

cd-6809

a resolução colorida



CODIMEX

APRESENTAÇÃO DO SEU MICROCOMPUTADOR DOMÉSTICO CD-6809

1. Detalhes:

1.1 - Memória é a área do microcomputador onde são armazenados os programas.

1.2 - Periféricos são unidades extras que podem ser acopladas ao seu computador. Vão aprimorar e facilitar a sua utilização.

O seu CD-6809 tem capacidade para 3 periféricos:

- Um gravador com fita cassete, para gravar seus programas, servindo como uma memória auxiliar.

- Uma impressora para listagem de seus programas.

- E ainda Joystick para você mesmo criar os programas e jogar.

1.3 - O seu CD-6809 lhe proporciona 255 sons diferentes; combinando-os você poderá criar quantos quiser.

Quanto a cor ele as possui em número de 9. Mais adiante explicaremos mais detalhadamente sobre esses recursos.

1.4 - A comunicação entre você e seu computador será feita através da linguagem BASIC. Caso você não a conheça, não se preocupe: continue lendo seu manual e você logo estará programando em BASIC.

2. Como conectar (ao seu micro) os PERIFÉRICOS para sua utilização:

2.1 - CONEXÃO DO CASSETTE (K-7)

O cabo para o gravador Cassete é constituído de um Plugue PHILLIPS numa das extremidades e 3 Plugues próprios para gravador na outra.

Conecte o Plugue PHILLIPS na parte traseira de seu microcomputador, no conector escrito Cassete.

IMPORTANTE:

Atenção ao conectar o cabo:

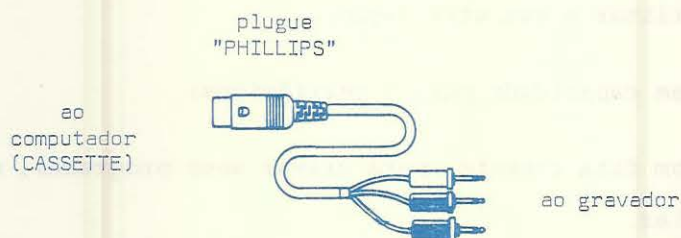
Ele deve ser colocado no conector correto (CASSETTE).

No gravador vão conectados os 3 pinos restantes nas seguinte ordem:

PINO VERMELHO - Conectado à saída de som do gravador (EAR)

PINO PRETO - Conectado à entrada de microfone (MIC)

PINO PRETO PEQUENO - Conectado ao controle remoto do motor (REMOTO)



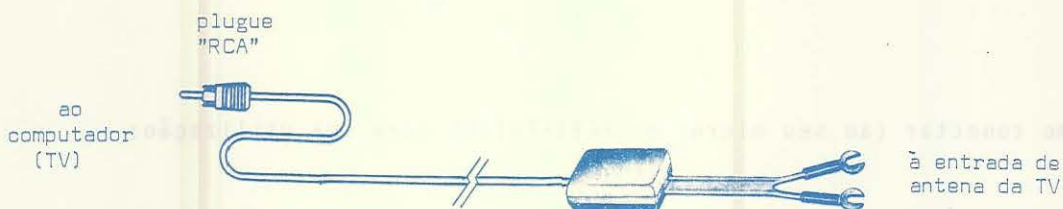
2.2 - CONEXÃO DO VÍDEO:

O cabo para o vídeo é constituído de um Plugue RCA em uma das extremidades, que irá ligado no conector correspondente, denominado "TV", na parte traseira do computador.

A outra extremidade é ligada à entrada de antena de seu televisor.

IMPORTANTE:

Desconecte a antena (Interna ou Externa) de seu televisor antes de ligar o cabo.



2.3 - CONEXÃO DA IMPRESSORA:

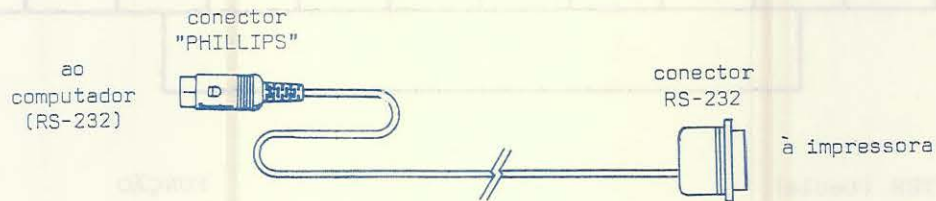
Junto com a Impressora acompanha o seu cabo de conexão. Trata-se de um plugue PHILLIPS a ser conectado ao computador (Conector RS-232) e um plugue RS-232 pa

ra a impressora.

IMPORTANTE:

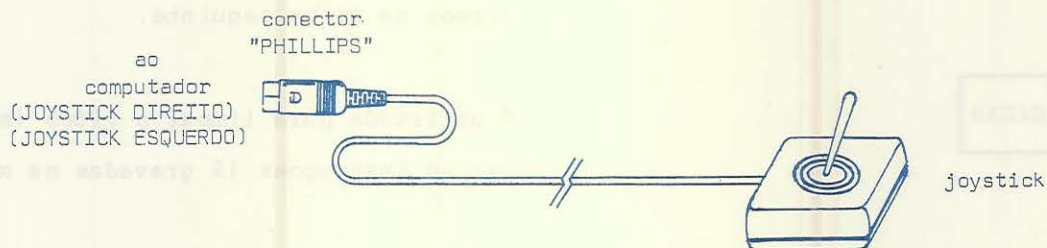
Atenção ao conectar o cabo no computador.

Ele deve ser colocado no conector identificado como RS-232.



2.4 - CONEXÃO DE JOYSTICKS:

O terminal do cabo do Joystick é constituído de um plugue PHILLIPS que deve ser conectado a uma das 2 entradas de "Joystick": JOYSTICK DIREITO ou JOYSTICK ESQUERDO.



3. Apresentação do teclado

O teclado de seu computador é a ligação entre você e o CD-6809. É através dele que você digitará os comandos para que ele os execute diretamente ou na forma de programas.

!	"	#	\$	%	&	'	()	∅	*	=	ESC
↑	Q	W	E	R	T	Y	U	I	O	P	←	→
↓	A	S	D	F	G	H	J	K	L	+;	CLEAR	RETURN
SHIFT	Z	X	C	V	B	N	M	<	>	?/	SHIFT	

3.1 - CHARACTER (tecla)

FUNÇÃO



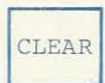
Retrocede o cursor.



Apaga toda linha digitada.



Informa ao computador que se chegou ao fim de uma linha ou comando, posicionando o cursor na linha seguinte.



É utilizada para limpar o vídeo (sem apagar as instruções já gravadas na memória).



Serve para operação de exponenciação.

Ex.:

Digite:

PRINT 2 + 3



8

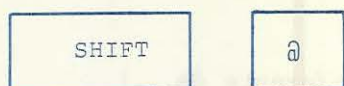
OK



Interrompe a execução de um programa.

TECLA

FUNÇÃO



Faz uma pausa na execução de um programa. O reinício se dará com o acionamento de qualquer tecla.



Troca o formato das letras de maiúscula para minúscula e vice-versa. No vídeo as letras minúsculas aparecem em fundo preto e letra verde.

3.2 - SÍMBOLOS E OPERADORES EM BASIC

3.2.1 - SÍMBOLO

FUNÇÃO

" "

Indica que os dados que estiverem entre "aspas" serão considerados constante alfanumérica para o computador.

:

Separa comandos em uma mesma linha.

()

Informa ao computador que a operação aritmética que estiver entre parênteses tem prioridade para ser executada.

3.2.2 - OPERANDO

FUNÇÃO

+

Adição, combinação de strings.

-

Subtração.

*

Multiplicação.

/

Divisão.

=

Igualdade.

>

Maior que ...

> = ou = >

Maior que ou igual a ...

< = ou = <

Menor que ou igual a ...

OPERANDO	FUNÇÃO
----------	--------

<	Menor que.
---	------------

<> ou ><	Não igual (diferente de...)
----------	-----------------------------

AND	Operação lógica 'E'
-----	---------------------

OR	Operação lógica 'OU'
----	----------------------

NOT	Operação lógica 'NÃO'
-----	-----------------------

Agora, após a apresentação do teclado do seu microcomputador e suas funções, você vai aprender a operá-lo.

Após conectar corretamente o seu microcomputador a uma TV e a um gravador, ligue seu computador. (Opcionalmente, uma impressora pode ser acoplada ao seu micro.)

Bem, vamos começar a operar.

Digite alguma coisa qualquer e após acione a tecla RETURN

Aparecerá no vídeo ? SN ERROR se não for uma instrução aceita pelo computador.

Como vimos anteriormente a comunicação entre o usuário e seu computador é feita através da linguagem BASIC.

Portanto, a partir de agora, vamos aprender essa linguagem para que possamos instruir o computador a executar nossos comandos.

Vamos ao primeiro comando.

Digite exatamente o que aparece abaixo.

PRINT "OLA, EU SOU O SEU COMPUTADOR"

Caso você cometa algum erro ao digitar a linha, basta teclar ← que o último caractere digitado será apagado, possibilitando a correção. O mesmo ocorrerá se você tiver cometido o erro no meio ou início da linha.

Se estiver correto, você terá na tela:

PRINT "OLA, EU SOU O SEU COMPUTADOR"

Aperte a tecla **RETURN** . O seu comando PRINT ____ será executado e em sua tela aparecerá

OLA, EU SOU O SEU COMPUTADOR

OK

Esta linha é a resposta ao seu comando.

STRING E NÚMEROS

Vamos iniciar com dois exemplos.

Digite:

PRINT "2 + 2" **RETURN**

Em resposta você terá:

2 + 2

Seu computador identificou como um STRING.

Agora digite:

PRINT 2 + 2 **RETURN**

Seu computador identificou como NÚMEROS e executou a operação dando como resultado

4

Vamos ver agora como o seu computador identifica um conjunto das duas operações:

Digite:

PRINT "4756 DIVIDIDO POR 3.9 É = A" 4756/3.9 **RETURN**

Se você digitar corretamente, o resultado será:

<u>4756 DIVIDIDO POR 3.9 É = A</u>	<u>1219.48718</u>
STRING	NÚMEROS

ERROS DE DIGITAÇÃO:

Se houver qualquer tipo de erro na digitação do COMANDO teclado pelo operador, o computador não irá aceitar a informação.

Faça um teste. Digite:

PRINT "ALO" RETURN

O computador dará esta resposta no vídeo:

? SN ERROR

Isto indica que houve um erro de SINTAXE. O comando PRINT não está no vocabulário pré-definido do computador, por isso ele não será aceito.

Faça mais um teste. Digite:

PRINT 5/Ø RETURN

O computador dará esta resposta no vídeo

?/Ø ERROR

Foi cometido um erro de LÓGICA.


É impossível dividir algo por ZERO.

Temos portanto exemplificados dois tipos de erro. O LÓGICO e o de SINTAXE.

A "COR" NO SEU MICROCOMPUTADOR

Você tem nove (9) diferentes possibilidades de cor na tela do seu microcomputador.

O comando que possibilita a troca é o CLS.

CLS (3)  O número indica a cor pretendida.

No caso é o azul.

Para obtermos as outras cores vamos a um exemplo.

CLS (7) O número sete indica a cor Violeta.

Para voltar ao verde normal, tecele:

CLS RETURN

Se você usar CLS sem o número código, o computador entenderá que você quer limpar a tela e retornar ao verde.

Cores:

Ø = Preto

3 = Azul

6 = Azul claro

1 = Verde

4 = Vermelho

7 = Violeta

2 = Amarelo

5 = Branco

8 = Laranja

O "SOM" DO SEU MICROCOMPUTADOR

Digite:

SOUND 1,100 RETURN

Se você não ouviu nenhum som, aumente o volume da sua TV (vídeo) e tente outra vez.

O comando SOUND vem acompanhado de dois valores. Estes servem para:

SOUND (X, Y)

X → Especifica o tom desejado.

Assume valores de 1 - 255

Y → Especifica por quanto tempo você deseja a emissão do som.

Assume valores de 1 - 255

Cada unidade corresponde a +/- 7/100 de segundo.

A MEMÓRIA DO SEU MICROCOMPUTADOR

A memória de um computador funciona como área de armazenagem de dados.

Tudo que for teclado sem erros de lógica ou sintaxe será gravado na memória do computador.

Vejamos um exemplo.

Digite:

A = 20 ou LET A = 20

No seu computador CD-6809, o comando LET do BASIC pode ser omitido, como o exemplificado acima.

O comando indicando que a variável 'A' é igual a 20 foi gravado na memória após ser digitada a tecla .

Este valor de 'A' permanecerá até que:

- 1 — novo valor seja dado a 'A'.
- 2 — sejam limpos os dados da memória.

Vamos verificar:

Digite:

PRINT A

Teremos como resposta:

20

ATENÇÃO:

VARIÁVEL é uma letra de A até Z, um conjunto de duas letras ou ainda uma letra e um algarismo, a qual se relaciona um valor.

Quando formos relacionar dados a variáveis, teremos que seguir esta regra.

Dados numéricos:

B = 20

Dados STRING:

C \$ = "ALÔ"

Para STRINGS deve-se usar o "DOLLAR" (\$) após o código da variável, que no caso é 'C'.

Digite:

```
B = 20 RETURN
```

```
C$ = "E A SOMA DE 10 + 10" RETURN
```

Para a variável B teremos o valor numérico 20, e para a variável C\$ teremos o valor STRING 'E A SOMA DE 10 + 10'.

Vamos verificar!

```
PRINT B; C$
```

A resposta deverá ser:

```
20 E A SOMA DE 10 + 10
```

Isto demonstra que na memória do computador temos armazenadas duas variáveis com valores relacionados.

VAMOS FAZER ESTES EXERCÍCIOS PARA TESTE.

```
F = 33.99 RETURN
```

```
XT$ = "MINHA MEMORIA TEM 16 KBYTES DE  
AREA UTILIZAVEL" RETURN
```

CB\$ = 5 Acusará TM ERROR, pois definimos uma variável STRING e lhe atribuímos um valor numérico.

M = "20.3" Esse acusará o mesmo erro anterior, só que aqui definimos uma variável numérica e lhe atribuímos um valor constante.

Entendido?

O computador aceitará:

```
F = 33.99
```

XT\$ = "MINHA MEMORIA TEM 16 KBYTES DE
AREA UTILIZAVEL"

R = F + F

Os restantes estão fora do padrão, portanto não serão gravados na memória.

VAMOS EM FRENTE

Digite:

NEW

O comando NEW serve para apagar tudo o que estiver na memória.

Digite a linha seguinte observando que antes do comando PRINT temos o número 1Ø.

1Ø PRINT "EU SOU O SEU COMPUTADOR"

Após o , a linha estará gravada na memória.

OBS.: Se fosse omitida a numeração teríamos:

EU SOU O SEU COMPUTADOR

E não estaríamos gravando absolutamente nada na memória.

A numeração antes do comando, indica ao computador que a linha faz parte de um programa.

A numeração deve aparecer em ordem crescente, pois é a partir dela que o computador se guiará para a execução de um programa.

Caso a numeração não observar esta regra, HAVERÁ ERRO.

Agora vejamos:

Digite:

RUN

O seu primeiro programa será executado

EU SOU O SEU COMPUTADOR

Digite mais estas linhas:

20 PRINT "MEU NOME E CD-6809" RETURN

30 PRINT "E O SEU?" RETURN

Agora digite:

LIST RETURN

Este novo comando LIST listará TUDO o que estiver gravado na memória.

VAMOS COMPLEMENTAR O NOSSO PRIMEIRO PROGRAMA.

Digite:

40 INPUT A\$ RETURN

Com o comando INPUT o computador ficará esperando dados de entrada pelo teclado.

Digite RUN e veja você mesmo.

Olhe onde está o cursor; ele está esperando um dado de entrada, no caso seu nome, devido ao comando INPUT.

O computador relacionará os dados de entrada, seu nome, a variável A\$.

Digite seu nome e continue aprimorando o programa:

50 PRINT "ALO," A\$

Vamos verificar a memória.

Digite:

LIST RETURN

M
E
M
O
R
I
A

```
1Ø PRINT "EU SOU O SEU COMPUTA  
DOR"  
2Ø PRINT "MEU NOME E CD-68Ø9"  
3Ø PRINT "E O SEU?"  
4Ø INPUT A$  
5Ø PRINT "ALO," A$
```

Agora vamos EXECUTAR o nosso programa.

Digite:

RUN

A resposta será:

```
EU SOU O SEU COMPUTADOR  
MEU NOME E CD-68Ø9  
E O SEU?
```

ROBERTO
ALÔ, ROBERTO

→ VALOR DADO AO A\$ NO COMANDO INPUT.

Caso você complemente o seu programa com esta linha:

```
6Ø GOTO 1Ø 
```

E digitar:

RUN

Seu programa será executado até que você teclé

Isto acontece devido ao comando GOTO, que faz com que o programa toda vez que passe pela linha 6Ø seja desviado para a linha 1Ø, recomeçando tudo outra vez.

Com isto poderemos entender melhor o porque da numeração e da seqüência.

Caso você queira deletar alguma linha digite o nº da mesma e tecle .

- Como utilizar e para o que serve a pontuação no final da linha PRINT:

1 - A vírgula posiciona em 2 colunas.

Ex.:

```
PRINT 1 , 2 , 3 
```

```
RESP.: 1 2  
       3
```

2 - Ponto e vírgula listará os comandos PRINT consecutivamente.

Ex.:

```
PRINT 1 ; 2 ; 3 
```

```
RESP.: 1 2 3
```

3 - Sem pontuação listará os comandos PRINT em fila, como aparecem nas instruções.

COMANDOS LIST E LLIST.

LIST - Listar os dados da memória no vídeo.

LLIST - Listar os dados da memória em uma impressora previamente conectada ao seu CD-6809.

CORREÇÃO OU ALTERAÇÃO DE LINHAS JÁ GRAVADAS NA MEMÓRIA

Neste caso digitar:

EDIT NN + número da linha a ser alterada.

Ex.: Se quisermos alterar a linha 10 digite:

EDIT 10

No seu vídeo aparecerá:

10 PRINT "EU SOU O SEU COMPUTADOR"

10

Para trocar:

Ex.: Vamos TROCAR QUALQUER LETRA:

- Posicione o CURSOR na letra a ser trocada (p. ex.: P).

- Tecle

- Digite a alteração.

- Tecle

Para deletar:

EDIT N N N ——— nº da linha

- Posicione no caracter a ser deletado

- Tecle

- Para continuar tecle ou

PARA INSERIR:

- EDIT NN

- Posicionar o cursor no caracter ao qual se deseja INSERIR algo.

- Tecle

- Digitar o que se deseja inserir.

- Se você apertar a tecla de espaçamento, você irá inserir espaços à sua linha.

- Para continuar a alterar tecla a combinação **SHIFT** **↑**, ou para terminar tecla **RETURN**.

Ex.: Digite: 1Ø PRINT "CDA-68"

EDIT 1Ø

1Ø

Aperte a tecla de espaço até o nº 8.

Tecla I

Insira os algarismos '0' e '9'.

Queremos, agora, deletar a letra 'A' do STRING. Para tanto tecla a combinação **SHIFT** **↑**, e volte o cursor até a letra A. Tecla **D**, para deletar 'A' e **RETURN** para finalizar a operação.

Quando for necessário deletar, inserir ou trocar ao mesmo tempo, temos a tecla **H** para ajudar.

- EDIT NN

- Posicione o cursor no início da alteração.

- Tecla **H**

- Coloque o que foi alterado

- Tecla **SHIFT** **↑** para continuar ou **RETURN** para terminar.

Ex.:

1Ø PRINT "EU SOU SEU COMPUTADOR"

Suponha que você queira trocar o que estiver após a palavra SEU:

1Ø PRINT "EU SOU SEU..."

Tecla H, e coloque a nova informação.

1Ø PRINT "EU SOU SEU MICRO".

PARA PROCURAR:

Suponha que você tenha:

```
4Ø PRINT "CD-68Ø9B"
```

Uma outra maneira de mover o cursor mais rapidamente é fazer com que o computador procure o carácter específico na linha.

Por exemplo:

Você deseja deletar a letra B dentro do STRING.

- EDIT 4Ø

- Tecle (SEARCH (procura)) e o carácter desejado, no caso o B.

O computador procurará o carácter B; ao encontrá-lo, o cursor irá parar. Então, tecle para deletá-lo.

Caso você tenha mais de um carácter igual:

```
1Ø PRINT "EU SOU O MICRO CD-68Ø90"
```

Você deseja deletar o último '0' - ele é o quarto da linha.

- EDIT 1Ø

- Tecle

- Tecle indicando o quarto '0'

- Tecle o carácter a ser deletado

No caso

PARA ACRESCENTAR CARACTERES NO FINAL DA LINHA:

- Tecle e o cursor irá para o fim da linha.

- Coloque os caracteres adicionais.

- para continuar ou para terminar.

MOVIMENTAÇÃO DO CURSOR

Às vezes você precisa movimentar o cursor sem querer deletar, inserir ou mudar a linha. Para isto tecle o nº de espaços que você deseja avançar e depois a tecla de espaçamento.

Ao contrário, se você estiver no final da frase e quiser retroceder, tecle o número que você quer voltar atrás e a tecla .

Ex.: NEW

3Ø PRINT 3Ø + 1Ø " = 4Ø "

Você quer voltar atrás 5 dígitos.

Tecle:

EDIT 3Ø.

Leve o cursor até o final da linha; agora tecle:

Resultado:

3Ø PRINT 3Ø + 1Ø

 CURSOR

VAMOS AGORA FAZER ALGUNS PROGRAMAS.

Primeiro vamos limpar a memória.

Digite:

NEW

A memória estará limpa, todos os dados anteriormente gravados estarão deletados.

Agora entre com os dados do seu programa:

```
1Ø PRINT "TROQUE O MEU SOM"
```

```
2Ø PRINT "TECLE UM NUMERO DE 1 A 255"
```

```
3Ø INPUT T
```

```
4Ø SOUND T, 5Ø
```

```
5Ø GOTO 1Ø
```

Neste momento o computador "abrirá" o teclado para entrada de dados.

O dado digitado será o valor da variável 'T'.

Para interromper, tecle

E digite NEW para limpar a memória.

ATENÇÃO:

Isto é muito importante. Quando você quiser LIMPAR A TELA tecle .
Agora, para LIMPAR A MEMÓRIA é necessário que seja digitado NEW .

Os dados contidos na memória não podem ser delimitados. Portanto se você, por engano, carregar mais de um programa, haverá erro na seqüência dos números indicativos das linhas e, conseqüentemente, erro na execução do seu programa.

Deixe bem viva a idéia da limpeza da memória (NEW), sempre que você for executar um novo programa.

ENTRE COM ESTE PROGRAMA:

```
1Ø FOR X = 1 TO 1Ø
```

```
2Ø PRINT "X =" X
```

```
3Ø NEXT X
```

APÓS CADA

LINHA

TECLAR

Neste programa vemos 2 novos comandos.

Eles são:

```
FOR          faz X variar de 1 à 10
NEXT        o próximo valor de X
```

Digite: RUN RETURN

O programa executará até que X seja igual a 10.

Digite: NEW RETURN

VAMOS A UM NOVO PROGRAMA:

```
10 FOR X = 255 TO 1 STEP -1
20 PRINT "SOM" X
30 SOUND X, 1
40 NEXT X
```

Este programa emitirá 255 tipos de som. Quando X for = 1 o computador interromperá a execução.

VEJAMOS:

- Linha 10 primeira passagem: X = 255
- Linha 20 X = 255
- Linha 30 255,1 → será emitido o som codificado como 255.
- Linha 40 será reduzido em 1 o valor de X, conforme o comando STEP da linha 10.

Digite RUN

Caso você queira interromper **MOMENTANEAMENTE** o programa tecle

SHIFT

@

ao mesmo tempo.

Para reiniciar digite qualquer tecla.

O comando STEP sô não é necessário quando o incremento for de 1 a 1 positivos.

A partir das informações dadas, você deve tentar novos exemplos do que foi mostrado até agora. Tente, você entenderá bem mais.

Vamos limpar a tela, se for o caso, e a memória.

CLEAR

NEW

RETURN

MONTE ESTE PROGRAMA, ELE LHE DARÁ UM RELÓGIO.

```
10 FOR H = 0 TO 23
20 FOR M = 0 TO 59
30 FOR S = 0 TO 59
40 CLS
50 PRINT H:" M ":" S
60 SOUND 150, 2
70 FOR T = 1 TO 375
80 NEXT T
90 NEXT S
100 NEXT M
110 NEXT H
120 GOTO 10
```

Para interromper, ESC

NÃO ESQUEÇA DE LIMPAR A MEMÓRIA.

Digite: NEW

Vamos sonorizar o nosso computador.

```
1Ø FOR X = 1 TO 2
2Ø SOUND 125,8
3Ø SOUND 108,8
4Ø SOUND 89,8
44 FOR Y = 1 TO 23Ø
46 NEXT Y
5Ø NEXT X
6Ø FOR X = 1 TO 2
7Ø SOUND 147,8
8Ø SOUND 133,4
9Ø SOUND 133,4
1ØØ SOUND 125,8
11Ø FOR Y = 1 TO 23Ø
12Ø NEXT Y
13Ø NEXT X
```

LEMBRE-SE:

O comando LIST **RETURN** possibilita uma demonstração do que nós temos na memória. Como começamos a escrever programas maiores, ele se torna imprescindível.

A partir da TABELA DE SONS você poderá montar novos programas com novas músicas. Tente!

TESTES QUE EXIGEM DECISÃO.

O comando de decisão é o IF/THEN.

Este comando pode aparecer várias vezes em um programa.

Vamos a um exemplo:

```
10 PRINT "VOCE QUER A TELA VERMELHA OU  
AZUL?"
```

```
20 INPUT C$
```

LEMBRE-SE: →

O TECLADO
FICARÁ ABERTO
PARA VOCÊ
ENTRAR COM
A INFORMAÇÃO

```
30 IF C$ = "VERMELHA" THEN 100
```

```
40 IF C$ = "AZUL" THEN 200
```

```
100 CLS (4) ←
```

```
110 END
```

```
200 CLS (3) ←
```

ATENÇÃO:

END é um novo comando que avisa o computador para interromper o processamento.

Se houvesse seqüência, ao executar a linha 200 a tela trocaria de cor.

COMANDO PARA ACESSO ALEATÓRIO.

O comando RND possibilita a montagem de programas de jogos de sorte.

RND (X) dará qualquer número aleatório de 1 a X.

Vamos a um exemplo:

Digite:

NEW

RETURN

```

100 CLS
200 A = RND (6)
300 B = RND (6)
400 S = A + B
500 PRINT @ 200, A
600 PRINT @ 214, B
700 PRINT @ 394, "VOCE ROLOU" S
800 PRINT @ 454, "VOCE QUER JOGAR NOVAMENTE?"
900 INPUT A$
1000 IF A$ = "SIM" THEN 100

```

RUN

Neste exemplo, o computador simula o lançamento de 2 dados simultaneamente. Após cada lance ele lhe questionará se deseja jogar novamente, e, a uma resposta SIM, todo o processo se desenrolará novamente.

"SALVAR" UM PROGRAMA

A partir deste ponto do Manual começaremos a montar programas maiores e mais interessantes.

Para que você possa SALVAR um programa, terá que utilizar uma área auxiliar.

No início deste Manual falamos sobre a conexão de periféricos auxiliares ao nosso microcomputador CD-6809. Um destes periféricos é o gravador cassete.

Vamos agora a uma demonstração de como passar um programa codificado na memória para uma fita cassete e vice-versa para que possamos rodar este programa ao nosso gosto.

COMO COPIAR DA MEMÓRIA PARA A FITA K-7

- O gravador deverá estar corretamente conectado.
(ver capítulo inicial).
- Digite qualquer programa no computador.

Digite RUN

Verifique se o mesmo está com o processamento OK.

- Posicione a fita no início do cassete.
- Dê ao programa um nome (máximo de 8 dígitos). Este nome será a identificação dos dados para uma futura utilização.
- Neste ponto teremos:
 - 1 - Um programa GRAVADO NA MEMÓRIA.
 - 2 - Um nome (identidade) escolhido.
 - 3 - Um gravador cassete com a fita posicionada no início.
- Coloque seu gravador para gravar.

Aperte PLAY e RECORD

DO GRAVADOR

- Para iniciar o "salvamento":

Nome dado por você
ao programa. Máxi-
mo 8 dígitos.

Digite CSAVE

"NOME"

RETURN

O motor do seu gravador começará a funcionar e seu programa passará em cópia da memória para a fita cassete.

Quando o gravador parar, a operação estará terminada e no vídeo aparecerá: OK

- Depois de gravado aperte o botão STOP do gravador.

COMO FAZER PARA CARREGAR NA MEMÓRIA

UM PROGRAMA QUE ESTÁ EM FITA:

- Conexão entre computador e gravador está OK?
- Aperte a tecla "PLAY" do gravador.
- Digite NEW para limpar a memória.

NÃO SE ESQUEÇA

- Para iniciar a CARGA:

Digite

CLOAD "NOME" RETURN

O motor do seu gravador começará a funcionar e o seu programa passará em cópia do cassete para a memória.

No canto superior esquerdo do vídeo aparecerá: **S**.

Isto indica que o computador está pesquisando na fita para achar o programa de terminado.

Se a operação for feita com sucesso teremos no vídeo:

F _____ ← o nome do seu Programa

e após a gravação na memória:

OK

A partir deste momento, o seu Programa estará gravado na memória e pronto para execução.

CASO DE ERRO:

Quando o computador estiver procurando o programa indicado na fita cassete, há verã uma MOSTRA constante do nome dos outros possíveis programas da fita.

Caso não haja um Programa com o nome indicado por você, o computador irá até o fim da fita procurando.

Para interromper a procura, tecle o botão STOP do gravador.

PARA "SALVAR" MAIS DE UM PROGRAMA

- Posicione a fita no início.
- Aperte o botão PLAY do gravador.
- Digite:

SKIPF e o nome do último programa gravado na fita.

Ex.: SKIPF "NOME" RETURN

O computador irá localizar o fim do programa indicado.

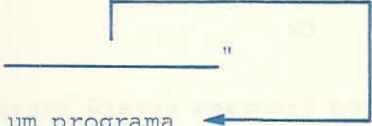
Vídeo  OK

- Aperte RECORD e PLAY do gravador.
- Digite:

CSAVE "NOME"

Quando você não estiver lembrado do nome do último programa gravado:

Digite:

SKIPF "  "
nome de um programa
que você saiba que não está gravado.

Ex.:

SKIPF "X"

O computador irá procurar o Programa "X" - quando o mesmo identificar na tela o último programa você poderá copiar o nome.

Se por acaso a leitura da fita chegar ao fim, o computador acusará I/O ERROR.

Você já deverá ter o nome do seu último programa, portanto:

- Posicione a fita no início.

- Digite: SKIPF "NOME"

- Etc. ...

LEMBRETE:

Quando o gravador não estiver sendo usado, não o deixe no estado de gravação ou leitura (RECORD ou PLAY). Aperte o botão STOP.

COMO COLORIR A TELA.

Uma das características do microcomputador CD-6809 é a confecção de desenhos coloridos.

Com esta facilidade você pode montar uma infinidade de desenhos, dependendo somente da sua criatividade.

Em uma parte posterior deste manual você aprenderá a montar gráficos, portanto a correta assimilação das informações que serão dadas agora, são de grande importância.

Para iniciar, digite:

1Ø CLS (Ø) → tela cor preta

2Ø SET (Ø, Ø, 3)

3Ø GOTO 3Ø

Digite agora RUN, para a execução.

Na parte superior esquerda da sua tela aparecerá um PUNTO azul.

A linha 3Ø deixa o computador em "LOOP", deixando a tela estática.

Para interromper:

Tecla → ESC

Caso você não usar o GOTO, o programa terminará e o ponto desaparecerá.

O novo comando SET serve para:

```
SET (31, 14, 3)
```

- Indicar ao computador a localização horizontal/vertical e a COR DO PONTO.

- O 1º número indica:

A localização horizontal com números de 0 até 63.

- O 2º número indica:

A localização vertical com números de 0 a 31.

- O 3º número indica:

A cor do PONTO (consulte tabela de cores).

VAMOS DESENHAR:

Verifique se a memória está limpa.

Digite LIST para verificar.

Tudo correto?

```
5 CLS (0)
```

```
10 FOR H = 15 TO 48
```

```
20 SET (H, 5, 5)
```

```
30 SET (H, 20, 5)
```

```
40 NEXT H
```

```
50 GOTO 50
```

→ RUN

O programa desviará da linha 40 para a linha 10 até que H seja igual a 49.

A linha 50, comando GOTO, deixará fixa a tela até que você teclasse ESC para interromper.

Bem, estamos começando a desenhar o "ROSTO" do computador.

Já temos uma linha superior e uma inferior.

Vamos agora às laterais.

Substitua a linha 5Ø e inclua:

```
5Ø FOR V = 5 TO 2Ø
```

```
6Ø SET (15, V, 5)
```

```
7Ø SET (48, V, 5)
```

```
8Ø NEXT V
```

```
9Ø GOTO 9Ø
```

```
—————→ RUN
```

Tecla ESC

Temos agora as laterais do rosto.

Vamos novamente substituir a linha 9Ø do comando GOTO, para a finalização.

Vamos fazer um nariz laranja, uma boca vermelha e os olhos azuis.

```
9Ø SET (32, 13, 8)
```

```
10Ø FOR H = 28 TO 36
```

```
11Ø SET (H, 16, 4)
```

```
12Ø NEXT H
```

```
13Ø SET (25, 1Ø, 3)
```

```
14Ø SET (38, 1Ø, 3)
```

```
15Ø GOTO 15Ø
```

```
RUN —————→ e o rosto estará pronto.
```

Caso você deseje que um dos olhos fique PISCANDO, substitua a linha 15Ø.

```
15Ø RESET (38, 1Ø)
```

```
16Ø GOTO 14Ø —————→ RUN
```


Para interromper → ESC

Para salvar em fita observe instrução anterior.

Para limpar a memória → digite:

NEW

A partir de agora vamos usar o microcomputador como auxiliar para a resolução de problemas.

Digite:

```
5 CLS
10 DATA MACA, LARANJA, PERA
20 FOR X = 1 TO 3
30 READ F$
40 PRINT "F$: " F$
45 FOR Y = 1 TO 300: NEXT Y
50 NEXT X
60 RESTORE
70 GOTO 5
```

Neste programa temos (3) três novos comandos.

Eles são:

DATA → Indica dados aproveitáveis na execução dos programas. Serão lidos um a um.

READ → Leitura dos dados relacionados no comando DATA.

RESTORE → Atua desta forma:

No Programa acima temos três dados informados no comando DATA.

Quando o comando NEXT for executado para X igual a 4, o comando RESTORE que é o próximo, posicionará a próxima leitura no primeiro dado, no caso MAÇÃS, e inicializa

a variável X para o primeiro valor do comando FOR.

Sem este comando (RESTORE), o computador acusaria erro no momento que fosse executar a linha 30, pois não haveriam mais dados para ler.

CLS → já foi indicado anteriormente. Serve para limpar a tela.

Digite: RUN RETURN

- Para interromper ESC .

O comando DATA pode ser colocado em qualquer lugar no programa. O computador se encarregará de fazer a identificação.

O primeiro dado do comando DATA será relacionado a primeira variável, o segundo dado a segunda variável, etc....

Quando houver mais dados do que variáveis, somente serão lidas as variáveis correspondentes aos primeiros dados.

Quando houver mais variáveis do que dados, o programa acusará erro.

No próximo Programa teremos o aparecimento dos comandos aprendidos anteriormente, e mais este: o INT (_).

INT → indica ao computador que só a parte inteira do número deve ser levada em conta.

DIGITE, mas antes verifique se há algo na memória.

Caso haja, e você não quiser aproveitar, digite NEW para limpar.

```
10 DATA RS, PORTO ALEGRE
```

```
20 DATA PARANA, CURITIBA
```

```
30 DATA GOIAS, GOIANIA
```

```
40 DATA BAHIA, SALVADOR
```

```
50 DATA MINAS, BELO HORIZONTE
```

```
60 N = RND (10)
```

```
65 IF INT (N/2) = N/2 THEN N = N-1
```

70 FOR X = 1 TO N

80 READ A\$

90 NEXT X

95 CLS

100 PRINT "O ESTADO E:" A\$

110 READ B\$

120 PRINT "SUA CAPITAL E:" B\$

125 SOUND 1,20

130 RESTORE

140 FOR A = 1 TO 100:NEXT A

150 GOTO 60

→ RUN

- Você pode incrementar neste programa com alteração das cores a cada novo comando PRINT executado.

96 W = RND (8)

97 CLS W

- Usar o comando PRINT @ para localizar no vídeo a resposta onde você entender.

- Colocar som, etc....

A criatividade é sua.

Digite:

RUN para a execução.

Lembre-se, para uma interrupção temporária:

TECLE SHIFT @

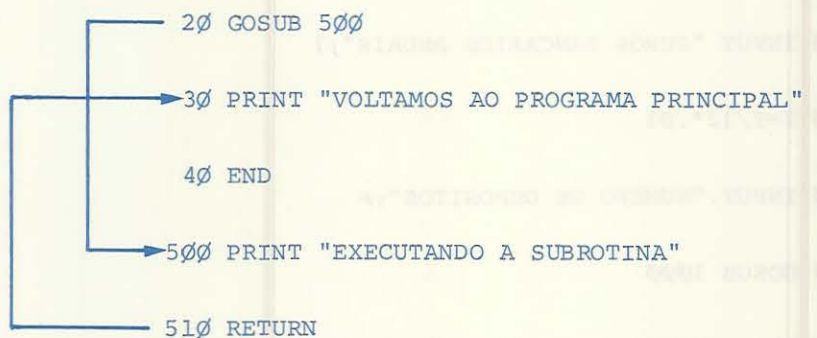
Para continuar:

Digite qualquer tecla.

Para a resolução de fórmulas complicadas ou problemas difíceis, podemos usar o microcomputador CD-6809.

Digite e execute este Programa:

```
10 PRINT "EXECUTANDO O PROGRAMA PRINCIPAL"  
20 GOSUB 500  
30 PRINT "VOLTAMOS AO PROGRAMA PRINCIPAL"  
40 END  
500 PRINT "EXECUTANDO A SUBROTINA"  
510 RETURN
```



A linha 20 indica ao computador que vá até a subrotina iniciada na linha 500.

RETURN informa ao computador que retorne a linha logo após o comando GOSUB.

Delete a linha 40 e preste atenção no que vai acontecer quando você executar o programa.

Na sua tela aparecerá:

```
EXECUTANDO O PROGRAMA PRINCIPAL  
  
EXECUTANDO A SUBROTINA  
  
VOLTAMOS AO PROGRAMA PRINCIPAL  
  
EXECUTANDO A SUBROTINA  
  
? RG ERROR IN 510
```

O comando END serve para delimitar o programa principal (conforme exemplo) e finalizar qualquer programa.

No caso do programa acima, quando o computador for executar o comando RETURN, não saberá qual é a subrotina a qual deverá retornar.

Limpe a tela

→

Limpe a memória → NEW

Programa agora uma aplicação para seu dinheiro, mensalmente a mesma importância.

Digite e execute este Programa:

```
10 INPUT "SEUS DEPOSITOS MENSAIS";D
20 INPUT "JUROS BANCARIOS ANUAIS";I
30 I=I/12*.01
40 INPUT "NUMERO DE DEPOSITOS";P
50 GOSUB 1000
60 PRINT "VOCE TEM CR$ "FV "EM" P "MESES"
70 END
1000 REM FORMULA DOS JUROS MENSAIS
1010 N=1+I
1020 GOSUB 2000
1030 FV=D*((E-1)/I)
1040 RETURN
2000 REM FORMULA DO AUMENTO
2010 E=1
2020 FOR X=1 TO P
2030 E=E*N
2040 NEXT X
2050 IF P=0 THEN E=1
2060 RETURN
```

O comando REM é aceito pelo computador como um simples código da linguagem BASIC, ele não tem sentido lógico; serve apenas como identificador dentro do programa, pa-

ra comentários.

Na linha 1030 do seu programa temos uma operação entre parêntesis - isto indica que ela está com prioridade para a resolução.

Tente fazer você mesmo algum programa utilizando os comandos aprendidos, isto será de grande valor para o entendimento completo do computador.

MAIS ALGUNS COMANDOS EM BASIC.

O primeiro é o STOP.

Digite e execute:

10 A = 1

20 A = A + 1

30 STOP

40 A = A * 2

50 STOP

60 GOTO 20

→ RUN

O computador imprimirá no vídeo:

BREAK IN 30

OK

O computador ficará esperando um comando seu.

Digite:

PRINT A

O computador imprimirá: 2

Agora digite:

CONT

Isto indicará ao computador que o processamento deverá continuar.

O próximo:

BREAK IN 50

Digite:

PRINT A

CONT

Digite NEW para limpar a memória e continue:

PRINT MEM

O comando MEM utilizado com o PRINT mostra quanto espaço de memória ainda sobra para ser ocupado.

Isto é de grande importância na elaboração de programas extensos.

Digite este Programa:

```
10 INPUT "TECLE 1, 2, OU 3"; N
```

```
20 ON N GOSUB 100, 200, 300
```

```
30 GOTO 10
```

```
100 PRINT "VOCE DIGITOU 1"
```

```
110 RETURN
```

```
200 PRINT "VOCE DIGITOU 2"
```

```
210 RETURN
```

```
300 PRINT "VOCE DIGITOU 3"
```

```
310 RETURN
```

Execute \longrightarrow RUN.

O comando ON ... GOSUB

Indica ao computador que identifique os números das linhas que seguem o comando GOSUB, com o valor da variável N.

```
Para N = 1 GOSUB 100
```

```
Para N = 2 GOSUB 200
```

```
Para N = 3 GOSUB 300
```

Se entrarmos com um valor maior que 3, o computador simplesmente trocará de linha.

O comando ON ... GOTO

A diferença do ON_GOSUB para o ON_GOTO é que o ON_GOSUB volta à linha do próximo comando quando for encontrado um RETURN na subrotina. O ON_GOTO vai para a linha especificada e NÃO VOLTA.

AND/OR

São operações lógicas a serem usadas com o comando IF.

Exemplo:

```
IF D$ = "SIM" AND E$ = "SIM" THEN...
```

somente se as duas igualdades forem satisfeitas será executada a instrução após o THEN.

```
IF D$ = "SIM" OR E$ = "NAO" THEN...
```

caso uma das igualdades ou ambas forem satisfeitas, será executada a instrução após o THEN.

SGN

Indica que o número é positivo ou negativo.

```
SGN ( )
```

Ex.:

```
10 INPUT "TECLE UM Nº"; X
```

```
20 IF SGN (X) = 1 THEN PRINT "POSITIVO"
```



```

3Ø IF SGN (X) = Ø THEN PRINT "ZERO"
4Ø IF SGN (X) = -1 THEN PRINT "NEGATIVO"
5Ø GOTO 1Ø

```

ABS

Indica o valor absoluto de um número.

Ex.:

```

1Ø INPUT "TECLE UM Nº"; N
2Ø PRINT "SEU VALOR ABSOLUTO E" ABS (N)
3Ø GOTO 1Ø

```

STR \$

Converte um número em um STRING.

Ex.:

```

1Ø INPUT "TECLE UM Nº"; N
2Ø A$ = STR$ (N)
3Ø PRINT A$ " E AGORA UM STRING"

```

VARIAÇÕES DO COMANDO PRINT.

PRINT TAB (X)

Move o cursor para a coluna especificada em (X) e imprime a partir daí.

Ex.:

```

1Ø X = 2 * 5
2Ø PRINT TAB (5) "DOBRO DE 5 E:"
3Ø PRINT TAB (2Ø) X
RUN

```

		Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	1	1	1	1	1	1	1	1	1	2	2		
Colunas	→	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
		D O B R O D E 5 E : 1 Ø																				

PRINT USING

É usado para a formatação de números, indicando quantas casas tem a parte inteira e a decimal.

O sinal # indica a quantidade de casas inteiras e decimais separadas pelo ponto.

PRINT USING "###,##"; A

→ A = O NÚMERO OU VARIÁVEL A SER IMPRESSA.

PRINT USING "#####": 44.2

→ 44

PRINT USING "#"; 44.2

→ %44 → % avisa que existe erro, no caso da formatação ser menor que o número.

O ponto decimal pode ser colocado em qualquer lugar. O computador arredonda o que for necessário.

PRINT USING "# # . #"; 58.76

→ 58.8

A vírgula colocada ao final da formatação, ordena os números da direita para a esquerda de 3 em 3.

PRINT USING "#####,"; 12345

→ 12,345

Colocando dois asteriscos no início da formatação, as casas não usadas à esquerda serão preenchidas com os asteriscos.

Os dois asteriscos aumentam duas posições no campo numérico.

```
PRINT USING "*** ### "; 44.0
```

```
→ **** 44
```

Para números que representam dinheiro, coloca-se o \$ a esquerda da formatação. O número será impresso com o "\$" na frente.

```
PRINT USING "$ ### . ## "; 18.6735
```

```
→ $ 18.67
```

Com o ponto de exclamação, apenas o primeiro STRING do caracter será impresso.

```
PRINT USING "!"; "COMPUTADOR"
```

```
→ C
```

Para imprimir o STRING inteiro usa-se %%.

Os sinais já contam como espaços

```
PRINT USING "% %"; "CASA"
```

```
→ CASA
```

Existe ainda uma outra forma de usar o PRINT USING.

Atribuimos o formato a uma variável.

Ex.:

```
1Ø A$ = "# # . ##"
```

```
2Ø Y = 5.5
```

```
3Ø PRINT USING A$; Y
```

```
RUN
```

```
→ 5.50
```

```
OK
```

Comando PRINT com números POSITIVOS ou NEGATIVOS.

```
( + ) PRINT USING "+ * * ### "; 300
```

```
***300
```

(-) PRINT USING " + # # # " ; - 300

-300

O sinal (-) colocado no fim da formatação, imprimirá um sinal negativo no final dos números negativos e deixará um espaço em branco nos positivos.

Dentro das especificações numéricas podemos citar como exemplo o comando PRINT FIX.

Ex.:

PRINT FIX (3.056)

Esta instrução serve para que haja somente o aproveitamento da parte inteira do número.

Para o comando acima teremos uma resposta assim:

3

OK

COMANDOS:

- 1 - LEN (X\$) — Informa o tamanho do STRING.
- 2 - RIGHT\$ (X\$, Y) — Este comando dá os "Y" últimos caracteres do STRING X\$ a partir da direita.
- 3 - LEFT\$ (X\$, Z) — Este comando dá os "Z" últimos caracteres do STRING \$ a partir da esquerda.
- 4 - MID\$ (X\$, A, B) — Este comando dá os caracteres situados entre "A" e "B".

EXEMPLOS:

1 — Digite:

```
10 PRINT "DIGITE UMA PALAVRA"
```

```
20 INPUT S$
```

```
30 PRINT "A PALAVRA TEM" LEN (S$) "CARACTERES"
```

```
40 INPUT "VOCÊ QUER TENTAR OUTRA"; A$
```

```
50 IF A$ = "SIM" THEN 10
```

LIMPE A MEMÓRIA — NEW

2 — Digite:

```
10 X$ = "ARMÁRIO"
```

```
20 PRINT RIGHT$ (X$, 3)
```

RUN

Resposta no

RIO

Vídeo

OK

3 — Digite:

```
20 PRINT LEFT$ (X$, 2)
```

RUN

Resposta no

AR

Vídeo

OK

4 — Digite:

```
20 PRINT MID$ (X$, 3, 4)
```

3º CARACTER

ARMÁRIO

CARACTERES

RUN

Resposta no

MARI

Vídeo

OK

A esta altura do Manual, já foram demonstrados praticamente todos os comandos da linguagem Basic, necessários para a operação-programação do seu Microcomputador CD-6809.

Vamos então colocar em prática estes conhecimentos.

Começaremos agora o desenvolvimento de programas/montagens no Vídeo, imagens com movimento, jogos; daí, partiremos para o desenvolvimento de gráficos mais complexos.

Bem, de início vamos aprender a testar telas com movimento.

Verifique se a memória do seu computador está limpa. Caso contrário, digite
NEW para limpá-la.

Após a verificação, digite:

```
PRINT ASC("A") 
```

O computador responderá:

65

65 é o código em ASCII para o caracter A

Digite:

```
 RETURN
```

O computador responderá:

A

Demonstrando que o Caracter (CHR\$) de código 65 representa a letra A.

ASCII - AMERICAN STANDARD CODE FOR INFORMATION INTERCHANGE.

É um dos tipos de códigos usados em sistemas computacionais.

Neste caso possibilita conectar seu computador com outro através de uma linha telefônica.

No Apêndice deste Manual temos a lista de caracteres/códigos em ASCII.

O uso destes códigos no desenvolvimento dos programas montadores de gráficos é muito importante.


Vejamos um exemplo. Digite:

(ATENÇÃO, a memória está limpa?)

```
1Ø CLS
2Ø H=63
25 SET(H,14,3)
3Ø A$=INKEY$
4Ø IF A$=CHR$(8) THEN6Ø
5Ø GOTO 3Ø
6Ø H=H-1
65 IF H<Ø THEN END
7Ø SET(H,14,3)
75 RESET(H+1,14)
8Ø GOTO 3Ø
```

Dê RUN no programa.



TECLE 

Cada vez que você teclar  , o computador fará o ponto azul voltar um espaço.


O ponto azul é designado no programa pelas linhas 25 e 7Ø.

O comando INKEY\$ faz o computador relacionar a tecla digitada com o caracter associado, no caso A\$.

No nosso Programa teremos:

Para INKEY\$  a tecla  digitada.

Na linha 4Ø CHR\$(8), indica em ASCII para o código 8 o sinal  .

Na linha 75 o comando RESET fará "desaparecer" o sinal indicado no último comando SET. 

Na linha 65 temos:

```
IF H<Ø THEN END
```

O END indica o fim do programa.

Agora já sabemos que o comando CHR\$ serve para imagens com movimento.

Vamos montar um programa que tenha movimento na imagem.


A partir disto tente tirar o maior proveito possível para montagens futuras.

(Lembre-se de verificar se a memória do seu computador está limpa!)

Digite:

```
10 CLS(6)
20 FOR H=0 TO 63
30 FOR V=22 TO 31
40 SET(H,V,1)
50 NEXT V,H
60 FOR H=0 TO 63 STEP 2
70 RESET (H,22)
80 NEXT H
90 FOR V=20 TO 21
100 FOR H=0 TO 15
110 SET(2+H,V,3):SET(20+H,V,3)
120 NEXT H,V
200 A$=INKEY$
210 IF A$=CHR$(9) THEN GOSUB 1000
215 IF A$=CHR$(8) THEN GOSUB 2000
220 GOTO 200
1000 REM =AVANCA
1010 IF F>26 THEN RETURN
1020 FOR V=20 TO 21
1030 FOR H=0 TO 1
1040 SET(2+F+H,V,6):SET(20+F+H,V,6)
1050 SET(18+F+H,V,3):SET(36+F+H,V,3)
1055 NEXT H,V
1060 F=F+2
1070 GOTO 1000
2000 REM VOLTA
2010 IF F<0 THEN RETURN
2020 FOR V=20 TO 21
2030 FOR H=0 TO 1
2040 SET(0+F+H,V,3):SET(18+F+H,V,3)
2050 SET(16+F+H,V,6):SET(34+F+H,V,6)
2060 NEXT H,V
2070 F=F-2
2080 GOTO 2000
```

ATENÇÃO!

No passo 210 o nº 9
indica a tecla 

Dê RUN neste programa.

Quando teclar  o seu TREM voltará.

Quando teclar  o seu TREM avançará.

Para interromper momentaneamente o seu programa, tecle:

SHIFT  .

Para parar definitivamente o programa, tecle:



Criado um programa com movimento, vamos desenvolver um programa de jogo.

Limpe a memória e digite:


```

5 CLS(Ø)
1Ø FOR X = 1 TO 5
2Ø SET (RND(64)-1,RND(3Ø)+1,8)
3Ø NEXT X
4Ø FOR V = 2 TO 31
5Ø FOR H = Ø TO 63
6Ø IF POINT (H,V) = Ø THEN GOSUB 1ØØ
7Ø NEXT H, V
8Ø END
1ØØ PRINT Ø Ø, "POSIÇÃO" H, V
11Ø RETURN

```

Na linha 6Ø o computador examinará cada POINT (PONTO) na definição vertical de 2 até 31 e horizontal de Ø até 63.

Agora delete as linhas 4Ø, 5Ø e 7Ø e faça alterações no programa a seu gosto e teste.



Limpe a memória e teste o colorido.

```
5 CLS (Ø)
1Ø C= RND (9)-1
2Ø SET (31,15,C)
3Ø IF POINT (31,15)=2 THEN PRINT @ Ø, "POSIÇÃO 31,15 E AMARELA"
4Ø IF POINT (31,15)=3 THEN PRINT @ 48Ø, "POSIÇÃO 31,15 E AZUL";
5Ø FOR T = 1 TO 1ØØØ: NEXT T
6Ø GOTO 5
```

Execute o programa — RUN

Na linha 3Ø o comando POINT (31,15)=2, testa se o ponto local (31,15) tem cor amarela (2).

Na linha 4Ø o comando POINT (31,15)=3 testa se o ponto localizado em (31,15) tem cor azul (3).

Se forem adicionadas novas linhas entre as linhas 4Ø e 5Ø poderemos testar outras cores e o efeito será muito interessante.

Lembre-se, todos os programas aqui demonstrados servem como base para a montagem (desenvolvimento) dos seus próprios programas. Tente sempre assimilar o máximo, fazer relações, comparações, etc.. O seu aprendizado será daí, completo.

Nunca esqueça que a memória do seu microcomputador não faz distinção entre comandos. Ela não separa programas. Se houver mais de um programa gravado e você der RUN, haverá erro. Sempre que um programa for digitado, verifique se há algo na memória (LIST). Em caso positivo, você poderá salvar em fita se lhe interessar ou deletar. Tenha sempre isto em conta.

Vamos montar um engarrafamento de tráfego?

Mãos a obra:

```
1Ø A=RND(7)*16:B=RND(7)*16
2Ø A$=CHR$(129+A)+CHR$(131+A)
3Ø B$=CHR$(133+3)+CHR$(143+B)+CHR$(13Ø+3)
4Ø IF RND(2)=2 THEN VE$=A$ ELSE VE$=B$
5Ø IF LEN(TR$+VE$+SP$)>32 THEN 1ØØ
6Ø TR$=TR$+VE$
7Ø GOTO 1Ø
1ØØ INPUT "VELOCIDADE (1-2ØØ) ";S
11Ø FOR P=0 TO 48Ø
12Ø PRINT @P,TR$
13Ø FOR X=1 TO S:NEXT X
14Ø CLS (Ø)
15Ø NEXT P
```

Se você der RUN no programa verá o tráfego de veículos.

Agora troque as seguintes linhas:

```
11Ø P= 32Ø
15Ø P= P+1
16Ø IF P = 351 THEN 11Ø
17Ø PRINT @P, LEFT$ (TR$,352-P);
18Ø PRINT @32Ø, RIGHT$ (TR$,P-32Ø);
19Ø GOTO 13Ø
```

Execute — RUN

Se você tiver uma lista ou um arquivo de qualquer tipo de informações e quiser a ajuda do seu microcomputador CD-68Ø9, ele poderá lhe dar esta ajuda.

Abaixo temos uma série de informações sobre as quais o seu computador poderá lhe ajudar:

- 1 - Super-Mercado
 - 2 - Biblioteca
 - 3 - Medicamentos
 - 4 - Endereços
 - 5 - Apontamentos Diversos
 - 6 - Telefones
 - 7 - Dados sobre coleções diversas
 - 8 - Lista de discos
 - 9 - Músicas
 - 10 - Jogos
- ETC...

Conte com seu microcomputador em quase tudo que você imaginar que pode ser guardado.

Ao armazenar dados e, a partir destes fazer relações, teremos que usar uma nova técnica, o ARRAY.

- ARRAYS são conjuntos de variáveis, todas representando algum valor, como as variáveis simples, mas todas compartilhando um nome comum.

Um índice é associado a este nome para individualizar os elementos do conjunto. Ao conjunto das variáveis que compartilham um nome comum, chamaremos de ARRAY ou TABELAS.

Para indicar ao computador o índice das tabelas e daí o número total das variáveis que formam o seu conteúdo, usaremos o comando DIM (DIMENSION).

Vamos a um exemplo.

Digite:

```
5 DIM S$(8)
10 DATA OVOS,BACON
20 DATA AÇUCAR,LEITE,MANTEIGA
30 DATA TOMATES,BATATAS,SAL
40 FOR X=1 TO 8
50 READ S$(X)
60 NEXT X
70 PRINT "LISTA DO SUPERMERCADO"
80 FOR X=1 TO 8
90 PRINT X;S$(X)
100 NEXT X
```

- O array "S" tem 8 variáveis
STRING

Leitura dos itens do ARRAY

RUN

A seguir, mais um exemplo, em que demonstraremos que um ARRAY pode ter várias dimensões:

Verifique se a memória do seu computador está pronta para receber um novo programa e digite:

```
5 DIM V (3,3,2)
10 DATA 143, 678, 215, 514, 125, 430
20 DATA 525, 54, 318, 157, 254, 200
30 DATA 400, 115, 124, 300, 75, 419
40 FOR G = 1 TO 3
50 FOR D = 1 TO 3
60 FOR C = 1 TO 2
70 READ V (G, D, C)
80 NEXT C
90 NEXT D
100 NEXT G
110 INPUT "ESCOLHA O GRUPO (1 - 3)"; G
120 IF G < 1 OR G > 3 THEN 110
130 INPUT "ESCOLHA COLOCAÇÃO (1 - 3)"; D
140 IF D < 1 OR D > 3 THEN 130
150 INPUT "ESCOLHA CANDIDATO (1 - 2)"; C
160 IF C < 1 OR C > 2 THEN 150
170 PRINT V (G, D, C)
180 GOTO 110
```

Neste programa teremos:

- Dois candidatos a um cargo X
- Três grupos

NESTE CASO TEREMOS:

ARRAY V

GRUPO 1

	CANDIDATO 1	CANDIDATO 2
COLOCAÇÃO 1	V (1, 1, 1) = 143	V (1, 1, 2) = 678
COLOCAÇÃO 2	V (1, 2, 1) = 215	V (1, 2, 2) = 514
COLOCAÇÃO 3	V (1, 3, 1) = 125	V (1, 3, 2) = 430

GRUPO 2

	CANDIDATO 1	CANDIDATO 2
COLOCAÇÃO 1	V (2, 1, 1) = 523	V (2, 1, 2) = 54
COLOCAÇÃO 2	V (2, 2, 1) = 318	V (2, 2, 2) = 157
COLOCAÇÃO 3	V (2, 3, 1) = 254	V (2, 3, 2) = 200

GRUPO 3

	CANDIDATO 1	CANDIDATO 2
COLOCAÇÃO 1	V (3, 1, 1) = 400	V (3, 1, 2) = 119
COLOCAÇÃO 2	V (3, 2, 1) = 124	V (3, 2, 2) = 300
COLOCAÇÃO 3	V (3, 3, 1) = 75	V (3, 3, 2) = 419

APLICAÇÕES PARA MONTAGEM DE GRÁFICOS.

Se você digitar:

PRINT CHR\$ (128) RETURN

terá como resposta um bloco de cor preta.



— PRETO

Tente agora estes números.

Digite:

PRINT CHR\$ (129)

RETURN

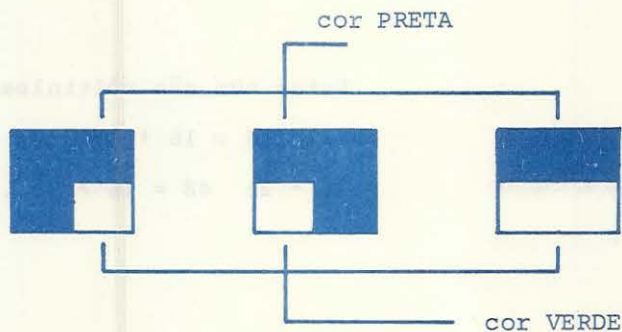
PRINT CHR\$ (130)

RETURN

PRINT CHR\$ (131)

RETURN

Terá:



Estes BLOCOS são chamados de caracteres gráficos.

Lembre-se do que aprendeu sobre o comando CHR\$. Ele faz a conversão do informado entre parêntesis em ASCII, para o caracter correspondente.

Ex.: CHR\$ (65) — 65 está em ASCII. O correspondente a 65 é A.

Digite este Programa para teste dos caracteres gráficos.

```
10 CLS(5)
20 FOR C=128 TO 143
30 PRINT @0,"PRESSIONE QUALQUER TECLA PARA CONTINUAR";
40 PRINT @173,C;
50 PRINT @240,CHR$(C);
60 K$=INKEY$:IF K$=""THEN 60
70 NEXT C
80 GOTO 10
```

A linha 50 indica a impressão dos caracteres gráficos com códigos de 128 até 143 na posição vertical 240 da sua tela.

Dê um RUN.

O caracter gráfico se apresenta com duas cores, preto e verde, mas há a possibilidade de alterarmos esta segunda cor.

Tente o exemplo seguinte

```
PRINT CHR$ (129 + 16) RETURN
```

A área que deveria ser verde agora é amarela, pois o nº 16 indica em código a cor amarela.

A seguir temos a lista dos códigos, e suas respectivas cores.

16 - AMARELO

32 - AZUL

48 - VERMELHO

64 - BRANCO

80 - AZUL CLARO

96 - VIOLETA

112 - LARANJA

Estes nºs são múltiplos de 16

(16 = 16 * 1; 32 =

16 * 2; 48 = 16 * 3;...)

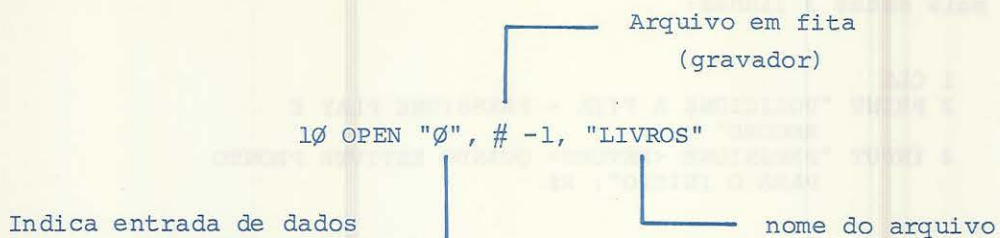
Faça alterações no seu Programa anterior incluindo caracteres coloridos e execute.

(Ao final do Manual, nos anexos você encontrará mais explicações sobre os caracteres gráficos)

COMO PASSAR PARA A FITA SUA LISTA DE DISCOS, LIVROS, ETC....

Para gravar qualquer tipo de dados para a fita do seu gravador auxiliar, você já sabe. Agora, para que estes dados sejam lidos conforme a sua necessidade, você precisará de um programa e alguns novos comandos que serão demonstrados agora.

Digite:



Tudo aquilo que for gravado na sua fita cassete auxiliar, como dados de informação, é considerado um ARQUIVO DE DADOS.

Cada ARQUIVO DE DADOS precisa de um identificador (um nome) para que o computador, ao acessá-lo, saiba onde é o início e o fim desse arquivo de dados.

Na linha acima digitada, o comando OPEN indica:

OPEN — Abertura do arquivo para a utilização dos dados.

-1 — Indica para o computador que o arquivo está no seu gravador auxiliar.

"LIVROS" — Indica o nome do arquivo.

Para que possamos gravar as informações no arquivo LIVROS, informe ao computador por meio de comandos esta operação.

Digite:

```
15 CLS : PRINT "ENTRE COM O TITULO DO LIVRO.
                PARA TERMINAR DIGITE <XX>"
20 INPUT "TITULO"; T$
30 PRINT # -1, T$
40 GOTO 15
```

Entrada do título do livro a ser armazenado.

Gravação do título no PERIFÉRICO # -1
(No caso o gravador de fitas K-7).

Adicione estas outras linhas:

```
15 IF T$ = "XX" THEN 50
50 CLOSE # -1.
```

Quando você entra com "XX" para terminar o arquivo, o computador desviará para a linha 50, comando CLOSE, e fechará o arquivo.

Adicione mais estas 3 linhas:

```
1 CLS
2 PRINT "POSICIONE A FITA - PRESSIONE PLAY E
  RECORD"
4 INPUT "PRESSIONE <RETURN> QUANDO ESTIVER PRONTO
  PARA O INICIO"; R$
```

Estas instruções para a gravação de arquivos em fita foram dadas pressupondo que o seu gravador estivesse conectado ao CD-6809. Caso não esteja, conecte-o.

Tudo pronto? ... ——— RUN

Quando você teclar RETURN o motor do seu gravador começará a funcionar, o arquivo será ABERTO (OPEN), você entrará com os dados até que seja digitado XX; após, o computador fechará o arquivo LIVROS.

Vamos agora adicionar algumas linhas para que possamos listar no vídeo o arquivo.

Digite:

```
60 CLS: PRINT "POSICIONE A FITA NO INICIO E PRESSIONE PLAY"
70 INPUT "PRESSIONE <RETURN> QUANDO ESTIVER PRONTO"; R$
80 OPEN "I", # -1, "LIVROS"
90 INPUT # -1, T$
100 PRINT T$
```

→ Neste caso o comando INPUT funciona como entrada de informações a partir da sua fita K-7 e não do seu teclado.

→ Imprimirá no vídeo o título do livro.

Adicione mais estas linhas:

```
85 IF EOF (-1) THEN 120
110 GOTO 85
120 CLOSE # -1
```

A linha 85 faz o teste do fim do arquivo. Se for fim o comando desviará para a linha 120 que contém instruções para o fechamento do arquivo.

Se você usar EOF (-1) em alguma linha comando antes do comando INPUT # -1, haverá erro (IE ERROR).

Digite LIST 6Ø- para listar no vídeo esta última parte do programa.

Dê uma analisada. Tudo certo?

Digite RUN 6Ø

↳ Isto indica ao computador que há uma execução a partir da linha 6Ø.

A seguir, uma condensação das informações dadas anteriormente sobre as operações de gravação e leitura de dados de um gravador K-7 auxiliar.

Para gravar dados na fita K-7:

1. OPEN — abrir o arquivo
2. PRINT — gravar dados no arquivo
3. Continuar a gravação até o final dos dados
4. CLOSE — fechar o arquivo

Para ler dados da fita K-7:

1. OPEN — abrir o arquivo
2. Usar o comando IF de teste para verificar se o arquivo chegou ao fim. EOF (-1).
3. INPUT — para a recepção dos dados da fita para a memória.
4. Continuação da recepção até que seja fim de arquivo.
5. CLOSE — para encerrar a comunicação (fechar o arquivo).

Como teste geral sobre as operações de criação e leitura dos dados de um arquivo, digite:

```

5 CLS: PRINT "POSICIONE A FITA - PRESSIONE PLAY E RECORD"
7 INPUT "TECLE <RETURN> QUANDO ESTIVER PRONTO"; R$
10 OPEN "O", # -1, "CHEQUES"
15 CLS: PRINT "ENTRADA DOS CHEQUES - TECLE <XX> PARA TERMINAR"
20 INPUT "NUMERO:"; N$
25 IF N$ = "XX" THEN 90
30 INPUT "DATA:"; D$
40 INPUT "PAGAVEL PARA:"; P$
50 INPUT "CONTA:"; S$
60 INPUT "VALOR: $"; A
70 PRINT # -1, N$, D$, P$, S$, A
80 GOTO 15
90 CLOSE # -1
92 CLS: T = 0
95 INPUT "QUAL CONTA"; B$
100 PRINT "POSICIONE A FITA - PRESSIONE PLAY"
110 INPUT "TECLE <RETURN> QUANDO ESTIVER PRONTO"; R$
120 OPEN "I"; # -1, "CHEQUES"
130 IF EOF (-1) THEN 170
140 INPUT # -1, N$, D$, P$, S$, A
150 IF B$ = S$ THEN T = T + A
160 GOTO 130
170 CLOSE # -1
180 PRINT "TOTAL DE GASTOS -" B$, "LS $" T

```

Antes de fazermos um resumo de todos os comandos, mensagens de erro e outras informações importantes dadas até agora, vamos mostrar mais alguma coisa em termos de instrução e programas. Vejamos:

Quando precisarmos tornar um STRING uma expressão numérica usaremos o comando VAL (X\$).

Exemplo:

Se X\$ = "5" então VAL (X\$) = 5

Se X\$ = "8" então VAL (X\$) = 8

No comando de teste "IF" podemos ter estes tipos de comparações:

IGUALDADE "="

Ex.: IF A = B

MAIOR QUE ">"

Ex.: IF A > B

MENOR QUE "<"

Ex.: IF A < B

DIFERENTE "<>" ou "><"

Ex.: IF A <> B

MAIOR OU IGUAL ">=" ou "=<"

Ex.: IF A >= B

MENOR OU IGUAL "<=" ou "=<"

Ex.: IF A <= B

Isto nos possibilitará uma maior versatilidade nos testes para desvio de linhas dentro de um programa.

IMPRESSORA

Como já foi dito no início deste Manual, você tem a facilidade de conectar ao seu microcomputador CD-6809, uma impressora auxiliar. Ela pode servir a diversos fins.

Para que o computador identifique o tipo de SAÍDA que você estiver desejando no desenvolvimento de alguma tarefa, use:

-2

E lembre-se, para a listagem direta de informações do computador (vídeo/impressora) use o comando LLIST.

Este é o exemplo de um programa que utiliza a impressora como SAÍDA de informações:

Digite este pequeno programa:

```
1Ø INPUT A$  
2Ø PRINT # -2, A$
```

└───────────> Identificação da impressora

Agora digite:

```
LLIST RETURN
```

Se a sua impressora estiver conectada e ligada, o seu programa será listado.

Agora execute o Programa — RUN

O que você digitar (linha 1Ø INPUT) será listado na impressora, pois o código # -2 indica ao computador que a SAÍDA é na impressora.

Você tem a opção de obter a saída de informações listadas na impressora em letra minúscula. Para tanto, siga estas instruções:

- Quando em seu programa houver entrada de dados (informações) pelo teclado, deixe o teclado em minúsculo (SHIFT Ø).
- Dê RUN no programa (o teclado deverá estar em MAIÚSCULO).
- A saída da listagem será em caracteres minúsculos.

Tente alterar algum dos seus Programas para que os dados de saída (PRINT) sejam impressos.

O COMPUTADOR FALANTE

Se o seu gravador K-7 estiver conectado poderemos começar as instruções, senão conecte-o.

- Posicione uma fita sem dados (limpa) no início e pressione PLAY e RECORD simultaneamente; conecte um microfone e grave algumas palavras.
- Posicione novamente a fita; agora digite:

```
5 CLS
1Ø INPUT "TECLE <RETURN> PARA OUVIR A GRAVAÇÃO"
2Ø MOTOR ON
3Ø AUDIO ON
```

- Pressione PLAY no gravador
- Dê volume na TV auxiliar (vídeo)
- Execute o programa → RUN

Você ouvirá o que foi gravado no áudio da sua TV.

Adicione mais estas linhas:

```
35 CLS
4Ø A$ = INKEY$
5Ø PRINT @225, "TECLE <X> PARA DESLIGAR O GRAVADOR"
```

```
6Ø IF A$ <> "X" THEN 4Ø
7Ø AUDIO OFF
8Ø MOTOR OFF
```

Prepare o gravador e → RUN

Após o aprendizado destes novos comandos, você já poderá dispor do seu gravador auxiliar para uma série de utilizações.

Como exemplo temos:

- Sincronismo de mensagens previamente gravadas com programas que tenham movimento.
- Simulação de voz para o computador, etc....

USO DO JOYSTICK:

Se você tiver disponíveis os JOYSTICKS (controles), poderá seguir estas novas instruções.

Para iniciar, conecte os JOYSTICKS ao seu microcomputador e digite este pequeno programa para uma demonstração da operação com este novo tipo de acessório.

```
1Ø CLS
2Ø PRINT @Ø, JOYSTK (Ø);
3Ø PRINT @5, JOYSTK (1);
4Ø PRINT @1Ø, JOYSTK (2);
5Ø PRINT @15, JOYSTK (3);
6Ø GOTO 2Ø
```

Execute o programa → RUN

Neste programa temos 4 posições do JOYSTICK operando.

- Movimento o eixo central do seu JOYSTICK, da direita. Acompanhe as alterações no vídeo. Se você movimentar o seu JOYSTICK da esquerda poderá observar também os efeitos da movimentação do comando central.

Nos comandos PRINT das linhas 2Ø e 3Ø do seu Programa temos:

Linha 2Ø — As instruções na horizontal

Linha 3Ø — As instruções na vertical

O mesmo acontece com as linhas 40 e 50, só que estas se relacionam com o controle (JOYSTICK) da esquerda.

Limpe a memória e digite:

```
10 CLS (0)
20 H = JOYSTK (0)
30 V = JOYSTK (1)
40 IF V > 31 THEN V = V - 32
80 SET (H, V, 3)
90 GOTO 20
```

Execute o programa → RUN

Mova o controle do seu JOYSTICK da direita lentamente, e tente formar algo na tela.

A linha 80 colocará um ponto azul em H (controle do horizontal do JOYSTICK) e em V (controle vertical).

A linha 90 proporcionará o aparecimento dos próximos pontos, horizontal e vertical, daí o efeito de "pintura na tela".

Adicione mais estas linhas no programa:

```
50 C = JOYSTK (2)
60 IF C < 31 THEN C = 3
70 IF C > = 31 THEN C = 4
80 SET (H, V, C)
```

Dê RUN.

Mova seu JOYSTICK esquerdo para a direita e o computador assumirá que C = 3. Isto fará os pontos da tela na cor vermelha. Movendo para a esquerda o valor de C será = 4; os pontos serão azuis.

Para usar o BOTÃO do seu JOYSTICK, adicione as próximas linhas:

```
100 P = PEEK (65280)
110 PRINT P
120 GOTO 100
```

e digite

RUN 100

O comando PEEK indica ao computador que assumo o valor que se encontra na "locação" 65280 de sua memória. Se você não tiver acionado o botão do JOYSTICK, os números deverão ser 255 ou 127.

Pressionando o botão direito os números modificarão para 126 ou 254.

Pressionando o botão esquerdo os números contados na memória (locação 65280), serão 125 ou 253.

Vamos fazer uma troca.

Digite:

```
110 IF P = 126 THEN 10
120 IF P = 254 THEN 10
```

Delete a linha 90 e adicione a linha

```
130 GOTO 20
```

LEMBRE-SE:

Para deletar uma linha basta teclar
o número da linha e dar **RETURN**

JOGOS NO VÍDEO.

CAMINHO ENTRE ASTERÓIDES

Este é o nosso próximo programa.

Neste jogo vamos usar o JOYSTICK da direita, portanto conecte-o.

Nós vamos criar asteróides que circularão na tela.

Limpe a memória → NEW ← Digite:

```
5 CLS (Ø)
1Ø FOR X = 1 TO 2ØØ
2Ø SET (RND(64) - 1, RND (3Ø) + 1,8)
3Ø NEXT X
4Ø FOR H = 54 TO 63
5Ø FOR V = 28 TO 31
6Ø SET (H, V, 3)
7Ø NEXT V, H
```

Para a atuação dos JOYSTICKS digite:

```
1ØØ A = JOYSTK (Ø)
11Ø B = JOYSTK (1)
12Ø B = B/2
13Ø B = INT (B)
2ØØ IF INT (A/2) <> A/2 THEN A = A -1
21Ø IF INT (B/2) <> B/2 THEN B = B -1
22Ø FOR H = A TO A + 1
23Ø FOR V = B TO B + 1
24Ø SET (H, V, 6)
25Ø NEXT V, H
999 GO TO 1ØØ
```

Execute o programa. RUN

Movimente o seu JOYSTICK. A coloração azul claro se movimentará em forma de uma linha, conforme o posicionamento do seu JOYSTICK.

Interrompa o jogo e digite:

```
212 FOR H = A TO A + 1
214 FOR V = B TO B + 1
```

```

216 IF POINT (H, V) = 8 THEN SOUND 128,1: T = T + 1
218 NEXT V, H

Execute. RUN

```

→ Cada vez que aparecer o ponto laranja o computador emitirá um SOM.

Para completar o programa, digite estas linhas:

```

235 IF POINT (H, V) = 3 THEN PRINT @ Ø, "SAUDAÇÕES": END
300 PRINT @ 28, T
310 IF T > 10 THEN 1000
1000 FOR X = 1 TO 40
1010 CLS (RND(8))
1020 SOUND RND (255), 1
1030 NEXT X
1040 PRINT @ 228, "SUA NAVE ESPACIAL EXPLODIU"

RUN

```

Para que haja mais um incremento no Programa, digite:

```

80 FOR X = 1 TO 8
82 READ A$
84 PRINT @ Ø, A$
86 FOR Y = 1 TO 1500: NEXT Y
88 NEXT X
90 R$ = INKEY$: IF R$ = "" THEN 90
92 FOR H = 4 TO 63
94 SET (H, Ø, 8): SET (H, 1, 8)
96 NEXT H
2000 DATA O SEU OBJETIVO E MAPEAR UM CURSO
2010 DATA PARA GUIAR SEU FOGUETE
2020 DATA ATRAVES DOS ASTEROIDES
2030 DATA ATE O PLANETA AZUL
2040 DATA SE MAIS DE 10 ASTEROIDES LHE ACERTAREM
2050 DATA SEU FOGUETE EXPLODIRÁ!!!
2060 DATA DIGITE QUALQUER TECLA PARA QUE
2070 DATA SEU FOGUETE SE POSICIONE NO TOPO SUPERIOR ESQUERDO

RUN ——— BOM JOGO !

```

Daremos a seguir uma série de programas para você montar.

Em alguns aparecerá o comando CLEAR na primeira linha. Este comando é usado para que o computador deixe disponível a maior área de memória possível para a execução do seu programa. Isto pode se tornar necessário, quando o programa for muito extenso ou os dados de entrada (comando DATA) sejam em grande número.

Estes programas são ótimos para que, em forma de exercícios, você se familiarize ainda mais com o seu microcomputador.

Vamos lá!

PROGRAMAS

CANHÃO ESPACIAL

```
10 CLEAR 1000
20 FOR Y = 0 TO 1
30 C = (Y+1)*16
40 S$(Y) = CHR$(131+C)+CHR$(139+C)+CHR$(130+C)
50 S2$(Y) = CHR$(128+C)+CHR$(136+C)
60 NEXT Y
100 FOR Y = 0 TO 1
105 C = JOYSTK(0)
110 A(Y) = JOYSTK(0+Y*2)
120 B(Y) = JOYSTK(1+(Y*2))
130 IF A(Y) > 59 THEN A(Y) = 59
140 B(Y) = INT(B(Y)/4)*4
150 L(Y) = B(Y) * 8 + INT(A(Y)/2)
160 IF L(Y) >= 480 THEN L(Y) = L(Y) - 32
170 NEXT Y
180 CLS(0)
190 FOR Y = 0 TO 1
200 PRINT @ L(Y), S$(Y);
210 PRINT @ L(Y)+32, S2$(Y);
220 NEXT Y
500 P = PEEK(65280)
510 IF P = 125 OR P = 253 THEN GOSUB 1000
530 GOTO 100
800 REM
900 REM      ROTINA DE TIRO
1000 V1 = INT(B(1)/2)+1
1010 H1 = A(1) + 2
1020 IF A(1) > A(0) THEN 1100
1030 FOR H = H1 + 3 TO 63
1040 IF POINT(H,V1) = 2 THEN SOUND 100,2
1050 SET(H,V1,4)
1060 IF H <= H1 + 4 THEN 1080
1070 RESET(H-2, V1)
1080 NEXT H
1090 RETURN
1100 FOR H = H1 TO 4 STEP -1
1110 IF H = H1 THEN 1160
1120 IF POINT(H-4,V1)=2 THEN SOUND 100,2
1130 SET(H-4,V1,4)
1140 IF H >= H1 - 2 THEN 1160
1150 RESET(H-2,V1)
1160 NEXT H
1170 RETURN
```

CALEIDOSCÓPIO

```
10 CLS 0
20 X = RND (57)
30 Y = RND (25)
40 C = RND (8)
50 FOR I = X TO X+6
60 SET (I, Y, C)
70 SET (I, 31-4, C)
80 SET (63-I, 4, C)
90 SET (63-I, 31-4, C)
```

```

100 Y = Y + 1
110 NEXT I
120 S = RND (20)
130 IF S=1 THEN FOR I=1 TO 600: NEXT: GOTO 10
140 GOTO 20

```

DADO ELETRÔNICO

```

4 CLEAR 2000
5 CLS (3)
6 DIM DB$(6)
8 DIM DF(21), P(6), D$(6)
10 REM      FACES DO DADO
20 FOR X = 1 TO 21
30 READ DF(X)
40 NEXT X
50 DATA 39
60 DATA 14, 64
70 DATA 14, 39, 64
80 DATA 14, 20, 58, 64
90 DATA 14, 20, 39, 58, 64
100 DATA 14, 20, 36, 42, 58, 64
105 FOR X = 1 TO 7
110 REM
130 FOR X = 1 TO 6
140 READ P(X)
150 NEXT X
160 DATA 1, 2, 4, 7, 11, 16
165 REM
175 FOR X = 1 TO 6
180 M = P(X)
185 FOR Y = 1 TO 7
190 FOR Z = 1 TO 11
192 IF (Y-1)*11+Z <> DF(M) THEN 200
194 D$(X) = D$(X) + CHR$(128)
196 M = M + 1
197 IF M = 22 THEN M = 0
198 IF M = X THEN M = 0
199 GOTO 230
200 D$(X) = D$(X) + CHR$(143+96)
230 NEXT Z
240 FOR Z = 0 TO 31-11
250 D$(X) = D$(X) + CHR$(143+32)
260 NEXT Z
270 NEXT Y, X
480 REM
490 REM      ROLAR O DADO
500 FOR T = 1 TO 10
510 A=RND(6): B = RND(6)
520 PRINT @ 35, D$(A);
530 PRINT @ 273, D$(B);
540 NEXT T
550 PRINT @ 113, "DIGITE";
560 PRINT @ 145, "QUALQUER TECLA";
570 K$=INKEY$: IF K$="" THEN 570
580 GOTO 500

```

PROGRAMAS COM EMPREGO DE SONS

Nº 1:

```

5 DIM A(25), S$(13), B(200): Y=1
10 FOR X = 1 TO 25: READ A(X): NEXT X
20 DATA 89, 99, 108, 117, 125
30 DATA 133, 140, 147, 153, 159

```

```

40 DATA 165, 170, 176, 180, 185
50 DATA 189, 193, 197, 200, 204
60 DATA 207, 210, 213, 216, 218
70 FOR X = 1 TO 13: READ S$(X): NEXT X
80 DATA A,W,S,E,D,F,T,G,Y,H,U,J,K
90 CLS
92 PRINT @ 167, "COMPONHA SUA CANCAO";
94 PRINT @ 292, "TECLE <X> PARA TERMINAR"
100 P$ = INKEY$
110 IF P$ = "" THEN 100
115 FOR X = 1 TO 13
120 IF P$ <> S$(X) THEN 150
130 SOUND A(X), 5
140 B(Y) = X
145 Y = Y + 1
150 NEXT X
160 IF P$ <> "X" THEN 100
165 CLS
170 PRINT @ 264, "QUAL TECLA (1-11)";
176 INPUT K
180 FOR X = 1 TO Y-1
190 SOUND A(B(X)+K), 5
200 NEXT X
210 GOTO 165

```

Nº 2:

```

10 DIM M(50), T(8)
20 FOR B = 1 TO 8
30 READ T(B)
40 NEXT B
50 X = 1
60 M(X) = RND(8)
70 FOR Y = 1 TO X
80 CLS(M(Y))
90 PRINT @ 239, M(Y);
100 SOUND T(M(Y)), 8
110 NEXT Y
120 CLS
140 FOR Y = 1 TO X
150 T = 1
160 K$ = INKEY$
170 T = T + 1
180 IF T > 150 THEN 310
190 IF K$ = "" THEN 160
200 K = VAL(K$)
210 IF K <> M(Y) THEN 310
220 CLS(K)
230 PRINT @ 239, K;
240 SOUND T(K), 3
250 NEXT Y
260 X = X + 1
270 CLS
280 FOR T = 1 TO 500: NEXT T
290 GOTO 60
300 CLS(0)
310 SOUND 1, 25
320 DATA 89, 108, 125, 133, 147, 159, 170, 176

```

COMPOSITOR MUSICAL

```

10 INPUT "PERIODO(1-10)"; M
20 M = M*4
30 INPUT "RITMO (1-4)"; T1
40 IF T1 = 4 THEN 60
50 T = T1 : GOTO 70
60 T = 8
70 FOR K = 1 TO M*8

```

```
80 GOSUB 1000
90 B = RND(3) * T
100 SOUND P, B
110 CLS(S)
120 NEXT K
130 IF RND(10) <=8 THEN 150
140 SOUND 125, 16*T
145 END
150 SOUND 90, 16*T
160 END
1000 X = RND(100)
1010 IF X <= 20 AND X <=25 THEN P = 90 : S = 1
1020 IF X > 20 AND X <=25 THEN P = 108 : S = 2
1030 IF X > 25 AND X <= 40 THEN P = 125 : S = 3
1040 IF X > 40 AND X <= 55 THEN P = 133 : S = 4
1050 IF X > 55 AND X <= 75 THEN P = 147 : S = 5
1060 IF X > 75 AND X <= 85 THEN P = 159 : S = 6
1070 IF X > 85 AND X <= 95 THEN P = 176 : S = 7
1080 IF X > 95 THEN P = 58 : S = 8
1090 RETURN
```


A partir desta parte do nosso Manual, começaremos a desenvolver programas para diversas aplicações, bem mais complexos que os anteriores. O máximo de atenção é necessário para a assimilação das novas instruções.

Como os programas terão um número bem maior de comandos, a probabilidade de erros ao digitar as instruções aumenta.

Para que seus programas não acusem erro de digitação (sintaxe) ao serem executados, experimente sempre que terminar a montagem de um programa, listar o mesmo no vídeo, ou na impressora, caso você a tenha conectado. Havendo algum erro, lembre-se do que foi ensinado anteriormente sobre o comando EDIT, o mesmo possibilita a alteração de uma linha determinada.

ATENÇÃO:

Os comandos que começaremos a usar agora, salvo indicação, são comandos para a montagem de gráficos.

Bem, vamos começar. Digite:

```
5 PMODE 1,1
1Ø PCLS
2Ø SCREEN 1,1
3Ø PSET (1Ø, 2Ø, 8)
4Ø GOTO 4Ø
```

Agora execute o programa → RUN

O resultado deverá ser uma tela com a cor branca e no canto esquerdo superior um ponto de cor laranja. O ponto é o resultado do comando PSET da linha 30.

Quando você precisar de um ponto colorido em qualquer lugar da tela, é o comando PSET a instrução a ser usada.

PSET (X, Y, C)

X — Indica a posição horizontal de 0 a 255

Y — Indica a posição vertical de 0 a 191

C — Indica a cor desejada, conforme código numérico de 0 a 8.

ATENÇÃO.

Nunca inverta as informações dentro do parêntesis.

Se você quiser colocar um ponto no centro da tela, anexe esta linha:

35 PSET (128, 96, 8)

Para um terceiro ponto, com a mesma cor dos dois (2) primeiros, no canto inferior direito, anexe esta linha:

33 PSET (255, 191, 8)

Dê RUN. Independente da opção que você tomou, aparecerá um novo ponto na tela.

Você tem nove opções de cores no seu microcomputador. Elas são:

Preto

Verde

Amarelo

Azul

Vermelho

Branco

Azul-claro

Violeta

Dependendo do ajuste de cores que você fizer em sua TV, as cores poderão sofrer alteração de tonalidade.

Através do comando visto anteriormente, o PSET, podemos criar um PONTO na tela. Ao contrário, o comando PRESET fará este ponto desaparecer.

PSET	—	POINT SET
PRESET	—	POINT RESET

Estes dois comandos têm relação com os vistos anteriormente, SET e RESET, lembre-se!

PRESET (X, Y)

X — Especifica a posição horizontal de 0 até 255.

Y — Especifica a posição vertical de 0 até 191.

No comando PRESET não há especificação da cor, pois o computador irá assumir a cor base da tela.

Os comandos PMODE e SCREEN usados nos exemplos anteriores, ficaram sem explicação. Para os mesmos teremos material mais adiante.

Você se lembra da função RND (RANDÔMICO) usada anteriormente neste manual? Se não, localize-a no ÍNDICE e dê uma olhada.

Vamos usar o RND no programa seguinte.

Digite:

```
5 PMODE 1,1
10 PCLS
20 SCREEN 1,1
```

```
3Ø X = RND (256)-1
4Ø Y = RND (192)-1
5Ø C = RND (9)-1
6Ø PSET (X, Y, C)
7Ø GOTO 3Ø
```

Este programa criará uma série de pontos coloridos na tela.

Vamos executá-lo. Dê RUN.

Agora substitua a linha 7Ø e inclua as seguintes:

```
7Ø IF X = 158 THEN 9Ø
8Ø GO TO 3Ø
9Ø PRESET (X, Y)
1ØØ GO TO 3Ø
```

Execute o programa.

O resultado será o desaparecimento de alguns pontos (linha 9Ø) criados na linha 6Ø.

Caso você deseje aumentar o efeito do comando PRESET, incremente a linha 7Ø com outros valores para a variável "X".

LINE

Já sabemos como colocar na tela pontos em posições diferentes. Agora vamos fazer algumas alterações. Para isto vamos usar o primeiro programa desta parte. Faça somente a alteração da linha 3Ø.

Digite:

```
5 PMODE 1, 1
1Ø PCLS
2Ø SCREEN 1, 1
3Ø LINE (Ø, Ø) - (255, 191), PSET
4Ø GOTO 4Ø
```

RUN

O resultado será uma linha laranja que cruza o vídeo do canto superior esquerdo até o canto inferior direito.

Mude novamente a linha 3Ø. Digite:

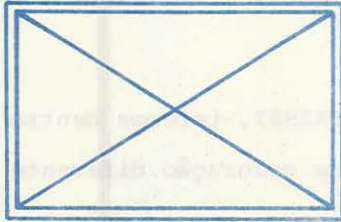
```
3Ø LINE (Ø, 191) - (255, Ø), PSET
```

e acrescente a linha 25. Digite:

```
25 LINE (Ø, Ø) - (255, 191), PSET
```

→ RUN

Agora teremos:



um X na tela

```
LINE (X1, Y1) - (XA, YZ), a, b
```

(X1, Y1) — Especifica o ponto do início da linha.

X1, é uma expressão numérica de Ø até 255.

Y1, é uma expressão numérica de Ø até 191.

(X2, Y2) — Especifica o ponto de fim da linha.

X2, é uma expressão numérica de Ø até 255.

Y2, é uma expressão numérica de Ø até 191.

a indica PSET ou PRESET

b indica B ou BF. (VALOR OPCIONAL).

IMPORTANTE:

Se os pontos de início (X1, Y1) não forem especificados, o computador iniciará a nova linha no fim da última linha especificada.

Exemplo:

Se o ponto final da sua linha era (128, 96), este mesmo ponto será o início da nova linha.

Se você não especificar as coordenadas de início (X1, Y1) o que indicará ao

computador qual tipo de ação a tomar será o hífen (-); portanto ele não pode ser omitido.

Vamos dar mais uma olhada nas duas linhas do programa anteriormente montado:

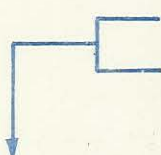
```
25 LINE (Ø, Ø) - (255, 191), PSET
3Ø LINE (Ø, 191) - (255, Ø), PSET
```

Estas instruções indicam comandos LINE que pela função PSET criam linhas na tela.

Se ao invés da função PSET usarmos PRESET, teremos dentro das coordenadas indicadas a cor de fundo, isto é, se houver alguma coloração diferente da cor de fundo, ela será retirada.

Vamos fazer um teste, digite:

```
36 FOR I = 1 TO 1ØØØ: NEXT
37 LINE (Ø, Ø) - (255, 191), PRESET
38 FOR A = 1 TO 1ØØØ: NEXT
39 LINE (Ø, 191) - (255, Ø), PRESET
```



Estas duas linhas foram anexadas somente para que haja um pouco mais de demora no processamento, propiciando uma melhor visualização na tela.

EXECUTE O PROGRAMA

Agora vamos ver qual são os efeitos das funções B e BF do comando LINE.

(Primeiro delete alguns passos do programa anterior DEL 36-39 RETURN)

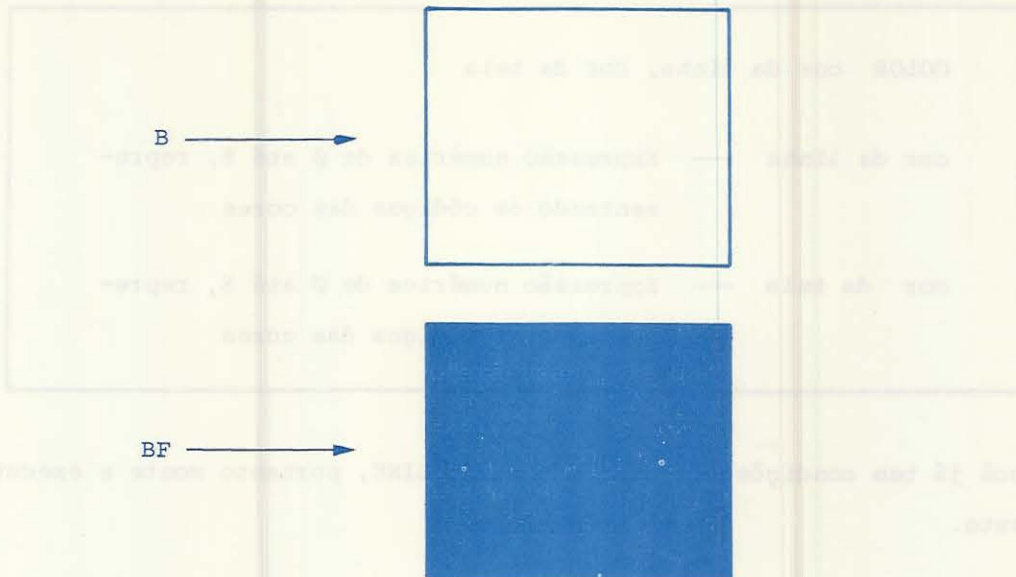
- Quando for usado o B, em lugar da linha especificada no comando, teremos um "quadrado" tendo por lado a dimensão e a cor da linha.

Altere as linhas 25 e 3Ø para:

```
25 LINE (Ø, Ø) - (255, 191), PSET, B
3Ø LINE (Ø, 191) - (255, Ø), PSET, B
```

- Quando for usado o BF, em lugar da linha especificada no comando, teremos um "quadrado" tendo por lado a dimensão da linha. A coloração interna do quadrado será a mesma da linha.

Altere as linhas 25 e 3Ø digitando BF ao invés de B.



Faça um teste você mesmo. Tente primeiro o B e depois o BF e veja o resultado.

Tenha sempre em conta, que estas novas informações podem servir para a montagem de diversos tipos de gráficos imaginados por você.

COLOR

Agora que estamos usando comandos para a formação de gráficos temos uma novidade: A COR DE FUNDO DA TELA.

Esta nova coloração depende de parâmetros de comandos que serão explicados mais adiante.

Vamos a um comando que você poderá usar para que seja definida uma certa coloração para a linha e para o fundo da tela.

Digite:

```
6 COLOR 2, 3
```

e execute o programa.

Teremos uma base de tela amarela e a linha em azul.

COLOR cor da linha, cor da tela

cor da linha — Expressão numérica de 0 até 8, representando os códigos das cores

cor da tela — Expressão numérica de 0 até 8, representando os códigos das cores

Você já tem condições de usar o comando LINE, portanto monte e execute um programa de teste.

Monte uma casa com porta da frente, janela,
e uma chaminé (não se esqueça da fumaça) !

Para que haja uma maior facilidade, você poderá usar os moldes gráficos que acompanham este Manual (lembre-se da função "BF" do comando LINE).

Após a montagem do programa, salve-o na fita K-7 para utilizá-lo posteriormente.

A resposta ao programa pedido está no final deste Manual sob o código: TESTE
1-1.

ATENÇÃO: Esta resposta para o teste pedido é apenas uma sugestão, pois existem várias maneiras de demonstrar o desenho solicitado.

PMODE

Como falamos anteriormente vamos agora a uma explicação sobre o comando PMODE.

Tomemos como base o programa exemplo (o mesmo utilizado para o tópico LINE)

```
5 PMODE 1,1
10 PCLS
20 SCREEN 1,1
25 LINE (0, 0) - (255, 191), PSET
30 LINE (0, 191) - (255, 0), PSET
40 GOTO 40
```

Na linha 5 temos o mais importante comando para a formação de gráficos no nosso microcomputador CD-6809.

```
5 PMODE 1,1
```

Experimente alterar esta linha para

```
5 PMODE 2, 1      RUN
```

```
ou 5 PMODE 3, 1      RUN
```

PMODE modo, nº de páginas de memória





modo — é uma expressão numérica de 0 até 4, ela é opcional. Quando não especificado o valor 2 é assumido. Indica o tipo de tela gráfica.

Nº de páginas de memória — é uma expressão numérica de 1 até 8. Cada unidade especifica uma página de memória com 1.5K BYTES. Se for omitido, será alocado o tamanho padrão de área de memória.

NOTA: Se PMODE for omitido, o computador assumirá os valores PMODE 2, 1

O comando PMODE não é necessário se você for usar a TELA com programas de textos.

TABELA DE RESOLUÇÃO DOS GRÁFICOS

PMODE	ÁREA DA TELA	Nº DE CORES	PÁGINAS USADAS	TAMANHO DO PONTO
4	256 x 192	2 cores	4	
3	128 x 192	4 cores	4	
2	128 x 192	2 cores	2	
1	128 x 96	4 cores	2	
0	128 x 96	2 cores	1	

O seu microcomputador divide a área da tela de sua TV em 256 pontos na horizontal e 192 pontos na vertical.

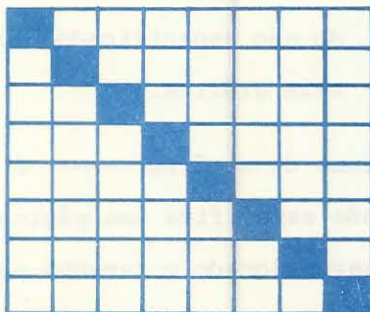
Este tipo de divisão é considerado normal se você estiver usando gráficos de alta resolução.

"PMODE 4" indica gráficos de alta resolução.

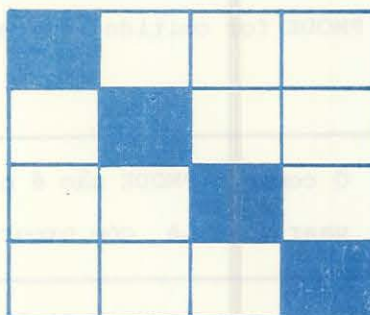
O seu computador tem 5 tipos de PMODE (0-4), cada um com um tipo diferente de capacidade.

Podemos ter dois tipos de resolução de tela:

ALTA RESOLUÇÃO



BAIXA RESOLUÇÃO



Dê uma olhada no gráfico anteriormente citado. (Tabela de resolução dos gráficos.)

Podemos fazer uma relação que se tornará satisfatória para o melhor entendimento das resoluções:

Imagine um lápis com a ponta grossa e um com a ponta fina.

Os traços feitos por estes dois lápis serão de espessuras diferentes:



A primeira (1) será de baixa resolução e a segunda (2) será de alta resolução.

Agora troque a linha 5:

```
5 PMODE 4, 1
```

```
RUN
```

A linha se tornará mais fina, pois estamos usando a alta resolução no gráfico.

TABELA DE CORES			
PMODE	CORES (veja comando do SCREEN)	DUAS COMBINAÇÕES DE CORES	QUATRO COMBINAÇÕES DE CORES
4	∅	preto/verde	-
	1	preto/branco	-
3	∅	-	verde/amarelo/azul/vermelho
	1	-	branco/azul claro/violeta/laranja
2	∅	preto/verde	-
	1	preto/branco	-
1	∅	-	verde/amarelo/azul/vermelho
	1	-	branco/azul claro/violeta/laranja
∅	∅	preto/verde	
	1	preto/branco	

Com o código 4 de resolução, você só terá a opção de DOIS tipos de combinação de cores.

Se o comando SCREEN da linha 2∅ for SCREEN 1,1 você terá a combinação de PRETO e BRANCO (PRETO para a base da tela e BRANCO para a linha).

Se o comando SCREEN estivesse com:

```
2∅ SCREEN 1,∅
```

a combinação seria PRETO e VERDE.

(PRETO para a base da tela e VERDE para a linha).

Isto indica que devemos seguir certos padrões. Por exemplo:

Se você quiser usar a cor vermelha não poderá ter quadros com as resoluções 4, 2 e 0 e, além disto, o comando SCREEN deverá estar com o seu segundo número de função (SCREEN n,n) com o valor 0 (zero).

Para tirar as dúvidas dê uma olhada na tabela de cores.

PCLS

Como são assumidas pelo computador as cores das linhas:

— comando PCLS

PCLS cor

cor → Expressão numérica com valores de 0 até 8 representando o código de cores.

Quando o valor for omitido a cor padrão será assumida para a tonalidade da tela.

Anteriormente o comando de cor da tela PCLS estava sem nenhum valor numérico de colorido.

As cores assumidas no caso eram:

Para a tela → BRANCO

Para as linhas → LARANJA

Se você trocar a linha 10 por:

→ 10 PCLS 6

RUN

a cor da tela de fundo será o AZUL CLARO, as linhas continuarão em LARANJA.

O comando PCLS se assemelha ao CLS, pois ele limpa a tela devolvendo a coloração nele indicada.

PCLEAR

Quando você estiver utilizando o seu computador para a configuração de gráficos com alguma complexidade, a área de memória a ser utilizada crescerá.

Normalmente o computador alocará uma área de 4 páginas na memória para os seus programas com comandos para gráficos. (Uma página contém 1536 locações de memória ou $\pm 1.5K$ BYTES)

Para que haja um aumento desta área usaremos o comando PCLEAR no início do programa.

Este comando alocará mais área de memória para o seu programa. (De uma até oito páginas).

Os problemas que podem surgir por falta de espaço na memória serão testados na execução do programa a seguir.

Digite, ou simplesmente altere o conteúdo da memória:

```
5 PMODE 3, 2
10 PCLS
20 SCREEN 1, 1
30 LINE (110, 20) - (120, 30), PSET, B
40 GOTO 40
```

Execute o programa.

```
? FC ERROR IN 5
```

Qual o erro? O comando PMODE com a função de modo igual a 3 necessita de 4 páginas de memória.

Quando você especificou 2 na função de nº de páginas, o computador tentará alocar as 4 páginas requeridas normalmente duas vezes. Daí o erro, pois somente quatro (4) páginas de memória são alocadas.

Quando houver necessidade de páginas extras para a execução de seus programas, você poderá recorrer ao comando PCLEAR; ele alocará de 1 até 8 páginas de memória para

o processamento.

Este comando deverá ser o primeiro ou segundo a aparecer em seu programa. (Se o comando CLEAR for usado, PCLEAR deverá vir logo após).

Digite:

```
4 PCLEAR 8
```

com isto você estará alocando as 8 páginas necessárias para a execução do seu programa.

```
PCLEAR n
```

n — é uma expressão numérica com valores de 1 até 8, que especifica o nº de páginas reservadas para gráficos na memória.

NOTA: Quando você for executar programas com comandos para gráficos, o computador irá gerar automaticamente um comando PCLEAR 4. Com isto fica bem definido que este comando somente deverá ser usado quando necessitarmos de um número diferente de área a ser alocada.

A linha 4 PCLEAR 8 já está inserida no programa? Então vamos executá-lo.

Após a execução limpe a memória e digite:

```
10 PCLEAR 8
20 FOR P = 1 TO 8 'P = PAGINAS, USAR TODAS AS 8
30 PMODE 0, P 'USAR UMA PAGINA DE CADA VEZ
40 PCLS
50 LINE (128, 0) - (138, 10 + (P-1) * 15), PSET
60 CIRCLE (128, P * 15), 15
70 NEXT P
80 FOR P = 1 TO 8: GOSUB 110: NEXT P
90 FOR P = 7 TO 1 STEP - 2: GOSUB 110: NEXT P
100 GO TO 80
110 PMODE 0, P
120 SCREEN 1, 0
130 FOR T = 1 TO 10: NEXT T
140 RETURN
```

Comentário ao comando

O processo de execução das imagens mostradas na tela é idêntica a de um dese-

nho animado; há uma repetição de mais ou menos 12.000 imagens diferentes propiciando daí o movimento.

Execute → RUN

PCOPY

PÁGINAS DE MEMÓRIA

Para copiar o conteúdo de uma página de memória de gráficos para uma outra página usa-se o comando PCOPY.

PCOPY de TO para

de e para são expressos numéricas de 1 até 8, indicando a página da memória.

NOTA: O comando PCOPY só poderá ser usado para páginas de número anteriormente reservado. Exemplo:

Após um PCLEAR 4, você só poderá copiar de 1 até 4 páginas.

Se você quiser copiar os gráficos da página 3 para a 8 digite:

```
PCOPY 3 TO 8
```

Uma das vantagens do PCOPY é a diminuição do tamanho dos programas eliminando daí as repetições.

O seguinte programa deverá sofrer uma redução de espaço de memória pela inclusão do comando PCOPY. Você deverá fazer esta redução e executar o programa.

```
4 PCLEAR 8
5 PMODE 3,4
10 PCLS
11 SCREEN 1,1
12 LINE (110, 20) - (120, 30), PSET, B
20 PMODE 3,3
21 SCREEN 1,1
22 LINE (110, 20) - (120, 30), PSET, B
30 PMODE 3,2
31 SCREEN 1,1
```

```
32 LINE (11Ø, 2Ø) - (12Ø, 3Ø), PSET, B
4Ø PMODE 3,1
41 SCREEN 1,1
42 LINE (11Ø, 2Ø) - (12Ø, 3Ø), PSET, B
5Ø GOTO 5Ø
```

A resolução deste problema está ao final deste Manual sob o código: TESTE 2-1.

PPOINT

A seguir temos o comando PPOINT.

O último comando a ser conhecido no que se relacione a comandos P (página) é o PPOINT.

PPOINT (X, Y)

X — Especifica as coordenadas X (horizontais) e é uma expressão numérica de Ø até 255.

Y — Especifica as coordenadas Y (verticais) e é uma expressão numérica de Ø até 191.

Os seguintes exemplos mostram como um comando PPOINT pode ser usado em seus programas.

```
5 PMODE 3,1
1Ø PCLS
15 SCREEN 1,1
3Ø X = RND (1Ø)
35 Y = RND (1Ø)
4Ø C = RND (8)
5Ø PSET (X, Y, C)
6Ø IF PPOINT (5,5) = 8 THEN GOTO 1Ø5
7Ø GOTO 3Ø
1Ø5 CLS
11Ø PRINT Ø 1ØØ, "A POSIÇÃO (5,5) E LARANJA"
```

Dê um RUN e fique atento ao canto superior esquerdo.

Na linha 6Ø é executado um teste no ponto (5,5) do seu vídeo: se ele for laranja (código = 8), o computador desviará para a linha 1Ø5, limpará a tela e imprimirá um mensagem.

O programa seguinte mostra um teste sucessivo de localizações:

```
5 PMODE 3,1
10 PCLS
20 SCREEN 1,1
30 PSET (10, 11, 5)
35 PSET (10, 12, 6)
40 PSET (10, 13, 7)
45 PSET (10, 14, 8)
50 PRINT PPOINT (10, 11)
55 PRINT PPOINT (10, 12)
60 PRINT PPOINT (10, 13)
65 PRINT PPOINT (10, 14)
```

Quando o programa for executado teremos os pontos

```
5
6
7
8
```

Usando o comando LINE e a paginação gráfica de memória, simule uma grande tempestade. Use linhas com diversas posições aleatórias em diferentes páginas de memória.

Uma das soluções para este programa está ao final deste Manual sob o código TESTE 2-2.

(Lembre-se, a resposta é uma sugestão, pois o que interessa é a sua criatividade).

SCREEN

O comando SCREEN

SCREEN tipo, cor

tipo — Quando for usado para texto = 0

Quando for usado para gráficos = 1

cor — Conforme "TABELA DE CORES" 0 ou 1

NOTA: Qualquer número que for teclado e exceder o valor absoluto 1 será assumido como 2 pelo computador.

Agora vamos a uma demonstração completa do comando SCREEN que já vinha sendo usado desde o início nas montagens de programas para gráficos.

Digite:

```
5 PMODE 1,1
10 PCLS
20 SCREEN 1,1
25 LINE (10, 0) - (255, 191), PSET
30 LINE (0, 191) - (255, 0), PSET
40 GOTO 40
```

Destes comandos, vamos particularizar a linha 20.

```
20 SCREEN 1,1
```

O comando SCREEN informa duas coisas ao computador:

1 — Informa se o programa é de texto ou de linhas, círculos, etc.

2 — Informa a coloração. Quanto à esta temos o gráfico a seguir:

	COLORAÇÃO DUPLA	COLORAÇÃO QUÁDRUPLA
Se for selecionado Ø	PRETO/VERDE	VERDE/AMARELO/AZUL/VERMELHO
Se for selecionado 1	PRETO/BRANCO	BRANCO/ ^{AZUL} CLARO/VIOLETA/LARANJA

Dependendo do que for usado na segunda informação do comando SCREEN você terá a opção de cores.

Por exemplo:

Se PMODE especificar 4, 2 ou Ø você terá somente a opção de dois tipos de combinações de cores. Se SCREEN for Ø teremos a opção de PRETO e VERDE (PRETO para a BASE e VERDE para a LINHA).

Se SCREEN for 1 teremos a opção de PRETO e BRANCO (PRETO para a BASE e BRANCO para a LINHA).

Se PMODE especificar 3 ou 1 você terá estas opções:

SCREEN com a função de cor = Ø:

verde, amarelo, azul e vermelho.

Onde:

Verde será a base e vermelho a linha.

As outras duas cores são opções para uso com estes parâmetros. (Caso você usar 'COLOR')

SCREEN com a função de cor = 1:

branco, azul claro, violeta e laranja.

Onde:

Branco será a base e laranja a linha.

As outras duas cores são opções para uso com estes parâmetros. (Caso você usar 'COLOR').

Se o comando SCREEN for omitido, o computador assumirá o valor \emptyset . (preto/verde).

CIRCLE

Foi usado em um programa exemplo anterior o comando CIRCLE. Pois vamos agora a sua definição.

CIRCLE (X, Y), R, C, hw, INIC, FIM

X — Especifica as coordenadas X (horizontais) do ponto central. Numeração de \emptyset até 255.

Y — Especifica as coordenadas Y (verticais) do ponto central. Número de \emptyset até 191.

R — Especifica o raio do círculo.

C — Especifica a cor, valores de \emptyset até 8.

hw — Especifica a altura/largura do círculo. É uma expressão numérica de \emptyset até 255. Se omitida assumirá o valor 1.

INIC — Especifica o ponto de início do círculo. Sendo omitido, o valor assumido será \emptyset . Os valores são: \emptyset e 1.

FIM — Especifica o ponto final do círculo; se omitido, o valor assumido será 1.

Os valores são: \emptyset e 1.

Nos exemplos seguintes temos demonstrações de como usar o comando CIRCLE. Tome como base o nosso programa padrão:

```

5 PMODE 1,1
1Ø PCLS
2Ø SCREEN 1,1
25 LINE (Ø, Ø) - (255, 191), PSET
3Ø LINE (Ø, 191) - (255, Ø), PSET
4Ø GOTO 4Ø

```

Delete a linha 25 e altere a linha 3Ø por:

```
3Ø CIRCLE (128, 96), 95
```

Execute o programa. → RUN

A tela da sua TV mostrará um círculo de cor laranja em uma base de coloração BRANCA.

Agora troque a linha do comando PMODE para uma linha indicativa de alta resolução

```
5 PMODE 4,1
```

Execute o programa após a alteração. O resultado será um círculo BRANCO em uma base PRETA.

O comando CIRCLE tem uma série de funções opcionais.

Se alguma destas não for utilizada, o seu espaço entre vírgulas deve continuar aparecendo no comando.

Exemplifiquemos um caso em que não iremos especificar a cor.

```
3Ø CIRCLE (128, 96), 3Ø,,1
```

este seria

o espaço para a definição da cor.

Vamos substituir a linha 3Ø por uma com o comando CIRCLE com várias opções e executar o programa.

Digite:

```
3Ø CIRCLE (128, 96), 3Ø, 1, 1,.25,.75
```

Execute o programa — RUN

NOTA: Quando forem usadas as funções inicio/fim, você deverá especificar também a função hw. Para um arco normal usar hw = 1.

Para testar os seus conhecimentos adquiridos até agora, tente montar um programa que crie um túnel formado por círculos (espiral). O resultado/sugestão deste problema aparecerá no fim deste Manual com o código 3-1.

Não podemos esquecer um detalhe. O nosso micro CD-6809 é um computador com opções de cores em seus programas. Se soubermos usar esta facilidade, teremos verdadeiras maravilhas em resposta no vídeo, pois há uma infinidade de programas possíveis a serem feitos com os comandos de formatação gráfica.

O próximo comando que vamos demonstrar é o PAINT.

PAINT

PAINT (X, Y), c, d

X ——— Especifica as coordenadas horizontais (X) de \emptyset até 255.

Y ——— Especifica as coordenadas verticais (Y) de \emptyset até 191.

c ——— Especifica o código de cores com as expressões de \emptyset até 8.

d ——— Especifica a coloração do ponto de interrupção.
Cores com os códigos de \emptyset até 8.

Digite ou simplesmente altere o conteúdo da memória.

```
5 PMODE 3,1
1 $\emptyset$  PCLS
2 $\emptyset$  SCREEN 1,1
3 $\emptyset$  LINE ( $\emptyset$ ,  $\emptyset$ ) - (255, 191), PSET
4 $\emptyset$  LINE ( $\emptyset$ , 191) - (255,  $\emptyset$ ), PSET
5 $\emptyset$  CIRCLE (128, 96), 9 $\emptyset$ 
6 $\emptyset$  PAINT (135, 125), 6, 8
7 $\emptyset$  GOTO 7 $\emptyset$ 
```

Antes de executar o programa, vamos dissertar um pouco sobre ele.

Se você analisar as linhas 3Ø e 4Ø, verá que elas formam um par de linhas que se cruzam. A linha 5Ø gerará um círculo cujo ponto de centro é a intersecção das linhas.

O comando PAINT, na linha 6Ø, com as indicações de parâmetros para X e Y (135, 125) e funções de cores (6, 8), indicará ao computador que a partir do ponto (135, 125) o mesmo deverá pintar na cor AZUL CLARO (6) todo o espaço que houver até onde for encontrada a coloração LARANJA (8).

Execute o programa e verifique.

Como teste, tente pintar o interior de um círculo usando como modelo o programa da memória. A resposta/sugestão está no fim deste Manual sob o código de TESTE 3-2.

DRAW

Até agora demonstramos como podemos criar linhas, círculos e quadrados (caixas); vimos também como pintar estes elementos.

Agora vamos ao comando DRAW.

Este comando possibilita a montagem de uma linha (ou série de linhas) com as especificações de direção, ângulo e cor. Tudo isto em uma mesma linha do programa.

DRAW line

line ——— É uma expressão "STRING" que poderá incluir comandos de movimentos:

M = movimentação

U = para cima

D = para baixo

L = para a esquerda

D = para a direita

E = em um ângulo de 45°

F = em um ângulo de 135°
G = em um ângulo de 225°
H = em um ângulo de 315°
X = executa um SUBSTRING e retorna

MODOS:

C = cor
A = ângulo
S = escala

OPÇÕES:

N = para que não desapareça a posição anterior
B = para que desapareça a posição anterior

Digite:

```
5 PMODE 3,1
1Ø PCLS
2Ø SCREEN 1,1
3Ø DRAW "BM 128, 96; U25; R25; D25; L25"
4Ø GOTO 4Ø
```

As funções do comando DRAW são tomadas pelo computador como um STRING.

As separações são feitas por "ponto e vírgula".

A função de movimento "M" é a primeira a aparecer no comando DRAW. Ela se compõe de especificações de orientação horizontal (X) e vertical (Y).

M x, y

x ——— coordenadas horizontais. Expressão numérica com valores de 0 até 255.

y ——— coordenadas verticais. Expressão numérica com valores de 0 até 191.

NOTA: Nas opções do comando DRAW temos o "B". Esta função deverá sempre preceder o "M" para que a linha (posição) anterior seja deletada quando houver movimento.

Vamos executar o programa: RUN

Agora troque a linha 3Ø. Digite:

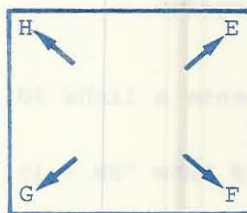
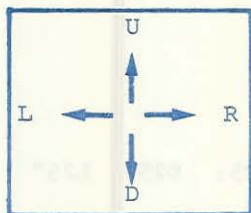
```
3Ø DRAW "C6; BM128,96; U25; R25; D25; L25"
```

Execute. — RUN

A função "C" indica a cor para este comando. No caso o azul claro (nº 6).

As coordenadas (x, y) 128, 96 indicam O CENTRO DA TELA se a tomarmos como um todo. Esta orientação é assumida como ponto de partida normal pelo computador.

Nas figuras a seguir, teremos uma demonstração da movimentação.



Dentro dos COMANDOS DE MOVIMENTO DO DRAW, temos a função M. Esta função poderá ter dois valores: VALOR ABSOLUTO e VALOR RELATIVO.

M sinal x, sinal y

x — é um número que especifica a distância do movimento a partir da posição x, tomando por base o ponto central da tela.

Se o sinal de x for (+) haverá o avanço de posicionamento. (Incremento).

Se o sinal de x for (-) haverá o retrocesso do posicionamento. (Decremento).

y — é um número que especifica a distância do movimento a partir da posição y, tomando por base o ponto central da tela.

Se o sinal for (+) haverá um avanço de posicionamento. (Incremento).

Se o sinal for (-) haverá um retrocesso do posicionamento. (Decremento).

Substitua a linha 3Ø. Digite:

```
3Ø DRAW "BM + 15, 15; U25; R25; D25; L25"
```

Quando for executada a linha 3Ø, tomando por base o ponto central (128, 96), teremos:

(X = 128 + 15 = 143, Y = 96 + 15 = 111)

Execute o programa.

Altere novamente a linha 3Ø. Digite:

```
3Ø DRAW "BM + 15, -15; U25; R25; D25; L25"
```

Execute.

Com a execução da linha 3Ø teremos:

(X = 128 + 15 = 143, Y = 96 - 15 = 81)

Com isto vimos a função "M".

Dentro da função de modos, temos o "S".

Sx

x — é uma expressão numérica de 1 até 62 e indicativa do fator de escala em unidades de 1/4.

l = 1/4 — Escala

2	=	2/4	—	Escala
3	=	3/4	—	Escala
4	=	4/4	—	Escala Completa
5	=	5/4	—	(125%) da Escala
8	=	8/4	—	Dobro da Escala
12	=	12/4	—	Triplo da Escala
etc....				

NOTA: Se não for especificado o "S", o computador assumirá uma escala de (4/4 = 1).

A função de modos "A" significa:

Ax

x — é uma expressão numérica de \emptyset até 3 e indica um dos seguintes ângulos:

\emptyset	—	\emptyset graus
1	—	9 \emptyset graus
2	—	18 \emptyset graus
3	—	27 \emptyset graus

Os ângulos obedecem o sentido dos ponteiros do relógio.

NOTA: Se o ângulo for omitido, o computador assumirá "A" igual a \emptyset .

Agora, um programa com o auxílio do comando DRAW. Ele criará uma série de linhas em quadros organizados no vídeo.

Digite:

```
5 PMODE 4,1
1 $\emptyset$  PCLS
2 $\emptyset$  SCREEN 1,1
```

```
25 FOR ESCALA = 1 TO 62
30 S$ = "S" + STR$ (ESCALA) + ";"
35 DRAW S$ + "BM 1Ø, 1ØØ U2Ø R2Ø D2Ø L2Ø
40 NEXT ESCALA
50 GOTO 5Ø
```

Execute.

LEMBRE-SE:

Os programas aqui demonstrados podem ser incrementados por você. A intenção do Manual é apenas exemplificar os comandos necessários à montagem dos programas.

Tente descobrir as inúmeras possibilidades do seu microcomputador !
Elas são bem maiores do que o apresentado.

COMANDOS GET/PUT

Estes comandos funcionam deste modo:

— O computador, com o GET, armazenará uma área retangular, onde está contido um gráfico, em ARRAY. Posteriormente, PUT recolocará esta área na tela, em outra posição.

GET Ponto de início - Ponto de fim, Destino, G

Ponto de início — (x1, y1) coordenadas do ponto superior esquerdo do retângulo em amostra no vídeo.

Ponto de fim — (x2, y2) coordenadas do ponto inferior direito do mesmo retângulo.

Destino — nome do ARRAY onde será destinada esta área. Este nome deve ser previamente definido.

G — função opcional. Indica ao computador para estocar o retângulo contendo o detalhe gráfico.

PUT - Ponto de início . Ponto de fim, Fonte, Ação

Ponto de início - Ponto de fim - Explanado anteriormente

Fonte - Nome do ARRAY

Ação - Determina aonde os dados do ARRAY devem ser impressos no retângulo.

PSET - Cria os pontos na montagem do ARRAY.

PRESET - Apaga os pontos na montagem do ARRAY.

A função "AÇÃO" é opcional em certas condições.

NOTA: O conjunto GET/PUT deve ser usado para um mesmo PMODE.

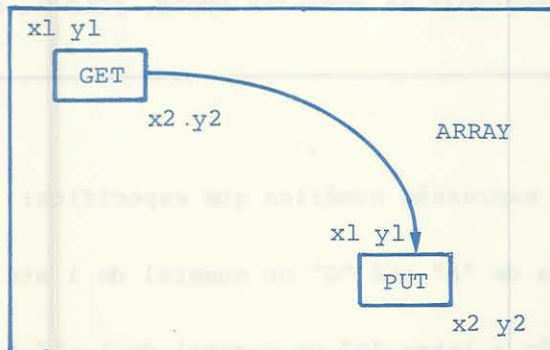
Digite este programa:

```
5 PCLEAR 4
10 PMODE 3,1
15 PCLS
20 SCREEN 1,1
25 DIM V (10, 20)
30 CIRCLE (20, 20), 10
35 GET (10, 10) - (30, 30), V
40 PCLS
42 FOR DL = 1 TO 300: NEXT DL
45 PUT (110, 110) - (130, 130), V
50 FOR DL = 1 TO 300: NEXT DL
60 GOTO 60
```

Execute.

Analise os passos do programa:

A criação do ARRAY (linha 25), círculo (linha 30), o comando GET (linha 35) com a área onde está contido o círculo, a limpeza da tela e o PUT (linha 45) devolvendo a tela à área armazenada.



Para a utilização dos comandos GET/PUT precisamos criar um ARRAY bidimensional.

Para um retângulo com área de $4\emptyset \times 2\emptyset$ precisamos de um ARRAY de $4\emptyset \times 2\emptyset$. Como o microcomputador inicia os valores para ARRAY em ZERO, a figura do ARRAY adequada será: DIM V (39, 19).

Mais um teste com GET/PUT:

Digite:

```
5 PCLEAR 4
10 DIM V (30, 30)
15 PMODE 3,1
20 PCLS
25 SCREEN 1,1
30 CIRCLE (128, 96), 30
35 PAINT (128, 95), 2, 4
40 PAINT (128, 97), 3, 4
45 GET (98, 81) - (128, 111), V, G
50 PCLS
55 FOR I = 150 TO 1 STEP - 1
60 PUT (I, 81 - I/5) - (I + 60, 111 - I/5), V, PSET
65 NEXT I
70 GOTO 70
```

Execute.

Agora limpe a memória e crie você mesmo um programa que mostre uma espaçona subindo na tela até o limite superior.

A resposta/sugestão está ao final deste manual sob o código TESTE 4-1.

O COMANDO PLAY

O comando 'PLAY' não é um comando específico para gráficos; portanto, você não precisará em seus programas os comandos PMODE, PCLS ou SCREEN para usá-lo.

PLAY música

música é uma expressão numérica que especifica:

NOTA: Letra de "A" até "G" ou numeral de 1 até 12.

OITAVA: Após a letra "O" um numeral de 1 até 5 indican-

do a oitava.

Se omitido é assumida a 2a. oitava.

TAMANHO DA NOTA: Após a letra "L" um numeral de 1 até 255. Se omitido, o tamanho normal da nota é assumido.

TEMPO: Após a letra "T" um numeral de 1 até 255. Se omitido, T2 é usado.

VOLUME: Após a letra "V" um numeral de 1 até 31. Se omitido, V15 é usado.

TAMANHO DA PAUSA: Após a letra "P", usar um numeral de 1 até 255.

EXECUÇÃO DE SUBSTRING: Após o prefixo "X" um "STRING" terminando por ";".

Exemplo: XA\$;

Vamos comparar as notas.

Dentro de uma música temos uma infinidade de notas musicais. Estas notas, nos nossos futuros programas, tomarão valores em letras de A até G e em numerais de 1 até 12.

As letras de "A" até "G" identificam as sete notas musicais.

As notas podem sofrer alteração de tonalidade. Para uma indicação tonal mais aguda usaremos a nota (numeral ou letra) seguida de "+" ou "#".

Para uma indicação tonal mais plana (aguda), usaremos a nota (numeral ou letra) seguida de "-".

Quando for usada a nota sem o sinal, teremos tonalidade normal.

Faça um teste. Digite:

PLAY "A"

Para que haja mudança na tonalidade, digite:

PLAY "A -; A; A#; A; A-"

O comando PLAY não reconhece as indicações "B#" ou "C-". Use os números 1 e 12 respectivamente, ou substitua "C" por "B#" e "B" por "C-".

A seguir temos a tabela de tonalidades. Ela indica a relação entre números e notas.

TABELA DE TONALIDADES	
NÚMEROS	NOTAS
1	C
2	C # / D-
3	D / E-
4	E - / D #
5	E / F-
6	F / E #
7	F # / G-
8	G
9	G # / A-
10	A
11	A # / B-
12	B

Quando houver a indicação de duas notas para um só número, isto indica que as duas representações alfabéticas, mais o sinal, tem o mesmo som musical.

Exemplo:

C# e D-

Digite e execute o próximo programa/demonstração das 12 tonalidades da escala.

```
5 CLS
10 FOR N = 1 TO 12 'N = NOTA
15 PRINT "NOTA #"; N
20 PLAY STR$ (N)
30 NEXT N
```

Para uma melhor comparação, adicione a seguinte linha:

```
25 FOR I = 1 TO 500: NEXT I
```

Podemos também controlar o tamanho das notas. Para tanto devemos anexar à frente da nota ou grupo de notas, a indicação deste tamanho.

TABELA DE TAMANHO DAS NOTAS	
L NÚMERO	TAMANHO DAS NOTAS
L 1	NOTA COMPLETA
L 2	METADE DA NOTA
L 3	3/4 DA NOTA
L 4	1/4 DA NOTA
L 8	1/8 DA NOTA
L 16	1/16 DA NOTA
L 32	1/32 DA NOTA
L 64	1/64 DA NOTA
.	
.	
.	
L 255	1/255 DA NOTA

Digite este teste:

PLAY "L1; A; A#; A-"

Após digite mais este:

PLAY "L255; A; A +; A-"

Deu para notar a diferença ?

Vamos agora a uma rápida explicação das outras funções do comando PLAY.

OITAVA — Proporciona uma variação musical da nota ou grupo de notas.

Exemplo:

PLAY "CDE F G; 03; ABA02; FEDCBA"

VOLUME — Especifica o volume da nota.

Exemplo:

PLAY "V5; A; V10; A; V15; A; V20; A"

PAUSA ——— Proporciona momentos de silêncio entre a execução das notas.

Exemplo:

```
PLAY "V5; A; P2; V1Ø; A; P2; V15; A;P2"
```

TEMPO ——— Tempo de execução da nota.

Exemplo:

```
PLAY "T1; V5; A; P2; V1Ø; A; P2; V15; A; P2"
```

EXECUÇÃO DO SUBSTRING ——— Proporciona o desvio para a execução de informações musicais contidas no STRING.

Exemplo:

```
5 CLS
1Ø A$ = "A; A #; A-"
2Ø B$ = "O5; XA$;"
3Ø C$ = "O1; XA$; XB$;"
4Ø PLAY C$
```

Execute o Programa.

Quando a linha 4Ø for processada, o computador irá executar o STRING C\$. Dentro da execução deste STRING temos a execução dos "SUBSTRINGS" (X) A\$ e B\$.

Dentro da função TEMPO podemos efetuar o incremento de um sinal; isto nos proporcionará uma alteração do valor inicial da função.

Vejamos o exemplo:

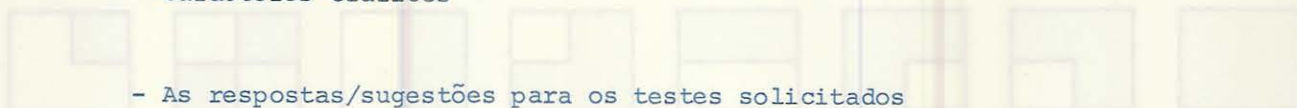
```
5 CLS
1Ø PLAY "T2"
2Ø PLAY "T +; A; A#; A-"
3Ø GOTO 2Ø
```

SUFIJO	PROPÓSITO
+	Adiciona 1 ao valor de "T"
-	Subtrai 1 do valor de "T"
>	Multiplica o valor de "T" por 2
<	Divide o valor de "T" por 2

A N E X O S

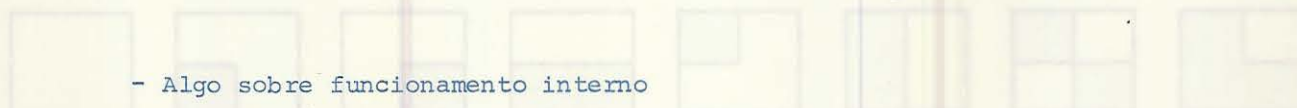
Nas próximas páginas você encontrará:

- Caracteres Gráficos



- As respostas/sugestões para os testes solicitados

- Programas utilizando os comandos do seu Microcomputador CD-6809



- Algo sobre funcionamento interno

- Mensagens de erro

- Sumário dos comandos

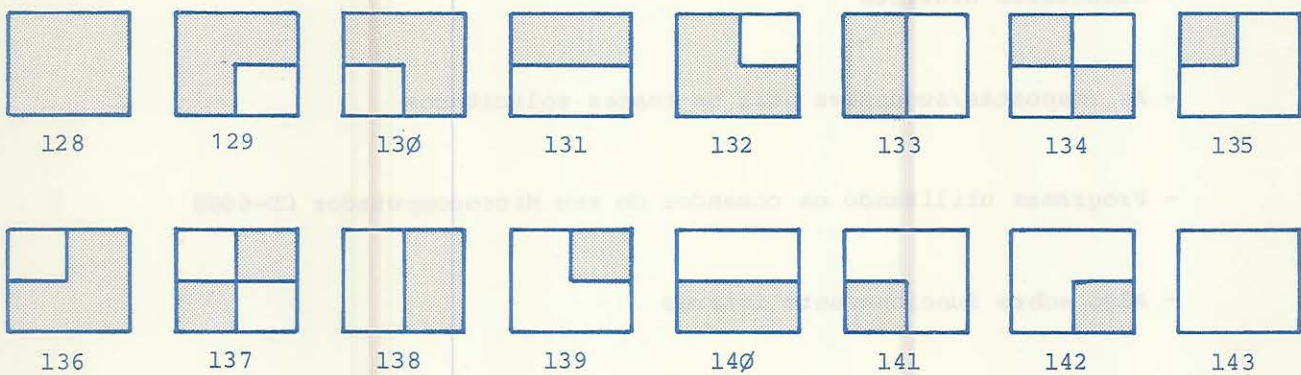
- Tabelas, Gráficos

- Índice

CARACTERES GRÁFICOS

Quando houver a necessidade do uso de caracteres gráficos, em um programa, você deve usar a função CHR\$ junto ao comando PRINT.

A seguir, exemplos dos caracteres gráficos e cores possíveis:



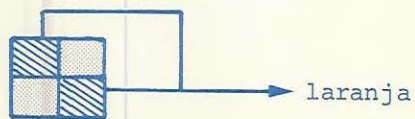
Código de cores:

	+ 16	amarelo
	+ 32	azul
	+ 48	vermelho
	+ 64	branco
	+ 80	azul-claro
	+ 96	violeta
	+ 112	laranja

Quando uma das cores (número) for somada ao número do caracter gráfico, teremos a substituição da área em verde pela nova cor definida.

Ex.:

```
PRINT CHR$ (137 + 112)
```



RESPOSTAS/SUGESTÕES PARA OS TESTES.

TESTE 1-1.

```
5 PMODE 1,1
10 PCLS
20 SCREEN 1,1
30 LINE (72,168)-(200,72), PSET, B
40 LINE (72,72)-(136,36), PSET
45 LINE (200,72)-(136,36), PSET
50 LINE (120,168)-(152,100), PSET, B
55 LINE (152,60)-(168,36), PSET, BF
60 LINE (165,128)-(191,100), PSET, B
65 LINE (178,128)-(178,100), PSET
70 LINE (165,114)-(191,114), PSET
75 LINE (85,128)-(111,100), PSET, B
80 LINE (85,114)-(111,114), PSET
85 LINE (98,100)-(98,128), PSET
90 GOTO 90
```

TESTE 2-1.

Faça a seguinte troca:

```
22 PCOPY 4 TO 3
32 PCOPY 3 TO 2
42 PCOPY 2 TO 1
```

e delete as linhas 11, 21 e 31.

TESTE 2-2.

```
10 PCLEAR 8
20 PMODE 4,1
25 PCLS
30 SCREEN 1,1
40 LINE (0,0)-(255,191), PSET
45 FOR Y = 1 TO 20: NEXT Y
50 PMODE 4,2
55 SCREEN 1,0
60 LINE (0,0)-(255,191), PSET
65 FOR 2 = 1 TO 20: NEXT 2
70 PMODE 0,3
75 SCREEN 1,1
80 LINE (0,0)-(255,191), PSET
85 FOR A = 1 TO 20: NEXT A
90 PMODE 1,4
95 SCREEN 1,1
96 PCLS
100 LINE (0,0)-(255,191), PSET
105 FOR R = 1 TO 20: NEXT R
110 GOTO 20
```

TESTE 3-1.

```
5 PMODE 1,1
10 SCREEN 1,0
```

```

15 PCLS 3
20 COLOR 1,0
25 CIRCLE (200,40),30,,1,.13,.63
30 CIRCLE (230,10),52,,1,.29,.48
35 LINE (100,185)-(180,125), PSET,B
40 LINE -(140,85), PSET
45 LINE -(100,125), PSET
55 LINE (110,160)-(125,130), PSET, B
60 LINE (155,160)-(170,130), PSET, B
70 LINE (130,130)-(149,185), PSET, B
75 PSET (134,157,1)
80 LINE (160,105)-(160,90), PSET
85 LINE -(175,90), PSET
90 LINE -(175,115), PSET
100'
105 X=167:Y=89
110 SP=0: EP=0
115 FOR R= 1 TO 50 STEP .05
120 EP=EP+.02: IF EP > 1 THEN EP = 0
125 CIRCLE (X+R, Y-R),R,4,1,SP,EP
130 NEXT R
200 GOTO 200

```

TESTE 3-2.

Adicione a linha 65:

```
65 PAINT (150,100),8,8
```

ou delete a linha 30.

TESTE 4-1.

```

5 PCLEAR 4
10 PMODE 4,1
15 PCLS
20 SCREEN 1,1
25 DIM V(35,35)
30 X = 10: Y = 10
35 DRAW "B10,10; S2; H10; R15; F10; R20; F10; G10; L20; G10; L15;
E10; U20; D4; NL8; D4; NL12; D4NL16; D4; NL12; D4; NL8"
40 GET (X-X, Y-Y)-(X*3.5, Y*3.5),V,G
45 A$ = INKEY$: IF A$= "" THEN 45 'DIGITE QUALQUER TECLA PARA INI-
CIAR
50 PCLS
55 FOR A = 10 TO 200 STEP 5
60 PUT (X+A, Y)-(X+A+35, Y+35),V, PSET
65 NEXT A
70 PCLS
75 GOTO 55

```

PROGRAMAS

PROGRAMA 1

```

5  '*FINANCAS*
10 DIM A(5,3,2), A$(5)
20 DATA UTILIDADES, PESSOAL, PREVISOES, RENDA, VIAGEM
30 FOR X = 1 TO 5
40 READ A$(X)
50 CLS
60 PRINT @ 139, "CUSTOS"
70 PRINT @ 175 - INT(LEN(A$(X))/2), A$(X)
80 PRINT
90 FOR Y = 1 TO 3
100 PRINT "DEPT" Y
110 INPUT "ORÇAMENTO"; A(X,Y,1)
120 INPUT "ATUAL"; A(X,Y,2)
130 NEXT Y
140 NEXT X
150 CLS
160 PRINT @ 133, "O QUE VOCE GOSTARIA DE VER"
170 L = 203
180 FOR X = 1 TO 5
190 PRINT @ L, X; A$(X)
200 L = L + 32
210 NEXT X
220 PRINT @ 460, "(1-5)"
230 INPUT X
235 C(1)=0:C(2)=0:LC(1)=0:LC(2)=0
240 FOR Y = 1 TO 3
250 C(1) = A(X,Y,1)+C(1)
260 C(2) = A(X,Y,2) + C(2)
270 NEXT Y
280 IF C(2) > C(1) THEN 310
290 LC(1)=30: LC(2)=INT(C(2)/C(1)*30)
300 GOTO 320
310 LC(2)=30: LC(1)=INT(C(1)/C(2)*30)
320 P = 129
330 CLS(0)
340 PRINT @ 11, "CUSTOS";
350 PRINT @ 47 - INT(LEN(A$(X))/2), A$(X);
360 PRINT @ 97, "ORÇAMENTO";
370 PRINT @ 257, "ATUAL";
380 PRINT @ 448, CHR$(159)+CHR$(159);
390 PRINT @ 451, "DEPT 1";
400 PRINT @ 459, CHR$(175)+CHR$(175);
410 PRINT @ 462, "DEPT 2";
420 PRINT @ 470, CHR$(191)+CHR$(191);
430 PRINT @ 473, "DEPT 3";
440 PRINT @ 480, "PRESSIONE QUAL TECLA PARA CONTINUAR";
450 FOR M = 1 TO 2
460 FOR N = 1 TO 2
470 P1 = P + 32
480 FOR Y = 1 TO 3
490 D(Y) = INT(A(X,Y,M)/C(1)*LC(1))
500 FOR O = 1 TO D(Y)
510 PRINT @ P1, CHR$(143+16*Y);
520 P1 = P1 + 1
530 NEXT O
540 NEXT Y
550 P = P + 32
560 NEXT N
570 P = 289
580 NEXT M
590 K$ = INKEY$: IF K$="" THEN 590
600 GOTO 150

```

PROGRAMA 2

```

1 ' **ENTRADA/SAIDA**
2 '
5 PMODE 3,1
10 PCLS 3
15 SCREEN 1,0
20 FOR I = 3 TO 7
25 FOR J = 2 TO 6
30 FOR S = 0 TO 3
35 FOR R = 0 TO 3
40 COLOR R,S
45 A = 0:B=255:C=0:D=191
50 LINE (A,C)-(B,D),PSET,B
55 A=A+J:B=B-J:C=C+I:D=D-I
60 IF A<255 AND C<191 THEN 50
65 NEXT R
70 NEXT S
75 NEXT J,I
80 GOTO 30

```

PROGRAMA 3

```

1 ' **VOLTA A BACH**
2 '
5 CLS
10 PRINT @ 96, STRING$(32,"*")
20 PRINT @ 320, STRING$(32,"*")
25 PRINT @ 201, "VOLTA A BACH"
40 FOR X = 1 TO 1000: NEXT X
55 A$ = "T6;O2;L2;G;L4;C;D;E;F;L2;G;C;P16;C"
60 B$ = "L2;A;L4;F;G;A;B;O3;L2;C;O2;C;P16;C;F;L4;G;F;E;D"
65 C$ = "L2;E;L4;F;E;D;C;L2;O1;B;O2;L4;C;D;E;C"
70 D$ = "L2;E;L1;D;L2;G;L4;C;D;E;F;L2;G;C;P16;C"
75 E$ = "L2;A;L4;F;G;A;B;O3;L2;C;O2;C;P16;C;F;L4;G;F;E;D"
80 F$ = "L2;E;L4;F;E;D;C;D;E;L2;F;O1;B;L1;O2;C"
85 X$ = "XA$;XB$;XC$;XD$;XE$;XF$;"
90 PLAY X$

```

PROGRAMA 4

```

1 ' **DANCA MEXICANA**
2 '
5 CLS
10 PRINT @ 96, STRING$(32,"*")
20 PRINT @ 320, STRING$(32,"*")
30 PRINT @ 199, "DANÇA MEXICANA"
40 FOR X = 1 TO 500: NEXT X
125 REM INICIO
130 O$ = "V15;T3;O2;"
135 P$ = "L8CFP8CFP8CFP4P8"
140 Q$ = "CFGFEP8FGP4P8"
145 X$ = "XO$;XP$;XQ$;"
150 PLAY X$
155 R$ = "CEP8CEP8CEP4P8"
160 S$ = "CEFEDP8EFP4P8"
165 Y$ = "XO$;XR$;XS$;"
170 PLAY Y$
180 REM SEGUNDO TEMPO
185 O$ = "V25;T3;O1"
190 PLAY X$
195 O$ = "T3;O4"
197 S$ = "CEFEDP8EFO4CO3AF"
200 PLAY Y$
210 A$ = "O3CO2BO3CO2AA-AFEFCP4"
220 B$ = "COLBO2CDEFGAB-O3CEG"
225 O$ = "V15;T4;"
230 Z$ = "XO$;XA$;XB$;"

```

```

235 PLAY Z$
240 C$="O3B-AB-GF+GED+ECEG"
245 D$="O4L16CP16CP16CP16L8DCO3B-AGFP4"
250 E$="XO$;XC$;XD$;"
255 PLAY E$
260 F$="O2L16GP16GP16GP16DP16DP16DP16EP16FP16L8EL16GP16O1GP16L8G"
265 G$="V15O2L16GP16GP16GP16DP16DP16DP16EP16FP16L8ECO1GC"
270 H$="XF$;XG$;"
280 PLAY H$
285 I$="XF$;"
290 PLAY I$
295 J$="O2L16GP16GP16GP16AP16GP16GP16AP16BP16O3L4CP8"
300 PLAY "XJ$;"
310 K$="O4L1DL4DEDEL8DEDEL16DEDEDEDEL32DEDEDEDEDEDEDEL64DEDEDEDED
EDEDEDEDEDEDEL32DD-CO3BB-AA-GF+FEE-DD-L4DD-"
320 PLAY "XK$;"
330 M$="T5L8DO2BB-BGF+GL4DP8"
340 N$="L8DC+DEF+GABO3CO2L4AP8"
350 AA$="O3L8CO2BO3CO2AG+AF+FF+L4DP8"
370 BB$="O3L8DDEDCO2BAO3DEDCO2BA"
380 CC$="O2DEDCO1BAO4DEDEDEDEDEF+GDO3BGT4DO2BGT3DO1T2BL4P2V3ØL1G"
400 PLAY "XM$;XN$;XAA$;XBB$;XCC$;"
500 PMODE 4,1
505 FOR Y = 1 TO 5
510 SCREEN 1,Ø
520 PCLS
550 CIRCLE (128,96),5Ø,1,.2,.85,.67
560 CIRCLE (128,96),25,1,2,.5,1
570 LINE (1Ø5,96)-(151,96),PSET
600 PMODE 4,1
610 SCREEN 1,Ø
620 PCLS
630 CIRCLE (128,75),5Ø,1,.2,.85,.67
660 CIRCLE (128,75),1,2,.5,1
670 LINE (1Ø5,75)-(151,75),PSET
675 NEXT Y
680 IF Y > 5 THEN 69Ø
685 GOTO 5ØØ
690 CLS
710 FOR X = 1 TO 6ØØ:NEXT X
720 GOTO 5

```

PROGRAMA 5

```

1 '**PROJECoes**
2 '
5 PMODE 4,1
1Ø PCLS
15 SCREEN 1,Ø
2Ø DRAW "BM5Ø,5ØR6ØD1ØNL2ØD2ØL2ØNU2ØL2ØNU2ØL2ØU2ØNR2ØU1Ø" 'VISAO DE
CIMA
25 DRAW "BM5Ø,1ØØR2ØND2ØR2ØND2ØR2ØD2ØNL2ØD1ØL6ØU1ØNR2ØU2Ø" 'VISAO
FRONTAL
3Ø DRAW "BM15Ø,1ØØR3ØD3ØL3ØU1ØNE2ØU2Ø" 'VISTA LATERAL
35 'VISAO DAS LINHAS OBLIQUAS 4Ø-6Ø
4Ø DRAW "BM15Ø,5ØU5E15R1ØBF2ØBD3ØNR5L2ØH25U1Ø
45 DRAW "BM15Ø,5ØU5F8U15R15H8F8L15F8NR15D15F8ND1ØE15NR1ØH8
5Ø LINE (175,3Ø)-(2ØØ,55),PSET
55 LINE -(2ØØ,8Ø),PSET
6Ø LINE (167,6Ø)-(183,46),PSET
65 GOTO 65

```

PROGRAMA 6

```

1 '**CUBO QUE DESMONTA**
2 '
5 PCLEAR 8
1Ø PMODE 3,1
15 PCLS

```



```

20 COLOR 6,5
25 DRAW "BM100,100U30NR30E15R30NG15D30G15NU30L30"
30 PAINT (105,95),8,6
35 PAINT (135,80),8,6
40 PAINT (110,65),8,6
45 SCREEN 1,1
50 FOR X = 1 TO 600: NEXT X
110 PMODE 3,5
112 PCLS
115 COLOR 6,5
120 DRAW "BM100,100U30NR30E20R30G20D30NL30F20L30H20"
125 LINE (100,100)-(70,95),PSET
130 LINE -(70,65),PSET
135 LINE -(100,70),PSET
140 LINE (70,95)-(40,65),PSET,B
145 LINE (130,100)-(160,95),PSET
150 LINE -(160,65),PSET
155 LINE -(130,70),PSET
160 PAINT (95,95),8,6
165 PAINT (105,95),8,6
170 PAINT (135,85),8,6
175 PAINT (45,85),8,6
180 PAINT (115,65),8,6
185 PAINT (125,114),8,6
190 SCREEN 1,1
195 FOR X = 1 TO 600: NEXT X
200 GOTO 10

```

PROGRAMA 7

```

1 '**ONDAS**
2 '
5 PMODE 4,1
10 PCLS
15 SCREEN 1,1
20 LINE (0,86)-(255,86),PSET
25 PI=3.14159
30 A1= -4*PI
35 A2= 4*PI
40 N=180
45 R=50
50 X=(A2-A1)/N
55 F=255/(A2-A1)
60 FOR I =A1 TO A2 STEP X
65 X=I*F
70 Y=R*SIN(I)
75 PSET ((X+140),(80+Y),1)
80 NEXT I
90 GOTO 90

```

PROGRAMA 8

```

1 '**SEN/COS**
2 '
10 PMODE 4,1
20 PCLS
30 SCREEN 1,0
40 LINE (127,5)-(127,185),PSET
50 LINE (7,95)-(247,95),PSET
60 FOR XSCALE=7 TO 247 STEP 20
70 PRESET(XSCALE,95)
80 NEXT XSCALE
90 FOR YSCALE=5 TO 185 STEP 10
100 PRESET(127,YSCALE)
110 NEXT YSCALE
130 FOR X=-180 TO 180 STEP 1.5
140 AX=X/57.29578
145 XP=X/1.5+127
150 F1=-(SIN(AX)*90)+95

```

```

160 F2=- (COS(AX)*90)+95
170 PSET(XP,F1,1): PSET(XP,F2,1)
180 NEXT X
190 GOTO 190

```

PROGRAMA 9

```

1 '**GRAFICOS ALEATORIOS**
2 '
10 PMODE 3,1
15 PCLS
20 SCREEN 1,1
25 F =RND(4):B=RND(8): IF B=F OR (B-4=F) THEN 25
30 COLOR F,B:PCLS B: FOR L = 0 TO 5
35 LINE -(RND(255),RND(191)),PSET
40 CIRCLE (RND(255),RND(191)),RND(100)
50 NEXT: FOR P=0 TO 10
55 PAINT (RND(255),RND(191)),RND(4),F
60 NEXT: FOR H= 1 TO 7
65 FOR T=0 TO 600: NEXT T: GOTO 10

```

PROGRAMA 10

```

1 '**COBERTOR INDIGENA**
2 '
5 PMODE 3,1
10 PCLS 4
15 SCREEN 1,0
20 COLOR 1,0
25 FOR X = 0 TO 255 STEP 10
30 OY = Y
35 Y = 30-OY
40 LINE (X,100-Y)-(X+10,100-OY),PSET
45 LINE (X,120+Y)-(X+10,120+OY),PSET
50 NEXT
60 FOR C = 2 TO 8
65 PAINT (0,110),C,1
70 NEXT
80 GOTO 5

```

PROGRAMA 11

```

1 '**LINHAS CRUZADAS**
2 '
5 PMODE 3,1
10 PCLS
20 SCREEN 1,1
30 DRAW"BM50,180U60BU20U60R60BR20R60D60BD20D60L60BL20L60
40 DRAW"BM50,180U60R40BR20R80D20BL20L60BL20L20D20R20BR60R20U20
50 DRAW"BM50,180R60U80BU20U40L40BD20D20BD60D20R20U60BU20U20L20
60 DRAW"BM50,180U60BU40BR20R60BR20R20U20L20D60BD20D20R20
70 DRAW "BM50,180BR80U40BU20U80
80 DRAW "BM50,180BU80R80BR20R40
90 PAINT (85,128),6,8
95 PAINT (95,78),6,8
97 PAINT (155,95),6,8
98 PAINT (135,145),6,8
99 PAINT (128,185),7,8
100 PAINT (75,150),7,8
101 PAINT (160,150),7,8
102 PAINT (75,75),7,8
103 PAINT (160,75),7,8
104 PAINT (120,110),7,8
110 FOR X = 1 TO 600: NEXT X
200 GOTO 5

```

PROGRAMA 12

```

1  ***LINHAS**
2  '
5  CLS
20 C = C + 1
25 IF C > 8 THEN C = 5
30 COLOR C,1
50 PRINT "TYPE X0,Y0";
60 INPUT X0,Y0
70 PRINT "TYPE X1,Y1";
80 INPUT X1,Y1
90 PMODE 3,1
95 PCLS
100 SCREEN 1,1
110 LINE (X0,Y0)-(X1,Y1),PSET
115 FOR X = 1 TO 2000: NEXT X
120 GOTO 20

```

PROGRAMA 13

```

1  ***LINHAS ALEATÓRIAS**
2  '
20 PMODE 4,1
25 PCLS
30 SCREEN 1,1
35 X = RND(255): Y = RND(191)
40 LINE -(X,Y), PSET
45 FOR X = 1 TO 200: NEXT X
50 GOTO 35

```

PROGRAMA 14

```

1  *** TREVO DE OITO FOLHAS ***
2  '
5  PCLEAR 8
10 PMODE 4,1
15 PCLS
20 SCREEN 1,0
25 PI=3,14159
30 A1=0: A2 = 2*PI
35 N=360:A=50
40 X = (A2-A1)/N
45 FOR I = A1 TO A2 STEP X
50 R = A * COS (4*I)
55 X = R * SIN(I)
60 Y = R * COS(I)
65 PSET(128 + X,96+Y,5)
70 NEXT I
75 GOTO 25

```

PROGRAMA 15

```

1  ***BOMBARDEIO**
2  '
10 PMODE 4,1
15 PCLS
20 SCREEN 1,1
25 CIRCLE (128,96),80
30 CIRCLE (128,96),90
35 PAINT (0,0),5
40 FOR T=30 TO -30 STEP -1
45 A=(2*3.1415)*T/60
50 LINE (128,96)-(75*SIN(A)+128,75*COS(A)+96),PSET
55 SOUND Q*2+1,20/(Q+1)+1
60 LINE (128,96)-(75*SIN(A)+128,75*COS(A)+96),PRESET

```

```

65 Q=60-2*T:FOR Y=Q TO 0 STEP -1:NEXT
70 NEXT
75 CLS
80 PCLS
85 PRINT @ 237,"BOOM!"
90 SOUND 1,30
95 PMODE 4,1
100 SCREEN 1,1
105 FOR I =2 TO 200 STEP 2
110 CIRCLE (128,96),I
115 NEXT I
120 SCREEN 1,1
125 FOR X =2 TO 200 STEP 2
130 CIRCLE (128,96),X,.3
135 NEXT X
140 FOR I = 2 TO 200 STEP 2
145 CIRCLE (128,96),I,3,.5
150 NEXT I
155 GOTO 155

```

PROGRAMA 16

```

1 '**
2 '
5 PCLEAR 8
50 GOTO 600
60 LINE ((255-X),(191-Y))-(X,Y),PSET
61 J = J+1:IF J>A THEN J=0:A=RND(50)
63 RETURN
601 FOR I = I TO 5 STEP 4
602 PMODE 3,I
603 PCLS
604 SCREEN 1,0
605 A=25:X=0: Y=0: J=0
610 FOR X =0 TO 254
612 COLOR X/32+1,5
615 GOSUB 60: NEXT X
620 FOR Y =0 TO 190
623 COLOR Y/24+1,5
625 GOSUB 60: NEXT Y
630 FOR X = 255 TO 1 STEP -1
633 COLOR X/32+1,5
635 GOSUB 60: NEXT X
640 FOR Y = 191 TO 1 STEP -1
643 COLOR Y/24+1,5
645 GOSUB 60: NEXT Y
650 NEXT I
660 FOR I = 1 TO 5 STEP 4
670 PMODE 3,I
680 SCREEN 1,0
690 FOR T = 1 TO 30: NEXT T
700 NEXT I
710 GOTO 660

```

PROGRAMA 17

```

1 '**TRIANGULOS MOVEIS**
2 '
10 FOR A = 90 TO 0 STEP -4
15 S1 = A*9: S2=191
20 A3 = A/57.29578
30 X1 = 0:Y1=191
40 X2 = S1+X1: Y2=Y1
50 X3 = X1+S2*COS(A3):Y3=Y1-S2*SIN(A3)
55 GOSUB 1000
90 NEXT A
99 GOTO 99
1000 PMODE 4,1
1005 PCLS

```

```

1010 SCREEN 1,0
1020 LINE (X1,Y1)-(X2,Y2),PSET
1030 LINE -(X3,Y3),PSET
1040 LINE -(X1,Y1),PSET
1060 RETURN

```

PROGRAMA 18

```

1  '**CONTADOR**
2  '
10 CLS
20 CLEAR 1000
30 PRINT "AONDE VOCE QUER INICIAR A CONTAGEM?"
35 INPUT A$
40 P=LEN(A$)
50 PRINT:PRINT A$
60 C=VAL(MID$(A$,P,1))+1
70 MS$=A$: MR$=RIGHT$(STR$(C),1): PS=P: GOSUB 200: A$=MS$
80 IF C<10 THEN 40
90 P=P-1
100 IF P=0 THEN IF LEN(A$)=255 THEN PRINT " " "": END: ELSE A$=
    "1"+A$: GOTO 40
110 GOTO 60
200 LS=LEN(MS$)
210 IF LS<> LEN(MR$)+LS-1 OR PS<1 THEN STOP
220 MS$=LEFT$(MS$,PS-1)+MR$+RIGHT$(MS$,LS-PS)
230 RETURN

```

PALAVRAS E SÍMBOLOS DE USO RESERVADO DO COMPUTADOR.

'	CLS	FOR	MID\$	PPOINT	SQR
*	COLOR	GET	MOTOR	PRESET	STEP
+	CONT	GOSUB	NEW	PRINT	STOP
-	COS	GOTO	NEXT	PSET	STR\$
REM	CSAVE	HEX\$	NOT	PUT	STRING\$
<	DATA	IF	OFF	READ	SUB
=	DEF	INKEY\$	ON	RENUM	TAB
>	DEL	INPUT	OPEN	RESET	TAN
ABS	DIM	INSTR	OR	RESTORE	THEN
AND	DLOAD	INT	PAINT	RETURN	TIMER
ASC	DRAW	JOYSTK	PCLEAR	RIGHT\$	TO
ATN	EDIT	LEFT\$	PCLS	RND	TROFF
AUDIO	ELSE	LEN	PCOPY	RUN	TRON
CHR\$	END	LET	PEEK	SCREEN	USING
CIRCLE	EOF	LINE	PLAY	SET	USR
CLEAR	EXEC	LIST	PMODE	SGN	VAL
CLOAD	EXP	LLIST	POINT	SIN	VARPTR
CLOADM	FIX	LOG	POKE	SKIPF	↑
CLOSE	FN	MEM	POS	SOUND	

Estas palavras e símbolos não podem ser usados como representação de variáveis, etc....

TABELAS EM CÓDIGOS ASCII

Estes são os códigos em ASCII. Você, com estas tabelas, poderá saber os códigos relativos a todos os caracteres do teclado do microcomputador.





Os códigos estão relacionados aos caracteres e são apresentados em dois tipos de conversão: a decimal (coluna da esquerda) e hexadecimal (coluna da direita).

Lembre-se:






Os códigos ASCII são usados com a função CHR\$.

CARACTER	DECIMAL	HEXADECIMAL
TECLA DE ESPAÇO	32	20
!	33	21
"	34	22
#	35	23
\$	36	24
%	37	25
&	38	26
'	39	27
(40	28
)	41	29
*	42	2A
+	43	2B
,	44	2C
-	45	2D
.	46	2E
/	47	2F
Ø	48	30
1	49	31
2	50	32
3	51	33
4	52	34
5	53	35
6	54	36
7	55	37
8	56	38
9	57	39
:	58	3A

CARACTER	DECIMAL	HEXADECIMAL
;	59	3B
<	60	3C
=	61	3D
>	62	3E
?	63	3F
@	64	40
A	65	41
B	66	42
C	67	43
D	68	44
E	69	45
F	70	46
G	71	47
H	72	48
I	73	49
J	74	4A
K	75	4B
L	76	4C
M	77	4D
N	78	4E
O	79	4F
P	80	50
Q	81	51
R	82	52
S	83	53
T	84	54
U	85	55

CARACTER	DECIMAL	HEXADECIMAL
V	86	56
W	87	57
X	88	58
Y	89	59
Z	90	5A
 *	94	5E
 *	100	0A
 *	8	08
 *	9	09
RETURN	03	03
CLEAR	12	0C
ENTER	13	0D

Se o teclado estiver em maiúsculo teremos:

	= 92 (decimal), 5C (hexadecimal)
	= 95 (decimal), 5F (hexadecimal)
	= 91 (decimal), 5B (hexadecimal)
	= 21 (decimal), 15 (hexadecimal)
	= 93 (decimal), 5D (hexadecimal)

TECLADO EM MINÚSCULO

Estes são os códigos em ASCII para o teclado em minúsculo.

CARACTER	DECIMAL	HEXADECIMAL
a	97	61
b	98	62
c	99	63
d	100	64
e	101	65
f	102	66
g	103	67
h	104	68
i	105	69
j	106	6A
k	107	6B
l	108	6C
m	109	6D

CARACTER	DECIMAL	HEXADECIMAL
n	110	6E
o	111	6F
p	112	70
q	113	71
r	114	72
s	115	73
t	116	74
u	117	75
v	118	76
w	119	77
x	120	78
y	121	79
z	122	7A

TABELA BÁSICA DE CONVERSÃO NUMÉRICA.

Esta tabela demonstra o valor numérico do BYTE em diferentes sistemas.

DEC.	BINÁRIO	HEX.	OCT.
0	00000000	00	000
1	00000001	01	001
2	00000010	02	002
3	00000011	03	003
4	00000100	04	004
5	00000101	05	005
6	00000110	06	006
7	00000111	07	007
8	00001000	08	010
9	00001001	09	011
10	00001010	0A	012
11	00001011	0B	013
12	00001100	0C	014
13	00001101	0D	015
14	00001110	0E	016
15	00001111	0F	017
16	00010000	10	020
17	00010001	11	021
18	00010010	12	022
19	00010011	13	023
20	00010100	14	024
21	00010101	15	025
22	00010110	16	026
23	00010111	17	027
24	00011000	18	030
25	00011001	19	031
26	00011010	1A	032
27	00011011	1B	033
28	00011100	1C	034
29	00011101	1D	035
30	00011110	1E	036
31	00011111	1F	037
32	00100000	20	040
33	00100001	21	041
34	00100010	22	042
35	00100011	23	043

DEC.	BINÁRIO	HEX.	OCT.
36	00100100	24	044
37	00100101	25	045
38	00100110	26	046
39	00100111	27	047
40	00101000	28	050
41	00101001	29	051
42	00101010	2A	052
43	00101011	2B	053
44	00101100	2C	054
45	00101101	2D	055
46	00101110	2E	056
47	00101111	2F	057
48	00110000	30	060
49	00110001	31	061
50	00110010	32	062
51	00110011	33	063
52	00110100	34	064
53	00110101	35	065
54	00110110	36	066
55	00110111	37	067
56	00111000	38	070
57	00111001	39	071
58	00111010	3A	072
59	00111011	3B	073
60	00111100	3C	074
61	00111101	3D	075
62	00111110	3E	076
63	00111111	3F	077
64	01000000	40	100
65	01000001	41	101
66	01000010	42	102
67	01000011	43	103
68	01000100	44	104
69	01000101	45	105
70	01000110	46	106
71	01000111	47	107

DEC.	BINÁRIO	HEX.	OCT.
72	01001000	48	110
73	01001001	49	111
74	01001010	4A	112
75	01001011	4B	113
76	01001100	4C	114
77	01001101	4D	115
78	01001110	4E	116
79	01001111	4F	117
80	01010000	50	120
81	01010001	51	121
82	01010010	52	122
83	01010011	53	123
84	01010100	54	124
85	01010101	55	125
86	01010110	56	126
87	01010111	57	127
88	01011000	58	130
89	01011001	59	131
90	01011010	5A	132
91	01011011	5B	133
92	01011100	5C	134
93	01011101	5D	135
94	01011110	5E	136
95	01011111	5F	137
96	01100000	60	140
97	01100001	61	141
98	01100010	62	142
99	01100011	63	143
100	01100100	64	144
101	01100101	65	145
102	01100110	66	146
103	01100111	67	147
104	01101000	68	150
105	01101001	69	151
106	01101010	6A	152
107	01101011	6B	153
108	01101100	6C	154
109	01101101	6D	155
110	01101110	6E	156
111	01101111	6F	157
112	01110000	70	160
113	01110001	71	161
114	01110010	72	162
115	01110011	73	163

DEC.	BINÁRIO	HEX.	OCT.
116	01110100	74	164
117	01110101	75	165
118	01110110	76	166
119	01110111	77	167
120	01111000	78	170
121	01111001	79	171
122	01111010	7A	172
123	01111011	7B	173
124	01111100	7C	174
125	01111101	7D	175
126	01111110	7E	176
127	01111111	7F	177
128	10000000	80	200
129	10000001	81	201
130	10000010	82	202
131	10000011	83	203
132	10000100	84	204
133	10000101	85	205
134	10000110	86	206
135	10000111	87	207
136	10001000	88	210
137	10001001	89	211
138	10001010	8A	212
139	10001011	8B	213
140	10001100	8C	214
141	10001101	8D	215
142	10001110	8E	216
143	10001111	8F	217
144	10010000	90	220
145	10010001	91	221
146	10010010	92	222
147	10010011	93	223
148	10010100	94	224
149	10010101	95	225
150	10010110	96	226
151	10010111	97	227
152	10011000	98	230
153	10011001	99	231
154	10011010	9A	232
155	10011011	9B	233
156	10011100	9C	234
157	10011101	9D	235
158	10011110	9E	236
159	10011111	9F	237

DEC.	BINÁRIO	HEX.	OCT.
160	10100000	A0	240
161	10100001	A1	241
162	10100010	A2	242
163	10100011	A3	243
164	10100100	A4	244
165	10100101	A5	245
166	10100110	A6	246
167	10100111	A7	247
168	10101000	A8	250
169	10101001	A9	251
170	10101010	AA	252
171	10101011	AB	253
172	10101100	AC	254
173	10101101	AD	255
174	10101110	AE	256
175	10101111	AF	257
176	10110000	B0	260
177	10110001	B1	261
178	10110010	B2	262
179	10110011	B3	263
180	10110100	B4	264
181	10110101	B5	265
182	10110110	B6	266
183	10110111	B7	267
184	10111000	B8	270
185	10111001	B9	271
186	10111010	BA	272
187	10111011	BB	273
188	10111100	BC	274
189	10111101	BD	275
190	10111110	BE	276
191	10111111	BF	277
192	11000000	C0	300
193	11000001	C1	301
194	11000010	C2	302
195	11000011	C3	303
196	11000100	C4	304
197	11000101	C5	305
198	11000110	C6	306
199	11000111	C7	307
200	11001000	C8	310
201	11001001	C9	311
202	11001010	CA	312
203	11001011	CB	313

DEC.	BINÁRIO	HEX.	OCT.
204	11001100	CC	314
205	11001101	CD	315
206	11001110	CE	316
207	11001111	CF	317
208	11010000	D0	320
209	11010001	D1	321
210	11010010	D2	322
211	11010011	D3	323
212	11010100	D4	324
213	11010101	D5	325
214	11010110	D6	326
215	11010111	D7	327
216	11011000	D8	330
217	11011001	D9	331
218	11011010	DA	332
219	11011011	DB	333
220	11011100	DC	334
221	11011101	DD	335
222	11011110	DE	336
223	11011111	DF	337
224	11100000	E0	340
225	11100001	E1	341
226	11100010	E2	342
227	11100011	E3	343
228	11100100	E4	344
229	11100101	E5	345
230	11100110	E6	346
231	11100111	E7	347
232	11101000	E8	350
233	11101001	E9	351
234	11101010	EA	352
235	11101011	EB	353
236	11101100	EC	354
237	11101101	ED	355
238	11101110	EE	356
239	11101111	EF	357
240	11110000	F0	360
241	11110001	F1	361
242	11110010	F2	362
243	11110011	F3	363
244	11110100	F4	364
245	11110101	F5	365
246	11110110	F6	366
247	11110111	F7	367

DEC.	BINÁRIO	HEX.	OCT.
248	11111000	F8	370
249	11111001	F9	371
250	11111010	FA	372
251	11111011	FB	373

DEC.	BINÁRIO	HEX.	OCT.
252	11111100	FC	374
253	11111101	FD	375
254	11111110	FE	376
255	11111111	FF	377

DEC.	HEX.	BINÁRIO	DEC.	DEC.	HEX.	BINÁRIO	DEC.
000	00	00000000	000	000	00000000	000	000
001	01	00000001	001	001	00000001	001	001
010	02	00000010	010	010	00000010	010	010
011	03	00000011	011	011	00000011	011	011

O bit é a denominação dada ao Dígito Binário 0 ou 1.

Para que possamos representar números e palavras, é preciso agrupar um certo conjunto de bits.

Em processamento de dados, uma palavra é um conjunto de bits.

Os microcomputadores usam palavras de 8 e 16 bits, dependendo do tipo do processador.

O conjunto de arranjos de bits compõem as instruções ou comandos e, por sua vez, estas instruções, em seqüência lógica, formam os programas.

Um BYTE é composto de 8 bits. Para cada byte são possíveis 256 combinações no estado ligado/desligado (0 ou 1). Estas combinações podem formar códigos de informação Binária de 00000000 até 11111111; em decimal de 0 até 255 e em hexadecimal de 0 até FF.

Para que os programas escritos em linguagem de alto nível, BASIC por exemplo, sejam entendidos pelo computador, devem sofrer uma "tradução". Esta, é feita da linguagem de programação para a de máquina, visto que o computador só aceita informações Binárias.

TRADUTORES

COMPILADORES - Traduzem todo o programa para o processamento.

INTERPRETADORES - Traduzem somente a linha da instrução a ser executada, isto é, linha a linha.

Nos microcomputadores, pela pouca área de memória disponível, o uso dos inter

pretadores proporcionam maior rendimento. Os compiladores utilizam mais espaço de memória (1,5 a 2 vezes) e gastam menos tempo no processamento, já que na execução, o programa estará totalmente traduzido.

TIPOS DE MEMÓRIA

MEMÓRIA RAM

(Random Access Memory) Memória de acesso aleatório.

É a memória que armazena os dados de entrada para o processamento (programas e dados de informação).

A memória RAM é totalmente apagada quando o computador é desligado. As memórias de um modo geral são divididas em endereços individuais, os bytes; estes endereços tomam a numeração de 0 até 16.383, caso o microcomputador tenha uma RAM de 16K. Cada endereço se refere a um byte (8 bits).

MEMÓRIA ROM

(Read Only Memory) Memória de leitura

A ROM é uma memória que não pode ser apagada. As instruções nela contidas, são previamente programadas pelo fabricante do equipamento.

Dentro das classes de memória ROM temos a EPROM.

MEMÓRIA EPROM

(Erasable Programmable Read Only Memory) Memória de leitura que pode ser programada.

Este tipo de memória pode ser apagada e reprogramada a partir de processos especiais.

Nesta área é que estão o sistema operacional e o interpretador Basic Residente.

SISTEMA OPERACIONAL

São instruções previamente programadas que atuam no controle geral do computador. O sistema operacional é o controlador da unidade central de processamento (CPU) do microprocessador.

BASIC RESIDENTE

São instruções previamente programadas que atuam como Interpretadoras, na linguagem Basic, dos comandos/instruções de seus programas.

Para que você possa saber o conteúdo da memória, em termos gerais, do seu microcomputador, utilize o comando PEEK (este comando já foi mencionado neste manual).

Exemplo:

```
PEEK (1543)
```

Assim procedendo você terá o conteúdo deste endereço da memória.

Com o comando POKE você altera o conteúdo de um endereço da memória RAM. Esta operação é utilizável na confecção de programas, quando necessária.

Exemplo:

```
POKE 1543,80
```

Isto atribui ao byte no endereço 1543 o valor 80.

Para confirmar digite:

```
PRINT PEEK 1543
```

Resposta:

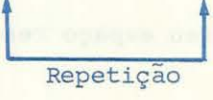
```
80
```

LINGUAGEM DE MÁQUINA

É uma linguagem de baixo nível usada internamente pelo computador; consiste de instruções do microprocessador.

A linguagem de máquina é usada em forma de subrotinas nos programas Basic, para que haja economia de tempo (acima de 1000 vezes mais rápida que o Basic) e do espaço utilizado de memória.

MENSAGENS DE ERRO

- /Ø Divisão por zero, é impossível dividir algo por zero.
- AO Foi feita uma tentativa de abrir (OPEN) um arquivo que já estava aberto.
- BS Erro no dimensionamento (DIM) de um ARRAY.
- CN Seu programa chegou ao fim, executou o comando END, e você digitou CONT para continuar.
- DD Dimensionamento duplo para um mesmo ARRAY.
- Ex.: DIM A (12) e DIM A (5Ø)
- 
- Repetição
- DN Uso incorreto do número indicativo da unidade auxiliar, impressora ou gravador.
- Verificar comandos OPEN, CLOSE, PRINT ou INPUT.
- DS Este erro é causado pela falta de numeração nas linhas/comandos de um programa, quando este for carregado (salvo) em fita.
- FC Função ilegal. Você está indicando algo fora dos padrões. Por exemplo SOUND (260, 260) ou CLS (10).
- FD O erro ocorreu quando você gravou dados no arquivo (PRINT), ou leu dados do arquivo (INPUT). Foi usada uma identificação errada para a variável. Por exemplo, INPUT #-1, A a variável "A" sendo um STRING.
- FM O erro ocorreu quando você tentou executar um comando INPUT num arquivo aberto para a saída de dados (OUTPUT (O)), ou PRINT em arquivo aberto para entrada de dados (INPUT (I)).

- ID Foi usado um comando INPUT sem o identificador da variável de entrada.
- IE Após ao fim do arquivo, foi usado um comando de leitura de dados.
- IO Erro de entrada/saída (I/O). Verifique a fita K-7: deve ter ocorrido um erro no acesso do arquivo.
- LS STRING com muitos caracteres. Excedeu a 255 possíveis.
- NF Comando NEXT sem o comando FOR na execução de um LOOP.
- NO O arquivo não está aberto e foi acessado.
- OD Instrução READ foi executada, mas não haviam dados no comando DATA para serem lidos.
- OM Falta de espaço de memória.
A memória está com todo o seu espaço reservado ou sendo usada.
- OS Falta de espaço para STRINGS; use o comando CLEAR.
- OV Excedeu a capacidade do computador.
- RG O comando RETURN foi codificado; GOSUB não.
- SN Erro de sintaxe. Algo incorreto foi digitado na linha comando. Liste o programa e verifique.
- ST Uma operação muito complexa para a execução. Separe esta operação em outras menores.
- TM Linha comando com uso incorreto.
Foi dado um valor numérico para um STRING.
Foi dado um valor STRING (ALFA) para uma variável numérica.
Ex.: (A\$ = 3), (A = "DATA").
- UL Linha não identificada. Foi executado um comando GOTO ou GOSUB para um desvio incorreto.

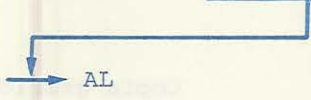
SUMÁRIO DOS COMANDOS EM LINGUAGEM BASIC UTILIZÁVEIS NO CD-6809

PALAVRA	DEFINIÇÃO	EXEMPLO
ABS	Valor absoluto de um nº.	PRINT ABS (-5)
ASC	Converte o primeiro caracter do STRING em código ASCII.	PRINT ASC ("B") X = ASC (T\$)
ATN	Indica o arcotangente em radianos.	Y = ATN(X/3)
AUDIO	Conecta e desconecta o gravador à saída de som da TV.	AUDIO ON AUDIO OFF
CHR\$	Converte em caracter ASCII o código especificado.	PRINT CHR\$ (143) PRINT CHR\$ (67) Y \$ = CHR\$ (32)
CIRCLE	Monta um círculo cujo centro são as coordenadas (X,Y).	CIRCLE (128,96),50,4,1, 5,75
CLEAR	Reserva espaço na memória do seu microcomputador para área de STRING. Sem o comando, o computador reserva uma área de 200 caracteres.	CLEAR CLEAR 500
CLOAD	Carrega o programa especificado da fita cassete. Se não for especificado o nome do programa, será carregado o 1º encontrado na fita.	CLOAD CLOAD "PROGRAMA"
CLOADM	Carrega um programa em linguagem de máquina da fita K-7.	CLOADM "PROG" CLOADM CLOADM "PROG", 1000

PALAVRA	DEFINIÇÃO	EXEMPLO
CLOSE	Fecha arquivo. Interrompe a comunicação entre o microcomputador e os periféricos.	CLOSE # -1 CLOSE # -2
CLS	Limpa a tela colocando a cor verde ou a especificada.	CLS CLS(2)
COLOR	Indica a cor a ser assumida pela tela (cor de fundo) e linhas.	COLOR 1,3
CONT	Continua a execução do programa depois de uma interrupção por ESC ou STOP.	CONT
COS	Indica o cosseno do ângulo em radianos.	Y = COS(7)
CSAVE	Salva o programa da memória para a fita cassete. Você deve usar um nome para o seu programa de no máximo 8 letras.	CSAVE CSAVE "PROGRAMA"
CSAVEM	Salva na fita K-7 um programa em linguagem de máquina.	CSAVEM X,4E,6F,5F
DATA	É o comando que armazena os dados de entrada que serão lidos no comando READ.	DATA 5,3, PERAS DATA PAPEL, LIVROS
DEF FN	Define uma função numérica.	DEF FN(X)=X * 3
DEFUSR	Define endereços da funçãoUSR.	DEFUSR5 = 45643
DEL	Deleta linhas do seu programa.	
	DEL deleta todo o programa	DEL -
	DEL n deleta linha especificada	DEL 3Ø
	DEL -n deleta linhas até o nº especificado	DEL -3Ø
	DEL n- deleta linhas após o nº especificado	DEL 3Ø-
	DEL n-n deleta linhas entre os nºs especificados	DEL 1Ø-3Ø

PALAVRA	DEFINIÇÃO	EXEMPLO
DLOAD	Carrega um programa Basic em um BAUD específico Ø = 3ØØ BAUD 1 = 12ØØ BAUD	DLOAD X,1
DIM	Reserva espaço na memória para o ARRAY.	DIM R(65), W(5Ø) DIM R\$(7,34)
DRAW	Faz a movimentação de uma linha a partir de ponto, tamanho e cor especificados. Se o ponto de início for omitido, (128, 96) ou a última posição, DRAW é usada como referência.	DRAW "BM 1ØØ,1ØØ;E25;F25; G25;H25"
EDIT	Edita no vídeo uma linha específica de seu programa que deve ser alterada.	EDIT 4Ø
END	Indica o fim do programa.	END
EOF	Verifica se o arquivo chegou ao fim.	IF EOF (-1) THEN CLOSE IF EOF (Ø) THEN INPUT #-1
EXP	Indica número exponencial.	Y = EXP(7)
FIX	Mesma função da instrução INT.	X = FIX(6.7)
FOR... TO STEP/NEXT	Estes comandos criam um (LOOP) ligação dentro do seu programa, repetindo os comandos do primeiro nº especificado até o último.	FOR M = 3Ø TO 15 STEP -5/ NEXT M
GET	Lê o gráfico e o armazena em um ARRAY para ser usado posteriormente por um PUT.	GET (5,2Ø)-(3,8),G

PALAVRA	DEFINIÇÃO	EXEMPLO
GOSUB	Desvia o processamento para a subrotina especificada.	GOSUB 500 GOSUB 5000
GOTO	Desvia o processamento para uma linha especificada.	GOTO 300
HEX\$	Computa números e retorna seu valor no sistema hexadecimal. Este número deve estar entre 0 e 65535.	PRINT HEX\$ (30) Y\$ = HEX\$ (X/16)
IF/THEN ELSE	O comando faz um teste relativo, executando então instruções especificadas. No caso de não haver o ELSE, a próxima linha será executada.	IF A = 5 THEN 300 IF B\$ = "YES" THEN PRINT "XYZ" IF A = 3 THEN PRINT "CORRETO" ELSE PRINT "ERRADO"
INSTR	Procura a primeira ocorrência do STRING Y\$.	PRINT INSTR(10,X\$,Y\$)
INT	Converte para um valor inteiro.	X = INT(6.7)
JOYSTK	Indica as coordenadas horizontais e verticais do JOYSTICK da direita e da esquerda.	M = JOYSTK (0) H = JOYSTK (K)
LEFT\$	Define a porção esquerda do STRING. Você deve especificar o tamanho.	P\$ = LEFT\$ (M\$,7)
LEN	Indica o tamanho do STRING.	X = LEN (M\$)
LET	Dá valor a uma variável. Opcional.	LET A\$ = "PROGRAMA"
LINE	Cria uma linha a partir de um ponto inicial até o ponto final especificado. Se for omitido o ponto inicial, (128,96) ou o último ponto é usado.	LINE (5,3)-(6,6),PSET LINE -(191,191),PSET,BF

PALAVRA	DEFINIÇÃO	EXEMPLO
LIST	Lista todo o programa ou as linhas do programa que você especificar, na sua tela.	LIST LIST 50-85 LIST 30 LIST -30 LIST 30-
LLIST	Lista da mesma forma do comando LIST, só que na impressora.	LLIST LLIST 20-50
LOG	Indica logaritmo.	Y = LOG(353)
MEM	Indica a quantidade de espaço que ainda resta na memória.	MEM
(1)MID\$	Especifica uma porção média do STRING. Você deve especificar o tamanho.	PRINT MID\$ ("PALAVRAS"2,3)  → AL
(2)MID\$	Troca uma porção de um STRING por outro STRING.	MID\$ (A\$,15,2) = "TA"
MOTOR	Liga ou desliga o gravador cassete.	MOTOR ON MOTOR OFF
NEW	Limpa a memória	NEW
ON... GOSUB	Especifica um desvio para uma das subrotinas.	ON Y GOSUB 190,200
ON... GOTO	Especifica um desvio para uma das linhas.	ON X GOTO 193,300
OPEN	Abre comunicação com um periférico em função de entrada (I) ou saída (O) de dados. Os periféricos usados são:	OPEN "I", # -1, "ARQUIVO" OPEN "O", # -1, "DADOS"

PALAVRA	DEFINIÇÃO	EXEMPLO
	# - 1 gravador cassete.	
	# - 2 impressora.	
	As indicações para o vídeo (tela da TV) e teclado não são necessárias.	
PAINT	Pinta gráficos a partir de parâmetros específicos.	PAINT (10,80),2,4
PCLEAR	Reserva número específico de páginas de memória para a montagem de gráficos.	PCLEAR 8
PCLS	Limpa a tela com a cor especificada. Se for omitido o código da cor, a coloração do fundo da tela será assumida.	PCLS 3
PCOPY	Copia gráficos de uma página de memória para outra.	PCOPY 5 TO 6
PEEK	Dá a uma variável o valor do conteúdo da memória em um local específico. No exemplo, o conteúdo da memória no local 32076 será atribuído à variável "A".	A = PEEK (32076)
PLAY	Toca música a partir de códigos específicos para as diferentes notas.	PLAY "L1;A#;P8;V10;T3;L2; B-;3;XA\$;"
PMODE	Seleciona o tipo de resolução do gráfico e as páginas de memória necessárias para o programa.	PMODE 4,3
POINT	Especifica o "PONTO" na tela com sua localização horizontal/vertical	IF POINT (15,12) = 3 THEN PRINT "AZUL"

PALAVRA	DEFINIÇÃO	EXEMPLO
PPOINT	Indica o código da cor que está sendo usado em determinado ponto da tela.	PPOINT (15,45)
POKE	Dá um valor a determinado endereço da memória.	POKE 15973,835
POS	Indica a posição do cursor.	PRINT TAB(8) POS(Ø)
PRESET	Retorna a cor de fundo da tela no ponto especificado.	PRESET (6,8)
PRINT	Imprime a mensagem especificada. Quando não houver indicação do periférico, a mensagem será impressa no vídeo.	PRINT "ALO" PRINT A\$ PRINT # -1, A\$ PRINT # -2, "ALO"
PRINT @	Imprime a mensagem em uma posição específica no seu vídeo.	PRINT @ 1Ø, "ALO"
PSET	Especifica a cor de um determinado ponto da tela.	PSET (6,8,4)
PUT	Retira o gráfico colocado no ARRAY pelo comando GET e o posiciona na tela.	PUT (3,2)-(5,6),V,PSET
READ	Lê o próximo item de dados do comando DATA.	READ A\$ READ C, B
REM ou *	Serve para a inserção de comentários nos programas.	REM ESTA E A PARTE PRINCIPAL
RENUM	Renumeras as linhas de um programa.	RENUM 1ØØØ,5,1ØØ
RESET	Apaga o PONTO do vídeo na localização especificada.	RESET (14,15)

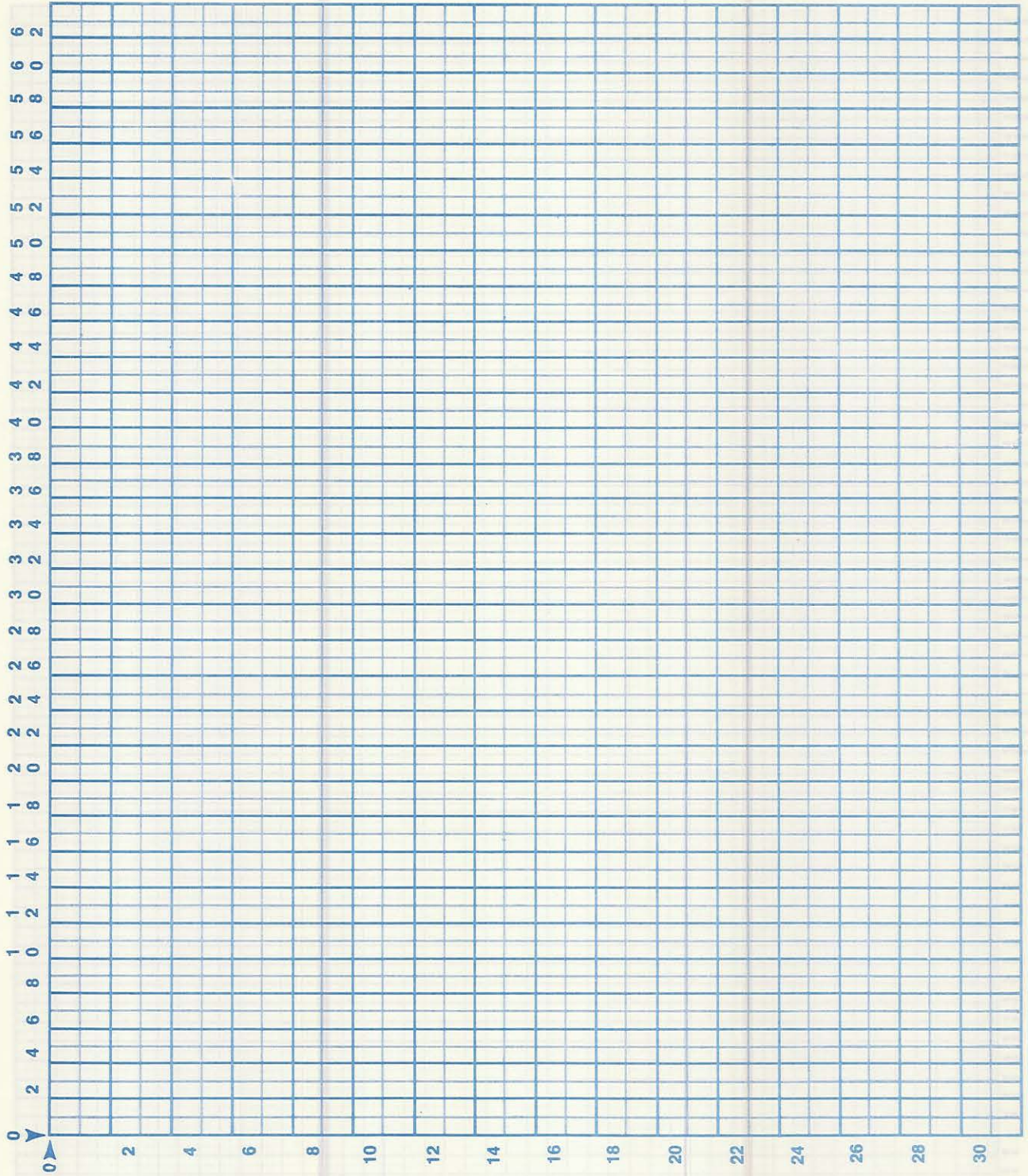
PALAVRA	DEFINIÇÃO	EXEMPLO
RESTORE	Indica ao computador o primeiro dado de um comando DATA. Possibilita o uso de uma mesma carga de dados repetidas vezes.	RESTORE
RETURN	Retorna o comando do computador a linha seguinte, após a execução do GOSUB.	RETURN
RIGHT\$	Especifica a porção direita do STRING. Você deve especificar o tamanho.	ZP\$ = RIGHT\$(AD\$,5) PRINT RIGHT\$("ONE",2)
RUN	Executa o programa.	RUN
SCREEN	Indica o tipo do programa. Texto ou gráfico e a cor usada.	SCREEN 1,0
SET	Cria o ponto no vídeo a partir de endereços na horizontal e vertical. (Dois primeiros dígitos). Especifica a cor do ponto no terceiro dígito.	SET (14, 13, 8)
SGN	Indica o sinal do número.	PRINT SGN (-4) X = SGN (A * B)
SIN	Indica o seno de um ângulo em radianos.	Y = SIN(5)
SKIPF	Pula para o fim do próximo programa na fita cassete, ou para o fim do programa especificado.	SKIPF SKIPF "PROGRAMA"
SOUND	Indica ao computador o tipo de som e a duração que o mesmo deverá ser emitido. Os sons vão de 1 até 255.	SOUND 128, 3

PALAVRA

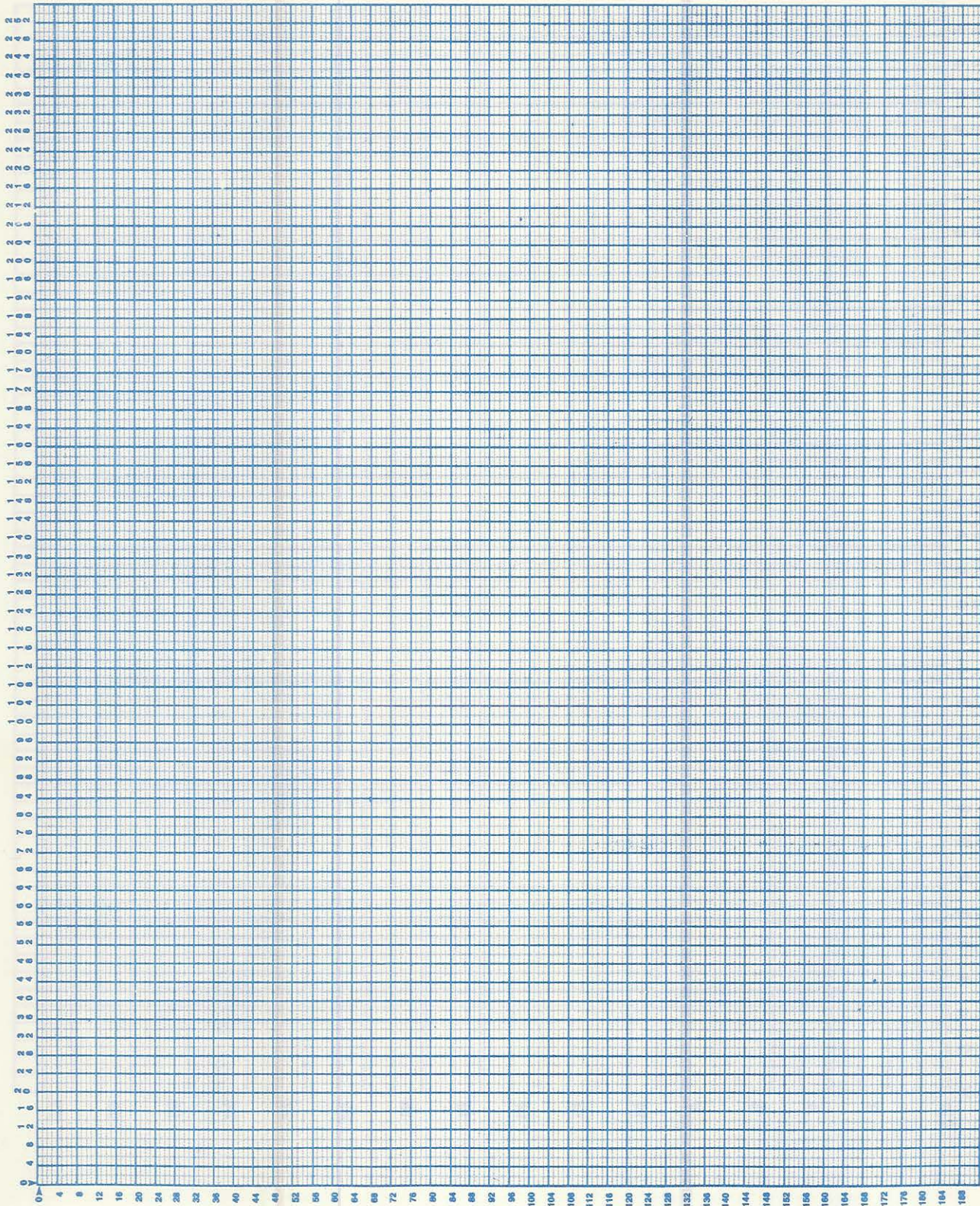
DEFINIÇÃO

EXEMPLO

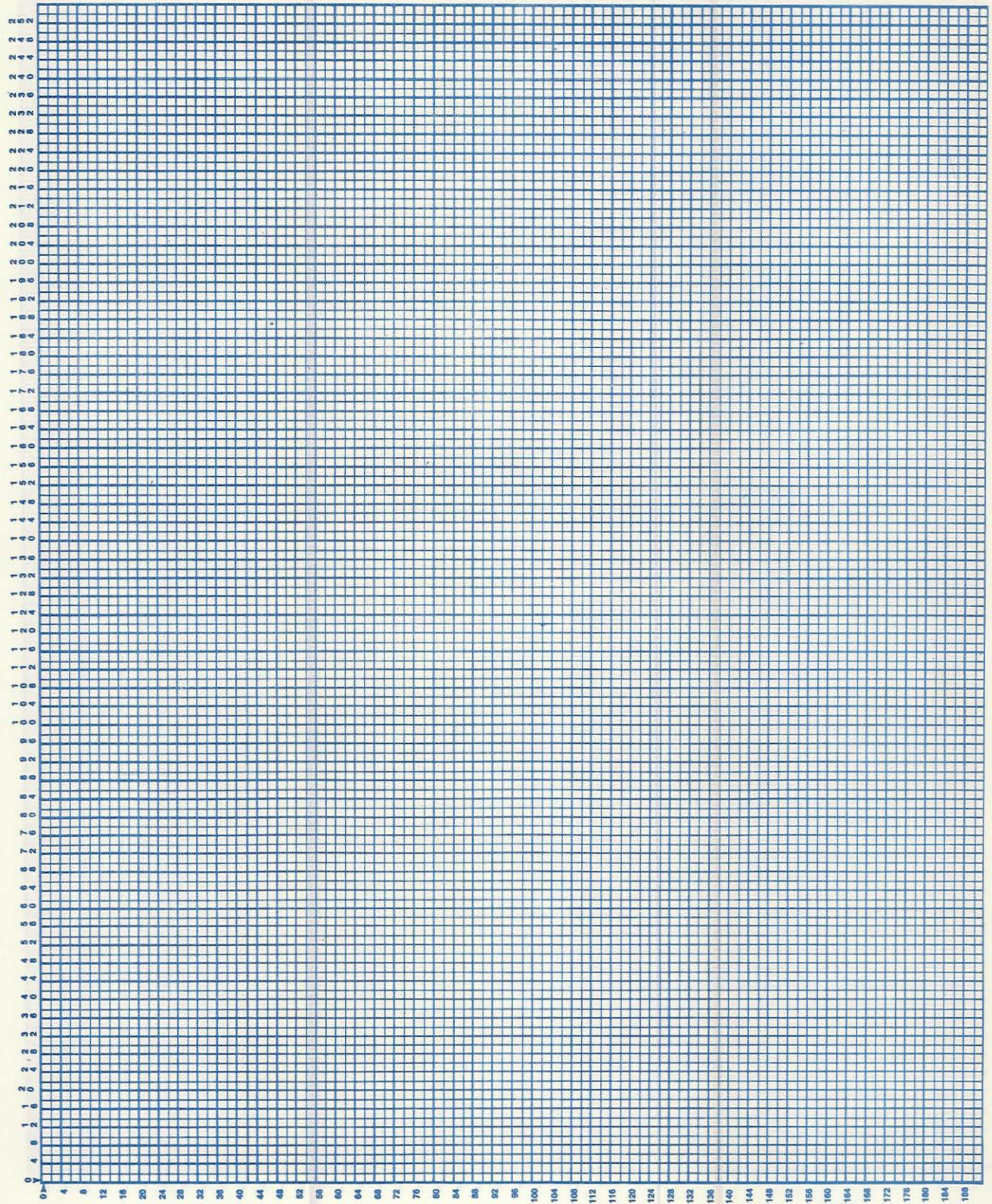
SQR	Indica a raiz quadrada do número.	$Y = \text{SQR}(5+3)$
STOP	Interrompe a execução de um programa.	STOP
STR\$	Converte um número em um STRING.	$X\$ = \text{STR}\(5) $X = \text{STR } \$ (4)$
TAB	Especifica a coluna em que deverá ser impressa a mensagem.	PRINT TAB(2) "ALO" PRINT # = -2, TAB(5) "ALO"
TAN	Indica a tangente de um ângulo em radianos.	$Y = \text{TAN}(45,7)$
TIMER	Retorna o conteúdo do temporizador interno.	? TIMER TIMER = \emptyset
TROFF	Interrompe a instrução ativada pelo comando TRON.	TROFF
TRON	Ativa a instrução que possibilita a visualização dos passos (linha após linha) na execução de um programa.	TRON
USR	Chama uma subrotina em linguagem de máquina.	$X = \text{USR}(Y)$
VAL	Converte um STRING em um NÚMERO.	$A = \text{VAL}(B\$)$
VARPTR	Indica o ponto de referência de uma variável especificada.	$Y = \text{USR}(\text{VARPTR}(X))$



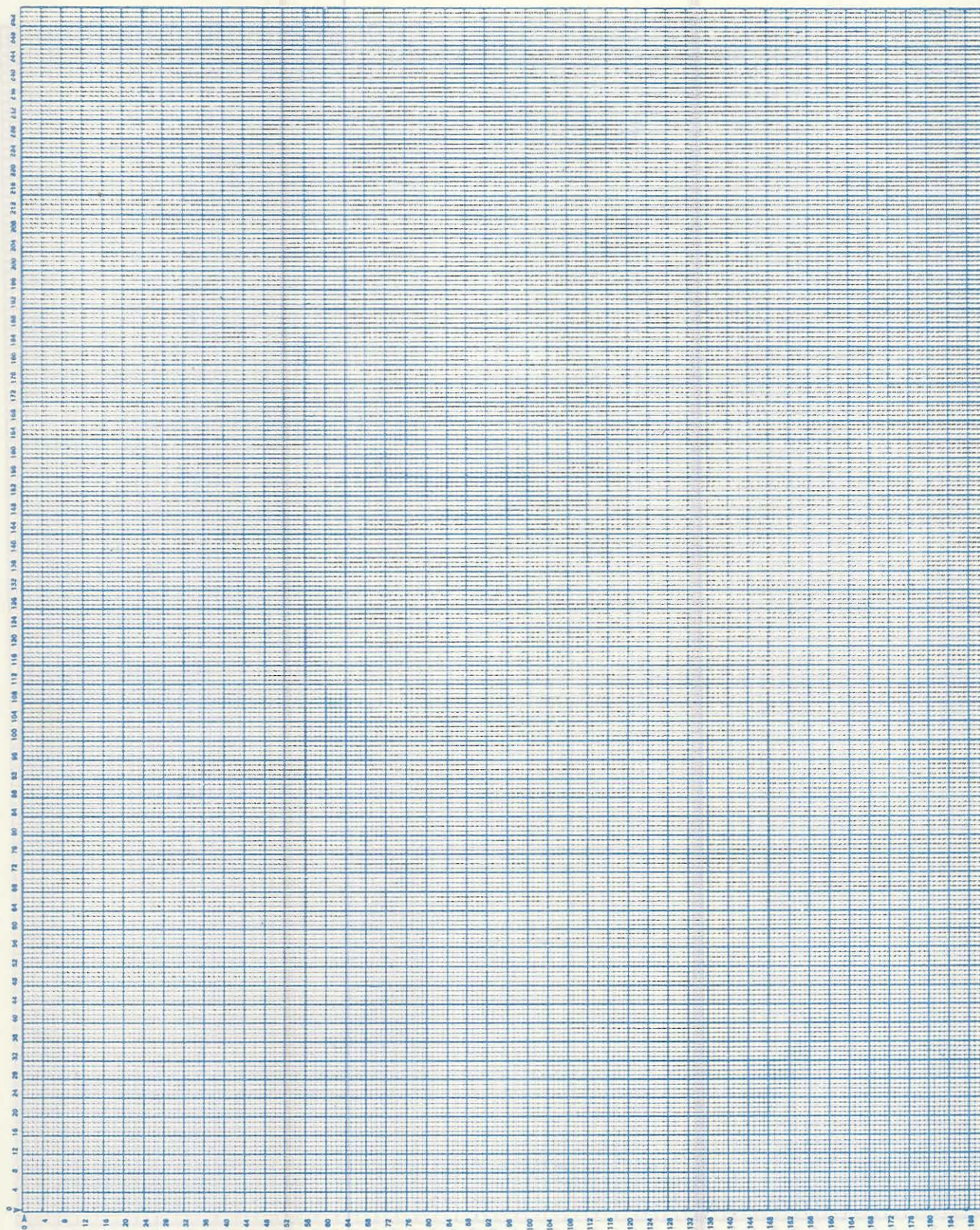
MODELO DE TELA PARA GRÁFICOS COM DIMENSÃO (128 x 192)



MODELO DE TELA PARA GRÁFICOS COM DIMENSÃO (128 x 96)



MODELO DE TELA PARA GRÁFICOS COM DIMENSÃO (256 x 192)



ÍNDICE

Este índice relaciona a primeira ou as mais importantes localizações de comandos, itens e funções apresentados no manual.

	PÁG.		PÁG.
ABS	40	MEM	38
AND/OR	39	(1)MID\$	43
ARRAY	50, 99	(2)MID\$	133
ATN	129	MOTOR	60
AUDIO	60	NEW	12
BIT	124	ON ... GOSUB	38, 39
BYTE	124	ON ... GOTO	39
CHR\$	52, 106	OPEN	55, 57
CIRCLE	84, 90	PAINT	92
CLEAR	4, 129	PCLEAR	83
CLOAD	27	PCLS	71, 82
CLOADM	129	PCOPY	85
CLOSE	55, 57	PEEK	62, 126
CLS	8	PLAY	100
COLOR	77	PMODE	71, 79
COMPILADOR	124	POINT	49
CONT	38	PPOINT	86
COS	130	POKE	135
CSAVE	26	POS	135
CSAVEM	130	PRESET	73
DATA	32	PRINT	6, 40, 45, 59
DEF FN	130	PRINT @	34
DEFUSR	130	PSET	71, 72
DEL	130	PUT	98
DLOAD	131	RAM	125
DIM	51, 100	READ	32
DRAW	93	REM OU '	36, 84
EDIT	16	RENUM	135
END	35	RESET	31
EOF	56	RESTORE	32
EPROM	125	RETURN	4, 35
ESC	4	RIGH\$	43
EXP	131	RND	24
FIX	43	ROM	125
FOR ... TO STEP/NEXT ...	20, 21, 30	RUN	12
GET	98	SCREEN	71, 81, 88
GOSUB	35	SET	30
GO TO	14	SGN	39
HEX\$	132	SIN	136
IF/THEN ELSE	24, 64	SKIPF	28
INKEY\$	46	SOUND	9, 20
INPUT	13, 36	SQR	137
INSTR	132	STOP	37
INT	33	STR\$	40
INTERPRETADOR	124	TAB	137
JOYSTICK	3	TABELAS	120
JOYSTK	61	TAN	137
LEFT\$	43	TIMER	137
LEN	43	TROFF	137
LET	10	TRON	137
LINE	74, 75	USR	137
LIST	13, 15, 23	VAL	137
LLIST	15, 59	VARPTR	137
LOG	133		

TERMO DE GARANTIA

O computador doméstico CD-6809 é garantido para o 1º comprador, contra eventuais defeitos de materiais e/ou de fabricação, pelo período de 6 meses a contar da data da compra. Esta garantia perderá sua validade se o computador sofrer uso inadequado, for violado/modificado por pessoas ou empresas não autorizadas ou danificado por acidente; fica subentendido que a CODIMEX não cobre qualquer outro tipo de garantia.

Para se beneficiar desta garantia, o computador deverá ser entregue ou enviado à nossa fábrica, juntamente com a apresentação da nota fiscal, correndo por conta do comprador quaisquer despesas acessórias, inclusive remessa do produto.

As instruções deste manual são constantemente revisadas. Erros e omissões são passíveis de ocorrer. A CODIMEX agradece por sugestões, adendos e correções que permitam aprimorar permanentemente a publicação.

O conteúdo deste manual é fornecido sem nenhuma garantia. A CODIMEX não se responsabiliza pela utilização e uso deste material em todo ou em parte, incluindo os programas, acrescidos a título informativo, vetando-se expressamente o seu uso comercial.

A reprodução total ou parcial deste manual, por quaisquer meios, somente será permitida após consulta prévia por escrito à CODIMEX.

CODIMEX

IMPORTAÇÃO, EXPORTAÇÃO E INDÚSTRIA DE COMPUTADORES LTDA.
AV. WENCESLAU ESCOBAR, 1549 - CEP 90.000 - PORTO ALEGRE (RS)
TELEFONE (0512) 49.8446.