do acionador? Mas ele funcionava ainda ontem! Será o disquete que "bichou"? Mas eu o usei hoje! O que fazer? Esta é uma boa pergunta e, possivelmente, nós possuímos a resposta.

Muitas vezes o problema é o desajuste da velocidade do acionador.

Como regular? Fácil. Basta seguir as instruções que se seguem.

Para esta tarefa você precisará de uma chave de fenda pequena, uma chave de fenda maior para abrir o gabinete do acionador de discos, o programa *Nibbles Away II* e um disquete que possa ser apagado.

- 1) Desligue todo o equipamento e vire cuidadosamente o gabinete do acionador de discos de cabeça para baixo e solte os quatro parafusos nas extremidades do mesmo.
- 2) Desvire o gabinete, voltando à posição original. Cuidado com o cabo

que liga a CPU. Observe que o fundo está solto. Retire a tampa do gabinete, deixando exposto o interior do mesmo.

- 3) Ligue todo o equipamento, primeiro o acionador, depois o vídeo e por fim a CPU.
- 4) Coloque o disquete de programas *Nibbles Away II* no acionador de discos de número 1. Espere que seja carregado.
- 5) Quando aparecer o menu, digite a letra **D** cuja opção é DISK DIAGNOSTICS. Ao aparecer o outro menu, escolha a opção **S** de SPEED TEST.

Quando a mensagem "DRIVE TO BE TESTED →" aparecer, digite **1** sem apertar o <CR>.

Neste instante, retire o disquete contendo o *Nibbles Away II* do acionador e coloque o de rascunho no mesmo. Aperte qualquer tecla para iniciar o teste.

Na tela aparecerá uma escala que varia de SLOW (lento) passando pro 0 (ideal) até FAST (rápido). O objetivo é colocar a "caixinha móvel" que fica abaixo do gráfico, perto do valor zero, ligeiramente abaixo. O ideal é entre o zero e o primeiro ponto de exclamação do lado esquerdo da escala.

Dentro da "caixinha" aparece a di-

ferença de velocidade a contar do zero. Este número está em hexadecimal.

- 6) Localize na parte traseira do *drive* de número 1, um potenciômetro verde com o parafuso voltado para cima. Este potenciômetro (peça eletrônica de forma retangular, de cor verde ou azul, com um pequeno parafuso na ponta e lacrado com esmalte) regula a velocidade do disco.

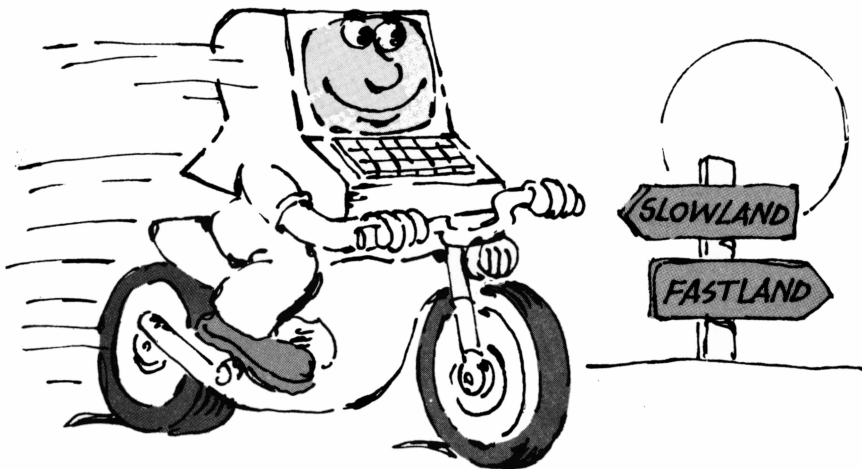
Pegue a chave de fenda pequena. Coloque-a na ranhura do parafuso do potenciômetro e gire gentilmente para um lado e depois para o outro.

Ao girar, observe a "caixinha móvel" no gráfico. Procure encontrar o lado que, ao girar o potenciômetro, a posicione no local apropriado.

Quando conseguir isto, já está pronto. Fácil. Agora aperte a tecla <ESC> e refaça o mesmo procedimento para o acionador de número 2 (se for o caso).

Quando terminar não esqueça de retirar os disquetes dos acionadores, desligar o equipamento e voltar a fechar o gabinete com os parafusos.

ROGÉRIO FAEZY
SDI/DEGD/DIREC
Ramal 2426



ESCOVANDO BITS

GLOBAL PLE

Você que lida com micros da linha APPLE como o Microengenh e outros, fatalmente já se deparou com dificuldades na correção e modificação de linhas de seu programa em BASIC. Se o seu problema é edição, saiba que este utilitário é uma aspirina para suas dores de cabeça. Ele é o GLOBAL PLE, desenvolvido por Neil Konzen.

Antes de carregar ou começar a digitar seu programa, carregue o disquete do GLOBAL PLE (*Program Line Editor*). Este contém programas encarregados de proporcionar ao usuário uma variada gama de comandos de edição.

Caso você, após ter carregado o utilitário, queira trazer seu programa de um disquete para a memória, coloque-o no *drive*, execute o comando LOAD (NOME DO PROGRAMA) e, após ser carregado, estará na memória com todas as possibilidades que o GLOBAL PLE oferece. Por outro lado, caso você queira iniciar a digitação de um programa, dê um NEW e comece.

Todos os comandos são da forma **CTRL-letra**, onde CTRL é a abreviação da tecla CONTROL (que fica geralmente no lado esquerdo do vídeo). Abaixo apresentamos um sumário do que eles são capazes de fazer:

- Para entrar no modo de edição pressione CTRL-E. Depois que a palavra EDIT aparecer, coloque o número da linha que você deseja editar. A linha aparecerá com o cursor no começo da sentença.
- O Editor de Linhas guarda o número da última linha editada; sendo assim, caso você queira editar a mesma linha novamente, pressione "CTRL-E". Isto fará com que a última linha editada reapareça.
- Se você atingir o tamanho máximo da linha (239 caracteres) um alarme soará.

Estando no modo de edição, todos os comandos que se seguem podem ser utilizados:

Inserir (CTRL-I = INSERT)

Este comando permite inserir caracteres na linha. Tudo o que for incluído após CTRL-I será inserido após a posição do cursor, "empurrando" o restante da linha, até que um caractere de controle (como outro comando de edição ou ←, →) seja teclado.

Apagar (CTRL-D = DELETE)

Apaga o caractere debaixo do cursor movendo o resto da linha no sentido de preencher o espaço criado.

Buscar (CTRL-F = FIND)

Este comando, seguido de um ca-

ractere teclado N vezes, posicionará o cursor na enésima ocorrência do caractere. Por exemplo: **CTRL-F** " " " posicionará o cursor na terceira aspas da linha. Tecle qualquer tecla para terminar a busca.

Matar (CTRL-Z = ZAP)

Apaga todos os caracteres da posição do cursor até um caractere especificado. Este comando, seguido de um caractere teclado N vezes, apagará todos os caracteres até a enésima ocorrência do caractere.

Passar por cima (CTRL-O = OVERRIDE)

Ele funciona como o comando de inserção, exceto pelo fato de que o primeiro caractere inserido pode ser um caractere de controle. Depois que o primeiro caractere for inserido, **OVERRIDE** funciona exatamente como o **INSERT**, isto é, a inserção é finalizada por um caractere de controle. No modo de edição, os caracteres de controle inseridos aparecem em **INVERSE**. Por exemplo: suponha que você queira dar um "CATALOG" em certo ponto de seu programa. Para isto basta colocar um **PRINT "CTRL-O CTRL-D CATALOG"** na linha desejada, que isto funcionará como o comando **CATALOG** dentro do seu programa. Lembre-se que **CTRL-D** corresponde a **CHR\$(4)** nos comandos do DOS.

Reeditar (CTRL-R = RESTART)

Este comando é usado para reeditar a linha original independente de qualquer modificação feita. Por exemplo, estando no modo de edição, você pode deletar qualquer caractere; teclando-se o **CTRL-R**, recupera-se a linha sem as deleções.

Compactar (CTRL-P = PACK)

Para evitar ultrapassar o tamanho máximo de linha permitido, pode-se usar o **CTRL-P** para apagar espaços desnecessários. Este comando reescreverá a linha com todos os espaços removidos, exceto os que estiverem entre aspas.

Posicionar no fim (CTRL-N = END)

Move o cursor para o final da linha. Ele é especialmente útil na inclusão de caracteres no final da linha.

Sair (CTRL-Q = QUIT)

Apaga todos os caracteres da posição do cursor até o final da linha e sai do modo de edição.

Terminar (CTRL-M = RETURN)

Esta é a maneira usual de deixar o modo de edição. Pode-se usar o **<CR>** também.

Além de lidar com a edição, o **GLOBAL PLE** também possui comandos de controle de listagem. São eles:

Parar listagem (CTRL-S)

Pára temporariamente a listagem de um programa ou **CATALOG**. Para continuar, basta pressionar **CTRL-S** novamente. Qualquer outra tecla ocasionará a saída do comando.

Interromper listagem (CTRL-C)

Serve para interromper uma listagem, voltando para o **BASIC**.

No Microengenho, **CTRL-H**, **J** e **U** são responsáveis pelo controle do cursor. Estes caracteres podem ser inseridos pelo comando de edição **OVERRIDE (CTRL-O)** dentro de uma linha.

CTRL-H — Esquerda

CTRL-U — Direita

CTRL-J — Para baixo

Como exemplo, um comando **PRINT** com um asterisco seguido de quatro **CTRL-J**'s e outro asterisco ocasionará o aparecimento de um asterisco seguido de quatro linhas entre eles, um espaço e o outro asterisco.

Além das funções definidas anteriormente, o **PLE** oferece outras possibilidades, agora utilizando o **ESCAPE (ESC)**. Esta função é chamada pressionando-se a tecla **ESC** seguida da tecla que está associada à função.

Destacamos as que se seguem:

ESC 1: Equivale ao comando CATALOG D1.
ESC 2: Equivale ao comando CATALOG D2.
ESC →: Move o cursor oito espaços para a direita.
ESC ←: Move o cursor oito espaços para a esquerda.
ESC I: Move o cursor para cima.
ESC M: Move o cursor para baixo.
ESC J: Move o cursor para a esquerda.
ESC K: Move o cursor para a direita.
ESC L: Equivale ao comando LIST.

ESC :: Entra no MONITOR funcionando como um CALL-151.

ESC /: Equivale ao comando PRINT.
OBS.: As funções de movimento do cursor são recursivas, ou seja, basta pressionar ESC uma vez para que as próximas teclas funcionem como anteriormente especificado.

Enfim, vale a pena lançar mão deste utilitário, evitando assim as tradicionais "chateações" na digitação de seus programas.

LUCIANE SCHÜTTE
SDI/DSI/DEGD/DIREC
Ramal 2426

LMOFFSET

O LMOFFSET é um utilitário que acompanha o sistema operacional NEWDOS/80 para o TRS-80.

A finalidade do LMOFFSET é ler programas de fitas ou discos, fornecendo dados do programa tais como:

- a) START POINT (início do programa).
- b) END POINT (fim do programa).
- c) ENTRY POINT (endereço de execução).

Estas informações são úteis para a reprodução de programas em fita.

Lembramos que os programas em fita a que nos referimos são gravados no formato SYTEM (linguagem de máquina).

A seguir daremos uma descrição de utilização do LMOFFSET:

1. LMOFFSET pergunta se o programa a ser carregado está em fita ou disco.

1.1. Para programas em disco ele guarda o nome do programa.

1.2. Para programas em fita ele guarda o nome.

1.2.1. Se a leitura for de fita, um * será apresentado no vídeo.

1.2.2. Durante a leitura surgirão *** no vídeo significando que a sincronização foi completada.

1.2.3. Em caso de erros teremos:

- a) **C** — para erro de *checksum*.
- b) **P** — para um *byte* de dado não reconhecido.
- c) **I** — inserção de um *byte* de dado errado.

Em qualquer um dos casos de erro repita o procedimento descrito anteriormente.

1.3. Após a leitura do programa o LMOFFSET mostrará os endereços de carga do mesmo. No caso do programa residir na mesma área do DOS, um aviso será mostrado.

1.4. Agora o LMOFFSET pergunta se você deseja mudar os endereços do programa a ser gravado. Isto é recomendado caso o mesmo resida na área do DOS. Caso contrário tecle apenas <ENTER>.

1.5. Se um novo endereço for especificado, será apresentado o endereço anterior e o LMOFFSET voltará ao passo 1.2..

1.6. Caso tenha modificado os endereços, o LMOFFSET perguntará se você deseja dar um "DISABLE DOS" ou seja, desabilitar o DOS, senão, será colocada uma subrotina ao final de seu programa

para recolocá-la durante sua execução.

1.7. Agora você deve responder se o programa será gravado em fita ou disco:

- a) Para disco o LMOFFSET pedirá o nome do programa.
- b) Para fita o LMOFFSET perguntará se o programa será guardado em velocidade (a)lta ou (b)aixa.

Para maiores informações sobre este utilitário aconselhamos uma consulta ao manual do NEWDOS/80 V 2.0..

GUSTAVO BENIGNO
SDI/DSI/DEGD/DIREC
Ramal 2426



BYTE ZAP-2

Estamos complementando agora o artigo sobre o BYTE ZAP iniciado em BYTE PAPO n.º 1.

MAPEANDO SEU DISQUETE

A tecla 8 causará a leitura do TRACK (trilha) 17, setor 0, chamado VTOC (Volume Table Of Contents), e apresentará este mapa em seu vídeo mostrando cada setor usado do disco (+) e cada setor livre (.). Você pode inclusive tirar um DUMP de seu VTOC.

ALTERANDO O CONTEÚDO DO BYTE

A tecla 9 permitirá que você troque o conteúdo de qualquer byte do setor atual. A alteração será apenas na memória de seu equipamento, até que você selecione a tecla de atualização do setor no disco (a seguir).

Existem três maneiras de se alterar o valor do byte:

HEXA — Digite um \$ seguido do novo valor (00-FF) e pressione <CR>.

DECIMAL — Digite o número decimal (0-255) e pressione <CR>.

ASCII — Você pode digitar um valor em ASCII nos seguintes formatos:

- N — Para caracteres normais (Ex.: NO é um O normal)
- I — Para caracteres INVERSE (Ex.: I% é um % em INVERSO)
- F — Para caracteres FLASH (Ex.: FG é um G em FLASH)
- L — Para caractere minúsculo (Ex.: LA é um A minúsculo)
- C — Para caractere de controle (Ex.: CM é um CONTROL-M, CARRIAGE-RETURN)

Se você desejar entrar com um espaço num determinado byte, basta digitar a letra correspondente ao formato, seguida de um SPACE BAR.

Ex.: "F ", "N " (sempre seguido de <CR>).

FINALIZANDO O BYTE ZAP

A tecla 0 (zero) determina o final da execução do BYTE ZAP.

GRAVANDO O SETOR NO DISCO

Pressionando a tecla "-" terá início a atualização do setor corrente no disco, efetivando as suas alterações. Antes de dar início à atualização, lhe será pedido que confirme a gravação.

MAIS SOBRE O CATALOG

Os setores de 1 (\$01) a 15 (\$0F) da trilha 17 (\$11) são reservados para o DIRETÓRIO ou CATALOG do disco. Ali são armazenadas as informações pertinentes aos programas e arquivos constantes no disco. Cada setor pode conter informações de até 7 arquivos. Portanto, a capacidade do diretório é para 105 arquivos. Outra peculiaridade é que o primeiro setor do diretório é o setor 15 (\$0F) e o último é o 0 (\$00).

OBS.: O setor 0 (\$00) da trilha 17 é reservado para o VTOC (Volume Table Of Contents).

RECUPERANDO ARQUIVOS DELETADOS

Uma das características interessantes do BYTE ZAP é a possibilidade de se recuperar arquivos previamente deletados; ao se deletar um arquivo do disco, o DOS na verdade não destrói o seu conteúdo. Para cada

arquivo, o DOS reserva as seguintes informações: um *byte* para a trilha de início do arquivo, um para o setor, um para o tipo do arquivo, trinta para o nome e quatro para o tamanho do arquivo.

Após a deleção, o DOS armazena o valor \$FF no *byte* que contém a trilha de início do arquivo. Mas como se recuperar o antigo valor deste *byte*?

A solução é a seguinte:

O DOS transfere este valor para o último *byte* do NOME do arquivo.

Delete um arquivo e verifique isto.

Assim, basta recuperar este valor e armazená-lo no *byte* correto.

Posicione o cursor no *byte* reservado para a trilha do arquivo deletado. Digite a tecla 9 (CHANGE BYTE) e I, seguido do caractere ASCII que consta no último *byte* do NOME. Se, por exemplo, este *byte* contiver o valor X, posicione o cursor no *byte* reservado para a trilha, digite 9 seguido de IX e tecle <CR>. Após isso, posicione o cursor no último *byte* do NOME e troque-o por um " " (branco), digitando 9 "N" <CR>. E, por fim, salve o setor no disco, usando a tecla "-".

Neste instante verifique o CATALOG, usando a tecla 6.

TRADUZINDO COMANDOS E MENSAGENS DE ERRO DO DOS

Se você tiver interesse em traduzir a seu gosto MENSAGENS ou COMANDOS do DOS, tais como CATALOG, INIT, OPEN, VERIFY etc., basta procurar na trilha 1, setores 7 a 9, que você encontrará os comandos e mensagens armazenados. Basta utilizar a tecla 9 para trocar o seu conteúdo para o desejado.

IMPORTANTE: note que estes valores estão armazenados em FLASH.

Portanto, para a troca dos caracteres utilize a tecla 9 seguida de "F <caractere desejado>". Todos os ca-



racteres devem estar em FLASH, com exceção do último, que deve ser um caractere NORMAL (9 "N <último caractere>"), para indicar o final do comando ou mensagem.

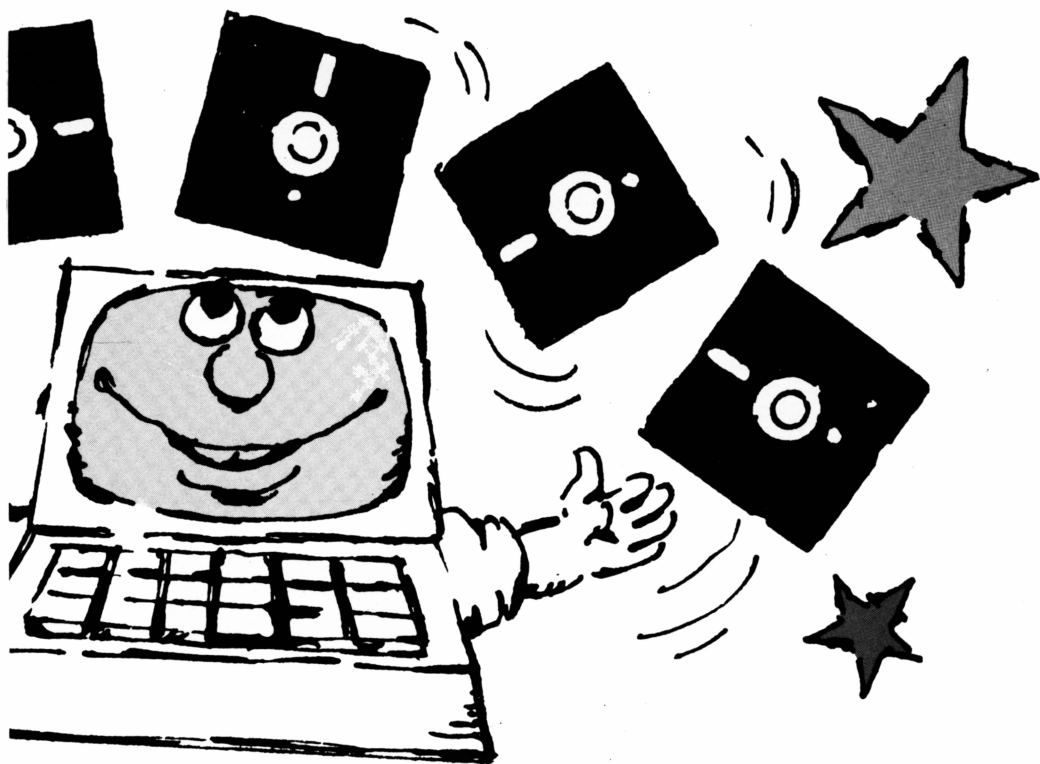
A melhor indicação é que, para melhor assimilação do BYTE ZAP, se teste cada comando aqui abordado.

EXTRAINDO O DOS

Uma boa aplicação para o seu BYTE ZAP é usá-lo como instrumento para se extrair o DOS de seu disco, liberando um espaço extra que muitas vezes pode ser útil.

Nem sempre é necessário o DOS em seu disco. Caso você o extraia, basta utilizar um outro disquete para carregá-lo.

Como mencionamos anteriormente, o setor 0 da trilha 15 contém o VTOC,



o qual mantém um controle dos setores livres no disco. Portanto, para livrarmos o DOS, basta alterar no VTOC os *bytes* que informam que os setores das trilhas 1 e 2 estão ocupados.

Não indicamos a liberação da trilha zero, pois o DOS a utiliza para indicar o fim do CATALOG e dos índices (*track/setor list*), o que inviabiliza o manuseio de dados nesta trilha, sem outras modificações mais complexas na estrutura do sistema, não significando que a trilha fique totalmente inútil. Ela pode ser usada para arquivos, desde que manuseada diretamente por linguagem de máquina.

O primeiro passo é ler o setor 0 da trilha 17. Lá você encontrará o mapa de conteúdo a partir do *byte* 56 (\$38). Para livrar as trilhas 1 e 2,

altere o valor dos *bytes* \$3C, \$3D, \$4D e \$41 para 255 (\$FF), que a esta altura devem conter zeros.

Após feita esta alteração, grave este setor usando a tecla “—”.

A partir deste instante, o DOS irá encerrar as trilhas 1 e 2 como livres para gravação, mesmo que o DOS ainda permaneça.

CONCLUSÃO

BYTE ZAP é uma ferramenta muito útil e oferece uma série de recursos que, sendo bem explorados, podem permitir um ótimo aproveitamento de seu disco.

RICARDO RIBEIRO DE F. CASTRO
SDI/DSI/DEGD/DIREC
Ramal 2426

INDICE

Editorial	3
Novidades	4
Legenda por computador	4
Vídeo & micro	4
Micro tagarela	4
Olha o passarinho!	4
Incrementando o seu micro	5
Wild Card	5
Micro-programador	6
Escrevendo na tela	6
Softpapo	7
Visiplot-Visitrend	7
Olympic Decathlon	9
Conversão	11
SET	11
PRINT AT	11
SIN	12
Artigos	13
A sociedade da informação	13
O microcomputador no ensino	14
Cursos	16
Tabela de Formas - 1	16
Desafio	18
Solução do desafio 1	18
Um programa que você não usa todos os dias mas é interes- sante	21
Macetes & Mutretas	22
Cuidados com o disquete	22
Dicas do Magic Window	23
Como regular a velocidade de seus acionadores de dis- quete	25
Escovando Bits	27
Global Ple	27
Lmoffset	29
Byte Zap - 2	31
Programa do mês: Downhill Racer	35
Peeks & Pokes	36

Programa do mês

Downhill Racer

Neste número trataremos de um programa que foi escrito em uma linha de BASIC.

Trata-se de um *one liner* para a linha do TRS-80 (CP-500, DIGITUS, NAJA, JR etc). Este programa é um jogo no qual você deverá controlar um carquinho (letra H) na pista usando as setas para isto.

Estes programas realmente cabem em uma linha de BASIC. Escreva-o sem espaços e usando o símbolo "?" no lugar de PRINT. Escreva o máximo

que puder e depois entre no modo editor para completar a linha. Para isto digite **EDIT 10** e aperte a letra **X** para deslocar o cursor ao fim da linha, entrando no modo de inserção permitindo que você escreva o restante.

Mãos à obra!



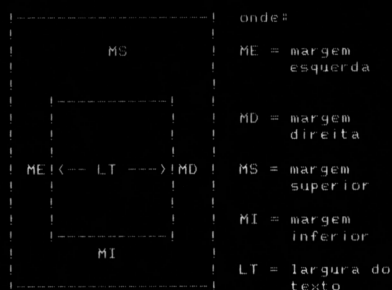
```
10 CLS:X=746:P=998:FOR I=1 TO 1000:PRINT@0,
"SCORE="S;:S=S+1:IF PEEK(15360+X+64)=4200
RPEEK(15360+X)=42 THEN PRINT@X,"H";:PRINT@
X,"H";:PRINT@896,"";:ENDELSEPRINT@X,"H";
:PRINT@P,"*      *":C=RND(3)-2:P=P+C:K=PE
EK(14440):IF K=32 THEN X=X-1:NEXTELSEIF K=64
THEN X=X+1:NEXTELSENEXT
20 REM DOWNHILL RACER:
CHRISTOPHER LAMPTON
```

PEEKs & POKES

Neste número vamos descrever quatro endereços da memória ROM do APPLE que desempenham papel fundamental na manipulação de telas.

Normalmente o micro utiliza 24 linhas e 40 colunas, mas podemos criar uma janela de texto no interior da tela.

O esquema a seguir ilustra o que é chamado de janela de texto.



Você poderá alterar qualquer um destes parâmetros, bastando para isto modificar o conteúdo dos seguintes endereços de memória (o comando **PEEK(end)**, retorna o valor atual do endereço **end** e o comando **POKE end, val** coloca o valor **val** no endereço **end**).

32 - Margem esquerda da janela de texto.

Neste endereço está a informação de onde inicia a margem esquerda na tela. Os valores permitidos para esta posição variam de 0 a 39. A troca do valor nesta posição não altera a largura da janela. Ambas as margens, esquerda e direita, se movem.

33 - Largura da janela de texto.

Neste endereço está a informação da largura do texto da janela. Os va-

lores permitidos estão entre 1 e 40 inclusive. A troca deste valor muda a posição da margem direita, que é dada pela soma do conteúdo dos endereços 32 e 33.

34 - Margem superior da janela de texto.

Neste endereço encontra-se a informação de onde inicia a margem superior da janela. Os valores permitidos estão entre 0 e 39 inclusive. O valor 0 indica a linha superior e o valor 39 a linha inferior da tela.

35 - Margem inferior da janela de texto.

Neste endereço podemos saber onde se encontra a margem inferior da janela. O valor permitido varia entre 0 e 23 inclusive.

Deve-se ter cuidado em observar os limites de valores dos parâmetros acima descritos. Tentar determinar janelas fora destes limites dará resultados imprevisíveis podendo haver até perda do programa.

Bem, por agora é só. No próximo número lançaremos mais uma lista de endereços para PEEKs e POKES e rotinas da ROM.

Lembramos que estamos prontos a receber qualquer contribuição. Se você utiliza aquela posição cheia de "tritic", faça um pequeno resumo, que teremos o maior prazer em publicá-lo.

RICARDO RIBEIRO
SDI/DEGD/DIREC
Ramal 2426