

elppa II

plus



**Manual
de
Instruções**

MANUAL DE INSTRUÇÕES

ÍNDICE

BENVINDO

CAPÍTULO 1 DESCRIÇÃO GERAL

- 2 Introdução
- 2 O que você precisará
- 3 Montando o sistema
- 4 Desembalagem
- 5 Removendo a cobertura do Elppa II plus
- 6 Uma olhada por dentro
- 8 Uma olhada no painel traseiro
- 10 Conectando cartões dentro do computador
- 11 Conectando o gravador cassette
- 11 Conectando uma unidade de disco
- 11 Instalando outro equipamento
- 12 Conectando um monitor
- 12 Conectando um televisor
- 13 Conectando os controladores de jogos
- 13 Fechando o computador
- 14 Verificação final

CAPÍTULO 2

FAMILIARIZANDO-SE COM O SISTEMA

- 16 Familiarizando-se com seu Elppa II plus
- 16 As partes básicas do sistema
- 18 Cuidando do sistema
- 18 Cuidando dos discos
- 19 Introduzindo um disco em uma unidade de disco
- 20 O teclado do Elppa II plus
- 23 Notação do teclado
- 24 Control e outros caracteres
- 25 Ajustando o gravador de fita
- 27 Procedimento normal para carregar fitas
- 28 Uma sugestão útil
- 29 Acionando o sistema com uma unidade de disco
- 31 O menu
- 31 Parando o computador
- 32 Ajustando as cores do TV

CAPÍTULO 3

INICIAÇÃO À LINGUAGEM ELPPASOFT

- 36 Iniciação à linguagem Elppasoft
- 36 A declaração PRINT
- 39 Formato do Elppasoft para números
- 40 Mais informações sobre o RETURN
- 41 Características que facilitam a edição ou: o que fazer antes de pressionar RETURN
- 43 Colorindo a tela
- 46 Mensagens de erro nos gráficos
- 47 Desenhando linhas
- 49 Os controles de jogos
- 50 Arquivos e outras possibilidades de cálculo
- 54 Prioridades de quem vem primeiro
- 56 Como evitar as prioridades

CAPÍTULO 4

PROGRAMAÇÃO ELEMENTAR

- 60 Execução adiada
- 64 Edição elementar
- 66 Acrobacias elementares
- 67 Comandos que facilitam a edição
- 68 O cursor móvel eliminando ou copiando o que foi escrito
- 70 Uma palavra sobre o aprendizado da linguagem Elppasoft
- 70 Como evitar acidentes
- 71 Verdadeiro ou falso
- 75 Ordem ou prioridade para operações
- 75 A declaração IF
- 76 Guardando programas em um disco
- 78 Guardando programas com um gravador cassette
- 78 Outros programas com gráficos
- 81 Loops com as declarações FOR e NEXT
- 83 Um programa errado
- 84 Um último exemplo de loops internos
- 84 Ganhando brilho
- 85 Imprimindo com detalhes

CAPÍTULO 5

GRÁFICOS

- 90 Conversando com um programa
- 94 Fora das paredes
- 95 Criando sons
- 97 Sons para a bola saltitante
- 97 Para notas mais altas múltiplas declarações em uma mesma linha
- 98 Números randômicos
- 100 Simulando um par de dados
- 101 Subrotinas
- 103 Verificações
- 105 Melhorando a subrotina que desenha um cavalo
- 108 Gráficos de alta resolução

CAPÍTULO 6

SEQUÊNCIAS E MATRIZES

- 116 Cadeias de caracteres
- 121 Concatenação
- 122 Mais funções sequenciais
- 124 Introduzindo matrizes
- 126 Mensagens de erro em matrizes
- 127 Conclusão

CAPÍTULO 7

COMO FUNCIONA O SEU ELPPA

- 130 Como funciona o seu Elppa
- 133 Hardware:
 - 133 Processador
 - 134 Memória principal
- 135 Sistema Software
- 136 Informação
- 136 Discos e arquivos

APÊNDICES

- 140 Apêndice A: Sumário dos comandos
- 151 Apêndice B: Palavras reservadas na linguagem Elppasoft
- 153 Apêndice C: Características de edição
 - 153 As teclas setas esquerda e direita
 - 153 Movimentando o cursor
 - 154 Apagando linhas de programa
 - 155 Sumário das características de edição
- 156 Apêndice D: Mensagens de erro
- 160 Apêndice E: Localizações de memórias usadas pelo DOS e pelo Elppasoft
 - 161 Valor do HIMEN fixado pelo carregamento do DOS
- 163 Apêndice F: O velho monitor ROM
 - 163 Usando o velho monitor ROM
 - 164 Recuperando-se de RESETs acidentais

BENVINDO

Este manual é destinado às pessoas que querem aprender a programar na linguagem BASIC do Elppa II plus. Com este manual, um Elppa II plus, um pouco de seu tempo e atenção, você descobrirá que não existe dificuldade em aprender como programar um computador. Em primeiro lugar, como tudo que é novo, programar não nos será familiar, porém este manual foi destinado para aliviar qualquer apreensão que você possa ter. Primeiro de tudo, não há segredos que você tenha que saber antes de ler este manual. Tudo é revelado, e somente uma coisa por vez é explicado. Se você começar pelo início, EXECUTAR TUDO que for aparecendo, e decidir-se a usar todo o seu tempo disponível, é bem provável que você aprenderá como programar.

O grande segredo é dispor de tempo e tentar tudo. Você não pode aprender programação apenas lendo este ou qualquer outro livro. Como aprender a andar de bicicleta ou tocar violino, você deve aprender praticando. Você tem que cometer erros e corrigi-los, e não sentir-se mal quando realmente cometê-los.

Se você já sabe como programar, uma rápida olhada neste manual o deixará familiarizado com as características do software BASIC do Elppa II plus, que doravante por nós será referido como Elppasoft. Desconfiamos que você ficará muito impressionado com a facilidade de se fazer gráficos de alta resolução e usar as outras características que fazem do Elppa um excelente computador para grande variedade de aplicações.

Este livro é um manual INSTRUTIVO. Siga atentamente este manual, o qual (sentimos prazer em dizer) tem sido muito bem recebido por clientes assim como por críticos. Esperamos que você o aprecie assim como sentimos prazer em escrevê-lo.

INTERFERÊNCIA EM RÁDIO E TELEVISÃO

O equipamento descrito neste manual gera e usa energia de radiofrequência. Se ele não for instalado e usado adequadamente, estritamente de acordo com as nossas instruções, pode causar interferência na recepção de rádio e televisão.

Você pode verificar se o seu computador está causando interferência desligando-o. Se ela cessar, foi provavelmente provocada pelo computador. Se ele realmente causar interferência na recepção de rádio ou televisão, você pode tentar eliminá-la usando uma ou mais das seguintes medidas:

- Gire a antena de TV ou rádio até que cesse a interferência.
- Desloque o computador para um lado ou outro do TV ou rádio.
- Desloque o computador o mais distante possível do TV ou rádio.
- Ligue o computador numa tomada que esteja num circuito diferente do TV ou rádio, isto é, tenha certeza que o computador e o TV ou rádio estão em circuitos controlados por interruptores ou fusíveis distintos.

Se necessário, você deve consultar o seu distribuidor ou um técnico experiente em rádio/televisão para sugestões adicionais.

CAPÍTULO 1

DESCRIÇÃO GERAL

- 2 Introdução
- 2 O que você precisará
- 3 Montando o sistema
- 4 Desembalagem
- 5 Removendo a cobertura do Elppa II plus
- 6 Uma olhada por dentro
- 8 Uma olhada no painel traseiro
- 10 Conectando cartões dentro do computador
- 11 Conectando o gravador cassette
- 11 Conectando uma unidade de disco
- 11 Instalando outro equipamento
- 12 Conectando um monitor
- 12 Conectando um televisor
- 13 Conectando os controladores de jogos
- 13 Fechando o computador
- 14 Verificação final

INTRODUÇÃO

Este manual lhe mostrará como ligar o seu Elppa II plus e também servirá como um guia para que você aprenda, facilmente, a programá-lo. Se você tem experiência em programação, descobrirá algumas novas características e conveniências no Elppasoft, que fazem a programação muito mais divertida. Se você já possui alguma experiência em programação, também descobrirá muitas características e conveniências no Elppasoft, que fazem a programação muito divertida. Mas, se você é um novato, fique ciente que programação, embora não seja difícil, pode somente ser aprendida praticando. Mais adiante, falaremos novamente sobre este tópico, mas lembre -- este é um livro para ser usado, não apenas estudado.

Se você adquiriu o seu Elppa de um distribuidor Elppa autorizado, ele permitirá que você o instale na loja, para ter certeza que você saberá como fazê-lo em casa. Não será difícil ligá-lo, é tão fácil como instalar um equipamento de som, e não exige conhecimento técnico para tanto.

O Elppa descrito neste manual possui a linguagem BASIC residente em ROM e o Autostart ROM instalados na placa lógica principal.

Atente para este símbolo que doravante aparecerá neste manual.



A intenção do mesmo, é alertá-lo para uma característica não usual do Elppasoft. A situação descrita pode causar-lhe a perda do seu programa.

O QUE VOCÊ PRECISARÁ

Este manual estava na caixa de acessórios. Esta caixa deveria também conter o cabo de força do Elppa (o cabo com o qual você o liga na tomada).

Em complementação ao Elppa II e ao conteúdo da caixa de acessórios, você precisará de mais dois itens escolhidos entre as opções abaixo (nenhum desses itens são fornecidos).

1. Você precisará de um dos seguintes itens (ambos têm utilidade, mas somente um é necessário).
 - a. Um gravador cassette e um cabo para conectá-lo ao Elppa.
OU
 - b. A unidade de disco Elppa, DISK II com um cartão de controle, interface DII.

2. Você também precisará de um dos seguintes itens:
- a. Um monitor de vídeo com cabo conveniente para ser acoplado ao conector fêmea tipo RCA na traseira do Elppa indicado como VIDEO OUT.

OU

- b. Um TV e um "Modulador de RF" com os cabos de conexão. O Modulador de RF muda a saída do sinal do Elppa de maneira que este seja compatível com o sinal de entrada normal do seu TV. Esses Moduladores estão disponíveis junto aos distribuidores.
- c. Um cartão PAL-M ENCODER, caso você queira ver o resultado de seus gráficos e jogos em cores no seu TV ou monitor colorido. Este cartão é instalado internamente no conector nº 7 do Elppa e contém nele o conector fêmea tipo RCA com saída de vídeo para o Monitor colorido ou Modulador de RF.

O Modulador vem com instruções de como interligá-lo. A capacidade do seu TV em receber programas normais, não será diminuída (ou aumentada) por ter o Elppa interligado a ele.

Um monitor branco e preto ou TV trabalhará bem, mas não deixará que você se aproveite da capacidade do Elppa para gerar quadros coloridos. As cores descritas neste manual aparecerão em matizes diferentes de cinza num TV ou monitor branco e preto.

MONTANDO O SISTEMA

Antes de começar, é conveniente que você tenha em mãos uma chave de fenda média e um alicate.

À medida que você for lendo este capítulo, siga o procedimento sugerido para desembalar e montar o sistema, como se segue:

- . Desembale o computador e sua caixa de acessórios.
- . Confira, peça por peça com a lista de conteúdo, para ter certeza de que nada falta.
- . Examine o computador, especialmente nas áreas destinadas aos dispositivos de conexão.

Se você tem um equipamento que não está descrito neste capítulo, e quer instalá-lo corretamente, leia as seções pertinentes do Apêndice

Este capítulo oferece apenas simples explicações de como desembalar e ajustar o sistema. Confie nos documentos que vieram com o equipamento para detalhes finais.

DESEMBALAGEM

Coloque a caixa de papelão do computador no chão ou sobre uma mesa. Retire os grampos da tampa da caixa. Retire a caixa de acessórios e deixe-a de lado.

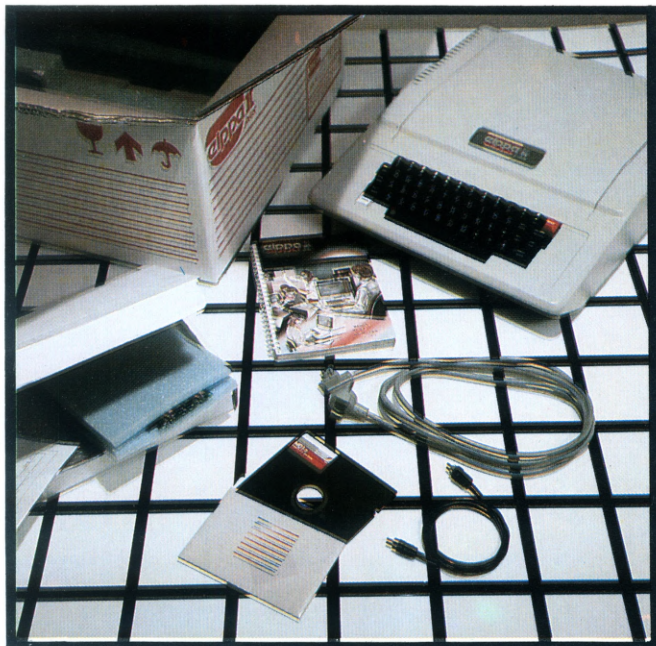
Enfie suas mãos entre a espuma protetora e os lados do computador. Empurrando a espuma para o lado com as costas das mãos, puxe o computador para fora e coloque-o sobre a mesa com o teclado de frente para você. Retire o saco de plástico que envolve o computador.

Coloque o conteúdo da caixa de acessórios sobre a mesa. Então junte todo material da embalagem e guarde-o para usá-lo em outra ocasião, como por exemplo, em caso de mudança.

Guarde e conserve todos os documentos que vêm na caixa de acessórios. Alguns podem ser suplementos (informações para serem adicionadas aos manuais) ou erratas (correções) que você pode precisar.

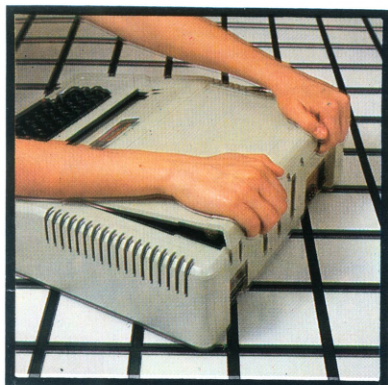
Verifique todos os itens com a lista que veio na caixa de acessórios. Use a fotografia para ajudá-lo a identificar as partes importantes. Se faltar alguma coisa, fale com seu fornecedor.





REMOVENDO A COBERTURA DO ELPPA II plus

Antes de conectar os dispositivos internos ao Elppa II plus, você deve remover a sua cobertura. Com ambas as mãos, agarre as abas que estão situadas na parte traseira da cobertura do computador. Puxe firmemente para cima até que os fechos do canto estalem. Deslize a cobertura traseira, para fora do teclado até que você possa levantá-la e tirá-la. Deixe-a de lado.



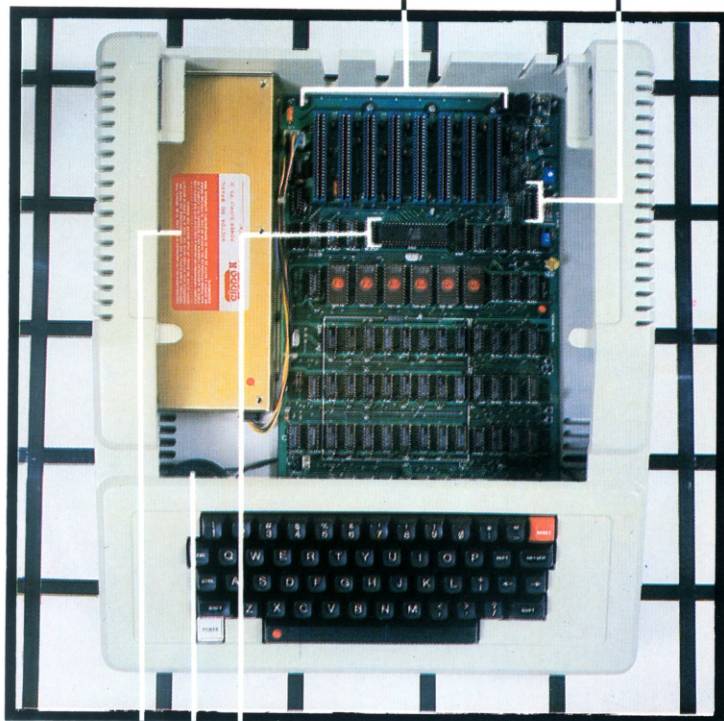
UMA OLHADA POR DENTRO

Compare o que você vê com os itens na fotografia .

- . PROCESSADOR 6502. Esta é a parte do Elppa II plus que realmente faz a computação.
- . MEMÓRIA PRINCIPAL (chamada de memória de acesso randômico ou RAM). Este é o lugar onde o computador armazena as informações com as quais o processador está trabalhando.
- . FONTE DE ALIMENTAÇÃO. Nela reside a alimentação do computador. Antes de tocar com as mãos em qualquer coisa dentro do Elppa, toque, primeiramente, na caixa da fonte de alimentação. Isto descarrega qualquer eletricidade estática que você possa ter em suas roupas ou corpo.
- . CONECTORES DE EXPANSÃO. Estes conectores, numerados de 0 a 7 (olhe e veja), são para CARTÕES PERIFÉRICOS, que tornam possível a conexão de uma grande variedade de equipamentos adicionais ao Elppa II plus. Cada cartão tem componentes ligados de um lado, e linhas de interconexão do outro lado. Instale sempre os cartões, de modo que os seus componentes NÃO fiquem de frente para a fonte de alimentação.
- . Conector do MODULADOR DE RADIOFREQUÊNCIA. Se você usa um televisor com o Elppa II plus, deve adquirir e ligar um modulador de radiofrequência (RF) à este conector, e o televisor ao modulador. Um modulador de RF torna possível para um televisor comum sintonizar os sinais vindos do computador. Leia as instruções que vieram com o modulador para ver qual canal usar.
- . Conector GAME I/O. Alguns produtos que usam um conector de 16 pinos podem ser adquiridos no seu fornecedor.
- . CONECTOR DE ENTRADA DO TECLADO, com o teclado embutido já interligado.
- . ALTO-FALANTE (e seu conector), que o Elppa II plus usa para emitir "bips" e tons musicais.

CONECTOR GAME I/O

CONECTORES DE EXPANSÃO



PROCESSADOR 6502

ALTO FALANTE

FONTE DE ALIMENTAÇÃO

UMA OLHADA NO PAINEL TRASEIRO

Vire o Elppa II plus de modo que o painel traseiro fique de frente para você. No lado direito do mesmo, está o interruptor de força. Tenha certeza de que ele permaneça na posição 0, até que você tenha terminado de ajustar o seu sistema, para evitar danos no circuito do Elppa.

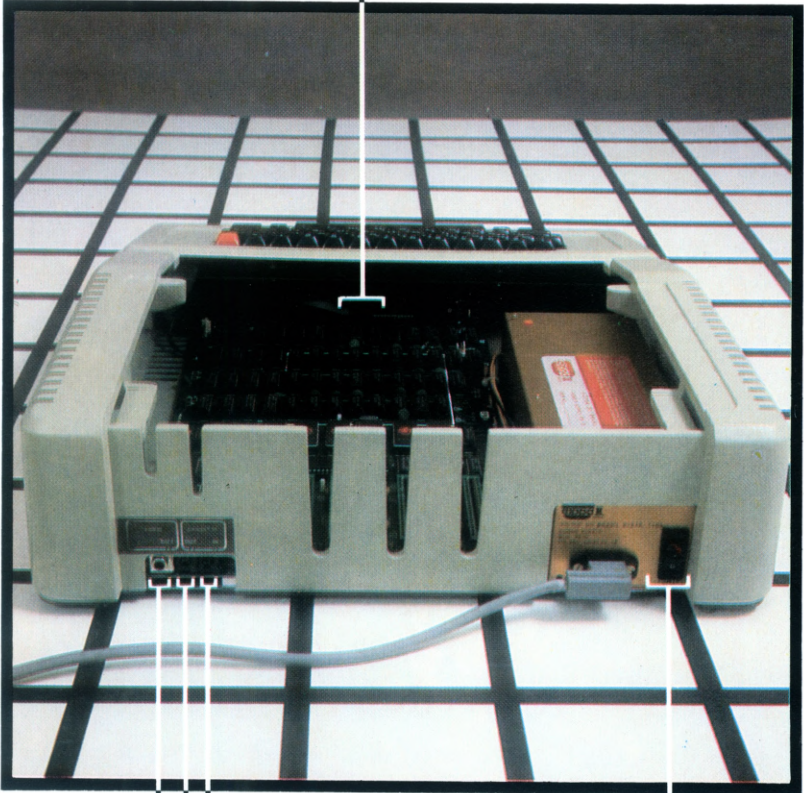
Conecte uma ponta do cabo de força no soquete próximo ao interruptor, e a outra ponta em uma tomada comum.

No painel traseiro do Elppa você irá encontrar:

- . TOMADA DE SAIDA DE VÍDEO. É o lugar por onde saem os sinais de vídeo para um monitor.
- . TOMADA DE SAIDA PARA CASSETTE. Se você usar um gravador cassette de audio para armazenar programas, utilizará este soquete para conectar o cabo que transporta a informação do Elppa para o gravador cassette (Capítulo 1).
- . TOMADA DE ENTRADA PARA CASSETTE. Este soquete conecta o cabo que transporta a informação do gravador cassette para o Elppa (Capítulo 1).

Quando você tiver terminado de examinar o painel traseiro, vire o computador de forma que o teclado fique de frente para você.

CONECTOR DE ENTRADA DO TECLADO



INTERRUPTOR DE FORÇA

CASSETTE IN

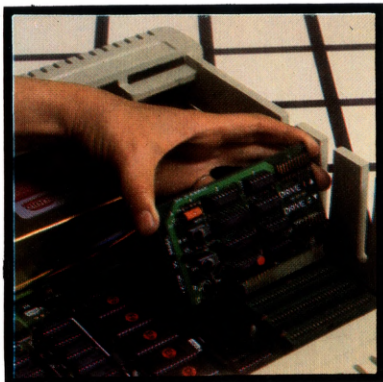
CASSETTE OUT

VÍDEO OUT

CONECTANDO CARTÕES DENTRO DO COMPUTADOR

Agora que você deu uma boa olhada no computador, é tempo de instalar os seus cartões e dispositivos periféricos. O procedimento padrão para se fazer isto, é o que se segue:

- . Tenha certeza de que o interruptor de força está DESLIGADO e o computador está ligado ao cabo de força.
- . Pegue o cartão por suas bordas. Manipule-o cuidadosamente, da mesma forma que você cuida de um LP importado.
- . Se o cartão tem chaves, ajuste-as de acordo com as instruções que o acompanham.
- . Se o cartão for conectado a um dispositivo periférico, siga as instruções de instalação do cabo que veio com o cartão e o dispositivo.
- . Toque na caixa da fonte de alimentação para descarregar qualquer eletricidade estática que possa estar em você ou em suas roupas.
- . Segure o cartão sobre o conector desejado, de modo que o lado dos componentes NÃO fique de frente para a caixa da fonte de alimentação de força e que quaisquer cabos de ligação não estejam no caminho entre o cartão e o conector.
- . Introduza delicadamente os contatos no conector até que o cartão esteja firmemente encaixado. Tenha cuidado para não entortá-lo.
- . Se o cartão não tem cabo, a instalação está completa.



CONECTANDO O GRAVADOR CASSETTE

(se você não está usando uma unidade de disco, ou se você vai usar ambos, o gravador e o disco)

Use um cabo especial que contenha dois plugs em cada extremidade para conectar o Elppa ao seu gravador de fita cassette. Conecte um plug preto ao borne de ligação marcado MIC ou MICROPHONE no gravador, e o outro plug preto (na extremidade oposta do cabo) ao borne de ligação situado na traseira do computador marcado CASSETTE OUT. Conecte o plug vermelho da extremidade do cabo ao borne de ligação do gravador marcado EAR ou EARPHONE ou MON ou MONITOR (marcas diferentes usam palavras diferentes). Conecte o outro plug vermelho (na extremidade oposta do cabo) ao borne de ligação do computador marcado CASSETTE IN. "OUT" quer dizer "saída do computador" e "IN" quer dizer "entrada do computador". Tudo que resta é conectar o cabo de alimentação do gravador cassette na tomada, e ele estará pronto para uso.

CONECTANDO UMA UNIDADE DE DISCO

Há várias formas de se conectar os cabos das unidades de discos Elppa DISK II. Para maiores detalhes siga atentamente as instruções que vieram com a unidade.

INSTALANDO OUTRO EQUIPAMENTO

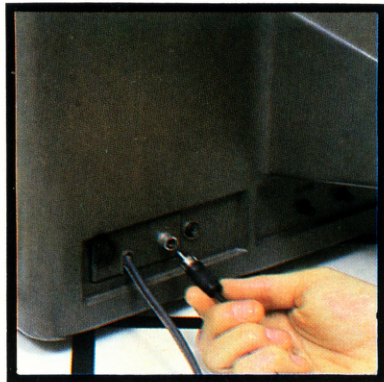
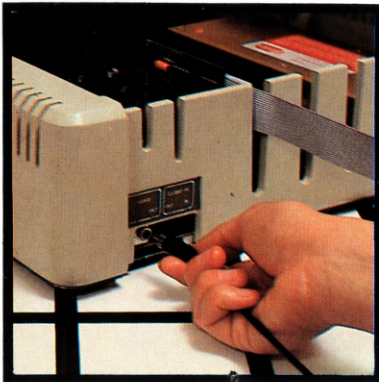
Se você tem outro equipamento que queira instalar agora, antes de fechar o computador, siga os procedimentos gerais contidos na documentação que veio com o equipamento. Tenha certeza de instalar os cartões nos conectores recomendados.

CONECTANDO UM MONITOR

Um monitor de vídeo é como um televisor, exceto pelo fato de que ele recebe as imagens, diretamente, através de um cabo, ao invés de recebê-las através de uma antena e um sintonizador.

Ligar um monitor ao Elppa é um procedimento simples:

- . Conecte uma das pontas do cabo de vídeo (que estava na caixa de acessórios) na tomada de saída de vídeo do Elppa II plus, e a outra ponta na tomada ou entrada de vídeo que está na traseira do monitor.
- . Se você precisa usar um adaptador, tenha certeza de alinhar os pinos do mesmo com o conector do monitor antes de apertar.



CONECTANDO UM TELEVISOR

Como já dissemos anteriormente, para ligar um televisor colorido ou branco e preto ao Elppa II plus, você precisa conectá-lo através de um MODULADOR DE RADIOFREQUÊNCIA (RF), o qual você pode adquirir no fornecedor do Elppa. Para instalar o seu modulador de RF, siga as instruções que o acompanham.

NOTA: O modulador pega os sinais do computador e os modifica de maneira que o televisor possa sintonizá-los.

Duas observações na instalação de moduladores:

- . Tenha certeza de que o TV e o computador estão desligados antes de prosseguir.
- . Esteja certo de que o modulador está conectado corretamente nos bornes de ligação da antena na traseira do TV e que o TV está sintonizado na estação especificada nas instruções de instalação do modulador.

Nota: Lembre-se que para obter imagens coloridas no seu TV ou monitor colorido, você precisará instalar um cartão PAL-M ENCODER, conforme descrito no item 2.c. da página 3.

CONECTANDO OS CONTROLADORES DE JOGOS

Se você adquiriu os controladores de jogos, proceda da seguinte forma para ligá-los ao seu Elppa: com a tampa aberta, ligue os controladores no soquete situado no canto traseiro direito da placa principal do computador. Tenha bastante cuidado e certifique-se que todos os pinos entraram no soquete. A posição correta é aquela na qual o conector tem seu ponto branco dirigido para a frente do computador.

FECHANDO O COMPUTADOR

Agora que você ligou o monitor de vídeo ou o televisor ao computador, feche o Elppa II plus:

- . Coloque a tampa no computador um pouco para trás de sua posição quando fechada.
- . Deslize a tampa na direção do teclado até que a borda da mesma fique dentro da caixa.
- . Empurre para baixo os cantos traseiros até que os fechos se encaixem, travando ambos os lados.



VERIFICAÇÃO FINAL

Aqui está um rápido resumo dos passos necessários à instalação.

- Desembale o computador e seus acessórios.
- Verifique-os minuciosamente com a lista que acompanha a embalagem.
- Examine o interior e o painel traseiro do computador.
- Instale todos os cartões de interface e seus cabos internos.
- Instale todos os dispositivos periféricos; faça a conexão dos seus cabos ao painel traseiro do computador.
- Conecte, como um DISPOSITIVO DE VISUALIZAÇÃO, ou um televisor com modulador de RF, ou um monitor de vídeo.

Se você executou todos estes passos, o seu sistema Elppa II plus está pronto para funcionar!

CAPÍTULO 2

FAMILIARIZANDO-SE COM O SISTEMA

- 16 Familiarizando-se com seu Elppa II plus
- 16 As partes básicas do sistema
- 18 Cuidando do sistema
- 18 Cuidando dos discos
- 19 Introduzindo um disco em uma unidade de disco
- 20 O teclado do Elppa II plus
- 23 Notação do teclado
- 24 Control e outros caracteres
- 25 Ajustando o gravador de fita
- 27 Procedimento normal para carregar fitas
- 28 Uma sugestão útil
- 29 Acionando o sistema com uma unidade de disco
- 31 O menu
- 31 Parando o computador
- 32 Ajustando as cores do TV

FAMILIARIZANDO-SE COM SEU ELPPA II plus

Agora que o seu sistema Elppa II plus está instalado, é hora de se familiarizar com ele. Este capítulo explica brevemente o que fazem as partes básicas do sistema e como ter cuidados com elas. Este capítulo também contém uma seção de referência sobre as características do teclado do Elppa.

AS PARTES BÁSICAS DO SISTEMA

Há sete componentes básicos do hardware do sistema Elppa II plus. O seu sistema pode ter mais componentes, mas as partes mais importantes estão aqui:

- . PROCESSADOR 6502. O processador executa sequências de instruções (chamadas programas) armazenadas em uma área da memória principal do computador.
- . MEMÓRIA PRINCIPAL. Este é o lugar onde o Elppa II plus armazena programas quando os está rodando. A porção da memória principal não ocupada pelos programas correntemente em processamento, está disponível para armazenar informação introduzida pelo teclado ou outra informação com a qual o programa está trabalhando.
- . TECLADO embutido. Este é o seu meio mais importante de enviar informação ao computador. Você pode digitar textos (caracteres que você pode ver) ou caracteres de controle.
- . ALTO-FALANTE embutido. O computador usa o alto-falante para emitir efeitos sonoros: por exemplo, um "bip" para indicar um acionamento bem sucedido ou um comando não reconhecido. O alto-falante pode também produzir tons musicais.

Todas estas quatro partes residem na caixa do computador. Três outras partes básicas completam o sistema.



- . MONITOR DE VISUALIZAÇÃO. Esta é a forma utilizada pelo computador para transmitir-lhe a informação, ou em palavras e números (texto) ou em figuras (gráficos).
- . UNIDADE DE DISCO. Uma unidade de disco lê e escreve informação em um disco magnético para uso repetitivo, da mesma forma que um gravador de fita toca e grava música.

Quando você grava informação em um disco, você diz ao computador para ESCREVER ou GUARDÁ-LA; quando você transfere informação do disco para a memória principal do Elppa II plus, você diz ao computador para LER ou CARREGÁ-LO.

Sempre que você desliga o computador, a informação na memória principal é perdida. Mas o que você guardar em um disco ainda está disponível para ser recarregado na memória principal depois que você liga o computador novamente.

DISCOS FLEXÍVEIS (chamados também "diskettes" ou "floppies"). Estes estão para uma unidade de disco, assim como fitas estão para o gravador de fita. Alguns discos têm programas ou outra informação já gravada neles (como fitas previamente gravadas). Outros são completamente virgens.

Doravante, para fins de simplificação, nomearemos os discos flexíveis apenas de DISCOS.

Com a sua unidade de disco você recebeu um programa que pode preparar discos virgens e neles escrever informações. O Capítulo 4 lhe mostra como usar este programa.

CUIDANDO DO SISTEMA

- . Instale seu sistema em lugar longe da luz direta do sol ou fontes de calor intenso (radiador, lareira) ou frio.
- . Tenha certeza que nenhum tipo de umidade (chuva, neve, etc.) entre no computador ou dispositivos conectados.
- . Não use, para ligar o computador e seus dispositivos, um circuito que já esteja carregado com utensílios de grande consumo, como ferro ou aquecedor elétrico.
- . Não bloqueie a circulação livre de ar através das ranhuras para ventilação, dispostas ao longo dos lados da caixa do computador.
- . Evite deixar cair poeira em excesso, sujeira ou cinza de cigarro sobre ou dentro do computador.

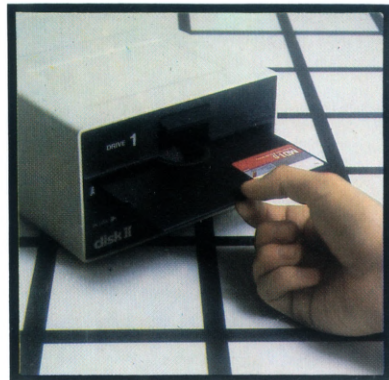
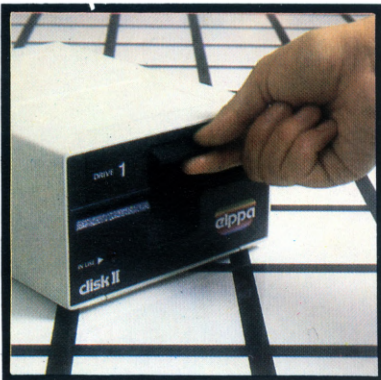
CUIDANDO DOS DISCOS

- . Manipule o disco pela sua capa e rótulo. Nunca toque no disco.
- . Para escrever sobre um rótulo já colocado à capa do disco, use caneta de ponta porosa. Não pressione com força. É melhor escrever sobre o rótulo antes de colocá-lo ao disco.
- . Nunca escreva sobre um rótulo colado com um lápis ou caneta esferográfica: isto pode formar sulcos na superfície de gravação, inutilizando o disco. Não use, também, uma borracha no rótulo. A poeira da borracha é abrasiva e pode danificar o disco.
- . Armazene os discos na posição vertical em suas capas. Não entorte ou prenda "clips" para papel nos discos.
- . Armazene os discos longe da luz direta do sol, umidade ou calor e frio em excesso.
- . Conserve os discos longe de dispositivos magnéticos ou elétricos, especialmente telefones, televisores e grandes motores.
- . Está tudo OK para colocar os discos temporariamente no computador ou unidade de disco.

INTRODUZINDO UM DISCO EM UMA UNIDADE DE DISCO

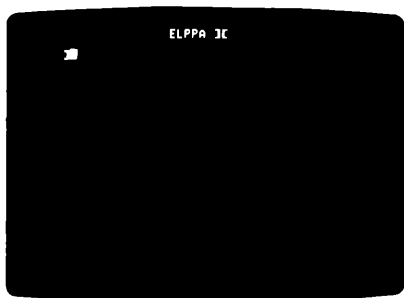
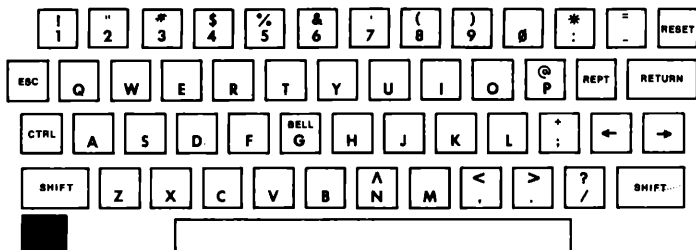
As seguintes instruções explicam o que fazer a qualquer momento que você queira introduzir um disco na unidade de disco Elppa DISK II.

1. Tenha certeza de que a luz IN USE na frente da unidade de disco está apagada. Nunca remova um disco enquanto esta luz estiver acesa: isto pode danificar o disco ou destruir parte da informação nele contida.
2. Levante a pequena porta na frente da unidade de disco. Se já há um disco na fenda, remova-o cuidadosamente, e coloque-o na sua capa.
3. Segurando, com o seu polegar, o rótulo do disco que você quer usar, remova-o da capa.
4. Gentilmente introduza o disco na fenda da unidade de disco, primeiramente o lado de forma ovalada, com o rótulo voltado para cima. Tome cuidado para não entortar ou forçar o disco. Se você sentir qualquer resistência, puxe-o vagarosamente para fora e tente outra vez.
5. Quando o disco estiver completamente introduzido, empurre a pequena porta da unidade de disco até que um "click" indique que ela está fechada.

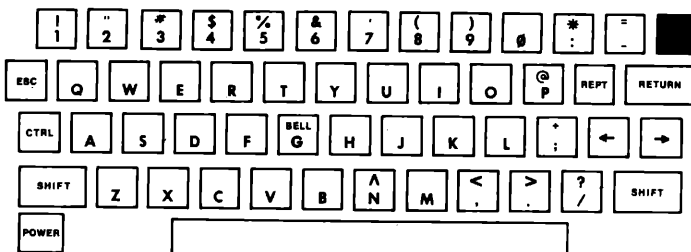


O TECLADO DO ELPPA II plus

A primeira coisa a fazer, agora que todas as conexões estão feitas, é ligar o Elppa. A chave está na traseira do computador próximo ao conector do cabo de alimentação. Acione-a. O indicador "POWER", situado na parte inferior esquerda do teclado, se iluminará. O indicador "POWER" não é uma tecla, e não pode ser pressionado. O título "ELPPA II" deve aparecer no topo da tela juntamente com o símbolo indicador de pronto] e um pequeno quadrado cintilante, chamado "cursor", no canto esquerdo da tela. Ambos indicam que o Elppa está na linguagem Elppasoft pronto para receber instruções.

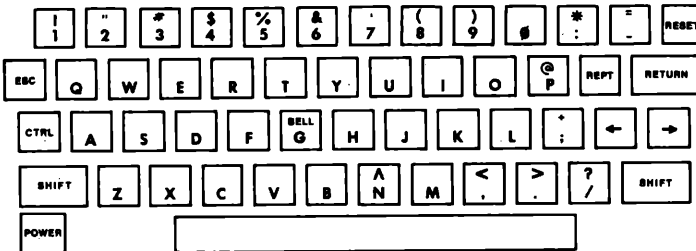


Se o seu Elppa não parece estar respondendo corretamente às suas instruções (você descobrirá o que são respostas corretas à medida que for se familiarizando com este manual), uma pressão na tecla **RESET** geralmente corrigirá o problema. Se isto não resolver, desligar o Elppa e então religá-lo, provavelmente solucionará o problema.



Se você tem uma unidade de disco, acoplada ao seu Elppa, quando o computador for ligado, alguns ruídos intermitentes surgirão, seguidos de um zumbido suave, e a luz vermelha "IN USE" aparecerá. A unidade de disco zumbirá sem parar, até que você pressione a tecla **RESET**. Faça-o agora. O título "ELPPA II" desaparece, e o indicador] e cursor aparecem na parte inferior esquerda da tela.

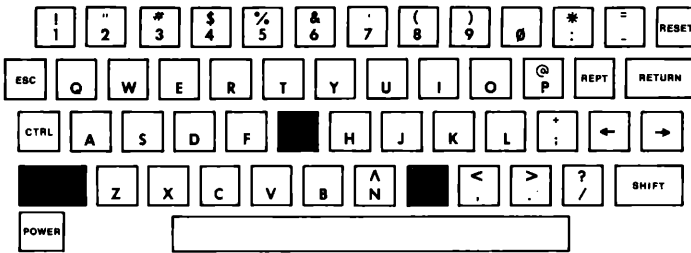
Estude o teclado. Se você está familiarizado com máquinas de escrever comuns, encontrará poucas diferenças entre o teclado do Elppa e um teclado de uma máquina de escrever. Primeiro, neste modelo não existem letras minúsculas (a não ser que, alternativamente, você tenha instalado uma ROM SPCL especial, que possibilita este recurso) porém, utilizando as letras maiúsculas disponíveis, é tudo que você necessita para programar o seu Elppa II plus.



Usando o diagrama, localize as duas teclas **SHIFT** no teclado. A razão da existência das teclas **SHIFT**, é permitir que tenhamos quase o dobro de caracteres, com o mesmo número de teclas. Um teclado com uma tecla separada para cada caracter seria muito grande, tornando-se difícil encontrar qualquer tecla desejada.

Se você pressionar uma tecla que possui dois símbolos, o símbolo inferior aparecerá na tela. Se você pressionar a mesma tecla enquanto mantém também pressionada uma das teclas **SHIFT**, o símbolo superior aparecerá na tela. Você descobrirá que uma vírgula e um ponto quando pressionados juntamente com a tecla **SHIFT**, produzirão respectivamente os símbolos < e >. Você também descobrirá outros símbolos no teclado do Elppa que não são encontrados em uma máquina de escrever comum. Sinta-se livre para tentar operar qualquer destas teclas.

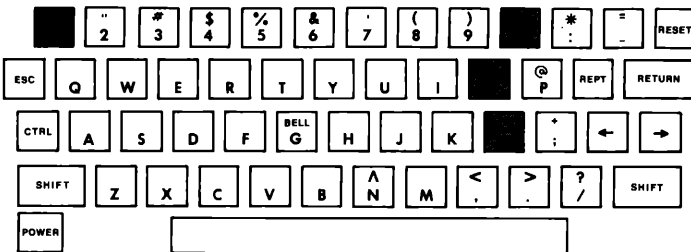
Se não há símbolos na parte superior de uma tecla, então pressione-a juntamente com a tecla **SHIFT**, não produzirá efeito algum. Há duas exceções: o **M** e o **G**.



A tecla **M** , juntamente com a tecla **SHIFT** , fornece o caracter] . A tecla **G** tem a palavra "BELL" acima do "G". Mas **SHIFT G** não coloca um sino na tela, mas sim um "G". O significado da palavra "BELL" (sino) sobre a tecla **G** será explicado mais tarde.

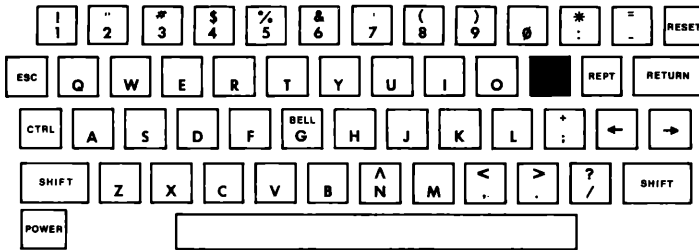
Uma diferença importante entre o uso do teclado do Elppa e a maioria das máquinas de escrever, é que você não pode empregar um "L" minúsculo para representar o número "1". É claro que não há o "L" minúsculo no Elppa, mas alguns datilógrafos terão que dominar o hábito de digitar a letra "L" quando quiserem representar o número "1".

Quando os matemáticos Hindus inventaram o círculo para representar o numeral zero, eles não usaram o alfabeto Romano. Então, escolheram um símbolo que, embora não se choque com o alfabeto deles, parece-se justamente com a nossa letra "O". O computador (e qualquer pessoa normal), desejará conservar a distinção entre zeros e oh's. O método usual de fazê-lo, no Elppa e quaisquer outros computadores, é colocar um corte no zero. Agora você conseguiu distingui-los. O teclado e a tela do TV fazem claramente a distinção. Experimente-os.



Após digitar um pouco, a tela tende a se encher de caracteres (ou palavras) inúteis.

Para limpar a tela, você precisa usar a tecla marcada **ESC** . ESC substitui a palavra "ESCAPE". A tecla **ESC** não é mostrada na tela do Elppa. Pressione **ESC** , e então digite um sinal (@) , pressionando a tecla **SHIFT** juntamente com a tecla marcada "P". Note que a tecla ESC, distintamente da tecla **SHIFT** , não precisa estar pressionada, enquanto uma outra tecla é digitada. Você tem que operar três teclas para limpar a tela. Primeiro pressione **ESC** e solte-a. Então, ao mesmo tempo que estiver pressionando o **SHIFT** , pressione P. Um instante de gratificação: o conteúdo da tela prontamente desaparece.



NOTAÇÃO DO TECLADO

Neste ponto introduziremos uma simples notação.

Como você tem visto, quando uma tecla é pressionada, tal como a tecla para a letra "A", o símbolo da tecla é mostrado: A. Para indicar várias teclas sendo pressionadas sucessivamente, nós simplesmente listamos as teclas na ordem a serem pressionadas:

A L O

Haverá situação, na qual você precisará segurar uma tecla, enquanto pressiona outra. Por exemplo, para digitar um sinal correspondente ao cifrão (\$), você deve segurar a tecla **SHIFT**, enquanto pressiona a tecla com o número 4. Sempre que esta dupla ação é exigida, nós mostraremos os símbolos para ambas as teclas, um acima do outro.

SHIFT

**\$
4**

A tecla superior deve ser segurada, enquanto a inferior é pressionada.

Eis como limpar a tela, usando a nova notação:

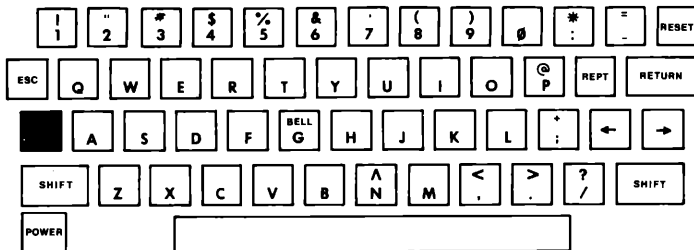
SHIFT

ESC **Ⓟ**

Experimente.

CONTROL E OUTROS CARACTERES

Quando você pressiona a tecla com o número **5**, o numeral 5 aparece na tela do TV. Você provavelmente acredita que isto seja verdade, mas, de qualquer maneira, experimente. Se você segura a tecla **SHIFT**, enquanto pressiona a tecla **5**, um sinal de porcentagem (%) deve aparecer na tela. Correto? A tecla **SHIFT** permite que algumas das teclas no teclado tenham duas funções diferentes. Diversas teclas também têm uma terceira função. A terceira função é obtida segurando a tecla **CTRL**, enquanto outras teclas são pressionadas. "CTRL" substitui a palavra "ContROl". Ao invés de colocar novos caracteres na tela, quando você usa a tecla **CTRL**, o computador responde executando certas ações. Os caracteres de controle nunca aparecem na tela.



Segure a tecla **CTRL** e pressione **G**

CTRL

**BELL
G**

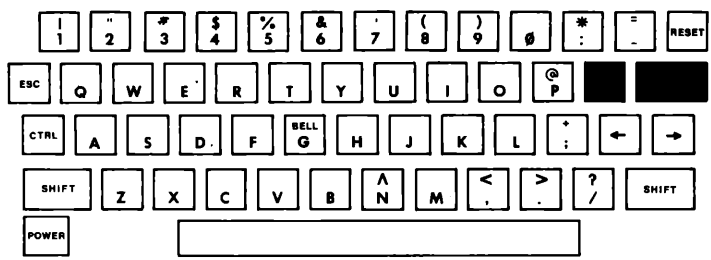
Você não vai ouvir o som de um "sino", mas sim um "bip". Sempre que o computador deseja chamar a sua atenção para alguma coisa, ele emitirá um "bip". **CTRL G** é chamado "BELL" (sino), por razões históricas: o modelo do teclado em questão, é baseado naquele do Teletipo. Naquela venerável máquina, **CTRL G** soa realmente como um sino.

Uma outra tecla que não é usualmente encontrada nas máquinas de escrever, é a **REPT**, que substitui a palavra "REPeat". Segurar a tecla **REPT**, enquanto você pressiona qualquer outra tecla, apenas faz com que o caracter desta tecla apareça repetidamente na tela. Você deve PRIMEIRO pressionar e segurar a tecla do caracter que deseja ver repetido e ENTÃO, segurar a tecla **REPT** Experimente.

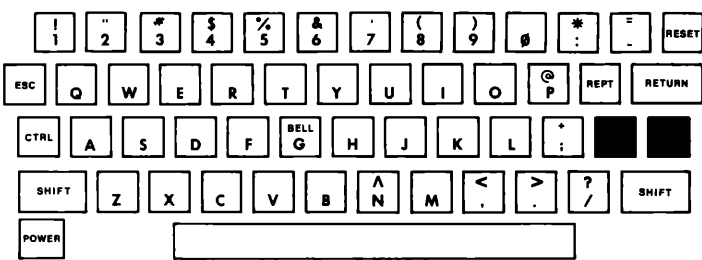
Há também uma tecla marcada **RETURN** no teclado. Nas máquinas do passado, esta era a tecla de "retorno do carro". No Elppa, isto faz com que o "cursor" retorne para o lado esquerdo da tela, mas é também uma mensagem especial para o computador. Mais informações sobre esta mensagem, serão vistas adiante. Se, por acaso, você pressionar o **RETURN**, algumas vezes você ouvirá um "bip" e a mensagem

ERRO:SINTAXE

aparecerá na tela. Por enquanto, ignore esta mensagem.



As únicas teclas que ainda não foram mencionadas, são as teclas **←** e **→**. Elas movem o cursor para a esquerda e para a direita e serão explicadas com mais detalhes adiante. Teste estas teclas e quaisquer outras que você possa encontrar. Não há nada que você possa fazer, digitando no teclado, que cause qualquer dano ao computador. À menos que você digite com um martelo. Sinta-se livre para experimentar. Com seus dedos!



AJUSTANDO O GRAVADOR DE FITA

(se você não está usando um gravador cassette, pule para a seção chamada "ACIONANDO O SISTEMA COM UMA UNIDADE DE DISCO")

Pressione a tecla **RETURN**. O indicador] e o cursor devem aparecer no lado esquerdo da tela, informando que você está "no Elppasoft" ou tem o "Elppasoft carregado". Agora você está pronto para ajustar o controle de volume no gravador de fita.

Quando você reproduz uma fita em um gravador, o faz usualmente com a intenção de produzir sons que deseja ouvir. Se ele está muito baixo, você perde algumas das palavras ou música. Se ele está muito alto, incomoda.

Quando você reproduz a fita no gravador para o Elppa, é com a intenção de colocar a informação da fita no computador. Se o volume ajustado está muito baixo, o Elppa perderá algumas das informações, e se queixará emitindo uma mensagem de erro. Se o volume ajustado está muito alto, o Elppa também reclamará.

Para encontrar o ajuste correto de volume, você usará um método experimental. Você reproduzirá uma fita Elppasoft baixinho no computador e verá se a informação está OK. Se isto não funcionar, você experimentará a fita novamente, um pouco mais alto desta vez, e assim sucessivamente, até que o volume ideal seja conseguido, quando então, o Elppa avisará emitindo um "bip".

Para limpar a tela digite:



Coloque a fita de demonstração no seu gravador. Para cada posição do controle de volume, você vai fazer o seguinte:

1. Rebobine a fita para o começo.
2. Comece a reproduzir a fita.
3. Digite:



Quando você fizer isto, o cursor desaparecerá. Isto pode levar 15 segundos antes que algo aconteça. Há estas possibilidades:

- a. A mensagem ERRO:SINTAXE aparece.
- b. Nada realmente acontece.
- c. A mensagem ERRO aparece (com ou sem um bip)
- d. O computador emite um "bip" e nada aparece.

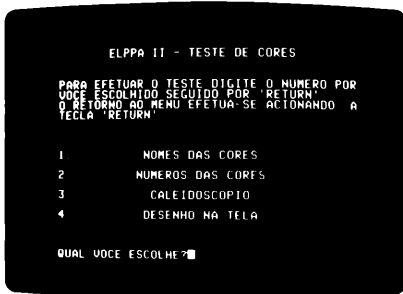
No caso a., não reajuste o controle de volume, mas volte ao passo 1. aonde você rebobina a fita.

Nos casos b. e c., tenha certeza que você esperou por 15 segundos antes de desistir. Se não aparecer o indicador] ou o cursor, e o Elppa não responder ao seu teclado, pressione **RESET**, ajuste o controle de volume um pouco mais alto e volte ao passo 1.. Ocasionalmente o comando LOAD pode não trabalhar adequadamente, e o cursor aparecerá na tela imediatamente após este comando, sem esperar que a fita seja "carregada" no computador. Se isto acontecer, desligue o seu Elppa, ligue-o novamente e reinicie pelo passo 1..

No caso d., você está no caminho certo. Quando você ouvir o bip, espere por mais 15 segundos. Ou você obterá uma mensagem de erro (caso c.), ou o indicador] e o cursor reaparecerão. Neste caso, pare e rebobine a fita. Marque a posição do controle de volume do gravador, de forma que você possa usar este ajuste cada vez que voltar a "carregar" uma fita. Então, digite

R **U** **A** **RETURN**

A tela deve assemelhar-se ao seguinte caso você tenha usado a fita com o programa DEMONSTRAÇÃO DE CORES:



PROCEDIMENTO NORMAL PARA CARREGAR FITAS

"Carregar" uma fita significa transmitir as informações da fita para a memória do computador. Para que isto seja feito, proceda da seguinte maneira (uma vez que o controle de volume do gravador foi ajustado corretamente):

1. Rebobine a fita.
2. Comece a reproduzir a fita.
3. Digite **LOAD**

Depois que você pressiona **RETURN**, o cursor desaparecerá. Nada acontece de 5 à 20 segundos, e então o Elppa emite um bip. Isto significa que a informação da fita está sendo transferida para o computador. Após algum tempo (dependendo da quantidade de informações contidas na fita, mas normalmente menos que uns poucos minutos) o Elppa emite um bip novamente e o indicador e o cursor reaparecem.

4. Pare o gravador e rebobine a fita. A informação foi transferida, e por ora você não mais usará o gravador.
5. Digite RUN e pressione **RETURN**, e o seu programa começará a ser executado.



Se o seu Elppa está na linguagem de computador Elppasoft, a fita que você está carregando deve estar também em Elppasoft. Tentar carregar uma fita numa linguagem de computador errada produz resultados que são bastante imprevisíveis. Mensagens de erro estranhas e caracteres desconhecidos podem aparecer na sua tela de TV, você pode perder o controle do teclado, ou um número qualquer de coisas estranhas pode acontecer. Se isto vier a ocorrer, desligue o seu Elppa e ligue-o novamente para que tudo volte ao normal.

Várias palavras são usadas para descrever o processo de tirar a informação de uma fita e colocá-la no computador. Diz-se que o computador "lê" a fita, e que a informação na fita foi "introduzida" ou "lida" no computador. O ato de ler uma fita é também chamado "carregar" uma fita no computador e diz-se que a informação na fita é "carregada" no computador. Todas essas expressões são formas de se dizer a mesma coisa.

UMA SUGESTÃO ÚTIL

O que é que o computador acha de tão interessante à respeito dessas fitas? Ouça uma delas. Não é música para seus ouvidos. Você ainda pode reconhecer alguns dos sons emitidos pelo computador. A informação começa com um tom constante. Então, há um pequeno "blip" seguido novamente pelo tom constante. O tom está numa frequência de 1000 Hertz. Após o tom, uma rajada de sons semelhantes a uma tempestade de chuva aparecerão.

Quando você está acostumado ao som de uma boa fita, você pode rapidamente, de ouvido, reconhecer se ela é ou não de computador. Se você puder dizer o que a fita contém escutando-a, você é um mutante e irá longe no mundo dos computadores.

ACIONANDO O SISTEMA COM UMA UNIDADE DE DISCO

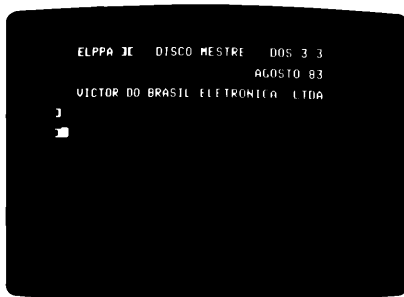
(Pule esta seção se você não está usando uma unidade de disco)

Uma unidade de disco é muito mais rápida e mais fácil de usar do que um gravador cassette, contudo, discos e unidades de disco são criaturas delicadas, e algum cuidado deve ser tomado para protegê-los. Você já deve ter lido à respeito de como cuidá-los e alimentá-los, nas páginas 18 e 19 deste capítulo.

Uma das características que fazem o disco Elppa, DISK II tão fácil de ser usado é sua habilidade para armazenar e recuperar diferentes grupos de informações. Os grupos de informações são arquivados no disco, sob nomes chamados "nomes de arquivo". Um programa que mantém pista de endereços, por exemplo, pode ser chamado no disco, de ENDEREÇOS.

Os programas que mantêm, memorizam e recuperam pistas de arquivos, e fazem uma porção de outras tarefas domésticas, são aqueles que compõem o Sistema Operacional do Disco ou DOS. O processo de adicionar as capacidades do DOS ao Elppasoft (ou à qualquer outra linguagem usada pelo seu Elppa) é chamado "carregar o DOS" ou "carregar o sistema".

Há muitas formas de carregar o DOS. Uma forma é simplesmente desligar o seu Elppa e ligá-lo novamente (o disco contido na unidade de disco 1 deve ser um disco inicializado, tal como o Disco Mestre). Você ouvirá um "bip", e o indicador de força (POWER), no canto inferior esquerdo do teclado se iluminará. Uma pequena lâmpada vermelha na frente da unidade de disco 1 se acenderá e a mesma emitirá alguns sons, na forma de zumbidos e "clicks". Após alguns instantes, os sons cessarão e a lâmpada vermelha apagará. O programa carregado do disco que está na unidade de disco 1 apresentará a sua abertura, que será exibida na tela do vídeo.



Quando você obtiver esta mensagem, saberá que o DOS está carregado.

Uma outra maneira de carregar o DOS é digitar

```
SHIFT
P R # 6 RETURN
```

no seu Elppa. Se o cartão de controle não está conectado no conector de número 6, então digite

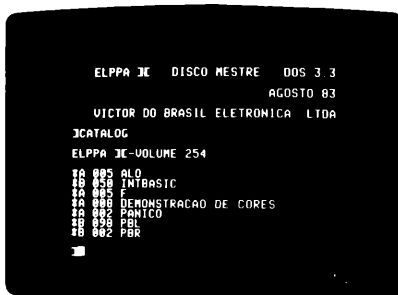
```
SHIFT
P R # 3
```

seguido do número do conector no qual o cartão está conectado, e então, **RETURN** .

O Sistema Mestre é um disco muito especial. Ele contém programas que você precisará, a fim de obter o máximo deste manual, bem como muitos outros programas úteis. Para ver que programas estão no disco, use o comando CATALOG. Simplesmente digite

```
CATALOG RETURN
```

e uma lista de nomes de arquivo aparecerá na tela.

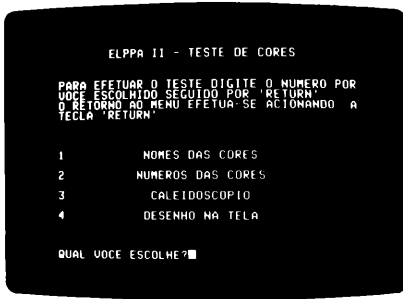


```
ELPPA IC DISCO MESTRE DOS 3 3
AGOSTO 83
VICTOR DO BRASIL ELETRONICA LTDA
ICATALOG
ELPPA IC-VOLUME 254
#A 005 ALO
#B 006 INTERASIC
#A 005 F
#A 000 DEMONSTRACAO DE CORES
#A 002 PANTICO
#B 006 PBL
#B 002 PBR
```

O primeiro programa que você precisa é chamado DEMONSTRAÇÃO DE CORES. Localize este nome no catálogo. Agora digite

```
RUN DEMONSTRACAO DE CORES
```

e então pressione **RETURN** . A tela deve parecer como a fotografia a seguir.



O MENU

Costuma-se chamar esta lista de descrições numeradas de "menu". Ela trabalha como o menu, por exemplo, de um restaurante chinês. Se você quer um frango xadrez, simplesmente pode dizer, "Eu quero o número 35". Tente selecionar uma das demonstrações coloridas digitando o seu número (seguido por um **RETURN** , certamente). Quando você estiver visualizando uma das demonstrações, simplesmente pressione **RETURN** para voltar ao "menu".

PARANDO O COMPUTADOR

Para parar o computador, use



Isto fará com que o indicador] e o cursor apareçam. O indicador lhe diz que está OK para introduzir informações digitadas ao computador. Por isso que ele é chamado de indicador de pronto: ele "prepara" você para digitar alguma coisa.

Uma vez que o computador está parado, ele pode ser ativado novamente digitando

RUN

(e, naturalmente, **RETURN** , mas você não mais precisa ser alertado sobre isto. De fato, não será de agora em diante).

Use



para parar o computador, e

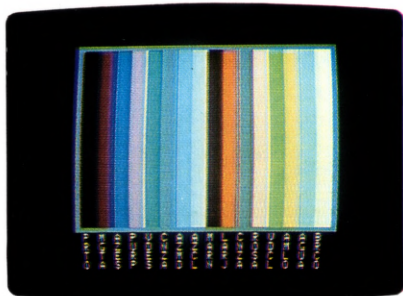
RUN

para ativá-lo novamente. Tente isto algumas vezes.

AJUSTANDO AS CORES DO TV

Se o "menu" não está na sua tela, carregue o DOS e rode (RUN) o programa chamado DEMONSTRAÇÃO DE CORES, se você estiver usando a unidade de disco. Ou, se você estiver usando um gravador cassette, siga o procedimento usual para carregar a fita marcada "DEMONSTRAÇÃO DE CORES". Um dos itens do menu é chamado NOMES DAS CORES. Nós usaremos esta DEMONSTRAÇÃO para ajustar as cores no TV. Digite o número 1 referente à esta DEMONSTRAÇÃO, e pressione **RETURN**. Um número de barras de luz (talvez em cores) aparecerão. Abaixo de cada barra está uma abreviação de quatro letras do nome da cor. Os nomes completos são:

- | | |
|-------------------------------|------------------------|
| Ø PRETO | 8 MARROM |
| 1 MAGENTA (levemente azulado) | 9 LARANJA |
| 2 AZUL ESCURO | 10 CINZA |
| 3 PÚRPURA | 11 ROSA |
| 4 VERDE ESCURO | 12 VERDE |
| 5 CINZA | 13 AMARELO |
| 6 AZUL MÉDIO | 14 ÁGUA (AZUL PISCINA) |
| 7 AZUL CLARO | 15 BRANCO |



Se você possui um televisor ou monitor preto e branco, ajuste o brilho e o contraste à seu gosto. Certamente, se a imagem está "rodando" pare da mesma forma como você o faria para qualquer programa de TV. Se você tem um TV colorido, um pouco mais de trabalho é necessário.

Lembre-se que este negócio de cor é muito subjetivo, e que você pode fazer o que quiser com a cor. As seguintes instruções lhe darão a imagem que nós gostamos, usando as cores padrão. Mas são seus olhos que você tem que agradar. Além disso, os ajustes ótimos variarão com diferentes quantidades de luz ambiente.

Desligue qualquer chave de Cor Automática. Em alguns aparelhos é marcado "AUTO COLOR" ou simplesmente "AUTO". Abaixar todo o controle de volume do TV (mas não o desligue). Quatro controles agora são importantes: Imagem, Brilho, Cor e Matiz. Alguns aparelhos possuem um botão marcado "Contraste" em vez de "Imagem", mas é a mesma coisa. Gire o controle de Imagem até à sua posição mais escura, e então diminua o Brilho até que fique completamente escuro. Gire o controle de Cor até o meio de sua faixa. Aumente o controle de Imagem de forma a clarear as coisas. Não deixe muito brilhante de modo que as cores se esparramem demais além dos limites das barras.

Agora ajuste o botão de Cor. Em um extremo, toda cor é perdida e a imagem se torna preto e branco. Este ajuste é manual, quando você está apenas mostrando o texto na tela. Ajuste o controle de Cor, até que as cores fiquem intensas mas não "fluorescentes" ou esparramando-se uma sobre a outra. Por último, ajuste o botão Matiz, até que todas as cores concordem com os seus nomes. Púrpura, Rosa e Amarelo são especialmente indicadores sensitivos. Também tenha certeza de que os três azuis estão distintos.

Quando as cores do aparelho de TV estiverem OK, pressione a tecla **RETURN** e o menu reaparecerá. Agora tente a DEMONSTRAÇÃO 2, que mostra as barras coloridas, com seus números de códigos. Tente também as outras demonstrações. Você nunca acreditará quão talentoso é o seu TV, até que você substitua as estações locais pelo seu Elppa.

CAPÍTULO 3

INICIAÇÃO À LINGUAGEM

ELPPASOFT

- 36 Iniciação à linguagem Elppasoft
- 36 A declaração PRINT
- 39 Formato do Elppasoft para números
- 40 Mais informações sobre o RETURN
- 41 Características que facilitam a edição ou: o que fazer antes de pressionar RETURN
- 43 Colorindo a tela
- 46 Mensagens de erro nos gráficos
- 47 Desenhando linhas
- 49 Os controles de jogos
- 50 Arquivos e outras possibilidades de cálculo
- 54 Prioridades ou quem vem primeiro
- 56 Como evitar as prioridades

INICIAÇÃO À LINGUAGEM ELPPASOFT

Se você está em Elppasoft, o indicador], seguido pelo cursor, aparecerão no lado esquerdo da tela cada vez que você pressionar **RETURN** . Comece a operar em Elppasoft e, se você tem uma unidade de disco, carregue o DOS.

A DECLARAÇÃO PRINT

Agora que você tem o indicador] e o cursor, (e o seu disco é carregado se você tem uma unidade de disco) você está pronto para começar a usar a linguagem Elppasoft. Digite

```
PRINT "ALO"
```

e o computador imprimirá a palavra

```
ALO
```

na próxima linha. Se ele não o fez, faça a si mesmo esta pergunta: "Eu não esqueci o **RETURN** ? Se digitar a palavra "PRINT" de forma errada, você obterá esta mensagem de erro:

```
ERRO:SINTAXE
```

Se você esquecer ainda a primeira ou ambas as aspas, o computador imprimirá um zero (você pode dizer que é um zero cortado):

```
0
```

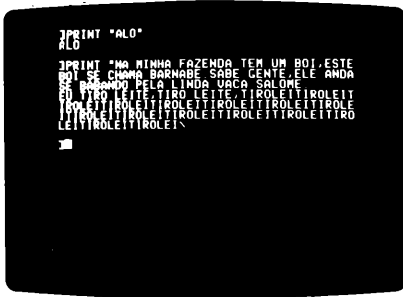
Se a aspa final é o último caracter antes do **RETURN** , você não tem que digitá-lo: a palavra "ALO" será impressa com ou sem ele digitado. De qualquer maneira, é uma boa idéia colocar a aspa final. O hábito de colocar a aspa final, tornar-se-á importante posteriormente. Este manual assumirá que você usa a aspa final.

A declaração

```
PRINT "ALO"
```

é uma instrução para o computador, informando-o para mostrar na tela todos os caracteres entre as aspas, neste caso uma palavra de saudação. Você pode usar a declaração PRINT para informar ao computador que você deseja mostrar uma determinada mensagem na tela. Contudo, se você digita acima de 240 caracteres, o

computador começará a emitir-"bips", interrompendo, através de uma barra (\), a sua mensagem e permitindo que você recomece.



Agora tente a declaração

```
PRINT "15Ø"
```

O computador, obediamente imprime o número 15Ø na próxima linha, como o esperado. Mas digite

```
FRINT 15Ø
```

e o computador novamente imprime o número, sem qualquer alarme ou mensagem de erro à respeito da não colocação das aspas. De fato, o Elppa deixará você imprimir qualquer número, sem que você precise colocá-lo entre aspas.

Sem um estudo mais aprofundado, o Elppa pode ser usado como uma simples calculadora de mesa com memória.

Tente isto no seu Elppa:

```
FRINT 4 + 5
```

A resposta 9, aparece na próxima linha. O Elppa pode fazer cinco diferentes operações aritméticas elementares:

1. ADIÇÃO. Indicado pelo usual sinal positivo (+).
2. SUBTRAÇÃO. Usa o convencional sinal negativo (-).
3. MULTIPLICAÇÃO. Muitas pessoas usam um "X" para representar a multiplicação. Isto poderia ser confundido com a letra "X". Algumas pessoas usam um ponto (.), mas isto poderia ser confundido com um ponto indicando final de sentença ou um ponto decimal. Então, o Elppa usa um asterisco (*). Para encontrar 8 vezes 9 (no caso de você não se lembrar da resposta), simplesmente digite

PRINT 8 * 9

e tenha sua memória refrescada.

4. DIVISÃO. Como é de costume, use uma barra (/). Para dividir 50 por 2, digite

PRINT 50 / 2

e a resposta correta aparecerá.

Tente dividir 7 por 2. A resposta é três e meio. O Elppa lhe fornece a resposta na forma decimal: 3.5.

Uma coisa que devemos destacar aqui, é que você pode fazer mais que uma operação aritmética na mesma instrução. Por exemplo, é válido dizer

PRINT 7 + 2 + 8 + 1

As regras exatas que governam tal uso, serão dadas posteriormente, mas você pode experimentar agora se quiser.

5. EXPONENCIAÇÃO. É frequentemente trabalhoso multiplicar um número por ele mesmo num dado número de vezes. Ao invés de se aborrecer em escrever

PRINT 7 * 7 * 7 * 7 * 7 * 7

você pode reduzir para

PRINT 7 ^ 6

O acento circunflexo é digitado da seguinte forma:

SHIFT

^
N

Não há nada de especial sobre exponenciação. É apenas uma abreviação para repetidas multiplicações! Em notação que não é de computador, isto seria escrito com um seis sobrescrito, tal como: 7⁶

Seu computador não arredondou o número, mas imprimiu da maneira que você o digitou. Loucura você diz? Ah, mas há um método para esta aparente loucura. Números são arredondados somente se eles têm mais do que nove dígitos. Qualquer número que tenha menos do que dez dígitos, não será arredondado. O computador faz o melhor que pode, mas ele tem somente nove dígitos com que trabalhar.

Se você digita uma declaração PRINT com um longo número como

```
1234567890
```

o Elppa responde com

```
1.23456789E+09
```

Os números 1234567890 e 1.23456789E+09 têm o mesmo valor. Realmente. O número impresso pelo seu computador está em "notação científica", ou seja, um número fracionário multiplicado por uma potência de 10. No exemplo, temos o número 1.23456789 multiplicado por 10 elevado à nona potência.

Experimente mais alguns números. Quantos dígitos pode ter um número sem um ponto decimal antes que o Elppa o transforme para a notação científica? Se a notação científica parece complicada, não se preocupe. Você provavelmente, não estará querendo usar números que a requeiram por algum tempo ainda. Lembre-se que qualquer número será impresso, da maneira que você o digitar, se o número está entre aspas. Contudo, o Elppa não pode usar números com aspas, para operações aritméticas.

MAIS INFORMAÇÕES SOBRE O RETURN

Até agora, você tem pressionado **RETURN** depois de cada linha, como um zumbi. Nós pensávamos que podíamos lhe dizer porque esta tecla trabalha excessivamente. A razão é simples: sem o **RETURN**, o computador não sabe quando você completou a instrução. Por exemplo, você podia começar digitando

```
PRINT 1 + 6
```

Se o computador imediatamente pulasse e imprimisse um 7, você se atrapalharia, porque tinha planejado digitar

```
PRINT 1 + 6 + 235
```

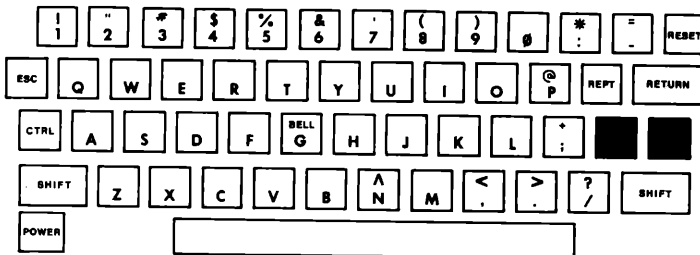
que teria dado uma resposta inteiramente diferente. Uma vez que o computador não pode dizer quando você terminou de digitar uma instrução, você deve informá-lo. Você o faz pressionando a tecla


RETURN . Desde que você sempre tenha que fazer isto, depois de digitar uma instrução, nós (como você sabe) paramos de mencionar **RETURN** após cada instrução. Pressionar **RETURN** após cada instrução deve ser um hábito a partir de agora, se você tem feito todos os exemplos.

Nós realmente esperamos que você tenha tentado todos os exemplos. Aprender a programar exige tanto quanto aprender a andar de bicicleta, tocar piano, ou jogar futebol. Você pode ler todos os livros do mundo à respeito de como andar de bicicleta, e ser um grande "teórico". Mas todo este aprendizado teórico é de pequena ajuda, quando você realmente monta numa bicicleta pela primeira vez. Uma vez que você aprendeu a andar, praticando (que pode ser um pouco doloroso), pode ir à qualquer lugar. O mesmo acontece em programação. Você pode ler este manual e achar que o entendeu. Porém não será capaz de programar. Somente fazendo os exemplos, à medida que forem aparecendo, é que você aprenderá a programar. Essa é a verdade.

CARACTERÍSTICAS QUE FACILITAM A EDIÇÃO OU: O QUE FAZER ANTES DE PRESSIONAR RETURN

Ninguém é um perfeito datilógrafo. Nós cometemos êros (Ôpa! Entende o que eu quero dizer?). O Elppa tem diversas características que auxiliam na correção de erros, evitando dessa forma, a necessidade de se redigitar uma linha inteira para cada erro cometido. Aqui é que entram as setas direita e esquerda no teclado.



A tecla  é semelhante à tecla de retrocesso em uma máquina de escrever, de forma que nós a chamaremos de "tecla de retrocesso". Umhas poucas experiências tornarão isto claro. Digite (exatamente como mostrado) a declaração:

PRINT TECLADO"

e, como de costume, pressione a tecla **RETURN**. O computador responderá

Ø

por causa da falta da aspa. Agora se tivéssemos digitado

```
PRINT "TECLADO"
```

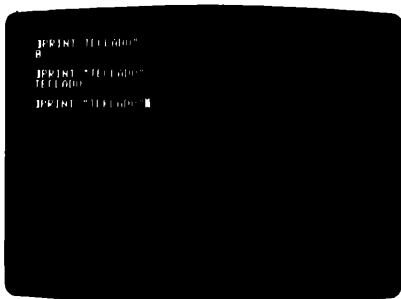
o computador teria respondido com

```
TECLADO
```

Não acredite neste manual. Tente. Agora, sem pressionar **RETURN** digite a instrução "errada":

```
PRINT "TEKLADO"
```

Desde que você pressionou **RETURN**, nada aconteceu ainda. Como mostrado na fotografia, o cursor está parado à direita da última aspa. (Sinto muito, nós não podemos fazer uma foto piscar)



Para mudar

```
TEKLADO
```

para

```
TECLADO
```

nós podemos usar a tecla **←**. Note que cada vez que você pressiona esta tecla, o cursor retrocede (para a esquerda) um espaço.

"Retrocasso" é também um verbo. Então, retroceda o cursor até o K. Digite um C. Como pode ver, o C substitui o K. Agora pressione **RETURN** . Você obteve

TEC

do computador? Isto é porque você retrocedeu além de "LADO". Qualquer caracter na linha que você está correntemente digitando que é retrocedido até o erro, não é enviado ao computador quando você pressiona **RETURN** . Uma solução seria corrigir o K retrocedendo até ele, e então digitar



Experimente.

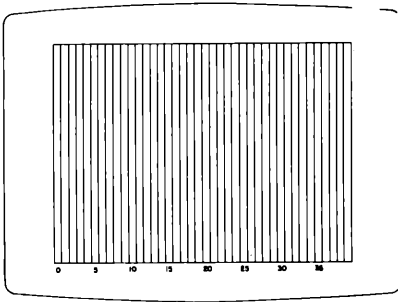
Funciona! Há, contudo, uma maneira mais fácil. Quando você pressiona a tecla **→** , o cursor se move para a direita. À medida que o cursor se move para a direita passando sobre um caracter, produz um efeito semelhante, como se o mesmo fosse novamente digitado. Nós podemos chamar a tecla **→** a tecla "avanço". Digite novamente

PRINT "TEKLADO"

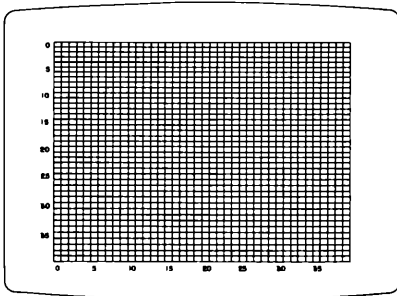
então retroceda até a letra K e troque-a por C. Para completar a correção, simplesmente pressione a tecla de avanço cinco vezes, e então pressione **RETURN** . Tudo funciona? O uso das teclas de retrocesso e avanço lhe ajudarão por muitas vezes. Faça questão de cometer erros intencionalmente, de modo que você as use tantas vezes quanto necessário, até que se tornem familiares.

COLORINDO A TELA

Para colocar gráficos coloridos na tela, precisamos de um meio para descrever qual das dezesseis cores disponíveis queremos, e onde pretendemos colocá-las. Para isso, dividimos a tela em 40 colunas verticais, numeradas de 0 a 39. A coluna 0 situa-se à extrema esquerda da tela, e os números aumentam para a direita. Você pode perguntar, por que os números não vão de 1 a 40 ao invés de 0 a 39. À medida que você se tornar mais experiente em programação, descobrirá que a escolha que fizemos é a mais adequada, mesmo que não pareça à primeira vista.



A tela também é dividida em 40 linhas horizontais, novamente numeradas de 0 à 39. As linhas horizontais começam com a linha 0 no topo da tela e aumentam até a linha 39 na base. Estas linhas se cruzam com as colunas, dividindo cada coluna em 40 "quadrados", numerados de 0 à 39. Para aqueles que gostam de terminologia acadêmica, reconhecerão que isto é meramente um sistema de coordenadas Cartesianas retangulares. Para aqueles que não gostam, pensem em termos de colunas de quadrados.



Para obter cores na tela, digite a seguinte instrução:

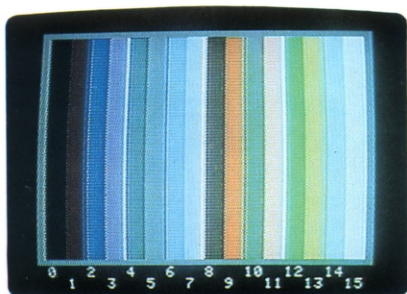
GR

Você se lembrou do **RETURN**, sem dúvida. Quando você usa este comando, a tela é automaticamente limpa, deixando apenas quatro linhas para texto, na parte inferior. "GR" substitui a palavra GRáficos. Para retornar ao conteúdo anterior (antes que você digitasse GR) use o comando

TEXT

Quando você digitar este comando, a tela rapidamente se encherá de uma porção de sinais (@). Isto é normal. Tente digitar a instrução TEXT, e então voltar aos gráficos, digitando a instrução GR.

Antes que você possa colocar um ponto colorido na tela, você deve dizer ao computador que cor deseja. Há dezesseis cores disponíveis. Você já as viu anteriormente: elas estão numeradas de 0 a 15, como mostrado na DEMONSTRAÇÃO DE CORES 2.



Suponha que você queira colocar um ponto verde em algum lugar da tela. Você deve primeiro digitar o comando GR, e então digitar

```
COLOR = 12
```

Isto significa que, qualquer ponto (ou um quadrado) colorido que você coloque, será verde. De fato, até que outra instrução seja formulada, tudo que o computador põe na tela será verde. Exceto, é claro, para a pequena área reservada na base da tela, para suas instruções. Para colocar um ponto colorido no canto superior esquerdo da tela (extrema esquerda ou coluna zero, topo ou quadrado zero), você digita

```
PLOT 0,0
```

Para colocar um ponto da mesma cor no canto superior direito da tela, você deve especificar: coluna 39, quadrado 0. Então digite

```
PLOT 39,0
```

Note que você sempre fornece primeiro a coluna. Agora coloque um quadrado cor de laranja no canto inferior esquerdo. Primeiro mude a cor. Lembre-se que você deve realmente estar fazendo estes exercícios, e não somente pensando neles. Então, prepare os seus dedos e digite

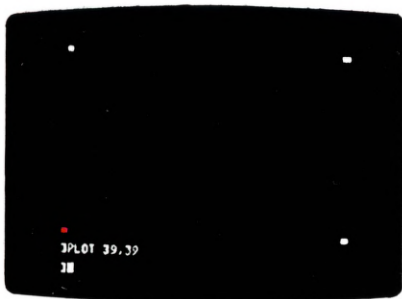
```
COLOR = 9
```

Nada acontece na parte superior da tela (mesmo que você se lembrou de pressionar **RETURN**). Mas o computador lembra que o seu próximo traçado, será em laranja, não em verde. Agora que você escolheu a cor, pode colocar um ponto no canto inferior esquerdo, que é a coluna Ø, quadrado 39:

PLOT Ø,39

Funcionou? Você esqueceu de pressionar **RETURN** ? É laranja a sua cor predileta?

Agora coloque um ponto magenta, no canto inferior direito. Compare a sua tela com a fotografia abaixo.



MENSAGENS DE ERRO NOS GRÁFICOS

Há duas mensagens de erro que podem facilmente aparecer, quando você está usando a declaração PLOT. Você já sabe que se digitou

PLAT

ou

PLDF

ao invés de

PLOT

você obteria a mensagem

ERRO:SINTAXE

Uma nova mensagem de erro ocorre quando você traça um número superior ou inferior que aqueles permitidos para coordenadas num comando PLOT. Digite

PLOT 13,79

e você obtém a mensagem

ERRO:QUANTID ILEGAL

Esta mensagem significa que você tentou traçar um ponto fora da faixa e da tela. Os maiores números que você pode usar numa declaração PLOT, são 39, para a primeira coordenada, e 47 para a segunda. O uso de números acima de 39, para a segunda coordenada, como em uma declaração, tal como

PLOT 20,44

somente lhe fornecerá caracteres peculiares na área de texto, na base da tela.

Tentar usar valores negativos em um comando PLOT, é uma outra forma de se obter a mensagem

ERRO:QUANTID ILEGAL

DESENHANDO LINHAS

Suponha que você queira desenhar uma linha horizontal azul clara, da coluna 5 até a coluna 9, na altura do quadrado 4. Você poderia digitar

```
COLOR = 7  
PLOT 5,14  
PLOT 6,14  
PLOT 7,14  
PLOT 8,14  
PLOT 9,14
```

Note que as junções entre os quadrados adjacentes, não aparecem, e formam uma linha contínua. Contudo, há uma maneira mais fácil de traçar linhas horizontais. Suponha que você deseja traçar uma linha horizontal branca no meio da tela. Usando a forma mais longa, você teria que digitar 40 declarações:

```
COLOR = 15
PLOT 0,20
PLOT 1,20
PLOT 2,20
```

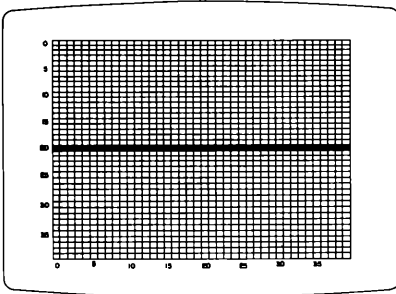
e assim por diante, até

```
PLOT 39,20
```

A maneira mais fácil é esta. Somente digite

```
COLOR = 15
HLIN 0,39 AT 20
```

Pressione a tecla **RETURN**, e você obtém: uma LINha Horizontal imediatamente traçada, desde a coluna 0 até a coluna 39, na altura do quadrado 20.



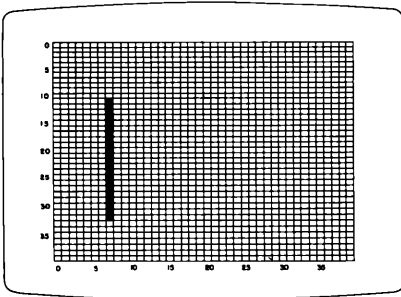
Agora tente colocar uma linha púrpura, da coluna 19 até a 28, na altura do quadrado 18. Experimente outras cores. Fazendo cerca de 6 linhas horizontais diferentes, você já estará familiarizado com o processo.

Note que, quando você coloca um ponto ou linha colorida, no mesmo lugar de outro (a) já existente, a nova cor toma lugar, e a anterior desaparece. Para limpar de uma vez a tela, de todos os gráficos, use o comando GR.

Há um recurso para traçar linhas verticais automaticamente, semelhante àquele utilizado para as linhas horizontais. Para desenhar uma LINha Vertical branca na coluna 7, do quadrado 12 ao 33, escrevemos:

```
COLOR = 15
VLIN 12,33 AT 7
```

Experimente esta declaração.



Pratique traçando diversas linhas verticais, mudando os números para as linhas e colunas. Você pode testar a sua habilidade com ambas as linhas horizontal e vertical, desenhando um contorno vermelho ao redor da tela com cinco declarações. Então, coloque uma cruz verde na tela. Tente desenhar algumas linhas com COLOR = Ø. Brinque com PLOT, HLIN e VLIN, por enquanto. Este manual se auto destruirá em cinco segundos, se você não experimentar estes comandos. Pfffsssss.

OS CONTROLES DE JOGOS

Caso você tenha os CONTROLES DE JOGOS, apanhe-os e vamos aprender a utilizá-los. Você já deve ter visto no primeiro capítulo, como conectá-los; agora faremos uma descrição sobre os comandos Elppasoft associados a esses controles. Apanhe um dos controles e digite

```
PRINT PDL(Ø)
```

e um número deve aparecer. Gire um pouco o controle. Agora digite novamente

```
PRINT PDL(Ø)
```

Experimente girar o controle, digitando

```
PRINT PDL(Ø)
```

Se o número nunca muda, você está com o controle de jogo errado. Quais são os números maiores e menores que você pode conseguir? Qual é a menor mudança de números que você pode obter?

Você pode descobrir a posição do outro controle digitando PRINT PDL(1). A abreviação "PDL" vem da palavra "PADDLE" (raquete), uma vez que esses controles são frequentemente usados para controlar "raquetes" em jogos. Há muitos outros usos para esses controles.

PDL é uma FUNÇÃO. Uma função, no Elppasoft, é uma instrução que usa um ou mais números e então, executa alguma operação sobre eles, fornecendo um único valor. Os números que a função usa, são chamados seus ARGUMENTOS e são sempre colocados entre parênteses após o nome da função. PDL é uma função que tem apenas um argumento. O número que a função encontra deve RETORNAR ao programa.

ARQUIVOS E OUTRAS POSSIBILIDADES DE CÁLCULO

Em inúmeras calculadoras, você pode guardar um número para mais tarde referir-se a ele ou até mesmo usá-lo. Para fazer isso, você coloca o número num lugar especial na calculadora - um lugar que chamaremos, por ora, um arquivo. Usualmente, isto é feito pressionando a tecla marcada "M" de "Memória". No Elppa você pode fazer a mesma coisa. Por exemplo, para guardar o valor 55, você digita

```
M = 55
```

O valor 55, não é impresso, apenas é armazenado no arquivo chamado M. Se agora, você digita

```
PRINT M
```

o computador imprimirá o valor de M. Tente digitar os dois comandos.

Agora digite

```
M = 247
```

e imprima (PRINT) o valor de M. É 247, certo? O que aconteceu ao 55? Ele se foi para sempre. O arquivo somente guarda um único valor por vez. Quando você põe um novo valor em M, o valor antigo é apagado.

Digite

```
PRINT "M"
```

O que acontece? Há uma grande diferença entre

```
M
```

```
e
```

```
"M"
```

É tal qual a diferença entre estas duas declarações:

```
GATO TEM QUATRO PATAS
```

```
"GATO" TEM QUATRO LETRAS
```

No primeiro caso, estamos nos referindo a um pequeno animal com longo rabo. No outro caso, estamos nos referindo à palavra GATO. Essa é a razão porquê as aspas são usadas em computadores. Quando digitamos

```
PRINT "M"
```

significa que queremos imprimir a letra M. Quando digitamos

```
PRINT M
```

significa que queremos imprimir o seu conteúdo. Você nunca confundiria o nome de alguém que você ama, com uma pessoa que possui o mesmo nome.

Você pode armazenar o resultado de uma operação aritmética num arquivo. Por exemplo:

```
M = 3 + 4
```

Você pode verificar que o resultado foi armazenado, mandando imprimir (PRINT) o valor de M.

Você pode também usar o valor de M numa operação posterior. Por exemplo, tente isto no seu Elppa:

```
PRINT M + 2
```

É a resposta que você esperava? Tente outros cálculos usando M.

Uma simples calculadora tem apenas um arquivo. Computadores têm centenas de arquivos(o seu Elppa tem 936). Utilizaremos o termo variável para designar um arquivo. Mas este termo é de certa forma impróprio, uma vez que os arquivos não se comportam como "variáveis" em matemática. Elas são muito mais simples. Cada variável é simplesmente um lugar onde um valor é armazenado. Porém, modificaremos o seu significado. Somente esqueça a matemática que você aprendeu. No Elppa todas as variáveis têm o valor zero, até que nelas sejam introduzidos outros valores.

Um arquivo, ou variável, pode ter quase todos os nomes que você desejar, desde que comece com uma letra. Por exemplo:

SOMA = 99 + 31 + 1785 + 6
RESULTADO = 66
JOGADOR2 = 3

Alguns nomes não são permitidos porque eles incluem uma palavra que tem um significado especial para o Elppa. Eles são conhecidos como palavras reservadas. Uma destas palavras é "COLOR". Deste modo, um nome de uma variável, não pode incluir a palavra "COLOR". Experimente digitar

COLORIDO = 7

ou

COLORADO = 2

Tudo que você obtém é uma mensagem de erro. Sempre que o nome de uma variável resulta em uma mensagem ERRO:SINTAXE, significa que você acidentalmente incluiu uma palavra reservada no nome. Não se preocupe. Escolha outro nome.

Uma lista de palavras "reservadas", que não podem ser usadas como variáveis ou parte de nomes de variáveis, pode ser encontrada no Apêndice B, no final deste manual.

Quando você estiver escolhendo nomes, faça com que eles expressem o significado que você pretende dar às variáveis. Isto as tornará mais fáceis de serem lembradas.

Experimente digitar

AVE = 11

e então

PRINT AVE

Você obteve o que esperava? Agora digite

PRINT AVIAO

O que acontece? Tente

PRINT AVELA

e

PRINT AVISO

Se você observar os nomes, notará que todos eles começam com "AV". O Elppasoft usa somente os dois primeiros caracteres de cada nome de variável para distinguir uma variável da outra. Então, o nome

AVE

se refere à mesma variável que

AVIAD

e

AVELA

e assim por diante.

Aqui está uma regra útil. Digamos que você tinha algum valor na variável PREÇO, e você queria aumentar este valor em 6 unidades. Uma forma de fazer isto, seria imprimir (PRINT) o valor de PREÇO, então adicionar 6 àquele valor, e finalmente armazenar o resultado, novamente em PREÇO. Por exemplo:

```
PRECO = 30
PRINT PRECO
PRINT 30 + 6
PRECO = 36
```

Mas veja como é bem mais fácil digitar

```
PRECO = 30
PRECO = 30 + 6
```

Tente digitar os próximos comandos, na seguinte ordem:

```
PRECO = 4
PRINT PRECO
PRECO = PRECO + 2
PRINT PRECO
PRECO = PRECO * 7
PRINT PRECO
PRECO = PRECO / 3
PRINT PRECO
```

Ao fim desta sequência de comandos, você provavelmente deverá obter o valor 14. Está correto? É isto que você esperava? Tente esta sequência:

```
PERAS = 88
BANANAS = 22
QUOCIENTE = PERAS / BANANAS
PRINT QUOCIENTE
```

Primeiro pense na resposta que você espera, então veja se acertou. Se não, descubra por que. Finalmente, tente estes comandos:

```
ALO = 320
PRINT "ALO"
ALO = ALO / 2
PRINT "ALO"
ALO = ALO / 2
PRINT ALO
```

O que você esperava? O que obteve?

PRIORIDADES OU QUEM VEM PRIMEIRO

Em certos banquetes formais, as pessoas eram servidas de acordo com um planejamento rigoroso: primeiro o convidado de honra, depois as mulheres (de acordo com a posição social dos maridos), então os homens (de acordo com a sua posição social), e por último o anfitrião. Não importa aonde estivessem sentados, o garçon andava entre eles, escolhendo quem deveria ser servido primeiro. Poderíamos dizer que havia uma certa prioridade entre os participantes do jantar. Num cálculo simples como

```
PRINT 6 + 9 / 3
```

você não pode dizer se a resposta deve ser 5 ou 9, até saber qual é a ordem (ou prioridade) das operações aritméticas. Se você somar 6 com 9, obtém 15. Se você então dividir 15 por 3, obterá 5. Esta é uma possível resposta. Contudo, se você somar 6 com 9 - dividido - por - 3, você tem 6+3, ou 9. Esta é uma outra possível resposta. Nove, é a resposta que o seu Elppa lhe dará. Veja a seguir, como o Elppa escolhe a ordem das operações aritméticas:

1. Quando o sinal de menos é usado para indicar um número negativo, por exemplo

$$-5 + 3$$

o Elppa, primeiro aplicará o sinal menos ao número ou variável assim designada. Então $-5+3$ resulta -2 . Se o Elppa fizesse a adição primeiro, $-5+3$ resultaria -8 . Mas isto não acontece. Um outro exemplo é

```
JOAO = 4
PRINT -JOAO + 12
```

A resposta é 8. (Note, contudo, que na expressão 6-2 o sinal de menos está indicando subtração, e não um número negativo).

2. Após aplicar os números negativos, o Elppa executa a exponenciação. A expressão

```
B + 4 ^ 2
```

é calculada, primeiro elevando 4 ao quadrado (quatro vezes quatro igual a dezesseis), e depois somando 8, dando um total de 24. Quando há um número de exponenciações, a operação é feita da esquerda para a direita, então:

```
2 ^ 4 ^ 2
```

é calculada, multiplicando 2 por ele mesmo quatro vezes ($2*2*2*2$) que é 16, e então multiplicando este valor por ele mesmo (16). A resposta é 256.

3. Depois que todas as exponenciações forem calculadas, todas as multiplicações e divisões são feitas, da esquerda para a direita. Operadores aritméticos de mesma prioridade são sempre calculados da esquerda para a direita. Multiplicação (*) e divisão (/) têm a mesma prioridade.
4. Por fim, todas as adições e subtrações são feitas, da esquerda para a direita. Adição (+) e subtração (-) têm a mesma prioridade.

Vamos resumir a ordem de prioridades do Elppa, para executar as operações matemáticas:

- Primeiro: - (sinal de menos, usado para indicar números negativos)
Segundo : ^ (exponenciação, da esquerda para a direita)
Terceiro: */ (multiplicação e divisão, da esquerda para a direita)
Quarto +- (adição e subtração, da esquerda para a direita)

A seguir, você encontrará algumas expressões aritméticas para calcular. Primeiro, execute cada uma delas, mentalmente (ou com o auxílio de uma calculadora, ou lápis e papel), e então tente no Elppa. Se a sua resposta não conferir com a do Elppa, procure descobrir o por quê. Nós apenas lhe daremos as expressões. Você terá que colocar um PRINT na frente de cada uma delas para obter a resposta do computador.

A menos que você tenha muita experiência com cálculos de expressões em computadores, você deve realmente fazer esses exemplos. Não faça tudo de uma vez só, para então checar com o computador. Faça um exemplo à mão e depois confira o seu resultado com o computador. Então, passe para o seguinte, e assim por diante.

$3 + 5$
 $2 + 7 - 4 + 3$
 $6 * 7$
 $3 ^ 2 + 1$
 $8 / 2 + 3$
 $8 - 6 / 3$
 $6 / 3 - 1$
 $5 * -2 + 8 / 2 + 8$
 $7 + -4$
 $2 ^ 3 - 3 + 2$
 $3 * 4 * 5 + 2$
 $4 * 5 + 2 * 2$
 $3 * 3 * 2 + 4$
 $9 / 3 / 3 / 1$
 $7 * 3 / 3 + 3 * 4 - 2 * 5$
 $15 / 3 * 4$

As respostas não são fornecidas neste manual. Seu Elppa lhe dará as respostas corretas.

COMO EVITAR AS PRIORIDADES

Suponha que você queira dividir 15 por três mais dois. Se você escreve

$15 / 3 + 2$

você obterá 15-dividido-por-três, depois somado com dois. Mas isto não é o que você desejava. Para efetuar o que você realmente estava querendo, pode escrever:

$15 / (3 + 2)$

O parênteses modifica a ordem da prioridade. A regra que o computador segue é simples: faça primeiro o que está entre parênteses. Se há parênteses dentro de parênteses, faça primeiro a operação dos mais internos. Aqui está um exemplo:

$40 / (4 + (3 + 1) - 2)$

Neste caso, fazendo a operação dos parênteses mais internos, você primeiro soma $3 + 1$. Agora a expressão fica, efetivamente,

$$40 / (4 + 4 \wedge 2)$$

Mas você sabe que $4 + 4 \wedge 2$ é $4 + 16$ ou 20 e portanto, a expressão fica simplificada para $40/20$, que é dois.

Num caso como $(7 + 3) * (4 + 6)$, onde há mais que um conjunto de parênteses, mas não estão agrupados um dentro do outro, você simplesmente trabalha da esquerda para a direita. Esta expressão se torna $10 * 10$, ou 100 .

Aqui estão mais algumas expressões para calcular. Novamente, se você não está familiarizado com computadores, os poucos minutos que você realmente passa trabalhando nisto e tentando resolvê-los no seu Elppa, lhe serão muito valiosos. Você se sentirá recompensado pelos seus esforços, ao ser capaz de usar o computador mais eficientemente. Casualmente, a maioria dessas regras de prioridade e parênteses são válidas para quase todos os sistemas de computadores do mundo, não somente para o seu Elppa.

$$66 / (3 + 3)$$

$$(66 / 3) + 3$$

$$6 + (-4 * 3)$$

$$(6 + -4) * 3$$

$$500 / (80 / (4 * (7 - 5)))$$

$$41 / (2 + (8 / 3) + (7 / 4))$$

CAPÍTULO 4

PROGRAMAÇÃO

ELEMENTAR

- 60 Execução adiada
- 64 Edição elementar
- 66 Acrobacias elementares
- 67 Comandos que facilitam a edição
- 68 O cursor móvel eliminando ou copiando o que foi escrito
- 70 Uma palavra sobre o aprendizado da linguagem Elppasoft
- 70 Como evitar acidentes
- 71 Verdadeiro ou falso
- 75 Ordem ou prioridade para operações
- 75 A declaração IF
- 76 Guardando programas em um disco
- 78 Guardando programas com um gravador cassette
- 78 Outros programas com gráficos
- 81 Loops com as declarações FOR e NEXT
- 83 Um programa errado
- 84 Um último exemplo de loops internos
- 84 ganhando brilho
- 85 Imprimindo com detalhes

EXECUÇÃO ADIADA

Não, esta seção não é uma suspensão temporária da aplicação de uma pena a um criminoso. Até agora, quando você digitava

```
PRINT 5 + 2
```

e pressionava a tecla **RETURN**, o computador fazia o que você dissesse para fazer, imediatamente. Quando um computador executa de acordo com a declaração que você emitiu, isto quer dizer que ele entendeu que era para executá-la. Dessa forma, você tem usado o computador para fazer uma EXECUÇÃO IMEDIATA para cada declaração que você digitar no teclado do Elppa.

Agora, você vai aprender como armazenar declarações para serem executadas posteriormente (EXECUÇÃO ADIADA). Para garantir que a memória do computador está limpa de quaisquer programas anteriores, digite

```
NEW
```

Como quase tudo que você já viu, NEW tem que ser seguido por um **RETURN**. Para mandar o computador armazenar uma declaração, simplesmente digite um número antes de digitar a declaração. Por exemplo, se você digita

```
100 PRINT 5 + 2
```

nada parece acontecer, mesmo se você pressiona **RETURN**. O Elppa armazenou a declaração. Para confirmar se a declaração foi armazenada, você digita a instrução

```
LIST
```

Tente. A menos que você digitou erradamente (e provavelmente obteve um

```
ERRO:SINTAXE
```

pelo seu esforço).

```
100 PRINT 5 + 2
```

aparece na tela. Agora digite a declaração

```
RUN
```

e a resposta

aparece na tela.

Ao digitar RUN, a instrução armazenada foi executada, mas o computador não se esqueceu dela. Você pode rodar (RUN) a mesma instrução quantas vezes desejar. Experimente.

Ademais, o computador não esquece a instrução armazenada mesmo quando você limpa a tela. Aqui temos uma nova maneira de limpar a tela:

HOME

O comando HOME tem o mesmo efeito como



que você aprendeu anteriormente, mas ele pode ser usado em execução adiada bem como em execução imediata. Para tentar isto, digite

100 HOME

Agora, quando você digita

RUN

o computador fielmente executa a instrução armazenada e limpa a tela. Digite

NEW

e depois

LIST

e veja o que acontece. Ao digitar NEW a instrução armazenada é perdida permanentemente. Digite

RUN

e nada aparece na tela. Isto se deve ao fato da sua instrução antiga ter sido apagada pelo comando NEW.

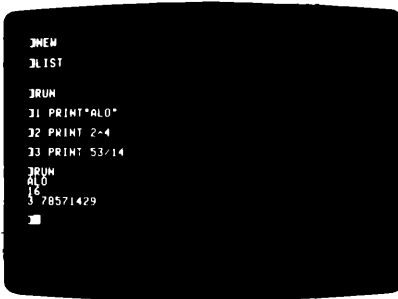
É possível armazenar várias instruções, dando a cada uma delas um número diferente. Experimente digitar isto:

```
1 PRINT "ALO"  
2 PRINT 2 * 4  
3 PRINT 53 / 14
```

Nada aconteceu até aqui. Mas, agora digite

RUN

e veja as respostas aparecerem.



Os números que colocamos na frente das instruções, com a finalidade de dizer ao computador para armazená-las, são chamados de NÚMEROS DE LINHAS. O computador armazena e executa as instruções em ordem crescente dos números de linhas. Para ver isto em ação, apague as instruções que você armazenou, digitando

NEW

e então digite estas declarações:

```
1 PRINT "L"  
0 PRINT "E"  
3 PRINT "A"  
2 PRINT "P"
```

Note que o zero é um número de linha permitido. O maior número de linha que você pode usar é 63999. Agora rode (RUN) estas instruções. O resultado deve aparecer como este:

```
E  
L  
P  
A
```

Para ver o que aconteceu dentro do Elppa, digite

LIST

Note que você não tem que listar (LIST) um conjunto de instruções antes de rodá-las (RUN), contudo, é uma boa idéia fazê-lo.

Um conjunto de instruções que são executadas quando você digita RUN, é chamado um PROGRAMA.

O programa deveria imprimir

```
E
L
P
P
A
```

mas, parece que uma declaração PRINT foi esquecida. Como você pode adicioná-la no programa? A única maneira é redigitar as instruções da linha 3 e acrescentar uma nova linha de número 4. Para fazer as correções, digite isto:

```
3 PRINT "P"
4 PRINT "A"
```

Para ver o que aconteceu, liste (LIST) o programa.

Note que, em qualquer ordem que sejam inseridas as instruções, o Elppa armazena-as com seus números de linhas em ordem numericamente crescente. Agora rode (RUN) o programa.

Foi um incômodo ter que redigitar aquelas instruções, a fim de simplesmente inserir uma instrução no meio. Portanto, é bom como prática de programação, deixar algum espaço de número de linha entre linhas, e antes da primeira linha. Digite

NEW

para eliminar o programa anterior, e colocar este:

```
100 PRINT "G"
110 PRINT "A"
120 PRINT "Q"
```

Quando você roda (RUN) o programa, este não imprime totalmente a palavra "GATO" na vertical. Mas agora, você pode retornar e digitar

```
115 PRINT "T"
```

Liste (LIST) e rode (RUN) este programa. De agora em diante, este manual iniciará todos os programas com número de linha razoavelmente alto e deixará uma quantidade de espaços entre números de linhas sucessivas, de modo que haverá espaço adequado para inserção de instruções.

EDIÇÃO ELEMENTAR

Anteriormente, você descobriu que a instrução

```
PRINT PDL(Ø)
```

imprimiria um número correspondente à presente posição de um dos controles de jogos. Foram necessários um número razoável de declarações PRINT, para se familiarizar com este controle. Agora que você pode escrever programas, a vida ficou mais fácil. Limpe o computador com a instrução

```
NEW
```

e digite

```
1ØØ PRINT PDL(Ø)
```

Agora, cada vez que você digita RUN, este pequeno programa é executado e você pode ver a posição do controle de jogo.

Para se fazer algo mais de uma vez, o programa armazenado já está lhe poupando algum trabalho. Antes, você tinha que redigitar uma declaração inteira ou grupo de declarações. Agora, você simplesmente redigita

```
RUN
```

A execução diferida dá outra vantagem. Você pode modificar parte de um programa e manter o restante, sem ter que redigitar tudo novamente. Por exemplo:

```
NEW
2ØØ P = PDL(Ø)
21Ø PRINT P
22Ø PRINT "MOVA O CONTROLE DE JOGO"
23Ø PRINT "PARA UMA NOVA POSICAO"
```

Rode (RUN) este programa algumas vezes, mudando a posição do controle de jogo entre um RUN e outro. Verifique se o programa responde a ambos os controles de jogos. Ele deve funcionar somente para um deles. Aproveite esta oportunidade para marcar este controle com o número zero.

Este mesmo programa pode ser usado, com uma leve mudança, para o outro controle de jogo. Liste o programa na forma que está agora, então digite

```
200 P = PDL(1)
```

Quando você digita uma declaração com o mesmo número de linha de uma outra já existente em um programa, a nova linha substitui a anterior. Liste (LIST) o programa para confirmar a mudança. Rode (RUN) o programa umas poucas vezes para ver o que acontece. Mova o outro controle de jogo entre um RUN e outro. Este programa responde a ambos os controles? Marque um número 1 no controle ao qual este programa responde.

Modificando um programa desta maneira, é um exemplo de editar um programa. Outra forma, é usar o que você já aprendeu em como apagar linhas, que você não mais quer em seu programa. Se você quisesse apagar a linha 230 no programa precedente, você digitaria

```
230
```

e então pressionaria **RETURN**

Você também poderia ter usado a instrução DEL. Para apagar (DEL) a linha 230 do seu programa você digitaria

```
DEL 230,230
```

As vantagens da instrução DEL, não são tão evidentes, até que revelemos para você que seções inteiras de programas podem ser apagadas com instruções como

```
DEL 200,230
```

que apaga (DEL) todas as declarações situadas entre os números de linhas 200 e 230 inclusive. Tente estes comandos, e liste (LIST) o programa para verificar o que ocorreu: A possibilidade de se apagar (DEL) blocos de declarações de linhas numeradas será muito útil quando você estiver escrevendo programas longos.

Como você já viu, há diversos comandos que o ajudam a lidar com programas inteiros. São eles:

NEW

que apaga programas,

LIST

que lista os programas, e

RUN

que executa programas, começando com a declaração que tem o menor número de linha. É possível, também, começar uma execução em qualquer outra linha, e listar (LIST) apenas parte de um programa. Estas possibilidades serão explicadas posteriormente.

ACROBACIAS ELEMENTARES

A partir deste ponto, você está começando a voar, pois esta seção discutirá os "loops", ou rotinas fechadas de programação.

A melhor maneira de ver como a função PDL trabalha -- e entender o programa "loops" -- é usar uma declaração que ainda não discutimos. Ela é muito simples. Digite as seguintes linhas (depois de digitar NEW, para apagar qualquer programa que ainda se encontre na memória):

```
11Ø PRINT PDL(Ø)  
12Ø GOTO 11Ø
```

A linha 11Ø deste programa imprime (PRINT) o número que representa o atual valor do controle de jogo. A linha 12Ø faz simplesmente o que parece dizer: ela obriga o programa a voltar para a linha 11Ø. O que acontece então? O programa imprime o valor atual do controle de jogo. Então, ele executa a linha 12Ø, que diz para executar a linha 11Ø novamente, e assim por diante. Isto é um "LOOP". Um "loop" é uma estrutura de programa que existe quando o programa inclui um comando para retornar a uma instrução executada previamente. Rode (RUN) o programa. Mexa com o controle de jogo. Na próxima seção, lhe diremos como parar este programa. Por enquanto, contemple o fato de que -- se você digitou RUN conforme foi dito há três sentenças atrás -- seu Elppa já executou a declaração PRINT PDL (Ø) algumas centenas de vezes. Agora, a força de um programa armazenado começa a aumentar significativamente muito além da sua habilidade manual. Suas possibilidades com o computador aumentarão dramaticamente nas próximas seções, agora que uma boa base de trabalho foi estabelecida.

COMANDOS QUE FACILITAM A EDIÇÃO

Mas primeiro, provavelmente você está desejando saber como parar o programa dos controladores. Você já deve ter notado como os números cintilam na tela à medida que você move o controle. Isto se deve ao fato dos números serem impressos na base da tela e à medida que cada número novo é impresso, todos os outros são movidos uma linha acima. Isto é chamado "rotação". Para parar o programa, simplesmente use

CTRL

C

O comando **CTRL C** permite que você saiba aonde a execução do programa foi interrompida, imprimindo no vídeo

INTERROMPIDO EM 110

ou, qualquer que seja o número de linha em que o programa foi interrompido. (Experimente). A propósito, esta é uma exceção à regra de teclar **RETURN** após todos os comandos. Normalmente não é necessário pressionar **RETURN**, quando um programa é interrompido com **CTRL C**. Você pode também usar a tecla **RESET** para interromper programas se assim o desejar, mas, com **RESET** não obterá a mensagem que lhe diz em que número de linha o programa foi interrompido.

Quando você interrompe um programa com **CTRL C** ou **RESET**, você pode continuar a execução do mesmo digitando a instrução

CONT

que significa "CONTINUE". Experimente, e então tente este programa:

NEW

```
100 X = FDL(0)
110 PRINT "CONTROLE DE JOGO ZERO ESTA EM"
120 PRINT X
130 Y = FDL(1)
140 PRINT "E CONTROLE UM ESTA EM"
150 PRINT Y
```

Anteriormente, dissemos que quando você digita RUN, o programa começa a ser executado no menor número de linha. Verdade! Contudo, se você quiser começar a rodar (RUN) a partir de outra linha, como por exemplo, a linha 130, simplesmente digite

RUN 130

Você também pode especificar números de linhas na declaração LIST. Se você digita

LIST 130

o Elppa listará a linha 130 (se houver uma, certamente). Se você digita

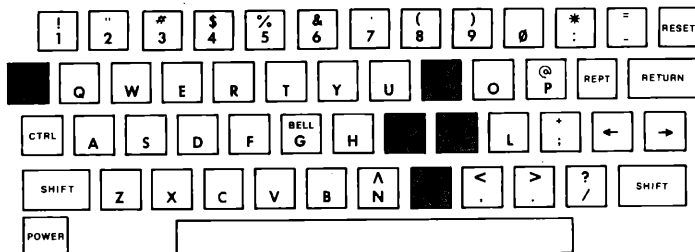
LIST 110,130

o Elppa listará todas as linhas do programa começando da linha 110 e continuando até a linha 130. Esta característica não é disponível com a instrução RUN.

O CURSOR MÓVEL ELIMINANDO OU COPIANDO O QUE FOI ESCRITO

Quando as teclas de retrocesso e de avanço são pressionadas, elas movimentam o cursor. Mas, também, eliminam ou redigitam caracteres, como você já aprendeu no Capítulo 3. É possível mover o cursor sem afetar nada (exceto a sua posição). Você faz isto simplesmente movendo o cursor.

Cinco teclas são usadas para executar os movimentos do cursor. Elas são: **ESC**, **I**, **J**, **K** e **M**. Aqui está, como usá-las.



Primeiro você coloca o Elppa em modo de edição, pressionando e soltando a tecla **ESC**, então use **I** para mover o cursor para cima, **J** para movê-lo para a esquerda, **K** para movê-lo para a direita, e **M** para movê-lo para baixo. Para mover o cursor repetidamente, pressione uma das teclas indicativas da direção do cursor (**I**, **J**, **K** ou **M**) ao mesmo tempo que pressiona a tecla **REPT**. O cursor andará rapidamente, enquanto as duas teclas estiverem pressionadas. Se o cursor atinge o topo da tela, ele irá parar, e se atingir a base da mesma, ele pára e o conteúdo que se encontra na tela se moverá para cima, uma linha por vez. Se ele atinge o canto direito da tela, ele desaparecerá e reaparecerá no

começo da linha seguinte. Pratique, movendo o cursor ao redor da tela com estas cinco teclas.

Quando você ficar cansado de movimentar o cursor, simplesmente pressione a barra de espaço, e descobrirá que tudo irá voltar ao normal.

Os simples movimentos do cursor causados pelas teclas **J** e **K**, quando vistos na tela, aparecem da mesma forma que os movimentos causados pelas teclas de retrocesso e avanço, mas o efeito é diferente como você verá quando listar (LIST) os resultados. Os simples movimentos do cursor não causam nenhuma mudança no texto sobre o qual eles passam, enquanto as teclas de retrocesso e avanço, apagam ou redigítam os caracteres sobre os quais elas passam.

Estes simples movimentos do cursor têm uma aplicação, não tão simples. Por exemplo, digite:

```
13Ø PRUNT "A QUALIDADE DO ELPPA"  
14Ø PRINT "E INCOMPARAVEL"
```

pressionando **RETURN** após cada linha. O seu Elppa parece ter aceitado as declarações, mas quando você tentar rodar (RUN) o programa, obterá a mensagem

```
ERRO:SINTAXE EM 13Ø
```

para a sua aflição.

Para fazer a correção necessária, você pode usar este artifício para evitar de ter que reescrever a linha toda. Primeiro, liste (LIST) o programa e então, pressione **ESC**. Agora, digite **I** tantas vezes quantas necessárias para mover o cursor para cima, até a linha com a declaração incorreta. Pressione a tecla **J**, para mover o cursor até o começo da linha, e então, use a tecla de avanço para redigitar todos os caracteres precedentes à "U" em "PRUNT". Digite um "I" sobre o "U", e continue com a tecla de avanço até o final da linha. Então, pressione **RETURN** e liste (LIST) o programa para ver se a linha foi devidamente corrigida. O computador costuma ter piedade dos péssimos datilógrafos.

Quando você precisa corrigir um pedaço de linha que aparece em algum lugar da tela, o simples movimento do cursor, e as teclas de retrocesso e avanço, podem ser usadas para corrigir com maior rapidez. Uns poucos minutos de treino com estas teclas agora, lhe ajudarão muito para economizar tempo mais tarde.



A tecla de retrocesso só funciona na linha em que você está

digitando. Se você digita uma linha de programa e, então, executa um simples movimento do cursor antes de pressionar **RETURN**, a linha de programa que você acabou de digitar, não os caracteres sobre os quais a tecla de retrocesso está passando, será afetada pelos movimentos da tecla de retrocesso. A tecla de avanço, contudo, redigita os caracteres sobre os quais ela realmente passa.

UMA PALAVRA SOBRE O APRENDIZADO DA LINGUAGEM ELPPASOFT

Muitas vezes surgem dúvidas que você pode perguntar sobre a linguagem Elppasoft, que não são diretamente respondidas neste manual. Por exemplo, na declaração

```
PRINT "ALO"
```

você precisa dar um espaço após a palavra "PRINT"? Ao invés de lhe darmos uma resposta, recomendamos que você simplesmente ligue o seu Elppa e tente os dois modos. Normalmente, uma simples experiência responderá à sua pergunta e, desde que você tenha dedicado o seu tempo para experimentar sozinho, se lembrará que é muito melhor do que se tivesse simplesmente lido a solução.

COMO EVITAR ACIDENTES

Anteriormente, neste capítulo você aprendeu a apagar uma linha, digitando o seu número e pressionando o **RETURN**. Esta é a forma favorita de se introduzir erros no seu programa. Suponha que você quisesse eliminar a linha 1100 do seu programa, mas se enganou e digitou

```
110
```

RETURN. Congratulações, você simplesmente acabou de eliminar a linha 110. Isto acontece. Ou então, você está pensando em reparar a linha 450 e então, digita

```
450
```

e pensa à respeito e decide não mudar mais a linha. Não pressione **RETURN**. Acione a tecla de retrocesso, passando sobre o número da linha, ou use o comando especial "esqueça esta linha",

CTRL

X

Usar **CTRL X** , coloca uma barra (\) no final da linha que você está digitando, e será como se nunca tivesse digitado esta linha.

```
3LIST
1 PRINT"ALQ"
2 PRINT 2 > 4
3 PRINT 53/14
74 PRINT "ISTO NAO SERA USADO"
3LIST
1 PRINT"ALQ"
2 PRINT 2 > 4
3 PRINT 53/14
```

VERDADEIRO OU FALSO

O Elppa pode distinguir entre o que é verdadeiro e o que é falso. Desde que isto é mais do que a maioria de nós podemos fazer, algumas explicações são necessárias. O símbolo > significa "maior que". A afirmação $6 > 2$ (que é lida "seis é maior que dois") é certamente verdadeira. O Elppa usa o número 1 para indicar a verdade.

Se você digita

```
PRINT 6 > 2
```

o computador responderá com um 1. A afirmação $55 > 78$ é falsa. O Elppa usa o número 0 para indicar o que é falso. Se você digita

```
PRINT 55 > 78
```

o computador responderá com um zero.

O símbolo < significa "menor que", e você pode, da mesma forma, usá-lo para fazer declarações. Aqui está uma relação completa de símbolos usados para se fazer afirmações:

> maior que

< menor que

= igual a

>= maior ou igual a

<= menor ou igual a

<> diferente de

Para digitar os símbolos de "maior ou igual a" e "menor ou igual a", no teclado do seu Elppa, você deve primeiro digitar um < ou um > e, então, digitar um =. Para digitar o símbolo "diferente de", você deve digitar um < e, então, um >.

Pense a respeito e, então, experimente para ver quais destas afirmações são verdadeiras, e quais são falsas.

```
3 <> 3
7 > 3
10 > 10
6 <= 5
7358 = 8537
6 < 9
41 >= -2
-6 < -3
-4 >= -6
5 <> -9
```

As afirmações podem incluir variáveis e expressões, tanto quanto números. Por exemplo

```
PRINT (37 * 5) <> (37 + 5)
```

imprimirá o valor 1, uma vez que 185 não é igual a 42 (lembre-se que 1 significa que a afirmação é verdadeira).

Você tem visto que o Elppa pode dizer a verdade da falsidade em simples afirmações sobre números. Entretanto, uma afirmação como CARTEIRO > PADEIRO pode ser verdadeira ou falsa, dependendo do valor das duas variáveis, CARTEIRO e PADEIRO. Se

```
CARTEIRO = 7
e
PADEIRO = 8
então a afirmação
CARTEIRO > PADEIRO
é falsa. Mas se
CARTEIRO = -6
e PADEIRO = -13
então a afirmação
CARTEIRO > PADEIRO
é verdadeira.
```

As afirmações têm os valores numéricos de zero ou um. Elas podem ser usadas em expressões aritméticas ao invés dos uns e zeros. Por exemplo,

```
PRINT 5 + (7 > 3)
```

imprimirá o valor 6. A declaração

T = 6 <> 4

dá a T o valor 1, desde que 6 não é igual a quatro, e então, a afirmação 6 <> 4 tem o valor 1. A declaração

QUENTE = 81 = 47

à primeira vista parece muito confusa, mas é facilmente entendida. Desde que 81 não é igual a 47, a afirmação 81 = 47 é falsa e tem o valor zero. O valor de Ø é dado à variável "QUENTE".

Como nós já vimos, o Elppa usa 1 para indicar verdadeiro, e Ø para indicar falso. Se algo não é verdadeiro, é falso. Se algo não é falso, é verdadeiro. Isto pode nem sempre ser o caso na vida real, mas com computadores é sempre assim. Experimente isto no Elppa:

PRINT NOT 1

e, então, experimente

PRINT NOT Ø

O computador concorda: não verdadeiro é falso e não falso é verdadeiro. Certamente, você pode usar expressões ao invés de uns e zeros. Por exemplo

PRINT NOT (37 > 4)

A sentença

QUADRADOS TÊM QUATRO LADOS.

é verdadeira. E a sentença

ESTE MANUAL ESTÁ EM PORTUGUÊS.

é verdadeira. Considere a sentença

QUADRADOS TÊM QUATRO LADOS E ESTE MANUAL ESTÁ EM PORTUGUÊS.

É esta sentença verdadeira ou falsa? É verdadeira. Considere a sentença

QUADRADOS TÊM NOVE LADOS E ESTE MANUAL ESTÁ EM PORTUGUÊS.

Esta sentença, como um todo, é falsa. Por último, considere a sentença

QUADRADOS TÊM NOVE LADOS E ESTE MANUAL ESTÁ EM ALEMÃO.

Esta sentença também é falsa. Em geral, quando você combina duas sentenças, ou afirmações, unindo-as com a letra E, você obterá

a. A nova sentença é verdadeira se as duas originais o forem.

b. A nova sentença é falsa se pelo menos uma das originais o for.

O Elppa sabe como determinar se uma afirmação contendo a palavra de conexão AND (E), é verdadeira ou falsa. Experimente o seu computador com as seguintes instruções; tente adivinhar cada resposta:

```
PRINT 1 AND 1
PRINT 1 AND 0
PRINT 0 AND 1
PRINT 0 AND 0
PRINT (5 > 3) AND 0
PRINT (NOT 0) AND (7 = 9)
```

Esta sentença é verdadeira ou falsa?

UM QUADRADO TEM QUATRO LADOS OU ESTE MANUAL ESTÁ EM LATIM.

É verdadeira. Um quadrado tem quatro lados, mesmo que este manual não esteja em Latim, então, a sentença como um todo é verdadeira. Quod erat demonstrandum. Em geral, quando você combina duas sentenças com a palavra OU, você obterá

- A nova sentença é verdadeira se pelo menos uma das originais o for.
- A nova sentença é falsa se ambas as originais o forem.

O Elppa também pode determinar se uma afirmação contendo OR (OU), é verdadeira ou falsa. Experimente cada uma dessas no seu Elppa, após pensar qual deverá ser a resposta.

```
PRINT 1 OR 1
PRINT 1 OR 0
PRINT 0 OR 1
PRINT 0 OR 0
PRINT (3 <> 4) OR (3 = 4)
PRINT 1 OR (0 AND 1)
PRINT ((5 > 8) OR (82 < 245)) AND (NOT 0)
```

AND, OR e NOT serão bastante úteis na próxima seção.

Você já descobriu que na declaração

```
PRINT 1 OR 0
```

o computador considera 1 como verdadeira e 0 como falsa. Agora experimente isto:

```
PRINT 17 OR 0
```

e isto:

```
PRINT -532 AND 55201.86
```

Em afirmações, o Elppa considera não apenas o 1, mas qualquer número que não seja o zero, como verdade. Contudo, quando o computador nos dá o valor de uma afirmação, aquele valor sempre será \emptyset ou 1.

Enquanto você não tiver conhecimento das regras de prioridades dos comandos AND, OR e NOT, recomendamos que você use parênteses para deixar bem claras as suas declarações.

ORDEM OU PRIORIDADE PARA OPERAÇÕES

1. ()
2. NOT - (para valores negativos)
3. ^
4. * /
5. + -
6. > < = >= <= <>
7. AND
8. OR

A DECLARAÇÃO IF

Suponha que você queira imprimir números inteiros de 1 a 10, sendo um número em cada linha. Um método óbvio para fazê-lo, é

```
NEW
210 PRINT 1
220 PRINT 2
230 PRINT 3
```

e assim por diante. Mas isto iria requerer 10 declarações, e se você quisesse imprimir os inteiros de 1 a 200, neste método, seriam necessárias 200 declarações. Usando o que você aprendeu, você pode imprimir (PRINT) inteiros de 1 em diante, com apenas 4 declarações, usando um loop:

```
200 N = 1
210 PRINT N
220 N = N + 1
230 GOTO 210
```

Há uma forma para controlar a extensão de um loop. O que você quer é uma declaração que execute um GOTO se N é, por exemplo, menor do que 11, mas não execute o GOTO se N é maior do que 11. A resposta para os seus desejos é a declaração IF. Se uma condição é satisfeita, o computador irá ignorar a instrução GOTO e executará a instrução da linha seguinte. Se não houver a próxima linha, o programa parará.

Aqui está um programa que conta de 1 a 10 e, então, para:

```
200 N = 1
210 PRINT N
220 N = N + 1
230 IF N < 10 THEN GOTO 210
```

Geralmente, a declaração IF funciona assim:

IF expressão aritmética THEN alguma declaração

Primeiro, a expressão aritmética é calculada. Se ela resulta em zero (falsa) todo o resto daquela linha de programa é ignorado, e o computador vai para a próxima linha. Se a expressão aritmética não for zero (verdadeira), o restante daquela linha de programa é executado.

A declaração IF é muito poderosa, e ela aparecerá em quase todos os programas que você escrever. Para treiná-la, experimente este programa:

```
NEW
400 GR
410 LINHA = 1
420 COLOR = LINHA
430 HLN 0,39 AT LINHA
440 LINHA = LINHA + 1
450 IF LINHA < 16 THEN GOTO 420
```

GUARDANDO PROGRAMAS EM UM DISCO

(Pule esta seção, se você não está usando uma unidade de disco)

Neste ponto, você pode desejar guardar em um disco, alguns desses programas que vem usando. Simplesmente, digite no programa (depois de digitar NEW)

SAVE

seguido do nome que você quer usar quando se referir ao programa, e então, certamente, um **RETURN** . Por exemplo, se você quisesse guardar o programa anterior, dando a ele o nome LINHAS, você digitaria

SAVE LINHAS

e o programa seria guardado no disco sob o nome LINHAS. Uma vez que o programa está guardado, digite

CATALOG

seguido de **RETURN** , para ver o nome do seu programa listado com os outros, no disco. Você, então, será capaz de rodar o programa chamado LINHAS, todas as vezes que desejar, digitando

RUN LINHAS

Experimente guardar (SAVE) o programa de sua escolha. Se você acidentalmente digitar errado o comando (e provavelmente obterá do Elppa ERRO:SINTAXE) simplesmente redigite a linha corretamente.

Algumas vezes é desejável carregar um programa na memória do Elppa sem realmente rodar (RUN) o programa. Por exemplo, você pode querer modificar um programa antes de rodá-lo. O comando LOAD é útil neste caso. Para usá-lo, simplesmente digite

LOAD

seguido do nome do programa que você deseja carregar (LOAD). Por exemplo, se você quisesse carregar (LOAD) o programa chamado LINHAS, digitaria

LOAD LINHAS

Se você modificar o programa e então desejar rodar a nova versão que está na memória do computador, mas não a que está gravada no disco, lembre-se de digitar somente

RUN

Se você esquece e digita

RUN LINHAS

a versão antiga armazenada no disco será recarregada, apagando a nova versão na memória.

Você pode usar os comandos LOAD e SAVE para transferir programas de um disco para outro, carregando um programa de um disco e guardando o programa em outro disco. Pratique usando os comandos LOAD e SAVE.

GUARDANDO PROGRAMAS COM UM GRAVADOR CASSETTE

(Pule esta seção se você não está usando um gravador cassette)

Para guardar (SAVE) um programa que você deseja usar mais tarde, numa fita cassette, primeiro coloque uma fita virgem no seu gravador e rebobine-a até o começo, onde seu programa gravado será facilmente localizado. No gravador, segure a tecla PLAY enquanto pressiona a tecla RECORD. Ambas devem permanecer abaixadas. De volta ao Elppa, digite

SAVE

Quando você pressiona **RETURN**, o cursor desaparecerá. Depois de 10 ou 15 segundos, o computador emitirá um "bip" avisando-o de que a gravação começou. Outro "bip" soará quando a gravação for completada, e o cursor reaparecerá. Aperte a tecla STOP do gravador, rebobine a fita até o começo, e você está pronto para voltar a programar. O seu programa contido no computador, não foi afetado em nada pela gravação.

OUTROS PROGRAMAS COM GRÁFICOS

Anteriormente, você colocou quatro cores (pontos coloridos) nos cantos da tela. Agora digite este programa:

```
NEW
190 GR
200 COLOR = 9
210 PLOT 0,0
220 PLOT 0,39
230 PLOT 39,39
240 PLOT 39,0
```

Liste (LIST) o programa para conferir se você o digitou corretamente, e então rode-o (RUN). Rápido, não? Para mudar as cores, apenas mude a linha 200, e rode (RUN) novamente o programa. Experimente listar (LIST) o programa. Note que a listagem passa através da estreita janela de texto na base da tela. Isto continuará a acontecer, à menos que você digite

TEXT

para retirar o modo GRÁfico, antes que você experimente listar (LIST).

O programa seguinte torna a tela inteira de uma só cor.

NEW

```
200 GR
210 COLOR = 9
220 COLUNA = 0
230 VLIN 0,39 AT COLUNA
240 COLUNA = COLUNA + 1
250 IF COLUNA < 40 THEN GOTO 230
```

Aqui está uma rápida explicação sobre o que acontece quando você roda (RUN) este programa. A linha 200 aciona o modo GRÁfico. A cor é selecionada na linha 210. O programa deve começar na coluna 0 e segue dessa forma até a coluna 39. A linha 220 garante que o programa comece na coluna 0. Na linha 230, uma linha vertical é desenhada na coluna 0. Agora que a coluna 0 está preenchida com a cor desejada, a linha 240 incrementa a coluna em uma unidade. O valor da COLUNA é 1 agora. A linha 250 verifica se o novo valor da COLUNA é menor que 40. Se o for, o programa retorna para a linha 230 para desenhar uma nova linha vertical naquela coluna. Contudo, quando o valor da COLUNA atinge 40 (na tela do Elppa, a coluna na extrema direita é a 39) o programa não retorna para a linha 230 e pára a execução porque chegou ao fim.

Para eliminar a necessidade de ter que digitar RUN, toda vez que você deseja preencher a tela com cor, digite

```
260 GOTO 210
```

Observe o que acontece. Quando este programa irá parar? Liste (LIST) o programa e procure entender o que ele faz antes de prosseguir adiante neste manual.

Quando você terminar de usar o programa anterior, limpe o computador e experimente o seguinte programa. Ele usa uma instrução nova e muito importante: a declaração REM. "REM" vêm de "REMark" e significa "COMENTÁRIO". Esta declaração permite que você coloque comentários num programa. O computador ignora quaisquer declarações REM; elas são feitas exclusivamente para beneficiar os humanos. Veja como é fácil acompanhar este programa

onde os comentários (REM) são usados liberalmente.

```
200 REM ACIONE MODO GRAFICO
210 GR
220 REM ESCOLHA UMA COR
230 COLOR = 1
240 REM LEIA CONTROLE ZERO
250 X = PDL(0)
260 REM DIVIDA POR 7 PARA QUE O MAXIMO VALOR DE X SEJA 36
270 X = Y / 7
280 REM LEIA CONTROLE UM
290 Y = PDL(1)
300 REM LIMITE A VARIACAO PARA MANTER Y TAMBEM NA TELA
310 Y = Y / 7
320 REM MARQUE O PONTO
330 PLOT X,Y
340 GOTO 250
```

Após digitar RUN, opere os controles de jogos. A divisão por sete é necessária uma vez que a função PDL fornece valores entre 0 e 255, considerando que a tela pode apenas aceitar valores para coluna e linha de 0 a 39. A divisão por sete, fornece valores de $(0 / 7) = 0$ à $(255 / 7) = 36.4285715$. Então, o modo Gráfico arredonda automaticamente valores de coordenadas para baixo até o inteiro mais próximo, cujo valor é menor ou igual ao valor dado. Em outras palavras, as coordenadas X e Y são arredondadas para baixo, para inteiros de 0 a 36, de modo que as coordenadas X e Y possam ser traçadas (PLOT). Este método não utiliza todo o espaço da tela. Para utilizar toda a largura da tela, ao invés de

```
270 X = X / 7
```

você poderia usar as duas linhas

```
270 IF X > 239 THEN X = 239
```

```
275 X = X / 6
```

Para utilizar toda a altura da tela, você poderia fazer a mesma coisa usando a coordenada Y.

A declaração IF limita o valor de X em 239. No modo Gráfico de baixa resolução do Elppa, 239 / 6 é arredondado para baixo, de 39.833333 para 39. Este uso da declaração IF para limitar a extensão de uma variável é muito comum.

LOOPS COM AS DECLARAÇÕES FOR E NEXT

Loops, se executados por aviões ou programas de computador, têm um topo e uma base. No programa

```
NEW
100 NUMERO = 0
110 PRINT NUMERO
120 NUMERO = NUMERO + 1
130 IF NUMERO <= 12 THEN GOTO 110
```

a linha 110 é o TOPO do loop, e a linha 130 é a BASE. O programa imprime os inteiros de 0 a 12, inclusive. O número 12 é o LIMITE do loop. Uma outra maneira de escrever um loop é usar uma declaração que ainda não discutimos: a declaração FOR. Nós podemos usar esta declaração para reescrever o programa anterior.

```
200 FOR NUMERO = 0 TO 12
210 PRINT NUMERO
220 NEXT NUMERO
```

Use

```
RUN 200
```

para executar este programa. Se você só digita RUN, o programa será executado a partir da linha 100 (o menor número de linha existente).

A linha 200 contém a nova declaração FOR. Ela começa dando a NÚMERO o valor 0. Esta é exatamente a mesma tarefa que a linha 100 realizou. Então, a linha 210 é executada. A base do loop com a declaração FOR, está na linha 220. A variável NÚMERO é aumentada em uma unidade e então comparada com o limite superior especificado na declaração FOR: 12. Se o NÚMERO não está acima do limite, a execução continua através da declaração imediatamente seguinte ao FOR. Se a variável está acima do limite, o programa pula (sai do loop) para a declaração após ao NEXT. Neste caso, o programa pula e, não encontrando outras linhas, encerra-se.

A vantagem mais óbvia do método FOR/NEXT, para a construção de loops, é que ele economiza uma declaração. A vantagem mais importante é que você não precisa pensar muito para escrever um loop, se você usa esse método. Se você quisesse desenhar uma série de linhas horizontais na tela, usando cada uma das 16 cores disponíveis, poderia digitar

```
3000 GR
3010 FOR N = 0 TO 15
3020 COLOR = N
3030 HLIN 0,39 AT N
3040 NEXT N
```

Uma outra vantagem, é que é muito mais fácil ler uma única declaração FOR, do que examinar três declarações para descobrir o que um loop está fazendo. Para achar a base de um loop FOR/NEXT, tudo o que você tem que fazer, é procurar um NEXT, que tem a mesma variável do FOR.

É importante mencionar que, embora você deva saber como funciona uma declaração FOR, você não tem que usá-la. Ela não acrescenta quaisquer novas possibilidades, àquelas que você já tem. Apenas torna alguns programas mais fáceis de escrever (para algumas pessoas).

Nesta altura, se você tem acompanhado as instruções deste manual, deve remover a porção do programa entre as linhas 3000 e 3040, inclusive. Então digite

```
DEL 3000,3040
```

e então, liste (LIST) para verificar os resultados do programa.

Para imprimir (PRINT) apenas os mesmos números de 0 a 12, você poderia usar o programa.

```
100 COISA = 0
110 PRINT COISA
120 COISA = COISA + 2
130 IF COISA <= 12 THEN GOTO 110
```

O segredo está na linha 120, onde 2 é adicionado em COISA. Dizemos que o loop tem PASSO (STEP) 2. Para ter passo 2 em um loop FOR, você digitaria

```
200 FOR COISA = 0 TO 12 STEP 2
```

O resto do programa permaneceria igual ao programa anterior a partir das linhas 210 até 220, exceto que o nome NÚMERO teria que ser trocado, sempre que ele ocorresse, para o nome COISA. Experimente. O PASSO (STEP) pode ser qualquer número dentro da faixa numérica do Elppa. O PASSO (STEP) pode também ser negativo, por exemplo

```
200 FOR COISA = 39 TO 15 STEP -3
```

Digite esta linha e experimente digitando

Você deve brincar com a declaração FOR, por enquanto, se você deseja aprender como usá-la. A partir deste ponto, diversos programas-exemplo usarão a declaração FOR.

Juntamente com a conveniência da declaração FOR, aparecem também, algumas limitações. Por exemplo, loops FOR/NEXT podem ser internos, mas não podem se cruzar. Aqui estão alguns exemplos que demonstram a idéia.

```

NEW
300 GR
310 FOR C = 1 TO 15
320 COLOR = C
330 FOR LINHA = 0 TO 39
340 HLIN 0,39 AT LINHA
350 NEXT LINHA
360 COLOR = C - 1
370 FOR COLUNA = 0 TO 39
380 VLIN 0,39 AT COLUNA
390 NEXT COLUNA
400 NEXT C
    
```

Este programa é um exemplo de dois níveis internos. Pense sobre isto e rode (RUN) este programa, antes de partir para o próximo. Lembre-se que, quando escrever programas usando declarações FOR, CADA FOR DEVE TER UM NEXT CORRESPONDENTE.

UM PROGRAMA ERRADO

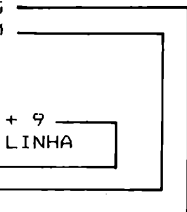
```

NEW
500 FOR N = 10 TO 20
510 PRINT N
520 FOR J = 30 TO 40
530 PRINT J
540 NEXT N
550 NEXT J
    
```

Este programa não funcionará. Seus loops estão cruzados, o que não somente fornece uma mensagem de erro, mas também, não faz qualquer sentido. Sempre que você descobrir que escreveu loops cruzados, refaça-os novamente. Se você tiver certeza do que está fazendo, e ainda quer cruzar loops, use loops feitos com as declarações IF. Você pode cruzá-las da maneira que desejar, para o seu próprio benefício.

UM ÚLTIMO EXEMPLO DE LOOPS INTERNOS

```
NEW
300 GR
310 C = 0
320 FOR COLUNA = 0 TO 35 STEP 5
330 FOR TRACO = 0 TO 30 STEP 10
340 C = C + 1
350 IF C > 15 THEN C = 0
360 COLOR = C
370 FOR LINHA = TRACO TO TRACO + 9
380 HLIN COLUNA, COLUNA + 4 AT LINHA
390 NEXT LINHA
400 NEXT TRACO
410 NEXT COLUNA
```



Este programa tem três níveis internos e desenha um xadrez. Note que COLOR não pode ser usada como uma variável de um FOR/NEXT. COLOR é uma palavra reservada no Elppasoft. Experimente rodar (RUN) o programa no modo de texto, retirando a linha 300. O que acontece?

GANHANDO BRILHO

Se você está cansado de imprimir caracteres brancos em um fundo preto, você apreciará muito esta seção. Digite

```
INVERSE
```

e dê uma olhada no indicador] e no cursor do Elppasoft. O indicador] deve estar preto sobre um fundo branco. Agora, digite um simples programa, tal qual

```
NEW
100 PRINT "PRETO E BRANCO EM CORES"
```

e rode-o (RUN). Não é excitante? Agora digite

```
FLASH
```

e rode (RUN) o programa novamente. Agora você verá a frase anterior PISCANDO.

Note que INVERSE e FLASH afetam somente a SAÍDA DO COMPUTADOR. Os caracteres que aparecem na tela permanecem os mesmos. Esses comandos podem ser usados tanto em execução imediata quanto em

execução diferida. Experimente-os. Depois de usar os comandos INVERSE e FLASH por enquanto, você pode achar que o branco no preto não é tão ruim assim. Se desejar voltar ao estado normal, digite

NORMAL

IMPRIMINDO COM DETALHES

Como experiência, digite este programa e veja o que acontece quando você o executa.

```
NEW
100 PRINT "ALO"
110 GOTO 100
```

Pare o programa com **CTRL C**. Então, mude a linha 100 apenas acrescentando uma vírgula

```
100 PRINT "ALO" ,
```

e rode (RUN) o programa novamente. Como você pode ver, a adição desta vírgula permite que a palavra seja impressa (PRINT) em colunas. Agora substitua a vírgula (,) pelo ponto-e-vírgula (;)

```
100 PRINT "ALO";
```

e rode (RUN) o programa novamente. Desta vez a saída é "EMPACOTADA". Isto significa que não há espaços entre o que você pediu ao computador para imprimir (PRINT). Ele imprime ALO após ALO, até que a tela esteja completamente preenchida.

Mude o programa adicionando esta declaração

```
99 V = 99
```

e altere a linha 100 para ler

```
100 PRINT V
```

rode (RUN) este programa. Agora mude a linha 100 para

```
100 PRINT "ALO",
```

e rode-o (RUN) novamente. Então, mude a linha 100 para

```
100 PRINT V;
```

e observe que a vírgula e o ponto-e-vírgula, também podem ser usados com valores numéricos. A habilidade de colocar números um após o outro sem espaços de separação, é algumas vezes muito útil. Vírgulas e ponto-e-vírgulas, podem ser usados entre uma declaração PRINT. Limpe o velho programa com o NEW, e digite

```
100 TIROS = 2
110 BOLAS = 3
120 PRINT TIROS, BOLAS
```

Você pode tornar as saídas mais claras, incluindo mensagens na declaração PRINT. Por exemplo, mude a linha 120, para

```
120 PRINT "OS TIROS E BOLAS SAO ";TIROS,BOLAS
```

Note que você provavelmente quer ter um espaço após a palavra SÃO, para que o número de tiros não seja impresso muito próximo a ela. Se você não acha que o grande espaço entre o número de tiros e bolas lhe parece agradável, você pode usar a declaração

```
120 PRINT "OS TIROS E BOLAS SAO  ";TIROS;" ";BOLAS
```

Nesta versão, um espaço em branco é colocado entre o número de tiros e bolas. Talvez, a melhor maneira de fazer isto (você está tentando tudo no seu Elppa?) seja

```
120 PRINT "TIROS ";TIROS;"    BOLAS ";BOLAS
```

Isto lhe fornece algo similar a um placar.

Vamos dizer que você quisesse imprimir (PRINT) a palavra AQUI, começando na décima coluna (a tela tem 40 colunas) você poderia usar esta declaração

```
PRINT "          AQUI"
```

(Você tem que confiar na nossa palavra de que há nove espaços em branco antes da palavra AQUI). Ou, você poderia usar a característica TAB. Como nas máquinas de escrever, você pode tabular no computador. A declaração

```
120 PRINT TAB(10)"AQUI"
```

produz o mesmo efeito que colocar nove espaços entre as aspas, como fizemos anteriormente. Experimente, você gostará disto.

Combinando o TAB com um loop FOR, você pode programar alguns agradáveis efeitos visuais. Por exemplo:

```
NEW
200 FOR N = 1 TO 24
210 PRINT TAB(N)"X"
220 NEXT N
```

Há 24 (não 40) linhas horizontais impressas, devido ao fato que o limite superior no loop do programa acima é 24. O TAB não pode ser usado para retroceder (para a esquerda) sobre uma linha. Somente avanços são executados. Para imprimir sobre uma determinada linha, você pode tabular a linha na vertical (VTAB). A primeira linha é a linha 1, e a última é a linha 24. VTAB, ao contrário de TAB, não é usado dentro de uma declaração PRINT.

Você pode tabular horizontalmente com HTAB, se você não quiser usar a declaração PRINT. HTAB funciona como TAB, exceto que ele também não é usado dentro de uma declaração PRINT. HTAB também pode fazer a impressão começar à esquerda ou à direita da posição que estava sendo feita. O caracter mais à esquerda em uma linha, está na posição 1, enquanto o caracter mais à direita, está na posição 40. A seguir, está um pequeno programa que demonstra o uso do HTAB e do VTAB:

```
NEW
590 HOME
600 FOR X = 1 TO 24
610 FOR Y = 1 TO X
620 HTAB X
630 VTAB Y
640 PRINT "ELFFA"
650 NEXT Y
660 NEXT X
670 GOTO 600
```

Antes de rodar (RUN) este programa, tente (não é fácil!) descobrir o que ele fará. Será uma surpresa agradável.

O TAB funciona pelo modo de execução imediata, mas você somente pode usar VTAB ou HTAB em programas. Enquanto TAB, HTAB e VTAB atuam como as coordenadas no PLOT, há porém, algumas diferenças. As 40 colunas da instrução TAB, são numeradas de 1 a 40, como seriam em uma máquina de escrever, enquanto que a primeira coordenada de uma instrução PLOT pode variar de 0 a 39, o que é mais conveniente para programar gráficos. Desde que os caracteres são maiores do que os "quadrados", com os quais construímos os gráficos, há espaço para somente 24 colunas de impressão na tela. Portanto, os limites do VTAB, são 1 e 24. Um zero ou um número que é muito grande, ou muito pequeno para TAB, VTAB ou HTAB, lhe fornecerão a seguinte mensagem

ERRO:QUANTID ILEGAL

O maior valor para VTAB é 24, mas o maior valor para TAB ou HTAB é 255. Ambos, TAB e HTAB ultrapassam a dimensão da linha e "pulam" para a seguinte. Para ver isto em ação, digite

```
NEW
300 FOR K = 1 TO 255
310 PRINT TAB(K) K
320 NEXT K
```

Então, tente recolocar as linhas 310 e 320 com

```
310 HTAB K
320 PRINT K
```

e adicionar

```
330 NEXT K
```

O que acontece ao programa, quando você coloca VTAB no lugar de HTAB?

CAPÍTULO 5

GRÁFICOS

- 90 Conversando com um programa
- 94 Fora das paredes
- 95 Criando sons
- 97 Sons para a bola saltitante
- 97 Para notas mais altas múltiplas declarações em uma mesma linha
- 98 Números randômicos
- 100 Simulando um par de dados
- 101 Subrotinas
- 130 Verificações
- 105 Melhorando a subrotina que desenha um cavalo
- 108 Gráficos de alta resolução

CONVERSANDO COM UM PROGRAMA

Aqui está um programa que faz um ponto colorido se mover através da tela, pulando para os lados direito e esquerdo.

```
NEW
400 REM ESCOLHA A COR DA BOLA
420 BOLA = 9
440 REM ACIONE MODO GRAFICO
460 GR
480 REM POSICAO INICIAL
500 XANT = 20
520 REM MOVA A BOLA PARA FRENTE E PARA TRAS
540 XMOV = 1
560 REM NOVA POSICAO DE X
580 XNOVO = XANT + XMOV
600 REM A BOLA ESTA NA TELA?
620 IF (XNOVO >= 0) AND (XNOVO < 40) THEN GOTO 720
640 REM MUDE A DIRECAO DE XMOV
660 XMOV = -1 * XMOV
680 GOTO 580
700 REM MARQUE A NOVA BOLA
720 COLOR = BOLA
740 PLOT XNOVO,20
760 REM APAGUE A BOLA ANTIGA
780 COLOR = 0
800 PLOT XANT,20
820 REM GUARDE A NOVA BOLA
840 XANT = XNOVO
860 REM MOVA NOVAMENTE
880 GOTO 580
```

Você deve sempre nomear as variáveis com certo cuidado. Pode parecer que a variável XNEW fosse um nome mais conveniente que XNOVO, até que você se lembra que NEW tem um significado especial no Elppasoft e é uma palavra reservada. A razão pela qual a variável XMOV foi chamada de "XMOV" será evidente se você mudar o seu valor. Experimente XMOV = 2, por exemplo. Se você der um valor muito alto à variável XMOV, a bola começará a pular violentamente através da tela, sem deixar nenhum traçado entre as posições.

Este tipo de programa é a base para muitos jogos típicos de TV. Vale a pena gastar um pouco de tempo brincando com o programa, mudando isto e aquilo, simplesmente para ver o que pode ser feito com ele. Seria uma ótima idéia guardar (SAVE) este programa, de forma que você não terá que redigitá-lo novamente, caso você faça uma mudança fatal.

Quando você lista (LIST) este programa, ele não cabe inteiro de uma só vez na tela, e as linhas do programa passam tão rapidamente que se torna difícil ler com facilidade. Uma forma de ver todas as linhas do programa é listá-lo (LIST) por partes. Por exemplo, você poderia digitar

```
LIST 400,680
```

e somente as linhas com números maiores que 400 e menores que 680 seriam impressas. Em seguida, você poderia listar (LIST) o resto do programa. Você pode também usar

```
CTRL
```

```
S
```

para interromper a listagem do programa. Experimente listá-lo (LIST) e, então, rapidamente digitar

```
CTRL
```

```
S
```

antes que as linhas do programa ultrapassem o topo da tela. Para continuar a listagem, digite **CTRL S** novamente. **CTRL C**, também pode ser usado para interromper a listagem. No entanto, **CTRL C** não permite que a listagem continue a partir do ponto em que havia parado.

Agora, você está em uma posição de compreender o programa da bola saltitante, mas você deve ter colegas que ainda não atingiram tal posição. Suponha que você quisesse um amigo que fosse capaz de escolher a cor da bola. Você poderia explicá-lhe em como mudar a linha 420, mas também, teria que explicar as possíveis mensagens de erros, e o que fazer se ... bem, levaria um pouco de tempo para ensinar-lhe. Seria melhor deixar o seu amigo trabalhar com o programa. Para fazer isto, você pode usar uma declaração INPUT. Mude a linha 420 para

```
420 INPUT BOLA
```

Quando o programa executa esta declaração, uma interrogação (?) aparecerá na tela, seguida do cursor. Então, o Elppa irá esperar até que alguém digite um número e pressione **RETURN**. O número digitado será o valor da BOLA e o programa recomeçará a execução. Pode ser uma boa idéia ter o computador dizendo ao seu amigo o que é esperado. Você poderia colocar a declaração PRINT tal como

```

280 REM ACIONE MODDO DE TEXTO
300 TEXT
320 PRINT "PARA SELECIONAR UMA COR PARA A BOLA SALTITANTE,"
340 PRINT "DIGITE UM NUMERO DE 1 A 15"
360 PRINT "APOS O PONTO DE INTERROGACAO."
380 PRINT "ENTAO PRESSIONE A TECLA ROTULADA 'RETURN'."

```

Você também pode incorporar uma mensagem à declaração INPUT:

```

420 INPUT "QUE COR VOCE GOSTARIA QUE A BOLA FOSSE (1-15)? ";BOLA

```

Note que em uma declaração PRINT a mensagem deve estar entre aspas e deve haver um ponto-e-vírgula (;) entre a mensagem e o nome da variável. Quando a declaração PRINT contém uma mensagem, não aparece uma interrogação junto à mesma. Se você quiser que apareça uma interrogação, você deve incluí-la na mensagem do INPUT.

Seus amigos podem usar as teclas de retrocesso e avanço para corrigir os erros que cometerem ao digitar, mas se ao cometerem um erro e então pressionarem **RETURN**, receberão uma mensagem de erro. Se o caracter inserido não é um número,

```

?REENTRE
?

```

aparecerá na tela. Se for inserido um número muito grande ou muito pequeno, ou o programa deixará a bola se mover para o lado direito da tela e depois parará, ou a mensagem

```

ERRO:QUANTID ILEGAL EM 720

```

aparecerá na tela, e o programa parará. A maioria dos usuários não saberá como reiniciar a utilização prática de seu programa. Por esta razão, você deve fazer o programa verificar se todos os números digitados pelo usuário estão corretos. Estas linhas farão a a verificação:

```

424 REM A BOLA ESTA ENTRE 1 E 15?
428 IF (BOLA > 15) AND (BOLA < 1) THEN GOTO 460
432 PRINT "ESTE VALOR NAO ESTAVA ENTRE 1 E 15."
436 GOTO 420

```

Você está começando a perceber por que nós o avisamos para deixar bastante espaço entre números de linhas?

É uma boa prática de programação fazer um programa tão à prova de erros quanto possível. Você avançou até o ponto em que está escrevendo mensagens de erro para outros lerem. É possível que esteja tudo sob controle para um programador como você ao ler uma

gíria como "ERRO:SINTAXE" , mas é quase certeza que para os usuários, na maioria leigos, não terá o menor sentido.

Cada vez que você usa uma declaração INPUT, seu programa deve verificar se o que o usuário digita está dentro de certos limites, para que o programa não "estoure" ou falhe de qualquer maneira. Lidando com usuários leigos (e você deve assumir que os usuários não são programadores) esta é uma arte que cabe a você. Use sentenças claras em português e, a verificação cuidadosa do que o usuário digita é sempre necessário.

A propósito, você pode introduzir diversos números com uma declaração INPUT. A declaração

```
3000 INPUT X,Y,Z
```

mostraria na tela um ponto de interrogação, como de costume, e então, aguardaria até que três números fossem digitados. O primeiro número seria armazenado na variável X, o segundo na variável Y, e o terceiro na variável Z. Os três números devem ser separados digitando a tecla **RETURN** ou, por vírgulas, e o último número deve ser seguido por um **RETURN** .

Guarde (SAVE) a sua melhor versão do programa "bola saltitante", só por precaução. Então, se ainda não o fez, experimente acrescentar um movimento vertical à bola. Use as novas variáveis Y1, Y2 e YMOV. Uma solução é dada a seguir, mas tente fazer isto sozinho antes de olhar.

Quando este programa estiver funcionando do jeito que você quer, guarde-o (SAVE) no seu disco ou fita cassette. Nós o usaremos mais tarde.

FORA DAS PAREDES

Aqui está uma maneira de fazer a bola saltar fora das quatro paredes. As declarações com um asterisco à sua frente, são aquelas que foram acrescentadas ou modificadas do programa no qual a bola saltava entre duas paredes.

```
*280 REM ACIONE MODO DE TEXTO
*300 TEXT
*320 PRINT "PARA SELECIONAR UMA COR PARA A BOLA SALTITANTE,"
*340 PRINT "DIGITE UM NUMERO DE 1 A 15"
*360 PRINT "APOS O PONTO DE INTERROGACAO,"
*380 PRINT "ENTAO PRESSIONE A TECLA 'RETURN'."
  400 REM ESCOLHA A COR DA BOLA
  420 INPUT "QUE COR VOCE GOSTARIA? ";BOLA
*424 REM A COR DA BOLA ESTA ENTRE 1 E 15?
*428 IF (BOLA > 0) AND (BOLA < 16) THEN GOTO 460
*432 PRINT "NAO ESTAVA ENTRE 1 E 15."
*436 GOTO 420
  440 REM ACIONE MODO GRAFICO
  460 GR
  480 REM POSICAO INICIAL
  500 XANT = 20
*510 YANT = 38
  520 REM MOVA A BOLA PARA FRENTE E PARA TRAS
  540 XMOV = 1
*545 REM MOVA A BOLA PARA CIMA E PARA BAIXO
*550 YMOV = 1
  560 REM NOVA POSICAO DE X
  580 XNOVO = XANT + XMOV
  600 REM A BOLA ESTA NA TELA?
  620 IF (XNOVO >= 0) AND (XNOVO < 40) THEN GOTO 686
  640 REM MOVA A BOLA PARA A ESQUERDA
  660 XMOV = -1 * XMOV
  680 GOTO 580
*684 REM NOVA POSICAO DE Y
*686 YNOVO = YANT + YMOV
*688 REM A BOLA ESTA NA TELA?
*690 IF (YNOVO >= 0) AND (YNOVO < 40) THEN GOTO 720
*692 REM MOVA A BOLA PARA CIMA
*694 YMOV = -1 * YMOV
*697 GOTO 686
  700 REM NOVA POSICAO DA BOLA
  720 COLOR = BOLA
  740 PLOT XNOVO,YNOVO
  760 REM APAGUE A ANTIGA POSICAO DA BOLA
  780 COLOR = 0
  800 PLOT XANT,YANT
  820 REM GUARDE A POSICAO DA BOLA
  840 XANT = XNOVO
*850 YANT = YNOVO
  860 REM MOVA NOVAMENTE
*890 GOTO 580
```

Você verá quando rodar (RUN) este programa, que o resultado é um pouco repetitivo. Você pode alterar o padrão de saltos, mudando os valores iniciais de XANT e YANT (linhas 500 e 510), mas aqui está uma mudança que você provavelmente gostará:

```
580 XNOVO = XANT + XMOV * PDL(0) / 70
686 YNOVO = YANT + YMOV * PDL(1) / 70
```

Para ver o que isto faz, jogue com os controles.

Mais uma sugestão. Por que não ter um outro INPUT, fornecendo um valor para uma variável chamada FUNDO? Preencha a tela uma vez com a cor da variável FUNDO no começo do seu programa (após GR). Então, para apagar a antiga posição da bola, use

```
780 COLOR = FUNDO
```

ou mesmo

```
780 COLOR = FUNDO + 3
```

Guarde (SAVE) sua versão favorita deste programa.

CRIANDO SONS

Clicks, ticks, tocks, e vários outros zumbidos são facilmente gerados. Você pode criar sons no seu Elppa dando uns tapinhas nele, arranhando-o, ou mesmo deixando-o cair, mas os sons descritos neste manual, são produzidos programando-o. Por isso, vá para um lugar quieto, e tente trabalhar em cima desta seção.

Para construir qualquer programa que produza som no Elppa, você precisará desta fórmula mágica:

```
150 SOM = PEEK(-16336)
```

Não há uma explicação fácil para esta fórmula. O número, -16336, está relacionado ao "endereço de memória" do alto falante do Elppa, e foi construído dentro da eletrônica do computador. Você vai ter que olhar para este número somente quando precisar dele.

O PEEK retorna o código numérico armazenado numa certa posição no computador. Na maioria das posições o PEEK somente retorna um valor numérico, mas em algumas posições, tal como -16336, ele pode ter uma reação diferente. Neste caso, ele faz o alto-falante emitir um "click". Toda vez que o programa executa esta declaração, o Elppa produzirá um minúsculo "click". Rode (RUN) o programa, e ouça o seu computador bem de perto.

Agora acrescente esta linha:

```
140 GOTO 150
```

e rode (RUN) o programa.

Para fazer o seu programa emitir "bips", por um limitado período de tempo, acrescente declarações tais como

```
140 FOR BIP = 1 TO 100  
160 NEXT BIP
```

Experimente.

Um tom é gerado por uma rápida sequência de clicks. Qualquer programa que usa PEEK (-16336) repetidamente, gerará algum tipo de ruído. Uma vez que -16336 é incômodo para digitar, nós introduziremos uma outra declaração que nos permitirá substituir um símbolo que é mais fácil de digitar. Entre com a declaração

```
100 S = -16336
```

Para produzir um click agradável e ressonante, mude a linha 150 para

```
150 SOM = PEEK(S) - PEEK(S) + PEEK(S) - PEEK(S) + PEEK(S) -  
PEEK(S)
```

Diferentes números de PEEKs na declaração, produzirá diferentes qualidades de clicks. Experimente rodar (RUN) algumas variações. Para produzir mais zumbidos, coloque uma das suas variações em um "loop". Em geral, quanto mais rápido é o loop, mais alto é o tom.

Agora, para usar estes sons, carregue (LOAD) o programa da bola saltitante, chamado FORA DAS PAREDES, de volta no computador. Tente acrescentar um som, cada vez que a bola bater na parede.

Uma solução possível, é dada a seguir, mas primeiro tente trabalhar sozinho. (Sugestão: uma batida sempre ocorre toda vez que XMOV ou YMOV muda de valor.)

SONS PARA A BOLA SALTITANTE

Aqui está uma maneira de tornar a bola audível. Acrescente estas linhas ao programa FORA DAS PAREDES:

```
240 REM DE A S O ENDEREÇO DO ALTO-FALANTE
260 S = -16336
663 REM SOM DE BATIDA
665 FOR B = 1 TO 5
670 SOM = PEEK(S) - PEEK(S) + PEEK(S) - PEEK(S)
675 NEXT B
695 REM SOM DE BATIDA
696 FOR B = 1 TO 5
697 SOM = PEEK(S) - PEEK(S) + PEEK(S) - PEEK(S)
698 NEXT B
```

Agora tente os seus próprios sons. Por que não fazer um som diferente para cada parede?

PARA NOTAS MAIS ALTAS MÚLTIPLAS DECLARAÇÕES EM UMA MESMA LINHA

Para conseguir ainda, tons mais altos, uma outra característica do Elppasoft pode ser introduzida. É possível colocar mais de uma declaração em uma mesma linha. Experimente este programa de uma linha:

```
NEW
50 S = PEEK(-16336) : GOTO 50
```

Os dois pontos (:) podem ser usados para separar declarações em qualquer programa onde você deseja ter mais que uma declaração em uma linha. Contudo, somente a primeira declaração na linha tem um número de declaração, de modo que você pode somente ramificar a primeira declaração com um GOTO.

Agora adicione

```
40 FOR PAUSA = 1 TO 2500 : NEXT PAUSA
```

As vantagens da multiplicidade de declarações com um número de linha comum são estas:

1. As declarações são executadas mais rapidamente. (Esta é uma vantagem somente se você precisar de maior velocidade.)
2. Você pode colocar na tela a maior parte do seu programa.
3. Você economiza espaço e tempo na digitação.
4. Você pode agrupar declarações que juntas, coletivamente, executam uma função, tal como a pausa na linha 40 do programa anterior.
5. Isto requer menos espaço na memória. (O que é uma vantagem somente se você está rodando um programa muito extenso, e o computador lhe dá uma mensagem do tipo ERRO:FALTA MEMÓRIA, à medida que você entra com o mesmo.)

Há também algumas desvantagens:

1. A leitura do programa se torna mais difícil.
2. A correção ou modificação do programa também é dificultada.
3. Você pode ramificar somente a primeira declaração na linha.
4. É muito desencorajador digitar uma longa e múltipla declaração, somente para obter uma mensagem de erro do tipo ERRO:SINTAXE quando o programa é rodado, tornando-se necessário redigitar a declaração.

NÚMEROS RANDÔMICOS

Experimente este pequeno programa no seu Elppa:

```
NEW
100 PRINT RND(1)
110 GOTO 100
RUN
```

A abreviação RND na linha 100 é proveniente de RaNDômico (aleatório). A função RND retorna números RaNDômicos (aleatórios). Pare o programa com

CTRL

C

Os números gerados por este programa foram frações decimais RaNDômicas entre zero e um.

Mude a linha 100 para

```
100 PRINT RND(6)
```

e rode (RUN) o programa.

Pare novamente o programa com **CTRL C** . Hmm, muito interessante! Os números randômicos ainda estão entre zero e um. Escreva o último número que foi gerado de forma que você possa lembrar-se dele.

Agora mude a linha 100 novamente, desta vez para

```
100 PRINT RND(0)
```

e rode-o (RUN). Pare o programa e compare o que você escreveu com o que está agora na tela. RND (0) retorna o último número Randômico (aleatório) que foi gerado.

As frações decimais randômicas entre um e zero podem ser um pouco desajeitadas. Frequentemente, os inteiros (números como 3, 6 e 10) são mais fáceis de usar. Para obter inteiros randômicos de 0 a 9, temos que acrescentar mais algumas linhas ao programa. Digite

```
NEW
 90 REM ATRIBUIR NUMERO RND PARA X
100 X = RND(1)
110 REM MULTIPLICAR X POR 10
100 X = X * 10
130 REM TIRAR PARTE FRACIONARIA
140 X = INT(X)
150 PRINT X
160 GOTO 100
```

A linha 140 introduz a função INT. A declaração INT (X) fornece o maior inteiro que é menor ou igual ao valor de X. Por exemplo, se o valor de X é 3,6754 , então INT (X) é igual a 3. Os parênteses que se seguem a INT podem conter qualquer expressão aritmética ou variável numérica.

Agora rode (RUN) o programa. Ele funciona do jeito que você esperava? Para mudar o programa de forma que gere números de um a dez, ao invés de zero a nove, apenas acrescente uma unidade ao valor de X, digitando, no seu programa, a seguinte linha:

```
145 X = X + 1
```

Experimente.

O programa pode parecer, à primeira vista, um pouco complicado. Para ver, passo a passo, o que acontece, você pode acrescentar diversas declarações PRINT.

Modifique seu programa para que se pareça com este a seguir..

```

90 REM ATRIBUIR NUMERO RND A X
100 X = RND(1) : PRINT "X = RND(1)",X
105 PRINT
110 REM MULTIPLIQUE X POR 10
120 X = X * 10 : PRINT "X=X*10",X
125 PRINT
130 REM TIRAR PARTE FRACIONARIA
140 X = INT(X) : PRINT "X=INT(X)",X
145 PRINT
150 REM SOMAR 1 AO VALOR DE X
160 X = X + 1 : PRINT "X=X+1"
170 PRINT X
175 PRINT
180 FOR PAUSA = 1 TO 2000 : NEXT PAUSA
190 GOTO 100

```

Rode (RUN) este programa e veja o que acontece.

Se você usar a sua imaginação, pode condensar este programa em apenas uma linha:

```
100 PRINT INT (10 * RND(1)) + 1 : GOTO 100
```

Você sabe como funciona a linha 100?

SIMULANDO UM PAR DE DADOS

Você pode usar o que aprendeu sobre números randômicos para escrever um programa que simula um par de dados.

```

NEW
100 PRINT "DADO BRANCO",
110 PRINT INT (6 * RND(1)) + 1
120 PRINT "DADO VERMELHO",
130 PRINT INT (6 * RND(1)) + 1

```

Este programa gera inteiros randômicos de um a seis para cada dado. Para "rolar" os dados novamente, reexecute o programa. Você pode escrever um programa que usa estes "dados" para fazer um jogo? Experimente.

Tente escrever um programa de apenas uma linha, que gere números randômicos de 1 a 50. De 0 a 25. Escolha seus próprios números. Lembre-se de somar (1) ao número randômico, se você não quiser gerar zeros.

Aqui está uma forma colorida de usar inteiros RaNDômicos.


```

NEW
200 GR
210 REM ESCOLHA UMA COR RANDOMICA
220 COLOR = INT (16 * RND(1))
230 REM ESCOLHA UM PONTO RANDOMICO
240 X = INT (40 * RND(1))
250 Y = INT (40 * RND(1))
260 REM POSICAO DO PONTO RANDOMICO
270 PLOT X,Y
280 REM FAÇA NOVAMENTE
290 GOTO 220

```

Experimente usar RND em outros programas. Você pode escrever um programa que desenhe na tela linhas em cores RaNDômicas?

SUBROTINAS

Imagine que exista um jogo para o qual, em um determinado momento, você precise de algo como um cavalo azul com patas cor de laranja e de cabeça branca. Aqui está um programa que faz isto:

```

NEW
1000 REM PROGRAMA PARA DESENHAR UM CAVALO AZUL COM CABECA BRANCA
E PATAS LARANJA
1010 GR
1020 COLOR = 7 : REM AZUL CLARO
1030 PLOT 15,15
1040 HLIN 15,17 AT 16
1050 COLOR = 9 : REM LARANJA
1060 PLOT 15,17
1070 PLOT 17,17
1080 COLOR = 15 : REM BRANCO
1090 PLOT 14,15

```

Não há nada errado com este programa; ele realmente desenha um cavalo azul com patas laranja e cabeça branca. Agora, suponha que você precisa desenhar um outro cavalo em algum outro lugar na tela. Você poderia reescrever este programa com novos valores de X e Y, mas seria uma perda de tempo. Deveria haver alguma forma de usar o mesmo programa para colocar uma figura em qualquer lugar da tela sem ter que reescrevê-lo novamente.

A chave para fazer isto é começar observando que você pode mover um ponto que está nas coordenadas (A, B) para a direita, acrescentado um valor à primeira coordenada, A neste caso. Por exemplo, o ponto (4,17) desloca-se 10 colunas para a direita se você somar 10 à primeira coordenada, tornando-o o ponto (14,17).

Analogamente, um ponto se move para a esquerda, se você faz uma subtração da primeira coordenada (ou adiciona um valor negativo). Uma simples experiência lhe mostrará que adicionando, ou subtraindo da segunda coordenada, o ponto se move para cima e para baixo respectivamente.

Com estes fatos em mente, você pode reescrever o seu programa para centrar o cavalo em quase todos os pontos (X, Y) na tela. Por que "quase" todos os pontos? Porque, se você escolher um ponto central em um canto da tela, o cavalo sairá da mesma, e isto pode lhe dar uma mensagem de erro do tipo ERRO:QUANTID ILEGAL EM 1030 (ou algum outro número de linha). Aqui está um programa melhorado.

NEW

```
1000 REM COLOQUE UM CAVALO EM QUALQUER LUGAR NA TELA
1010 COLOR = 7 : REM AZUL CLARO
1020 PLOT X,Y - 1
1030 HLIN X,X + 2 AT Y
1040 COLOR = 9 : REM LARANJA
1050 PLOT X,Y + 1
1060 PLOT X + 2, Y + 1
1070 COLOR = 15 : REM BRANCO
1080 PLOT X - 1, Y - 1
```

Note que GR foi suprimido. Nós queremos usar esta parte do programa para colocar diversos cavalos na tela. Um GR aqui, limparia a tela antes que cada novo cavalo fosse desenhado.

Este programa não pode ser rodado, apenas como está. Primeiro você deve acionar o modo GRáfico, e escolher X e Y. Uma boa primeira tentativa de se usar o programa cavalo pode ser:

```
20 GR
30 REM CENTRAR O PRIMEIRO CAVALO
40 X = 12
50 Y = 35
```

Se você tentar rodar (RUN) isto, você vai conseguir um cavalo no lugar desejado, mas o programa termina lá. Nós queremos colocar dois cavalos na tela. Suponhamos que você pudesse escrever

```
60
70 REM CENTRAR O SEGUNDO CAVALO
80 X = 33
90 Y = 2
100
```

Faça a parte do programa da linha 1000 e então, termine.

Não seria isso melhor e mais fácil? Você sabe que o computador não pode ler aquelas estranhas instruções, nas linhas 60 e 100. Pode, contudo, ler

```
GOSUB 1000
```

em Elppasoft. Um programa como este, começando da linha 1000 é

chamado de uma SUBROTINA. GOSUB 1000, diz ao computador para ir (GO) para a SUBrotina, começando na linha 1000 e inicia executando naquela declaração. Ela também diz ao computador para voltar à linha imediatamente após a declaração GOSUB, quando ele terminou de executar a subrotina. O computador sabe que a subrotina termina, quando ela encontra uma declaração RETURN. Para tornar a parte do seu programa que desenha um cavalo na tela, uma completa subrotina, acrescente a linha

```
1090 RETURN
```

Agora você pode reescrever aquele programa com aquelas "estranhas instruções":

```
20 GR
30 REM CENTRAR O PRIMEIRO CAVALO
40 X = 12
50 Y = 35
60 GOSUB 1000
70 REM CENTRAR O SEGUNDO CAVALO
80 X = 33
90 Y = 2
100 GOSUB 1000
```

Agora rode (RUN) este programa. Você obtém uma mensagem de erro:

```
ERRO:RETURN SEM GOSUB EM 1090
```

mas, por outro lado, o programa parece rodar bem. Com efeito, você adicionou uma nova declaração no Elppasoft: uma declaração desenhar-cavalo. Agora, você pode usar a declaração

```
GOSUB 1000
```

para desenhar um destes cavalos especiais, em qualquer posição X, Y que você tenha escolhido.

VERIFICAÇÕES

A parte do programa da linha 1000 até a linha 1090, é chamada uma SUBROTINA ou SUBPROGRAMA. A parte do programa da linha 20 até a linha 100 é chamada de PROGRAMA PRINCIPAL.

Para ver o fluxo do programa, ou o caminho da execução, você pode invocar uma característica especial chamada TRACE (VERIFICAÇÃO). Esta característica especial pode lhe mostrar porque o Elppa forneceu uma mensagem de erro, quando o programa desenhar-cavalo foi executado. Acrescente esta linha ao programa principal.

```
10 TRACE
```

e, por um momento, apague a linha 20. Coloque o Elppa em modo de texto (TEXT) e rode (RUN) o programa.

Os números que você vê na tela, são os números de linha de cada declaração que é executada. Você pode ver como o programa começa na linha 10, continua através do programa principal até a chamada da subrotina, então executa a subrotina, volta ao programa principal, executa a subrotina novamente, e não encontrando nenhum número de linha menor, vai para a linha 1000 e executa a subrotina novamente. Aqui é onde ocorre o problema. Agora você entende o porquê da mensagem de erro?

Para remediar o problema, adicione esta nova linha ao programa:

```
110 END
```

Quando o programa atinge a linha 110, ele fará justamente o que ela diz: fim. Rode (RUN) o programa mais uma vez. Não há mais mensagem de erro. Como você já viu, TRACE é muito útil quando você está tendo problemas com um programa. Se você quiser verificar (TRACE) somente uma parte de um programa, você pode usar a declaração NOTRACE. Acrescente esta linha:

```
65 NOTRACE
```

e o programa será verificado (TRACE) somente a partir da execução da linha 65.

TRACE também pode ser usado no modo de execução imediata. Simplesmente digite

```
TRACE  
RUN
```

e seu programa será verificado (TRACE).



Uma vez que você usou o comando TRACE, se no modo de execução imediata ou como uma declaração em um programa, ele será verificado (TRACE), toda vez que for rodado. Para parar o TRACE, você deve usar um comando NOTRACE, ou em uma linha do seu programa, ou no modo de execução imediata.

MELHORANDO A SUBROTINA QUE DESENHA UM CAVALO

As subrotinas devem ser escritas de modo que os problemas de possíveis erros não aumentem quando o programa é rodado (RUN). Um problema com a nossa subrotina desenhar-cavalo, é que alguns valores de X e Y farão com que o cavalo fique fora da tela. Isto pode ser prevenido por um conjunto de declarações, tais como:

```
1012 IF X < 1 THEN X = 1
1014 IF X > 37 THEN X = 37
1016 IF Y < 1 THEN Y = 1
1018 IF Y > 38 THEN Y = 38
```

(Por que o máximo valor de Y deve ser 38, enquanto X deve ser limitado a 37?)

Se houver qualquer tentativa para colocar um cavalo para fora da tela, ele será deslocado para a lateral mais próxima. Há outras possíveis estratégias, tais como, dar uma mensagem de erro e parar o programa. Contudo, nossa escolha tem a vantagem de que ela não pára o programa, e você pode ver que alguma coisa está acontecendo.

Algumas vezes você quer mudar os valores em uma subrotina para diferentes programas GOSUB. Por exemplo, um segundo jogador pode querer colocar um cavalo de cor diferente. Uma maneira de se fazer isto, seria digitar a subrotina inteira novamente, com cores diferentes. Contudo, vamos tentar usar as variáveis ao invés de números. Ao invés da linha 1010 dizer COLOR = 7, ela poderia dizer

```
1010 COLOR = C : REM C = CORPO
```

Analogamente, você poderia escrever:

```
1040 COLOR = PES
1070 COLOR = CABECA
```

Então, o programa principal poderia ser assim:

```
20 GR
30 REM A COR DO CAVALO DO PRIMEIRO JOGADOR
40 C = 7 : REM AZUL CLARO : REM C = CORPO
50 PES = 9 : REM LARANJA
60 CABECA = 15 : REM BRANCO
70 REM CENTRAR O CAVALO DO PRIMEIRO JOGADOR
80 X = 15
90 Y = 30
100 GOSUB 1000
```

e assim por diante (tenha certeza de que você colocou uma declaração END, antes de tentar rodá-lo (RUN)). Várias declarações são necessárias, cada vez que você quiser um cavalo, mas é ainda bem menos do que se você tivesse que redigitar todo o programa. Para uma facilidade adicional na programação, um truque sutil é ter uma subrotina que determina as cores para o cavalo de cada jogador -- e que cada uma dessas rotinas chame a subrotina que desenha um cavalo.

```
2000 REM DESENHE O CAVALO AZUL COM PES LARANJA E CABECA BRANCA
2010 C = 7 : REM AZUL CLARO : REM C = CORPO
2020 PES = 9 : REM LARANJA
2030 CABECA = 15 : REM BRANCO
2040 GOSUB 1000
2050 RETURN
```

```
2500 REM DESENHE O CAVALO LARANJA COM PES COR DE ROSA E CABECA
VERDE
2510 C = 9 : REM LARANJA : REM C = CORPO
2520 PES = 11 : REM ROSA
2530 CABECA = 12 : REM VERDE
2540 GOSUB 1000
2550 RETURN
```

Agora, tudo o que você precisa, para colocar um cavalo azul com cabeça branca e pés laranja em (10, 11), é

```
30 REM CAVALO DO PRIMEIRO JOGADOR
40 X = 10
50 Y = 11
60 GOSUB 2000
```

Para colocar um cavalo laranja em (19,2), tudo o que precisa é

```
70 REM CAVALO DO SEGUNDO JOGADOR
80 X = 19
90 Y = 2
100 GOSUB 2500
```

Ambas as subrotinas que desenham um cavalo, da página anterior, começando com as linhas 2000 e 2500, chamam uma outra subrotina que começa na linha 1000. As coisas começam a se tornar eficientes neste estágio. Uma vez que você escreveu uma boa subrotina, que verifica erros e usa variáveis que você pode determinar no programa que for chamado (que pode ser o programa principal ou uma outra subrotina), então você pode sobrepor outras subrotinas sobre ela. Isto torna os programas principais mais fáceis de escrever. Usando as três subrotinas, é muito fácil colocar na tela uma atrativa exibição de cavalos.

Mas primeiro, uma outra rotina útil:

```
3000 REM ESCOLHA X,Y ALEATORIOS
3010 X = INT (RND(1) * 37) + 1
3020 Y = INT (RND(1) * 38) + 1
3030 RETURN
```

E, agora para o programa principal

```
10 REM ACIONE MODO GRAFICO
20 GR
30 REM ESCOLHA UM PONTO ALEATORIO
40 GOSUB 3000
50 REM PONHA UM CAVALO AZUL NESTE PONTO
60 GOSUB 2000
70 REM ESCOLHA OUTRO PONTO ALEATORIO
80 GOSUB 3000
90 REM PONHA UM CAVALO LARANJA NESTE PONTO
100 GOSUB 2500
110 REM FAÇA TUDO NOVAMENTE
120 GOTO 30
```

Isto é como deve parecer um programa principal, se você é um bom programador: incluindo vários REM e GOSUB. O trabalho deve ser feito em subrotinas relativamente curtas, cada uma delas fácil de ser escrita, e completas em si mesmas. Sinta-se livre para usar TRACE para verificar como este programa trabalha.

GRÁFICOS DE ALTA RESOLUÇÃO

Nós chamamos o tipo de gráficos que você utilizou até agora, de gráficos de BAIXA RESOLUÇÃO. Nesta seção, você aprenderá a usar um outro tipo de gráficos, chamados gráficos de ALTA RESOLUÇÃO. Este novo tipo de gráficos, permite que você desenhe com maiores detalhes, do que poderia obter com uma tela de baixa resolução (40 por 40). A nova tela de gráficos de alta resolução é constituída de 280 por 160 pontos traçados. As coordenadas horizontais, começam com 0 à esquerda da tela e terminam com 279 à direita. Analogamente, as coordenadas verticais vão de 0 no topo da tela a 159 na base.

Os gráficos de alta resolução não são difíceis de entender. Frequentemente, os comandos dos gráficos de alta resolução são os mesmos correspondentes aos gráficos de baixa resolução, exceto pela adição de um H (de High-resolution). Um conhecimento completo dos gráficos de baixa resolução lhe será útil nesta seção.

Digite

HGR

para introduzir os gráficos de alta resolução. Este comando limpa a tela, deixando quatro linhas na base da mesma para texto. Assim, como nos gráficos de baixa resolução, os gráficos de alta resolução permitem que você use coordenadas verticais que estariam na área de texto (192 é o máximo), mas estes pontos não são mostrados na tela. Se o cursor não for visível, pressione **RETURN** umas poucas vezes, até que ele apareça perto da base da tela.

Os gráficos de alta resolução, são realmente maravilhosos, mas você tem que fazer algum sacrifício se quiser usá-los. Há bem poucas cores disponíveis neste novo tipo de modo gráfico. As cores de alta resolução, vão de 0 (preto) a 7 (branco). São elas

0	preto 1	4	preto 2
1	verde	5	laranja
2	violeta	6	azul
3	branco 1	7	branco 2

Estas irão variar de TV para TV e de acordo com as suas posições na tela. Um ponto de alta resolução traçado com a cor número 3, por exemplo, será azul se a coordenada horizontal do ponto for par, verde se a coordenada horizontal for ímpar, e branco somente se AMBAS as coordenadas horizontais par e ímpar forem traçadas. Esta é exatamente a forma de funcionamento dos TVs normais.

A única instrução para o traçado (PLOT) de gráficos em alta resolução é HPLOT. Para tentar isto, uma vez que você já executou o comando HGR, digite


```
HCOLOR = 3  
HPLOT 130, 100
```

A última linha traçará um ponto branco de alta resolução na posição X = 130, Y = 100



É mais fácil desenhar linhas em gráficos de alta resolução do que em gráficos de baixa resolução. Você simplesmente digita o comando HPLOT de um ponto na tela PARA (TO) outro ponto. Para desenhar uma linha no topo da tela, digite

```
HPLOT 0, 0 TO 279, 0
```

Se você quer desenhar uma linha de um canto no ponto 279,0 até o outro canto da tela, tudo que você tem que fazer é digitar

```
HPLOT TO 279, 159
```

e uma linha aparece ao longo da margem direita da tela. Quando você usa esta última declaração, a nova linha adota como, seu ponto inicial e sua cor, à do último ponto anteriormente traçado (mesmo se você introduziu um novo comando HCOLOR desde que o ponto foi traçado). Você pode "encadear" estes comandos e desenhar diversas linhas em uma mesma declaração (HPLOT) se assim o desejar.

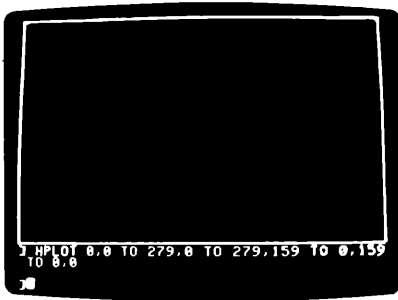


Elppasoft em cassette ou disco não pode usar esta característica de "encadeamento". Se você tem Elppasoft em cassette ou disco, você deve usar declarações HPLOT, especificando no máximo, dois pontos para desenhar linhas.

Limpe a tela com HGR, e tente isto no seu Elppa:

```
H PLOT 0, 0 TO 279, 0 TO 279, 159 TO 0, 159 TO 0, 0
```

Deveria aparecer uma linha ao redor da tela. Se ela não for contínua, primeiramente verifique se você digitou corretamente. Se a linha ainda não apareceu ou não for contínua, mude HCOLOR e tente outra vez. Algumas partes da tela somente aparecem em certas cores em determinadas TVs.



Desenhar linhas diagonais também é fácil em gráficos de alta resolução. Para desenhar uma linha do canto esquerdo superior ao canto direito inferior da tela, apenas digite

```
H PLOT 0, 0 TO 279, 159
```

Pratique, desenhando linhas de alta resolução variando seus comprimentos e cores.

Aqui está um programa que torna o seu Elppa uma tela de alta resolução.

```
NEW
200 HGR
210 HCOLOR = 3
220 X = PDL(0)
230 Y = PDL(1)
240 IF Y > 159 THEN Y = 159
250 H PLOT X,Y
260 GOTO 220
```

A linha 240 foi incluída porque a função PDL pode retornar valores para o controlador de jogo acima de 255, e a coordenada Y estaria fora da tela se o seu valor fosse maior que 159. Rode (RUN) este programa.

Este programa funciona, mas seria melhorado se fosse mais fácil desenhar uma linha contínua. Aqueles espaços entre os pontos traçados, nem sempre são desejados. Você pode melhorar o programa digitando as seguintes linhas:

```
220 GOSUB 1000
230 HPLOT X,Y
240 GOSUB 1000
250 HPLOT TO X,Y
260 GOTO 240
1000 X = FDL(0) / .913
1010 Y = FDL(1) / 1.6
1020 RETURN
```

Liste (LIST) o programa e verifique-o cuidadosamente para ter certeza que você digitou tudo corretamente. Aqui está o que o programa faz. Os gráficos de alta resolução e HCOLOR são introduzidos, e então o programa vai para a subrotina que começa na linha 1000. A subrotina determina o valor das coordenadas X e Y. Cada um dos controles de jogos, retorna um valor máximo de 255. Devido ao fato que, os gráficos de alta resolução usam coordenadas horizontais de 0 a 279, os valores retornados por PDL (0), são divididos por .913 para estender a sua faixa em todo o comprimento da tela. Analogamente, os valores retornados por PDL (1), são divididos por 1.6, para comprimir a sua faixa para a das coordenadas verticais de alta resolução: 0 a 159. Assim como nos gráficos de baixa resolução, a coordenada realmente traçada, é o valor do inteiro mais próximo, menor ou igual ao valor dado. A linha 1020 retorna (RETURN) ao programa principal, e então a linha 230 traça o ponto X, Y. Depois o programa retorna à subrotina, dá novos valores a X e Y e retorna (RETURN) à linha 250 no programa principal, onde uma linha muito pequena é desenhada do antigo ponto X, Y ao novo X, Y. O GOTO na linha 260, repete todo o programa, exceto as instruções HGR e HCOLOR, obtendo novos valores para X e Y da posição dos controles de jogos, e então traça (PLOT) a nova posição X, Y. Use **CTRL C** para parar o programa.

Há uma razão para desenhar linhas ao invés de traçar cada ponto separadamente. Leva algum tempo, traçar ponto por ponto, e quando o Elppa traça pontos um por vez, ele não pode manter os controles de jogos. Isto é porque haviam espaços entre os pontos, quando você movia os botões nos controles de jogos rapidamente no primeiro programa. Desenhando uma pequena linha para cada nova posição especificada pelos botões do controle de jogo, podemos consertar isso: ao desenhar uma linha de um ponto para outro, todos os pontos intermediários são traçados automática e muito mais rapidamente do que se eles tivessem sido traçados um por vez.

Agora guarde (SAVE) o programa, e então, rode-o (RUN).

Aqui está um programa que desenha lindas figuras na sua tela.

```
NEW
 90 HOME
100 VTAB 24 : REM MOVA O CURSOR PARA A ULTIMA LINHA
120 HGR : REM ACIONE O MODDO GRAFICO DE ALTA RESOLUCAO
140 A = RND(1) * 279 : REM FIXE UM "A" PARA CENTRO
160 B = RND(1) * 159 : REM FIXE UM "B" PARA CENTRO
180 N = INT (RND(1) * 4) + 2 : REM FIXE O TAMANHO DE UM PASSO
200 HTAB 15: PRINT "PASSO = ";N;
220 FOR X = 0 TO 278 STEP N : REM PASSO DETERMINADO POR VALORES
DE A
240 FOR S = 0 TO 1 : REM 2 LINHAS, DE X E X + 1
260 HCOLOR = 7 * S : REM PRIMEIRA LINHA PRETA, PROXIMA BRANCA
280 REM DESENHE UMA LINHA DO "CENTRO" AO LADO OPOSTO
300 HPLOT X + S,0 TO A,B TO 279 - X - S,159
320 NEXT S,X
340 FOR Y = 0 TO 158 STEP N : REM PASSO DETERMINADO POR VALORES
DE B
360 FOR S = 0 TO 1 : REM 2 LINHAS, DE B E B + 1
380 HCOLOR = 7 * S : REM PRIMEIRA LINHA PRETA, PROXIMA BRANCA
400 REM : DESENHE UMA LINHA DO "CENTRO" AO LADO OPOSTO
420 HPLOT 279,Y + S TO A,B TO 0, 159 - Y - S
440 NEXT S,Y
460 FOR PAUSA = 1 TO 1500 : NEXT PAUSA : REM ATRASO
480 GOTO 120
```

Este é um programa bem longo, por isso, digite-o cuidadosamente, e liste-o (LIST) por partes (por exemplo, usando (LIST 0, 320)) para verificar o que você digitou. Quando estiver certo que tudo está correto, rode (RUN) o programa.



Se você tem o Elppasoft em disco ou cassette, terá que digitar as linhas 300 e 420, usando declarações HPLOT separadamente. Aqui está um exemplo de como fazê-lo:

```
300 HPLOT X + S,0 TO A,B
310 HPLOT TO 279 - X - S,159

420 HPLOT 279,Y + S TO A,B
430 HPLOT TO 0,159 - Y - S
```

Como você viu nas linhas 320 e 440, uma instrução NEXT pode conter variáveis de mais de uma declaração FOR. Tenha certeza de que você listou as variáveis NEXT na ordem certa, para evitar loops cruzados.

Para voltar a programar, pare a figura, digitando

A image of a black rectangular key with the word "CTRL" in white capital letters.A image of a black rectangular key with the letter "C" in white capital letters.

e então

TEXT

Você tem idéias para modificar o programa? Após guardar (SAVE) esta versão no seu disco ou cassette, tente fazer mudar o valor de HCOLOR randomicamente. Tente desenhar primeiro linhas laranja e depois azuis, ou somente linhas azuis.

CAPÍTULO 6

SEQUÊNCIAS E MATRIZES

116	Cadeias de caracteres
121	Concatenação
122	Mais funções sequenciais
124	Introduzindo matrizes
126	Mensagens de erro em matrizes
127	Conclusão

CADEIAS DE CARACTERES

Você gostaria de ver o seu nome soletrado de trás para frente? Até o momento, temos manipulado com gráficos e números, mas os computadores também podem manipular letras e símbolos. Seu computador pode lidar com um único caracter, ou com uma cadeia (STRING) inteira de caracteres, de uma vez. Isto parecerá muito natural, uma vez que nós, humanos, também lidamos com caracteres agrupados para formar palavras, nomes, etc. Variáveis que contêm cadeias de caracteres, como variáveis numéricas, também têm nomes. Os nomes das variáveis em cadeia, seguem as mesmas regras dos nomes das variáveis numéricas, exceto que elas terminam com o símbolo do cifrão (\$). Aqui estão alguns exemplos de nomes de variáveis sequenciais (ou, em cadeia):

```
NOME$  
A$  
FRASE$
```

A variável A é diferente da variável A\$, e ambas podem ser usadas no mesmo programa.

Se você deseja que a variável sequencial chamada NOME\$ (pronunciada "NOME-cifrão"), contenha as letras "DEODORO DA FONSECA", você pode digitar

```
NOME$ = "DEODORO DA FONSECA"
```

Note que os caracteres que você coloca em uma variável sequencial devem estar entre aspas. A declaração

```
PRINT NOME$
```

imprimirá o conteúdo da variável NOME\$: neste caso, o nome do primeiro PRESIDENTE DO BRASIL. Portanto, quando você tem uma sequência de caracteres que você frequentemente precisa, pode armazená-la em uma variável com um nome curto.

Há diversas instruções no Elppasoft que manipulam sequências. Suponha que você queira saber o comprimento de uma sequência (quantos caracteres ela contém). Você pode digitar

```
PRINT LEN ("DEODORO DA FONSECA")
```

ou você digita a declaração equivalente,

```
PRINT LEN (NOME$)
```


e o Elppa imprimirá (PRINT) o comprimento (LENGth) da sequência, neste caso 18. Note que os espaços também são contados como caracteres.

O número de caracteres em uma sequência pode variar de 0 a 255. Se você tentar usar mais do que 255 caracteres em uma sequência, obterá as mensagens de erro: ERRO:SINTAXE ou ERRO:STRING LONGO. Uma sequência com 0 caracteres, é chamada de uma sequência NULA.

Em algumas ocasiões, você pode querer imprimir (PRINT) somente uma parte da variável NOME\$. Para fazer isto, você pode utilizar três funções muito úteis: LEFT\$, RIGHT\$ e MID\$.

Se, por exemplo, você quer imprimir (PRINT) as sete primeiras letras que estão em NOME\$, digite

```
PRINT LEFT$ (NOME$,7)
```

e

```
DEODORO
```

aparecerá na tela. Se você digita

```
PRINT RIGHT$ (NOME$,6)  
ONSECA
```

aparecerá. Para cada programa que você escreve, que usa variáveis sequenciais, você deve determinar o valor da sequência DENTRO do programa. Cada vez que você roda (RUN) um programa, todas as variáveis numéricas são primeiramente zeradas e todas as variáveis sequenciais são igualadas à nula (ou contém a sequência nula). Aqui está um pequeno programa que usa as funções LEN e LEFT\$.

```
NEW  
90 NOME$ = "DEODORO DA FONSECA"  
100 FOR N = 1 TO LEN (NOME$)  
110 PRINT LEFT$ (NOME$,N)  
120 NEXT N
```

Rode (RUN) este programa. O comando RIGHT\$ é idêntico ao comando LEFT\$, exceto que usa os caracteres mais à direita na sequência. Agora escreva um outro programa substituindo LEFT\$ por RIGHT\$. O que acontece quando você o roda (RUN)?

Se você quer usar os caracteres começando do meio da sequência, ao invés do início ou do fim, a função MID\$ é o que você precisa.

Digite

```
PRINT MID$ (NOME$,9)
```

Seu computador respond com

```
DA FONSECA
```

uma vez que a letra D é o nono caracter na sequência. Agora tente este programa.

```
NEW
190 NOME$ = "DEODORO DA FONSECA"
200 FOR N = 1 TO LEN (NOME$)
210 PRINT MID$ (NOME$,N)
220 NEXT N
```

Você obtém o esperado, quando o programa é rodado (RUN)?

Suponha que você quer imprimir (PRINT) apenas as letras "O DA F", da sequência chamada NOME\$. Para fazer isto, você deve acrescentar um outro argumento à função MID\$.

```
PRINT MID$ (NOME$,7,6)
```

O primeiro número (7) especifica a posição do caracter, na qual o Elppa deve começar a imprimir (PRINT). O segundo número (6) diz quantos caracteres devem ser impressos (PRINT). Portanto, a instrução é interpretada pelo Elppa como "encontre o sétimo caracter, em NOME\$, e imprima (PRINT) seis caracteres começando pelo sétimo e movendo-se para a direita". Mude a linha 210 do programa anterior como se segue, e rode-o (RUN).

```
210 PRINT MID$ (NOME$,N,6)
```

Não prossiga neste manual, até que você tenha testado completamente as funções LEFT\$, RIGHT\$ e MID\$.

Considere este programa.

```
NEW
200 A$ = "ABCDEFGHIJKLMNPOQRSTUVWXYZ"
210 PRINT
220 PRINT "DIGITE UM NUMERO, DE 1 ATE ";LEN(A$);" "
230 PRINT "E EU LHE DIREI QUE LETRA TEM AQUELA POSICAO NO ALFABETO.  ";
```

```

240 INPUT P
250 IF P > LEN(A$) OR P < 1 THEN GOTO 210
260 PRINT
270 PRINT MID$(A$,P,1);" E A LETRA NUMERO ";P;" NO ALFABETO. "
280 PRINT : PRINT
290 PRINT "DIGITE UMA LETRA, E EU LHE DIREI"
300 INPUT "ONDE ELA ESTA NO ALFABETO. ";X$
310 FOR N = 1 TO LEN(A$)
320 IF MID$(A$,N,1) = X$ THEN GOTO 370
330 NEXT N
340 PRINT
350 PRINT "ISTO NAO E UMA LETRA DO ALFABETO. " : PRINT
360 GOTO 290
370 PRINT
380 PRINT X$;" E A LETRA NUMERO ";N;" DO ALFABETO. "
390 PRINT
400 GOTO 210

```

Este programa ilustra algumas práticas comuns de programação. Note como ele encontra a posição de um caracter em uma sequência. Este método de usar um loop para examinar minuciosamente, uma posição por vez, em uma sequência é muito comum. Note também a função dos espaços em branco, nas declarações PRINT. O que aconteceria com a saída sem estes espaços em branco? Finalmente, observe que os limites do programa são sempre determinados pelo comprimento da variável A\$ (LEN (A\$)), que é o número de caracteres do alfabeto. Isto permite que o programa funcione mesmo se você especificar um "alfabeto" diferente na linha 200. Experimente e veja.

Você pode substituir uma sequência por outra, com uma declaração de substituição, tal como

```
X$ = A$
```

Esta declaração copia o conteúdo de A\$ em X\$. Contudo, você não pode usar as funções de notação parcial no lado esquerdo de uma declaração de substituição. Por exemplo, a declaração

```
MID$ (X$,3,3) = "XYZ"
```

é ilegal, mas a declaração

```
X$ = MID$ (A$,24,3)
```

está correta. Somente uma variável pode estar no lado esquerdo de uma declaração de substituição.

Oh sim -- você ainda quer ver o seu nome soletrado de trás para frente? O programa a seguir fará justamente isto.

```

NEW
100 REM PROGRAMA PARA SOLETRAR O SEU NOME DE TRAS PARA FRENTE
110 INPUT "DIGITE SEU NOME E EU O MOSTRAREI A VOCE SOLETRADO DE
TRAS PARA FRENTE ";N$
120 REM INVERTENDO A ORDEM DAS LETRAS
130 FOR T = LEN(N$) TO 1 STEP -1
140 R$ = R$ + (MID$(N$,T,1))
150 NEXT T
160 PRINT : PRINT "SEU NOME SOLETRADO DE TRAS PARA FRENTE E ";R$
170 PRINT : PRINT
180 GOTO 110

```

Rode (RUN) este programa, tentando diversos nomes diferentes. Depois que o programa for executado algumas vezes, você notará que há alguma coisa errada. A linha 140 é a chave do problema. Se o seu nome é PEDRO, por exemplo, você o digitaria quando pedido e então colocaria PEDRO em N\$ e ORDEP em R\$. Talvez uma amiga sua chamada MÁRCIA esteja aí com você, e queira também ver o seu nome de trás para frente. A próxima vez que o programa solicitar por um nome, ela digitaria o seu nome, colocando MÁRCIA em N\$. A linha 140, colocaria então em R\$, o conteúdo anterior de R\$ MAIS N\$ de trás para frente, em outras palavras, ORDEPAICRAM. Há, portanto, a necessidade de um comando que "zere" as variáveis sequenciais, de forma que R\$ possa ser recarregado com novos caracteres após cada GOTO.

Felizmente, este comando existe no Elppasoft. É o comando CLEAR. O comando CLEAR "zera" todas as variáveis de qualquer tamanho, forma e cor. Acrescente esta linha ao seu programa.

```
175 CLEAR
```

Agora rode (RUN) o programa novamente.

O comando CLEAR, também pode ser usado em execução imediata. Digite

```
N = 254
PRINT N
```

Agora digite

```
CLEAR
```

e então

```
PRINT N
```

novamente. O seu Elppa deu 0 como o valor de N?

CONCATENAÇÃO

É possível acrescentar uma segunda sequência no fim de uma já existente, usando o sinal (+). Este processo é chamado CONCATENAÇÃO. Tente o seguinte no seu Elppa

```
C$ = "BOM DIA"  
D$ = C$ + " " + "MARCIA"  
PRINT D$
```

O seu Elppa responderá com

```
BOM DIA MARCIA
```

A concatenação é especialmente útil se você deseja retirar parte de uma sequência, e então, colocá-la de volta com pequenas modificações. Por exemplo, se você quiser criar uma nova sequência semelhante à anterior (D\$), exceto que os espaços entre as palavras, fossem substituídos por traços, você poderia digitar

```
E$ = RIGHT$(D$,6) + "-" + LEFT$(D$,3) + "-" + MID$(D$,5,3)  
PRINT E$
```

e

```
MARCIA-BOM-DIA
```

apareceria na sua tela.

Aqui está um programa que usa concatenação.

```
NEW  
100 INPUT "DIGITE METADE DE UMA FRASE. ";PRIMEIRA$  
110 INPUT "AGORA DIGITE A SEGUNDA METADE DA FRASE";SEGUNDA$  
120 FRASE$ = PRIMEIRA$ + SEGUNDA$  
130 PRINT  
140 PRINT FRASE$  
150 PRINT : PRINT : PRINT : GOTO 100
```

E é desta maneira que você faz a concatenação.

MAIS FUNÇÕES SEQUENCIAIS

Sequências podem ser compostas de quase todos os tipos de caracteres, incluindo números. Assim como na declaração PRINT, os caracteres entre aspas em uma sequência, não podem ser aritmeticamente interpretados, mesmo se forem números. Para ver o que acontece quando você tentar fazê-lo, digite

```
C# = "123"  
PRINT C# + 7
```

O seu Elppa imprimirá

```
ERR0:VAR IMPROPRIA
```

e não será capaz de executar a última declaração. Nós precisamos da ajuda da função

```
VAL
```

(abreviatura de VALue) para resolver este problema.

A função VAL retorna o valor (VALue) do conteúdo de uma sequência. Digite

```
PRINT C#
```

e então digite

```
PRINT VAL(C#)
```

Ambos os comandos aparentemente fornecem o mesmo resultado; contudo, as aparências enganam. Você já sabe que se digitar

```
PRINT C# + 5
```

o seu Elppa responderá com

```
ERR0:VAR IMPROPRIA
```

Tente digitar

```
PRINT VAL(C#) + 5
```

e

aparece na tela. Note que o nome da variável sequencial que é o argumento da função VAL, deve estar entre parênteses.

O que acontece se você quiser colocar o valor de C\$ menos 21 em uma variável comum (não sequencial)? É simples. Apenas digite

```
Q = VAL(C$) - 21
```

Agora digite

```
PRINT Q
```

e veja o que você obtém. O conteúdo de Q é o que você espera? Você pode mesmo usar VAL para acrescentar o valor numérico de duas sequências diferentes. Para tentar isto, crie uma nova sequência

```
K$ = "12"
```

e então digite

```
P = VAL(C$) + VAL(K$)
PRINT P
```

Tente VAL com diferentes sequências, incluindo sequências que começam ou terminam com letras.

Algumas vezes é necessário mudar um número em uma sequência. A função STR\$, que funciona como o inverso da função VAL, pode ser usada para fazer esta mudança. Suponha que você queira mudar a variável numérica P, para uma variável sequencial. Digitar

```
P$ = STR$(P)
PRINT P$
```

lhe mostrará como funciona a função STR\$. Aqui está um programa que usa STR\$ e VAL.

```
300 INPUT "DIGITE UM NUMERO DE 1 A 999999999. ";N$
310 N = VAL(N$)
320 IF N < 1 OR N 999999999 THEN GOTO 300
330 N$ = STR$(N)
340 FOR T = LEN(N$) TO 1 STEP -1
350 P$ = P$ + MID$(N$,T,1)
360 NEXT T
370 PRINT : PRINT "ORIGINAL",N$
```

```

380 PRINT : PRINT "INVERSO",P$
390 P = VAL(P$)
400 PRINT : PRINT "ORIGINAL + INVERSO = ",N + P
410 CLEAR
420 PRINT : PRINT : GOTO 300

```

Você entende como funciona o programa? Por que há vírgulas nas linhas 370 e 380? Tente apagar a linha 330 para ver qual é o seu efeito. As primeiras quatro linhas deste programa demonstram os primeiros passos para fazer uma rotina de entrada verdadeiramente "à prova de erros". Veja que entradas podem ainda parar este programa, e então, procure descobrir meios de detetá-las antes que façam o programa parar.

INTRODUZINDO MATRIZES

Nesta seção sobre matrizes, usaremos exemplos matemáticos, mas eles são da matemática recreativa e não requerem nada além do conhecimento básico da aritmética elementar.

Matrizes permitem que você selecione qualquer elemento em uma tabela de números, e as vantagens que elas oferecem na programação, compensam o pequeno esforço que você deve fazer em pensar um pouco mais até estar familiarizado com elas.

Uma MATRIZ é uma tabela de números. O nome desta tabela, chamado de NOME DA MATRIZ, é qualquer nome de uma variável legal: A, por exemplo. O nome da matriz A é distinto e separado da variável simples A.

Para criar uma matriz, você deve primeiro dizer ao computador o máximo número de elementos que você quer atribuir à ela. Para fazer isto, você usa a declaração DIM (abreviatura de DIMension). Os elementos em uma matriz são numerados a partir de 0. Assim para dimensionar (DIM) uma matriz chamada A, que terá no máximo 16 elementos, digite

```
DIM A(15)
```

A declaração DIM anterior, nos dá 16 novas variáveis. Elas se comportam exatamente como as variáveis que você já aprendeu. Elas são:

```

A(0)
A(1)
A(2)

```

e assim por diante, até

```
A(15)
```


Embora você possa achá-las desajeitadas para digitar, elas podem ser usadas da mesma forma como qualquer outra variável.

A declaração

```
A(9) = 45 + A(12)
```

é perfeitamente correta. O número entre parênteses, é chamado um ÍNDICE, e a notação A(12) é lida "A-doze". O índice pode ser uma expressão aritmética ou pode ser representado por uma variável.

Digite o seguinte programa. Ele ilustra o uso de variáveis no índice e imprime na tela o conteúdo de cada elemento da matriz.

```
100 REM DIMENSIONE A MATRIZ CHAMADA "DIAS", PARA CONTER 7 NUMEROS
110 DIM DIAS(6)
120 REM PREENCHA A MATRIZ
130 FOR NUM = 0 TO 6
140 DIAS(NUM) = NUM + 1
150 NEXT NUM
160 REM IMPRIMA O CONTEUDO DOS ELEMENTOS DA MATRIZ
170 FOR I = 0 TO 6
180 PRINT "DIAS(";I;") = ";DIAS(I)
190 NEXT I
```

Se uma matriz é usada em um programa, antes de ter sido dimensionada (DIM), o Elppasoft reserva espaço para 11 elementos (índices de 0 a 10). Contudo, é boa prática de programação dimensionar TODAS as matrizes.

Suponha que você queira escrever um programa que gera os números de um a oito, de uma forma desordenada. Para realizar isto, você precisa manipular com tabelas de dados. É justamente o tipo de coisa para a qual as matrizes são excelentes. O programa a seguir realiza isto.

```
NEW
200 REM DIMENSIONE A MATRIZ
210 DIM COFO(8)
220 REM PREENCHA A MATRIZ
230 FOR I = 1 TO 8
240 COFO(I) = I
250 NEXT I
260 REM DESORDENE A MATRIZ E ESCOLHA CADA ELEMENTO
270 FOR VINHO = 1 TO 8
280 REM ESCOLHA ALGUM OUTRO ELEMENTO
290 LEITE = INT (RND (1) * 8) + 1
300 REM LEITE ERA DIFERENTE DE VINHO?
310 REM SE NAO, TENTE OUTRA VEZ
320 IF LEITE = VINHO THEN GOTO 280
330 REM TROQUE COFO(VINHO) E COFO(LEITE)
```

```

340 TEMP = COPO(VINHO) : COPO(VINHO) = COPO(LEITE) : COPO(LEITE)
= TEMP
350 NEXT VINHO
360 REM IMPRIMA CONTEUDO DA MATRIZ
370 FOR C = 1 TO 8
380 PRINT COPO(C)
390 NEXT C

```

Você entendeu como funciona este programa? Primeiramente, ele preenche uma matriz com números e então, desordena os conteúdos da matriz. Note que você não tem que começar a preencher a matriz a partir de zero. Aqui está uma descrição do que fazem algumas linhas deste programa. As linhas 230 a 250 preenchem a matriz e atribuem a cada elemento um número correspondente ao seu índice (COPO (1) = 1, etc.). A linha 270 introduz a nova variável VINHO para números de 1 a 8. A linha 280 introduz a variável LEITE para inteiros RANDÔMICOS (aleatórios) de 1 a 8. Então, a linha 300 verifica se o valor de VINHO não é igual ao valor de LEITE, em um dado momento. O conteúdo das variáveis COPO (VINHO) e COPO (LEITE) são trocados na linha 310. Finalmente, a matriz é impressa com as linhas 330 e 350.

A troca que ocorre na linha 310, pode ser imaginada da seguinte maneira: vamos dizer que temos dois copos -- um é o copo de vinho (VINHO) e o outro é um copo de leite (LEITE). Oh não, houve um engano. O leite está no copo de vinho e o vinho está no copo de leite. Felizmente, nós temos um copo extra, chamado TEMP. Nós podemos colocar o leite no copo extra, e então colocar o vinho no copo de vinho e, finalmente, colocar o leite no copo de leite. Desta forma, ambas as bebidas foram trocadas para os seus copos apropriados.

MENSAGENS DE ERRO EM MATRIZES

Aqui estão algumas mensagens de erro que você pode gerar enquanto trabalha com matrizes.

ERRO:MATRIZ REDIM

Esta mensagem de erro ocorre quando uma matriz é dimensionada mais de uma vez em um mesmo programa. Este erro ocorre frequentemente quando deixamos de dimensionar uma matriz, e posteriormente, acrescentamos ao programa uma declaração de dimensionamento.

ERRO:ÍNDICE ILEGAL

Esta mensagem de erro ocorrerá, se for feita uma tentativa de usar um elemento da matriz que está fora da dimensão da mesma. Por exemplo, se A foi dimensionada para ter 25 elementos com a declaração DIM A(25), referir-se ao elemento A(52) ou qualquer outro elemento cujo índice é menor do que 0 ou maior do que 25,

lhe dará a mensagem de erro ERRO:ÍNDICE ILEGAL

ERRO:QUANTID ILEGAL

Você obterá esta mensagem se tentar usar um número negativo como índice em uma matriz.

Estas são algumas das formas que você pode usar matrizes. As matrizes utilizadas aqui, são todas matrizes de uma dimensão. Você também pode usar matrizes que tenham duas ou mais dimensões.

CONCLUSÃO

Este manual apresentou o essencial da linguagem Elppasoft. Se você tornar a lê-lo, escrevendo os seus próprios programas com as declarações que foram apresentadas, você solidificará consideravelmente o seu conhecimento. O Elppa tem muitas outras habilidades, e uma vez que você assimilou as que foram aqui apresentadas, terá ainda várias outras para explorar e descobrir.

CAPÍTULO 7

COMO FUNCIONA O SEU ELPPA

130	Como funciona o seu Elppa
133	Hardware
133	Processador
134	Memória principal
135	Sistema software
136	Informação
136	Discos e arquivos

COMO FUNCIONA O SEU ELPPA

Se, até o momento, você realmente USOU este manual, e não apenas o estudou, como dissemos no início, provavelmente já deve estar bem familiarizado com o seu microcomputador e sabe como programá-lo.

Este capítulo explica em termos gerais como funciona o sistema Elppa II plus. Se você achar a explanação e todos os novos termos ligados à computação difíceis de compreender, não se preocupe. Para dirigir um carro, você não precisa saber exatamente como funciona o seu sistema de combustão interna. Da mesma forma, você não precisa entender exatamente o que está acontecendo dentro do Elppa II plus, para ser capaz de colocá-lo em funcionamento.

À medida que você for lendo este capítulo e se familiarizando com o sistema, muitos desses conceitos se tornarão mais claros. Seja paciente!

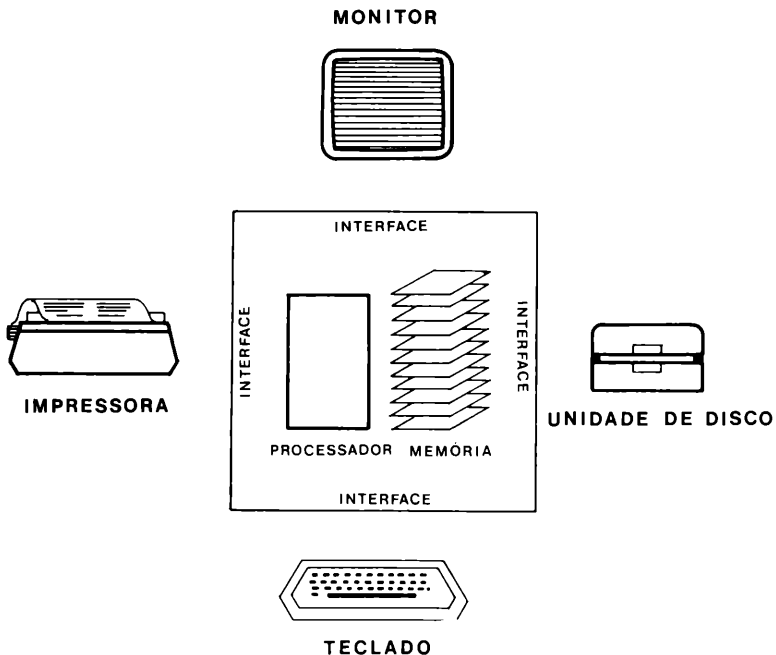
O sistema Elppa II plus, é composto por três grupos fundamentais: hardware, software e informação.

HARDWARE inclui o computador e todos os dispositivos físicos acoplados à ele.

As partes mais importantes do hardware no computador são a memória principal e o processador.

Logo após, estão os circuitos que adaptam e conduzem os sinais entre o computador e os DISPOSITIVOS PERIFÉRICOS. Esses circuitos são chamados de INTERFACES. Alguns estão embutidos; outros, tais como o cartão controlador de disco, podem ser instalados nos conectores de expansão do computador.

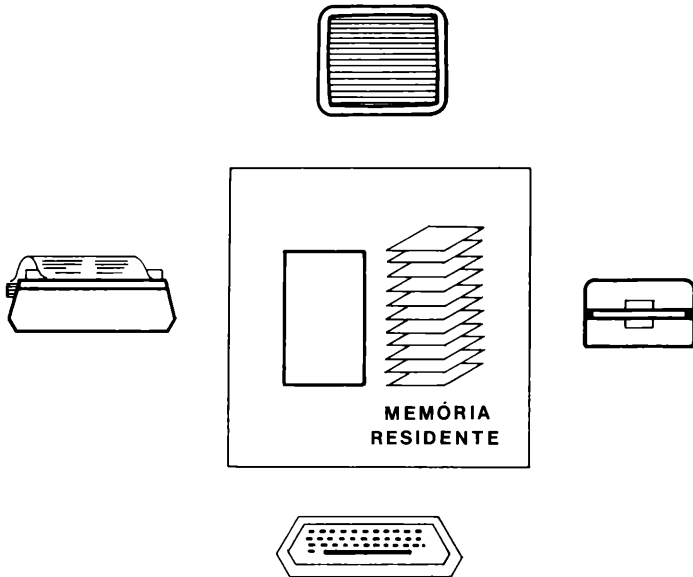
Finalmente, há os dispositivos periféricos externos: DISPOSITIVOS DE ENTRADA para introduzir a informação no computador e DISPOSITIVOS DE SAÍDA para retirar a informação do mesmo. Alguns dispositivos, tais como as unidades de disco, são uma combinação dos DISPOSITIVOS DE ENTRADA/SAÍDA (E/S), usados para armazenamento.



SOFTWARE é constituído de uma sequência de instruções (programas) que o processador pode executar, quando aquelas instruções estão na memória principal. As instruções podem ser facilmente alteradas.

Alguns programas são gravados durante o processo de fabricação em ROM (read-only memory). Estes pequenos programas residentes não podem ser alterados, e são chamados de FIRMWARE (suporte lógico inalterado).

A maioria dos programas são carregados, através de um disco, na memória principal, e aqueles que transportam a informação e facilitam o uso do sistema, juntamente com o firmware residente, são conhecidos como SISTEMA SOFTWARE. Programas levados à memória principal para fazer o sistema executar alguma coisa - organizar informações, fazer cálculos, conectar o computador com uma rede de informação distante, e assim por diante - são chamados de PROGRAMAS DE APLICAÇÃO.



INFORMAÇÃO inclui o que você digita no teclado, o que você vê na tela ou na impressora, ou o que o programa processa. O computador lida com todas essas informações na forma de números - de fato, na forma de cadeias de 1's e 0's. Como o computador interpreta estes números, depende de onde ele os encontra e o que pretende fazer com eles.

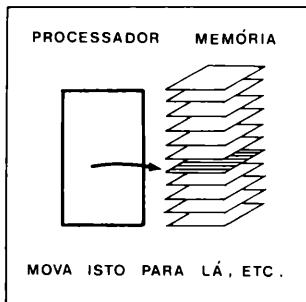
HARDWARE

O hardware no sistema, como vimos, divide-se em três categorias: o processador mais a memória principal, as interfaces para os dispositivos periféricos e os dispositivos periféricos propriamente. Estes são fáceis de compreender: fazem coisas que podemos ver, ouvir e tocar. Mas, e sobre o processador e a memória principal?

Processador

O centro de toda atividade no sistema, é o processador. A matéria prima com que ele lida, está na memória principal. O processador é um robô - como uma máquina elétrica que processa continuamente um procedimento fixo: logo que você aciona o sistema, ele localiza a primeira posição na memória e executa a instrução que lá se encontra; então, muda para a próxima posição e executa a instrução lá encontrada; e, assim por diante.

Em outras palavras, a menos que uma das instruções que o processador executa, lhe diga para ir a mais algum lugar (o que frequentemente acontece), o processador se desloca continuamente de uma posição de memória à outra. Tudo que ele realmente faz, é encontrar, mover, comparar e combinar números e ao mesmo tempo seguir para a próxima instrução ou, pular para alguma outra posição de memória para continuar. Isso não é tão simples como aparenta, desde que, aproximadamente, meio milhão de tais passos elementares são executados por segundo.



Memória Principal

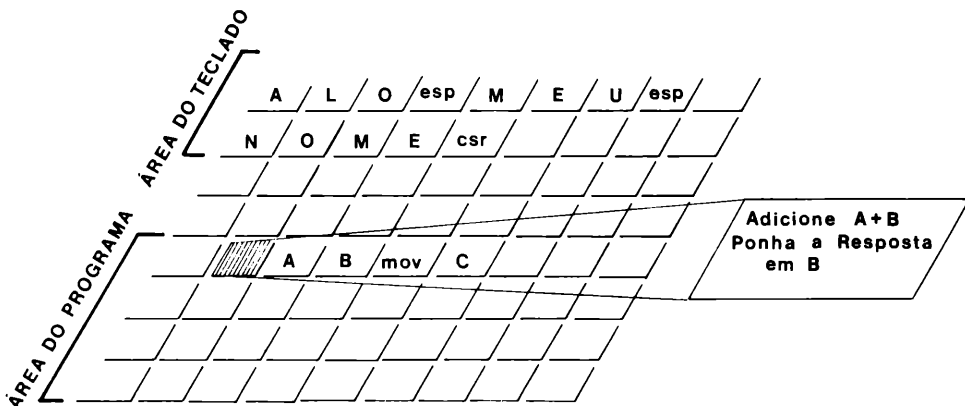
A memória principal é a área de armazenamento do processador. Ela é frequentemente chamada RAM (random-access memory) porque você pode ter acesso a qualquer posição dada, tão facilmente, quanto qualquer outra.

A memória principal é como uma grade de milhares de arquivos ou caixas de correspondências, cada uma delas identificada por um número chamado seu ENDEREÇO. Cada caixa pode reter um código que representa uma instrução, o endereço de um arquivo de memória, ou um caracter que você digita no teclado (por exemplo, a letra M), e assim por diante.

Os programas normalmente ocupam áreas da memória principal separadamente das informações, com as quais, eles estão trabalhando, para reduzir a probabilidade de confundir instruções com informações. Por exemplo, se um erro (chamado de BUG) ocorre no seu programa, ele pode tentar interpretar o número do telefone de alguém como uma instrução de programa.

Outro aspecto importante, da memória principal, é que enquanto todos os arquivos são acessíveis para o processador, alguns também são acessíveis para um outro dispositivo tais como, o teclado ou o dispositivo de representação visual (monitor de vídeo ou TV). Estas posições especiais de memória são chamadas posições de entrada/saída (E/S).

Finalmente, um aspecto significativo à respeito da memória principal é que, quando você desliga o computador, os dados e informações com os quais você estava trabalhando são perdidos; a menos que, naturalmente, você use o sistema software, especificamente um sistema operacional, para guardar as informações em um disco.



SISTEMA SOFTWARE

O sistema software, para os propósitos desta explanação, é de três tipos:

1. O PROGRAMA MONITOR RESIDENTE, que aciona o sistema, quando você liga e fornece pequenos grupos de instruções (ROTINAS) para serem utilizadas por outros programas: rotinas que transportam informações ao longo de um programa, rotinas que esperam que você pressione uma tecla; e assim por diante.
2. TRADUTORES DE LINGUAGEM, que convertem as palavras ou abreviações que você digita em um teclado, em instruções para o processador. Cada LINGUAGEM DE PROGRAMA inclui algumas dezenas de verbos imperativos mais os seus objetos, como IMPRIMA ISTO ou ADICIONE ESTES NÚMEROS ATÉ QUE VOCÊ ALCANCE O FIM DA LISTA. Contudo, o processador somente entende instruções em linguagem de máquina (como 01101001, que diz ao processador para somar dois números). Se você quer programar sem ter muito trabalho, você precisa de um tradutor que estabeleça um vínculo entre o que é digitado e a linguagem de máquina.
3. SISTEMAS OPERACIONAIS, que facilitam a nossa vida. Ao invés de ter que aprender como escrever um programa, para usar a unidade de disco, ou copiar discos, e assim por diante, você pode carregar um sistema operacional no computador, digitar uma ou duas palavras facilmente lembradas, e deixar o programa do sistema operacional fazer o árduo trabalho. A figura ilustra o que faz um típico sistema operacional de um computador pessoal.

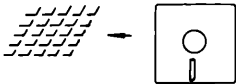
1. Prepara o disco



4. Copia um programa ou um disco inteiro



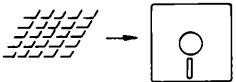
2. Carrega a informação do disco



5. Mostra a você o que está num disco



3. Guarda a informação no disco



6. Transfere a informação para outros dispositivos



O sistema operacional que vem com as unidades de disco ELPPA DISK II, é chamado de DOS (Disk Operating System), ou Sistema Operacional de Disco. O próximo capítulo lhe dará alguns exemplos práticos do DOS. As linguagens de programação Integer BASIC e o Elppasoft, assim como muitos programas prontos, usam comandos e discos DOS preparados pelo DOS.

INFORMAÇÃO

À medida que você digita as suas informações - o relatório ou dados financeiros ou respostas às mensagens de programas - o programa com o qual você está conversando, armazena a informação numa área livre da memória.

Ocasionalmente você encontrará o termo "buffer". Um BUFFER na memória principal é uma área de memorização temporária, onde a informação pode ser armazenada vagarosamente e retirada rapidamente, ou vice versa.

Outro conceito importante é o de ARQUIVOS. Quando o computador armazena informações em um disco, ele o faz na forma de arquivos.

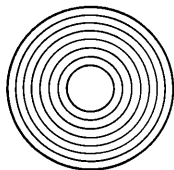
DISCOS E ARQUIVOS

Um dos serviços de um programa do sistema operacional é pegar os discos virgens e prepará-los para receber informação. Isto é chamado FORMATAÇÃO ou INICIALIZAÇÃO. Quando você formata um disco, o programa escreve círculos concêntricos de zeros (chamado pistas) no disco, como linhas divisórias numa pista de corrida, para que a unidade de disco saiba onde escrever a informação mais tarde.

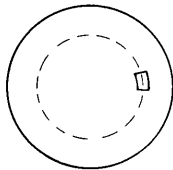
Para aumentar a velocidade de leitura ou escrita, a unidade de disco divide cada trilha em diversas partes chamadas SETORES. Os sistemas operacionais lêem e escrevem as informações no disco em formas diferentes, usando diferentes números de setores por pista; é por isso que um disco formatado para um sistema operacional pode não funcionar com outro sistema.

Quando você diz ao computador para guardar informações no disco, um sistema de programa divide primeiramente a informação em partes correspondentes à dimensão de um setor, para que ela possa ser transferida e armazenada eficientemente. Uma vez armazenada, a informação é chamada um ARQUIVO.

Alguns setores no disco contêm um índice com o nome do disco e os nomes e posições de todos os arquivos nele armazenados. Alguns sistemas operacionais chamam este índice de um CATÁLOGO, outros chamam de um DIRETÓRIO.



35 pistas por disco



16 setores por pista



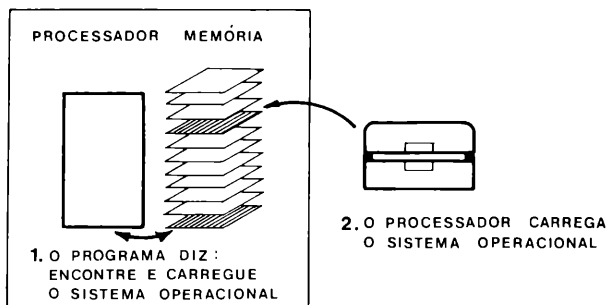
256 bytes por setor

COMO FUNCIONA O SISTEMA

Quando você inicia ou reinicia o processamento no computador, o processador começa a executar instruções no programa Monitor - instruções que dizem ao processador para procurar um disco na unidade de disco 1 e carregar um programa do sistema operacional caso ele o encontre. Isto é o que acontece quando você inicia o disco do SISTEMA MESTRE DOS 3.3 (Capítulo 2). Então, você pode usar os comandos do sistema operacional para carregar e rodar (usar) programas selecionados.

Muitos programas de aplicação têm, ou um sistema operacional ou partes necessárias dele em um disco, de forma que eles possam manipular todas as suas necessidades. Um disco completo - isto é, aquele que tem em si mesmo um sistema operacional - é chamado um DISCO DE PARTIDA. Se um programa de aplicação está em um disco de partida, logo que você liga o sistema, você está conversando diretamente com o programa (Capítulo 2).

Se um sistema operacional não é acessível quando você inicia o sistema, então tudo que está na memória é o software residente - o programa Monitor e um tradutor para a linguagem Elppasoft. Por exemplo, quando você usa um gravador cassette, ao invés de uma unidade de disco, o programa Monitor pode carregar, guardar, e rodar os programas em cassette, mas isto é tudo.



APÊNDICES

- 140 Apêndice A: Sumário dos comandos
- 151 Apêndice B: Palavras reservadas na linguagem Elppasoft
- 153 Apêndice C: Características de edição
 - 153 As teclas setas esquerda e direita
 - 153 Movimentando o cursor
 - 154 Apagando linhas de programa
 - 155 Sumário das características de edição
- 156 Apêndice D: Mensagens de erro
- 160 Apêndice E: Localizações de memórias usadas pelo DOS e pelo Elppasoft
 - 161 Valor do HIMEN fixado pelo carregamento do DOS
- 163 Apêndice F: O velho monitor ROM
 - 163 Usando o velho monitor ROM
 - 164 Recuperando-se de RESETs acidentais

APÊNDICE A: SUMÁRIO DOS COMANDOS

(Este apêndice contém ambos os comandos, Elppasoft e DOS)

Apresentamos a seguir um sumário dos comandos que podem ser usados na linguagem de programação Elppasoft.

TECLAS COM SETA → ←

As teclas marcadas com setas apontando para a direita e para a esquerda, são usadas para editar programas em Elppasoft. A tecla seta para a direita, move o cursor para a direita; ou seja, cada caracter sobre o qual ela passa na tela, é interpretado como se tivesse sido escrito. A tecla seta para a esquerda, move o cursor para a esquerda; à medida que ela se move, um caracter é apagado da linha do programa que você está digitando.

CALL -151

Causa o aparecimento do asterisco, indicando que o Elppa está agora respondendo em sua linguagem nativa, chamada linguagem de máquina. Se você não é um programador experiente, provavelmente não precisará usar este comando.

CATALOG

Coloca na tela uma lista de todos os arquivos contidos no disco que você inseriu na unidade de disco especificada. O tipo de arquivo e o número de setores ocupados pelo mesmo, são indicados à esquerda do nome do arquivo. Os tipos de arquivos são:

- I O arquivo é um programa Integer BASIC.
- A O arquivo é um programa em Elppasoft.
- T O arquivo consiste em um Texto: foi criado por um comando WRITE.
- B O arquivo é uma imagem bit-a-bit de uma parte da memória do Elppa.

O comando CATALOG, é um comando do DOS (Sistema Operacional do Disco), não um comando do Elppasoft.

CLEAR

Introduz zero em todas as variáveis e torna nulas todas as sequências.

COLOR = 12

Introduz a cor para traçar no modo de gráficos de baixa resolução. No exemplo, a cor colocada é verde. O comando GR coloca a cor em zero. Os nomes das cores e seus respectivos números são:

Ø preto	5 cinza	10 cinza
1 magenta	6 azul médio	11 rosa
2 azul escuro	7 azul claro	12 verde
3 púrpura	8 marrom	13 amarelo
4 verde escuro	9 laranja	14 água (azul piscina)
		15 branco

CONT

Se a execução do programa for interrompida por STOP, END ou CTRL C, o comando CONT faz com que a execução recomece a partir da próxima INSTRUÇÃO (como GOSUB) -- NÃO do próximo número de linha. O comando CONT não pode ser usado se você tem

- uma linha de programa modificada, adicionada ou apagada, ou,
- obtido uma mensagem de erro e a execução do programa foi interrompida.

CTRL C

Pode ser usado para interromper um programa em execução ou uma listagem. Pode também ser usado para interromper um INPUT se este é o primeiro caracter inserido. O INPUT não é interrompido, até que a tecla RETURN seja pressionada.

CTRL X

Diz ao Elppa para ignorar a linha que está sendo digitada sem apagar qualquer linha anterior do mesmo número de linha. Uma barra (\) é mostrada no fim da linha a ser ignorada.

DEL 23,56

Remove do programa, as linhas compreendidas entre os dois números. No exemplo, as linhas 23 a 56 serão apagadas (DElete) do programa. Para apagar (DEL) uma única linha, digamos a linha 35Ø, use a forma DEL 35Ø, 35Ø ou simplesmente digite o número da linha e então pressione a tecla RETURN.

DIM NOME\$(5Ø)

Quando uma declaração DIM é executada, ela cria um espaço para a matriz especificada com índices variando de Ø até o índice dado.

No exemplo, NOME\$ (50) reservará espaço para 51 seqüências de qualquer tamanho. Se um elemento de matriz é usado em um programa antes de ser dimensionado, um índice máximo de 10 é reservado para cada dimensão no índice do elemento. Elementos de matriz são zerados quando RUN ou CLEAR é executado.

END

Provoca o cancelamento da execução do programa, e retorna o controle ao usuário. Nenhuma mensagem é impressa.

ESC I OU ESC J OU ESC K OU ESC M

A tecla ESC (abreviatura de ESCape) pode ser usada juntamente com as teclas de letras I, ou J ou K ou M, para mover o cursor sem afetar os caracteres sobre os quais ele passou. Para mover o cursor, PRIMEIRO pressione e então SOLTE a tecla ESC para entrar em modo de edição. Então, pressione a tecla da letra apropriada uma vez para cada movimento na direção desejada. A tecla REPT, pode ser usada para acelerar o movimento, pressionando a tecla da letra apropriada e então a tecla REPT, ENQUANTO segura a tecla da letra.

<u>Comando</u>	<u>move o cursor um espaço para</u>
ESC I	cima
ESC J	esquerda
ESC K	direita
ESC M	baixo

FLASH

Faz com que o conteúdo do vídeo fique piscando, ou seja, alternando entre caracteres brancos com fundo preto e caracteres pretos em fundo branco. Use NORMAL para retornar à tela os caracteres brancos em fundo preto.

```
FOR W = 1 TO 20 ... NEXT W
FOR Q = 2 TO -3 STEP -2 ... NEXT Q
FOR Z = 5 TO 4 STEP 3 ... NEXT Z
```

Permite que você escreva um "loop" para realizar quaisquer instruções entre o comando FOR (o topo do loop) e o comando NEXT (a base do loop) um especificado número de vezes. No primeiro exemplo, a variável W conta quantas vezes deve executar as instruções; as instruções dentro do loop serão executadas para W igual a 1, 2, 3, ... 20, então o loop termina (com W = 21) e a instrução após NEXT W é executada. O segundo exemplo ilustra como indicar que o tamanho do passo (STEP) que você conta, seja diferente de 1. A verificação ocorre no FINAL do loop, então no

terceiro exemplo, as instruções no interior do loop são executadas uma vez.

GOSUB 250

Faz o programa pular para a linha indicada (250, no exemplo). Quando uma declaração RETURN é executada, o programa volta à declaração imediatamente seguinte após o último GOSUB executado.

GOTO 250

Faz o programa pular para a linha indicada (250, no exemplo).

GR

Aciona o modo GRáfico de baixa resolução (40 por 40) para a tela do TV, deixando quatro linhas na base para texto. A tela é limpa, o cursor é movido para a área de texto, e COLOR é assinalado no valor 0 (preto).

HCOLOR = 4

Determina a cor dos gráficos de alta resolução de acordo com a cor especificada em HCOLOR. Os nomes das cores e seus valores são:

0 preto 1	4 preto 2
1 verde	5 laranja
2 violeta	6 azul
3 branco 1	7 branco 2

HGR

Aciona o modo gráfico de alta resolução (280 por 160) para a tela, deixando quatro linhas para texto na base. A tela é limpa, e a página 1 da memória é mostrada. Nem HCOLOR nem a memória do texto da tela são afetados quando HGR é executado. O cursor não é movido para a área de texto.

HGR2

Aciona o modo gráfico de alta resolução para toda a tela (280 por 192). A tela é limpa, e a página 2 da memória é mostrada. A memória do texto da tela não é afetada.

HLIN 10,20 AT 30

Usado para desenhar linhas horizontais em modo gráfico de baixa resolução, usando a última cor especificada em COLOR. A origem ($x = 0$ e $y = 0$) para o sistema, é o ponto no canto superior esquerdo da tela. No exemplo, a linha é desenhada a partir de $x = 10$ até $x = 20$ em $y = 30$. Uma outra forma de dizer isto: a linha é desenhada do ponto (10, 30) até o ponto (20, 30).

HOME

Move o cursor para o canto superior esquerdo da tela, dentro da janela de texto, e limpa todo o texto da janela.

```
HFPLOT 10,20
HFPLOT 30,40 TO 50,60
HFPLOT TO 70,80
```

Traça pontos e linhas em modo gráfico de alta resolução, usando o último valor especificado em HCOLOR. A origem é o ponto no canto superior esquerdo da tela ($x = 0$, $y = 0$). O primeiro exemplo traça um ponto de alta resolução em $x = 10$, $y = 20$. O segundo exemplo traça uma linha de alta resolução a partir do ponto $x = 30$, $y = 40$ até o ponto $x = 50$, $y = 60$. O terceiro exemplo traça uma linha a partir do último ponto traçado até o ponto $x = 70$, $y = 80$, usando a cor do ÚLTIMO PONTO TRAÇADO, que não é necessariamente o último valor de HCOLOR.

HTAB 23

Move o cursor ou para a esquerda ou para a direita, até a coluna especificada (1 até 40) na tela. No exemplo, o cursor será posicionado na coluna 23.

```
IF IDADE < 18 THEN A = 0: B = 1: C = 2
IF ANS# = "SIM" THEN GOTO 100
IF N < MAX THEN GOTO 25
```

Se a expressão que sucede o comando IF for verdadeira (i.e. não-zero), então as instruções que sucedem THEN, na mesma linha, serão executadas. Caso contrário, quaisquer instruções que sucedem THEN serão ignoradas, e a execução passa para a instrução no próximo número de linha do programa. Expressões sequenciais são avaliadas de acordo com a ordem alfabética.

```
INPUT A
INPUT "DIGITE IDADE DEPOIS UMA VIRGULA DEPOIS NOME. "; B, C$
```

No primeiro exemplo, INPUT imprime um ponto de interrogação e espera que o usuário digite um número, que será atribuído à variável numérica A. No segundo exemplo, INPUT imprime a sequência opcional exatamente como o mostrado, e então espera que o usuário digite um número (que será atribuído à variável B) e então uma vírgula, e então uma entrada sequencial (que será atribuída à variável sequencial C\$). Entradas múltiplas num comando INPUT podem ser separadas por vírgulas ou RETURNS.

```
INPUT (NUM)
```

Fornece o maior inteiro menor ou igual ao argumento dado. No exemplo, se NUM é 2.389, então 2 será fornecido; se NUM é -45.123345, então -46 será fornecido.

```
INVERSE
```

Coloca o vídeo no modo em que a SAÍDA do computador é impressa com letras pretas em fundo branco. Use o comando NORMAL para retornar à impressão com letras brancas em fundo preto.

```
LEFT$("ELPPASOFT",5)
```

Fornece o número especificado de caracteres mais à esquerda da sequência. No exemplo, ELPPA (os 5 caracteres mais à esquerda) serão fornecidos.

```
SETA PARA A ESQUERDA
```

Veja "Teclas com Setas".

```
LEN("UMA BANANA POR DIA")
LEN(B$)
```

Fornece o número de caracteres em uma sequência, entre 0 e 255. No primeiro exemplo, 18 serão fornecidos.

```
A = 23.567
A$ = "DELICIOSO"
```

Ao nome da variável à esquerda de = é atribuído o valor da sequência ou expressão à direita de =.

```
LIST
LIST 200,3000
LIST 200-3000
```

O primeiro exemplo faz com que o programa inteiro, seja mostrado na tela do TV; o segundo e o terceiro exemplos, fazem com que as linhas 200 a 3000 sejam mostradas. Para listar do início do programa até a linha 200, use LIST 200; para listar da linha 200 até o fim do programa, use LIST 200-. A listagem pode ser interrompida por um CTRL C, e o comando CONT não pode ser usado. Para parar o programa temporariamente, em algum ponto na listagem use CTRL S. Use CTRL S outra vez, para recomeçar a listagem.

LOAD

Lê um programa Elppasoft da fita cassette para inserir na memória do computador. Nenhum indicador] é dado: o usuário deve rebobinar a fita e pressionar "play" no gravador, antes de carregar (LOAD). Um bip é ouvido, quando a informação é encontrada na fita, que está sendo carregada (LOAD). Quando o carregamento é satisfatoriamente completado, um segundo bip é ouvido e o indicador] do Elppasoft retornará. Somente o RESET pode interromper um LOAD.

LOAD AMOSTRAGEM

Procura encontrar um arquivo de programa com o nome AMOSTRAGEM no disco, na unidade de disco especificada. Se o programa é encontrado, ele será carregado na memória do Elppa. O LOAD apaga qualquer programa no Elppa, antes de colocar o novo programa na memória. Este comando, quando seguido pelo nome de um arquivo, é um comando DOS.

```
MID#("UMA BANANA POR DIA",5)
MID#(DIAS,5,12)
```

Retorna a subsequência especificada. No primeiro exemplo, do quinto até o último caracter da sequência será retornado: BANANA POR DIA. No segundo exemplo, os doze caracteres começando pelo quinto caracter na sequência, serão retornados: BANANA POR D

NEW

Apaga o programa corrente na memória e todas as variáveis.

NEXT

Veja a explanação sobre FOR ... TO ... STEP.

NORMAL

Coloca o vídeo no modo usual, ou seja, letras brancas em um fundo preto, tanto para a entrada como para a saída.

NOTRACE

Desliga o modo TRACE. Veja TRACE.

PDL(1)

Retorna o valor corrente, um número de 0 a 255, indicado na raquete (paddle) do controle de jogo. Raquetes de jogos com números de 0 a 3 são válidos.

FLOT 10,20

No modo gráfico de baixa resolução, coloca um ponto no lugar especificado. No exemplo, o ponto estará em $x = 10$, $y = 20$. A cor do ponto é determinada pelo último valor de COLOR, que é 0 (preto) se não for previamente especificada.

PRINT

```
PRINT A$; "X = "; X
```

O primeiro exemplo, cria uma linha em branco e fornece RETURN para ser executado na tela. Itens de uma lista a ser impressa, devem ser separados por vírgulas se cada um for mostrado separadamente em um campo tabulado. Os itens devem ser separados por ponto e vírgulas se eles forem impressos um após o outro, sem deixar espaços em branco. Se A\$ contém "CORE" e X é 3, o segundo exemplo fornecerá a impressão de

```
COREX = 3
```

REM ISTO E UM COMENTARIO

Permite que textos sejam inseridos em um programa, como comentários .

REPT

Se você segura a tecla de repetição, denominada REPT, enquanto pressiona qualquer outro caracter, este será repetido.

RETURN

Retorna à declaração imediatamente seguinte ao último GOSUB executado.

```
RIGHT$( "ELPPASOFT", 4)  
RIGHT$( S$, 2)
```

Retorna o número especificado dos caracteres mais à direita da sequência. No primeiro exemplo, SOFT (os 4 caracteres mais à direita) serão retornados.

SETA PARA A DIREITA

Veja "Teclas com Setas".

RND(5)

Retorna um número real randômico (aleatório) maior ou igual a 0 e menor do que 1. RND(0) retorna o último número randômico gerado. Cada argumento NEGATIVO gera um número randômico particular, que é o mesmo toda vez que RND é usado com aquele argumento, e SUBSEQUENTES comandos RND com argumentos positivos seguirão sempre uma sequência particular e repetitiva. Cada vez que RND é usado com qualquer argumento positivo, um NOVO número randômico de 0 a 1 é gerado, a menos que ele seja parte de uma sequência de números randômicos iniciada por um argumento negativo.

RUN 500

Limpa todas as variáveis, apontadores e pilhas, e começa a execução no número de linha indicado (500, no exemplo). Se nenhum número de linha é especificado, a execução começa no menor número de linha no programa.

RUN ANTENA

Carrega o arquivo chamado ANTENA da unidade de disco especificada e então roda o programa carregado. Quando seguido por um nome de arquivo, RUN é um comando do DOS, e não um comando do Elppasoft.

SAVE

Armazena na fita cassette o programa corrente na memória. Nenhum sinal é fornecido. O usuário deve pressionar "record" e "play" no gravador, antes de executar o comando SAVE. O SAVE não verifica se os botões apropriados foram pressionados. O computador emite um "bip" no início e no fim de uma gravação.

SAVE ENDERECOS

Guarda (SAVE) o arquivo corrente na memória. Se nenhum arquivo chamado ENDEREÇOS é encontrado no disco na unidade de disco especificada, um arquivo é criado naquele disco e o programa corrente na memória é armazenado sob o nome do arquivo dado. Se o disco contém um arquivo com o nome do arquivo especificado, e na mesma linguagem, o conteúdo do arquivo original é perdido e o programa corrente é guardado (SAVE) em seu lugar. Nenhum aviso é fornecido. O comando SAVE, quando seguido por um nome de arquivo, é um comando do DOS.

STR*(12.45)

Retorna uma sequência que representa o valor do argumento. No exemplo, a sequência "12.45" é retornada.

TAB(23)

Deve ser usado em uma declaração PRINT; o argumento deve estar entre 0 e 255 e fechado entre parênteses. Para argumentos de 1 a 255, se o argumento é maior do que o valor da posição atual do cursor, então o comando TAB move o cursor para a posição de impressão especificada contando da margem esquerda da linha atual do cursor. Se o argumento é menor do que o valor da atual posição do cursor, então este não é deslocado. TAB(0) coloca o cursor na posição 256.

TEXT

Faz com que a tela fique no modo de texto normal, não gráfico, com 40 caracteres por linha e 24 linhas. Também recoloca a janela do texto em toda a tela.

TRACE

Quando executado, causa o aparecimento do número de linha de cada declaração. TRACE não é desligado por RUN, CLEAR, NEW, DEL ou RESET. O único comando que desliga TRACE é NOTRACE.

VAL("-3.7E4E5FFA")

Interpreta uma sequência, até encontrar o primeiro caractere não-numérico, como um número real ou inteiro, e retorna o valor daquele número. Se nenhum número ocorre antes do primeiro caractere não numérico, o valor 0 é devolvido. No exemplo, o resultado será -37000.

VLIN 10,20 AT 30

No modo gráfico de baixa-resolução, desenha uma linha vertical na cor indicada na última declaração COLOR. A linha é desenhada na coluna indicada pelo terceiro argumento. No exemplo, a linha é desenhada de $y = 10$ até $y = 20$ em $x = 30$.

VTAB(15)

Move o cursor na tela para a linha especificada pelo argumento. A primeira linha é a linha 1 e a última linha é a linha 24. VTAB moverá o cursor para cima ou para baixo e não para a esquerda ou direita.

APÊNDICE B: PALAVRAS RESERVADAS NA LINGUAGEM ELPPASOFT

A lista a seguir contém TODAS as palavras reservadas na linguagem Elppasoft. No entanto, a maioria destas palavras não aparecem neste manual, a lista é útil como um guia para nomear variáveis.

&	GET	NEW	SAVE
	GOSUB	NEXT	SCALE=
ABS	GOTO	NORMAL	SCRN (
AND	GR	NOT	SGN
ASC		NOTRAC.	SHLOAD
AT	HCOLOR=		SIN
ATN	HGR	ON	SPC (
	HGR2	ONERR	SPEED=
CALL	HIMEM:	OR	SQR
CHR\$	HLIN		STEP
CLEAR	HOME	PDL	STOP
COLOR=	HPlot	PEEK	STORE
CONT	HTAB	PLOT	STR\$
COS		POKE	
	IF	POP	TAB (
DATA	IN #	POS	TAN
DEF	INPUT	PRINT	TEXT
DEL	INT	PR #	THEN
DIM	INVERSE		TO
DRAW		READ	TRACE
	LEFT\$	RECALL	
END	LEN	REM	USR
EXP	LET	RESTORE	
	LIST	RESUME	VAL
FLASH	LOAD	RETURN	VLIN
FN	LOG	RIGHT\$	VTAB
FOR	LOMEM:	RND	
FRE		ROT=	WAIT
	MID\$	RUN	
			XPLOT
			XDRAW

O símbolo & é projetado somente para o uso interno do computador; ele NÃO é um comando particular do Elppasoft. Este símbolo, quando executado como uma instrução, causa um salto incondicional para a posição \$3F5.

XPLOT é uma palavra reservada que não corresponde a um comando corrente do Elppasoft.

Algumas palavras reservadas são reconhecidas pelo Elppasoft somente em certos contextos.

COLOR, HCOLOR, SCALE, SPEED, e ROT
são analisados como palavras reservadas somente se o próximo carácter não-espaço é o sinal de substituição, =. Isto é de pouca utilidade no caso de COLOR e HCOLOR, pois como ela contém a palavra reservada OR, não podem ser usadas como nomes de variáveis de qualquer forma.

SCRN, SPC e TAB
são analisados como palavras reservadas somente se o próximo carácter não-espaço é um parêntese esquerdo, (.

HIMEM: deve ser seguido dos dois pontos (:) para ser analisado como uma palavra reservada.

LOMEM: também requer os dois pontos (:) se é para ser analisado como uma palavra reservada.

ATN é analisado como uma palavra reservada somente se não houver espaço entre o T e o N. Se houver um espaço entre o T e o N, a palavra reservada AT é analisada, ao invés de ATN.

TO é analisado como uma palavra reservada A MENOS QUE seja precedido por um A e houver um espaço entre o T e o O. Se ocorrer um espaço entre o T e o O, a palavra reservada AT é analisada ao invés de TO.

Algumas vezes, os parênteses podem ser usados para contornar as palavras reservadas:

```
1000 FOR A = LOFT OR CAT TO 15
Lista como 1000 FOR A = LOF TO RC AT TO 15
           mas 1000 FOR A = (LOFT) OR (CAT) TO 15
Lista como 1000 FOR A = (LOFT) OR (C AT) TO 15
```

APÊNDICE C: CARACTERÍSTICAS DE EDIÇÃO

As Teclas Setas Esquerda e Direita ← →

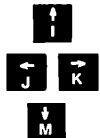
A tecla seta esquerda, também chamada tecla de retrocesso, move o cursor um espaço para trás (esquerda), apagando o caracter sobre o qual ele passa. Se você não pressionou a tecla **RETURN**, no final da linha que você digitou, a tecla de retrocesso somente afeta os caracteres naquela linha.

Pressionar a tecla seta direita, também conhecida como a tecla de avanço, faz o cursor mover-se para frente (direita), redigitando o caracter sobre o qual ele passa. Se você redigita uma linha com a tecla de avanço, e então pressiona a tecla **RETURN**, o Elppa se comporta como se você tivesse digitado a linha novamente.

Você pode fazer com que o cursor se mova mais rapidamente pressionando a tecla **REPT**, enquanto pressiona uma das teclas com seta.

Movimentando o Cursor

As teclas **ESC**, **I**, **J**, **K**, e **M** são usadas para mover o cursor sem afetar qualquer um dos caracteres na tela. Imagine setas desenhadas nas teclas das letras como mostra a ilustração:



Pressionando a tecla **ESC**, você entra no modo de edição. Uma vez que você está no modo de edição, pressionando uma das teclas com as letras anteriormente mencionadas, fará com que o cursor se mova um caracter na direção da seta correspondente. Você pode usar estas teclas para mover o cursor para qualquer lugar da tela.

Para movimentos mais rápidos do cursor, pressione uma destas teclas e então pressione a tecla **REPT**. O cursor irá circular através da tela, enquanto ambas as teclas estiverem pressionadas. Se o cursor atinge o topo da tela, ele parará. Se o cursor atinge a base da tela, ele parará, e as linhas mostradas na tela serão deslocadas para cima. Se ele atingir o canto direito, o cursor irá desaparecer e reaparecer no canto esquerdo, mas na próxima linha. No canto esquerdo, ele reaparecerá no direito, uma linha acima. Para retornar ao modo normal, pressione a barra de espaço uma vez.

Apagando Linhas de Programa

Um método fácil de apagar uma linha de programa é simplesmente digitar o número da linha que você deseja apagar e a seguir pressionar a tecla **RETURN**. Se você tem mais de uma linha para apagar, você deve utilizar o comando DEL. Para apagar, por exemplo, as linhas de 100 a 200, você digitaria

```
DEL 100,200
```

Todas as linhas do programa de 100 a 200 inclusive, serão apagadas.

Digitando



a linha que você está digitando, será apagada. Isto é útil, quando você percebe que cometeu um erro antes de pressionar a tecla **RETURN**.

Limpendo a Tela

Os comandos a seguir, afetam somente o que você vê na tela, não o que está armazenado na memória do Elppa. Pressionando a tecla **ESC** uma vez, coloca você em modo de edição.

Para limpar a tela, pressione a tecla **ESC** seguido por um sinal "@".



O cursor aparecerá no canto superior esquerdo da tela sem o indicador] : O indicador] aparecerá quando você pressionar a tecla **RETURN**.

Se você já se encontra em modo de edição, pode limpar a tela simplesmente pressionando o sinal "@". O Elppa retornará ao modo normal.

O comando HOME também é usado para limpar a tela. Simplesmente digite

```
HOME
```

e o cursor aparecerá no canto superior esquerdo da tela.

É possível, também, limpar somente pedaços da tela. Para limpar de um ponto da tela até o fim da mesma, entre em modo de edição pressionando a tecla **ESC**. Então, use as características de movimento do cursor para movê-lo para o primeiro caracter que você deseja limpar. Pressione a tecla **F**, e todos os caracteres daquele ponto até o fim da tela serão eliminados. Para limpar caracteres até o fim de uma linha, você deve primeiro estar em modo de edição. Então, mova o cursor até o primeiro caracter a ser eliminado, e pressione a tecla **E**. Em ambos os casos, o Elppa retornará ao modo normal após o comando ser executado.

Sumario das Características de Edição

Entrar em modo de edição	Pressione ESC
Sair do modo de edição	Pressione a barra de espaço
Mover o cursor	Pressione I , J , K , ou M
Apagar um caracter	Pressione ←
Redigitar um caracter	Pressione →
Limpar do cursor até o fim da linha	Pressione ESC , depois E
Limpar do cursor até o fim da tela	Pressione ESC , depois F
Limpar toda a tela	Pressione ESC , depois SHIFT e Ⓟ
Parar a listagem	Pressione CTRL e S
Continuar a listagem	Pressione CTRL e S

APÊNDICE D: MENSAGENS DE ERRO

Todas as mensagens de erro que podem ser geradas na linguagem Elppasoft, estão listadas e descritas neste Apêndice.

Após a ocorrência de um erro, o Elppasoft retorna ao nível de comando indicado pelo indicador] e cursor.

Os valores das variáveis e o texto do programa permanecem intactos, mas o programa não pode continuar (CONT) e todos os contadores de loop GOSUB e FOR, são zerados.

Quando ocorre um erro em uma declaração de execução imediata, nenhum número de linha é impresso.

Formato das mensagens de erro:

Declaração de execução imediata	ERRO:XX
Declaração de execução diferida	ERRO:XX EM YY

Em ambos os casos anteriores, "XX" é o nome do erro específico. "YY" é o número de linha da declaração de execução diferida, onde ocorreu o erro. Erros em uma declaração de execução diferida não são detectados até que a mesma seja executada.

São os seguintes, os possíveis erros e seus significados:

ERRO:NAO PODE CONT

Tentativa de continuar um programa não existente, ou após a ocorrência de um erro, ou depois que uma linha foi apagada de um programa ou acrescentada ao mesmo.

ERRO:DIVISAO POR ZERO

Dividir por zero é um erro.

ERRO:FORMULA MUITO COMPLEXA

Mais do que duas declarações da forma IF "XX" THEN foram executadas.

ERRO:DIRETO ILEGAL

Você não pode usar uma declaração INPUT, DEF FN, GET ou DATA como um comando de execução imediata.

ERRO:QUANTID ILEGAL

O parâmetro utilizado em uma função matemática ou em uma sequência, estava fora do limite permitido. Os erros QUANTID ILEGAL podem ocorrer devido a:

- a) um índice negativo em uma matriz (por exemplo, $A(-1) = \emptyset$)
- b) uso de um LOG com um argumento negativo ou zero
- c) uso de um SQR com um argumento negativo
- d) $A \wedge B$ com A negativo e B não inteiro
- e) uso de MID\$, LEFT\$, RIGHT\$, WAIT, PEEK, POKE, TAB, SPC, ON ... GOTO, ou qualquer das funções gráficas com um argumento impróprio.

ERRO:NEXT SEM FOR

A variável em uma declaração NEXT não correspondeu à variável em uma declaração FOR que ainda era válida, ou um NEXT sem nome correspondeu a qualquer FOR que ainda era válido.

ERRO:SEM DADOS

Uma declaração READ foi executada mas todas as declarações DATA, no programa, já tinham sido lidas. O programa tentou ler muitos dados ou dados insuficientes foram incluídos no programa.

ERRO:FALTA MEMORIA

Quaisquer dos seguintes casos podem causar este erro: programa muito extenso; variáveis em excesso, loops internos com FOR, contendo mais do que 10 níveis; GOSUBs internos com mais de 24 níveis; uma expressão muito complicada; parênteses internos com mais de 36 níveis; tentativa de fixar LOMEM: muito alto; tentativa de fixar LOMEM: bem abaixo do seu valor atual; tentativa de fixar HIMEM: muito baixo.

ERRO:OVERFLOW

O resultado de um cálculo foi muito grande para ser representado no formato usado para números no Elppasoft. Se ocorre um underflow (quando o resultado de um cálculo for muito pequeno), zero é dado como resultado e a execução continua sem a impressão de qualquer mensagem de erro.

ERRO:MATRIZ REDIM

Após uma matriz ser dimensionada, uma outra declaração de dimensionamento para a mesma matriz foi encontrada. Este erro frequentemente ocorre se uma matriz foi usada sem ter sido dimensionada, isto é, o sistema considerou DIM (1Ø) e logo depois apareceu o dimensionamento desta matriz. Esta mensagem de erro pode ser útil se você desejar descobrir que linha de programa em uma certa matriz foi dimensionada: apenas introduza uma declaração DIM para cada matriz na primeira linha, rode (RUN) o programa e o Elppasoft lhe dirá onde está a declaração DIM original.

ERRO:RETURN SEM GOSUB

Uma declaração RETURN foi encontrada sem que uma correspondente declaração GOSUB esteja sendo executada.

ERRO:STRING LONGO

Foi feita uma tentativa de concatenação para criar uma sequência (STRING) com mais de 255 caracteres.

ERRO:INDICE ILEGAL

Foi feita uma tentativa de se referir a um elemento de uma matriz que está fora das dimensões da mesma. Este erro pode ocorrer se o número errado de dimensões são usadas em uma matriz de referência; por exemplo, LET A (11) = Z quando A foi dimensionado usando DIM A (2).

ERRO:SINTAXE

Falta de parênteses em uma expressão, caracter ilegal em uma linha, pontuação incorreta, etc.

ERRO:VAR IMPROPRIA

O lado esquerdo de uma declaração de atribuição era uma variável numérica e o lado direito era uma sequência, ou vice-versa (por exemplo, B = 5+2); ou a uma função que esperava uma sequência como argumento foi dado um argumento numérico ou vice-versa.

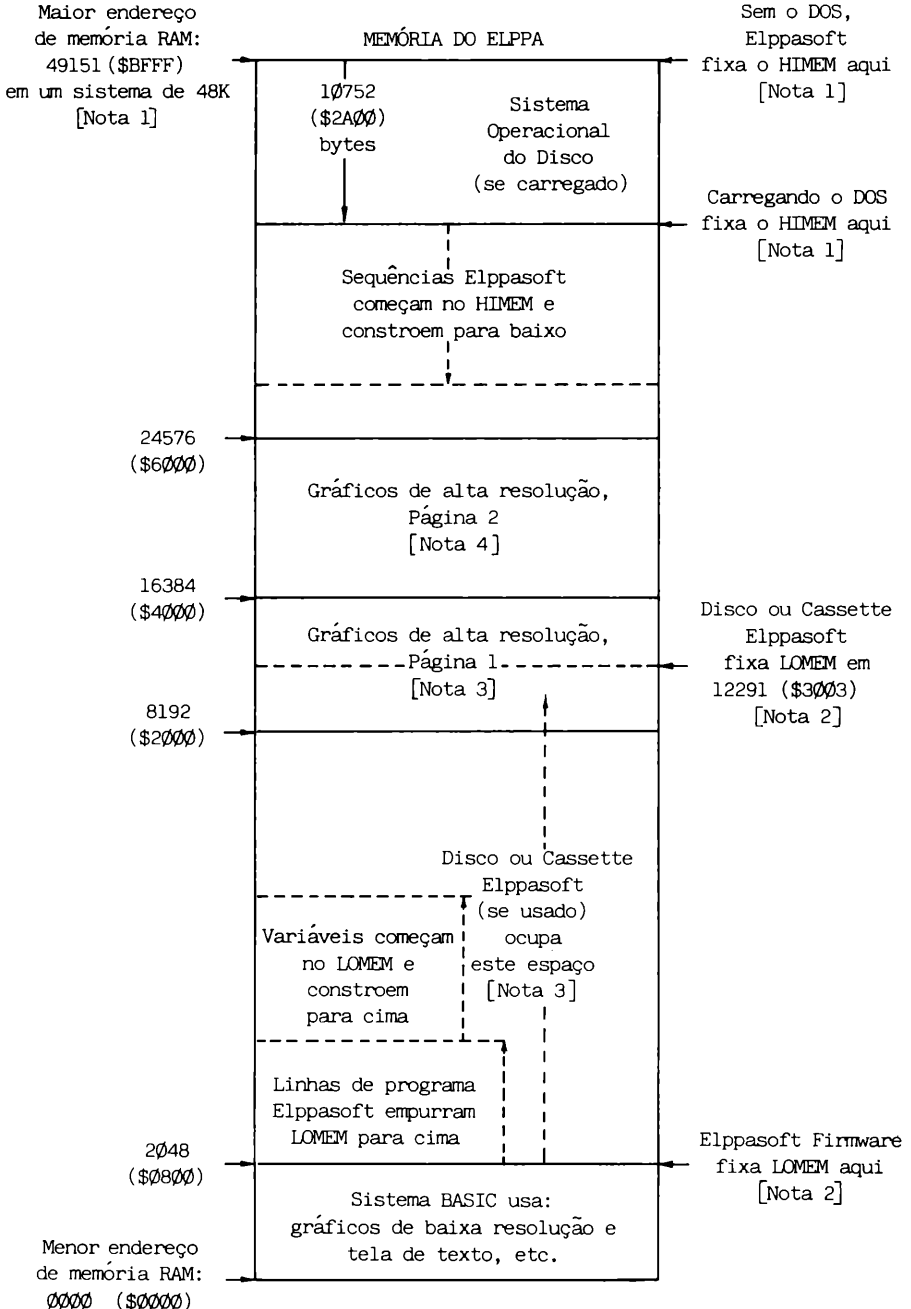
ERRO:LINHA NAO EXISTE

Foi feita uma tentativa de usar GOTO, GOSUB ou THEN para uma declaração, cujo número de linha não existe.

ERRO:FUNCAO INDEFINIDA

Foi feita uma referência a uma função especificada pelo usuário, que nunca foi definida.

APÊNDICE E : LOCALIZAÇÕES DE MEMÓRIAS USADAS PELO DOS E PELO ELPPASOFT



Nota 1. HIMEM é o endereço da localização da maior memória disponível para um programa Elppasoft. Se o seu sistema está em Elppasoft, o valor de HIMEM pode ser encontrado (primeiro byte mais baixo, então byte mais alto) em localizações decimais 115 e 116 (\$73-\$74, hexadecimal). Para ver o valor corrente de HIMEM, digite

```
PRINT PEEK (115) + PEEK (116) * 256
```

Consulte a seguinte tabela para o valor de HIMEM fixado pelo carregamento do DOS, para sistemas com várias quantias de memória. Aumentar MAXFILES moverá HIMEM para baixo 595 bytes adicionais para cada arquivo de memória intermediária (file buffer) acrescentado.

Valor do HIMEM Fixado pelo Carregamento do DOS

Tamanho do Sistema	Maior endereço de RAM		HIMEM: fixado pelo DOS carregado	
	Decimal	Hexadecimal	Decimal	Hexadecimal
16K	16383	\$3FFF	5632	\$1600
20K	20479	\$4FFF	9728	\$2600
24K	24575	\$5FFF	13824	\$3600
32K	32767	\$7FFF	22016	\$5600
36K	36863	\$8FFF	26112	\$6600
48K	49151	\$BFFF	38400	\$9600

O equivalente negativo de qualquer endereço decimal positivo n é (n - 65536).

Nota 2. LOMEM é o endereço da menor localização de memória disponível para um programa Elppasoft. Em Elppasoft, o valor de LOMEM pode ser encontrado (primeiro byte menor, então byte maior) em localizações decimais 105 e 106 (\$69-\$6A, hexadecimal). Para ver o valor corrente de LOMEM, digite

```
PRINT PEEK(105) + PEEK(106) * 256
```

O Elppasoft automaticamente fixa LOMEM após a última linha do programa corrente armazenado, e a primeira variável começa em LOMEM.

Nota 3. Usar gráficos de alta resolução Página 1 (com HGR) apaga os conteúdos das localizações da memória de 8192 até 16383.

Se você está usando disco ou cassette Elppasoft, uma tentativa de usar gráficos de alta resolução Página 1, sempre apagará parte do Elppasoft. Com o disco ou cassette Elppasoft, você somente pode usar gráficos de alta resolução Página 2. Contudo, veja a Nota 4.

Nota 4. Usar gráficos de alta resolução Página 2 (com HGR2) apaga os conteúdos das localizações de memória de 16384 até 24575. Se você está usando o DOS, uma tentativa de usar gráficos de alta resolução Página 2 pode apagar parte do DOS a menos que o DOS fixe HIMEM para um valor maior do que 24575. Isto significa que você não pode usar o DOS e gráficos de alta resolução Página 2 ao mesmo tempo, a menos que o seu sistema contenha no mínimo 36K de memória.

APÊNDICE F: O VELHO MONITOR ROM

A maior parte deste manual assume que o seu Elppa contém o ROM de partida automática (Autostart ROM), que instantaneamente permite que o seu Elppa faça o processamento na linguagem BASIC quando você liga o interruptor de força.

Quando você liga o seu Elppa, se ele limpa a tela e imprime

ELPPA[]

no topo (e carrega o seu disco, se você tem um), então o seu Elppa contém um usual ROM de partida automática.



A pedido do usuário poderá ser fornecida opcionalmente uma ROM programada com o velho monitor.

Usando o Velho Monitor ROM

Cada vez que você liga o seu Elppa, verá um asterisco (*) indicador no canto inferior esquerdo da tela seguido pelo cursor. Isto indica que você está no programa Monitor, que é usado por programadores muito experientes, quando estão trabalhando em "linguagem de máquina". Para começar a processar em BASIC, após ligar o seu Elppa, você sempre deve seguir esta sequência mágica:

1. Pressione a tecla **RESET**
2. Segure a tecla **CTRL** enquanto digita a letra **B**
3. Pressione a tecla **RETURN**

Em notação simbólica, temos:



Recuperando-se de RESETs Acidentais

Se o seu Elppa contém a ROM de partida automática, pressionar a tecla **RESET** não causa problemas: você permanece na linguagem BASIC que está usando.

Com o velho monitor ROM, contudo, pressionar acidentalmente a tecla **RESET**, lhe jogará repentinamente para o programa Monitor. Você verá o asterisco (*) indicador no canto inferior esquerdo da tela, seguido pelo cursor. Para retornar ao BASIC, sem perder qualquer programa armazenado, você deve se utilizar de uma das seguintes opções:

1. Sem o Disco

Se você está usando o Integer BASIC, você pode retornar ao seu programa, depois de pressionar acidentalmente ou intencionalmente a tecla **RESET**, digitando

CTRL

C **RETURN**

Isto lhe retornará ao BASIC que você estava usando quando pressionou a tecla **RESET**, sem perder o seu programa.

Se você está usando o Elppasoft em cassette, depois de pressionar a tecla **RESET**, digite

O **G** **RETURN**

2. Com o Disco

Se você carregou o DOS, não importa qual a versão do Elppasoft que você está usando, após pressionar a tecla **RESET** deve digitar

3 **D** **O** **G** **RETURN**

Isto lhe retornará ao DOS e ao BASIC que você estava usando quando pressionou a tecla **RESET**, sem perder o seu programa.



Se você está usando disco ou cassette Elppasoft,

CTRL

RESET **C** **RETURN**

tentará estabelecer o Integer BASIC como sua linguagem de programação. Isto pode apagar o Elppasoft e qualquer programa na memória, e o Integer BASIC não inicializado, não funcionará corretamente.



Victor do Brasil Eletrônica Ltda

RUA AIMBERÉ 931, SÃO PAULO SP
BRASIL