

**TK 2000/III ENTENDENDO A ROM GERALDO COEN**

# **TK 2000/III**

## **ENTENDENDO A ROM**



**GERALDO COEN**



**micromega**

**DIVISÃO MICRODIGITAL**

# **TK 2000/II**

## **ENTENDENDO A ROM**

Um estudo completo da ROM  
de um computador  
baseado no 6502

Geraldo Coen

## Geraldo Coen

É um matemático que, desde 1967 trabalha em Processamento de Dados. Atuou nas áreas de metodologia, suporte e software de base em equipamentos de médio e grande porte. Realizou interpretadores e sistemas em tempo real. A partir de 1976, dedicou-se ao desenvolvimento de projetos de microcomputadores, tendo projetado alguns dos produtos mais difundidos no mercado nacional. Atualmente é consultor independente em projetos de software e sistemas.

### **Advertência:**

Esta obra é resultado de um trabalho do autor, pessoal e independente, de pesquisa da ROM do TK 2000 e não constitui uma listagem oficial ou aprovada pelo fabricante.

Apesar do esforço neste sentido e das inúmeras revisões que foram feitas, não se certifica a absoluta precisão das informações aqui contidas nem do uso que se venha a fazer delas.

Os Editores

## Índice

Prefácio .....	5
Introdução .....	7
Definições .....	9
I - O Conteúdo da ROM .....	11
1. As Rotinas da ROM .....	11
2. A Distribuição dos Módulos .....	11
II - O Uso da RAM .....	13
1. A Distribuição das Áreas da RAM .....	13
III - A Operação do Interpretador .....	15
1. A Entrada de Linhas e Sua Análise .....	15
2. A Interpretação e Execução .....	16
3. As Rotinas de Execução de Comandos .....	17
4. As Rotinas Aritméticas e Funções .....	17
4.1. Formatos de Variáveis .....	17
5. As Rotinas de I/O .....	22
5.1. Rotina de Varredura do Teclado .....	22
5.2. Rotinas de Interpretação do Buffer IN .....	22
5.3. Saída de Dados Pelo Vídeo .....	23
6. Os Utilitários .....	28
IV - Pontos de Entrada de Rotinas .....	29
V - As Tabelas .....	35
Listagem da Rom .....	37
Bibliografia .....	210



## **Prefácio**

Desde o seu lançamento, o TK 2000 tem comprovado na prática ser um computador versátil. Por estas e outras características, o TK 2000 rapidamente se tornou o mais popular computador baseado no 6502.

Seus usuários, já bastante numerosos, sentiam a necessidade de conhecer as entranhas deste computador, para poder tirar o máximo proveito em seus programas em linguagem de máquina.

Com este livro, milhares de usuários poderão satisfazer sua curiosidade a respeito de importantes detalhes do comportamento do equipamento que possuem, evoluindo mais um passo no completo domínio do TK 2000 e de um dos microprocessadores mais utilizados no mundo todo, o 6502.

O autor teve a preocupação de não transformar este livro em um tratado técnico-científico, mas de fornecer ao usuário sua coleção de notas de trabalho, compiladas de forma prática.

Estas notas são fruto de um árduo trabalho de um especialista de *software* de renome, o Sr. Geraldo Coen, e serão entregues "de bandeja" aos aficionados de computação que desejavam desenvolver bons programas em linguagem de máquina.

É uma grande satisfação ver surgir uma obra desta envergadura, verdadeiramente nacional, o que representa um fato inédito na literatura técnica brasileira.

Parabéns a Geraldo Coen pela sua contribuição à tecnologia nacional de *software*, através da sua disposição em por em livro o fruto de seu trabalho pessoal na análise minuciosa do TK 2000.

Os Editores



## Introdução

Este é um livro de *software* dirigido aos usuários do TK 2000 e de outros equipamentos baseados no microprocessador 6502, e a pessoas simplesmente interessadas na matéria. Resultado de uma análise minuciosa da ROM do TK 2000, o programa, nela contido, foi estudado e comentado, em detalhes, usando minha própria experiência profissional e de bibliografia sobre a matéria, em particular no que se refere aos fundamentos de desenvolvimento de *software* básico, teoria de interpretadores, algoritmos numéricos.

Com a publicação completa desta análise, é meu objetivo propiciar ao leitor uma melhor compreensão de algumas técnicas importantes de *software*, ao mostrar um programa longo e não trivial, que inclui, entre outras a implementação de um monitor, um miniassembler, um interpretador BASIC, rotinas de comunicação com o operador, e, por último, rotinas básicas de entrada e saída (*drives* de I/O).

Àqueles que quiserem alterar o *software* básico de guia para compreensão da listagem detalhada da ROM, incluindo também a distribuição dos módulos do *firmware*, sua estrutura básica, além de um esquema de funcionamento e uso que é feito da RAM.

Advertimos o leitor que, para o entendimento da listagem é pressuposto o conhecimento do microprocessador 6502 e das rotinas básicas do TK 2000.

Agradeço à minha irmã Lorette, pelo incentivo, e à minha esposa Gladis pelo apoio neste trabalho.





## DEFINIÇÕES

Para a exata compreensão das rotinas da ROM, descritas na listagem, o leitor deverá ter conhecimento das definições dos seguintes termos:

DESCRITOR - cadeia de caracteres que descreve uma função

DP - ponto decimal (Decimal Point)

FP - ponto flutuante (Floating Point)

LITERAL - seqüência de caracteres, por ex., "[caractere]"

MICROLINHA - linha de vídeo de alta resolução

NASCII - código ASCII negativos (\$80 a \$FF)

PARSE - análise da gramática; forma pela qual o computador interpreta o que está sendo digitado e coloca em ordem de execução

PASCII - código ASCII positivos (\$00 a \$7F)

SINTAXE - refere-se à estrutura dos comandos do computador, à sua ordem e às várias partes destes comandos.

STRINGS - literal fechada por aspas, por ex., "[caractere]"

TOKEN - código de palavras-chave referentes aos comandos da linguagem



## I - O CONTEÚDO DA ROM

### 1. AS ROTINAS DA ROM

A ROM incorpora os seguintes grupos de rotinas:

- . Rotinas de I/O físico ("drivers")
- . Implementação do interpretador BASIC
- . Rotinas de execução dos operadores aritméticos e funções BASIC
- . Rotinas de apoio ao BASIC: alocação de memória para variáveis e tabelas, acesso a variáveis e tabelas, acesso a periféricos e outras
- . Monitor do teclado, comunicação com o operador
- . Rotinas utilitárias: LIST, LOAD etc.
- . Rotinas de inicialização: RESET etc.
- . Monitor
- . Mini-assembly

### 2. A DISTRIBUIÇÃO DOS MÓDULOS

A distribuição dos módulos da ROM está esquematizada no mapa a seguir (a distribuição detalhada deve ser acompanhada na listagem, pois muitas rotinas são comuns a módulos diferentes):

```

: Mini-assembler
: Comando SOUND
: Comando PDL
-----
: Tabelas:
: Endereços de funções
: Palavras-chave
: Mensagens
-----
: Leitura de linhas
: Emissão de mensagens
-----
: Análise de linhas
: Conversão de tokens
: Comandos: GOTO
:           LIST
:           FOR-NEXT
:           STOP
:           GOSUB ...
: Impressão de strings
-----
: Avaliação de fórmulas:
: Obter variáveis
: Obter elementos de arrays
: Funções unárias
: Comparação
: Multiplicação inteira
: Execução de funções FN
: Obter espaço para strings
: Recuperar espaço livre
: Funções para strings:
:   LEFT$, ...
: PEEK, POKE
: Aritmética ponto flutuante:
: Soma, Subtração
: Multiplicação, Divisão, Log
: Movimentação de números
: Arredondamento
: Normalização
: Comparação
: Conversão em string
: SQR, EXP, RND
: SIN, COS, TAN, ATN
-----
: Rotinas gráficas:
: Baixa resolução
: Cálculo de coordenadas
: HLINE, VLINE, PLOT, ...
: Manipulação de formas
: DRAW, XDRAW, ...
-----
: Rotinas de I/O

```

## II - O USO DA RAM

### 1. A DISTRIBUIÇÃO DAS ÁREAS DA RAM

Durante a execução normal de programas BASIC, o TK 2000 utiliza áreas bem definidas da RAM. O mapa à esquerda mostra a distribuição destas áreas. Deve ser usado em conjunto com o mapa físico, à direita, que fornece os endereços das áreas de texto, de alta resolução, de uso das páginas 0, 1, 2, ...

: ROM	:
: \$73 - \$74 (MENSIZ)	:
: Area de RAM	:
: Strings	:
: Arrays de strings	:
: \$6F - \$70 (FRETOP)	:
:	:
: Area Livre	:
:	:
: \$6D - \$6E (STREND)	:
: Arrays	:
:	:
: \$6B - \$6C (ARYTAB)	:
: Variáveis simples	:
:	:
: \$69 - 6A (VARTAB)	:
: \$AF - \$B0 (PRGEND)	:
: Programa	:
:	:
: \$67 - \$68 (TXTTAB)	:
:	:
: Uso do Sistema	:
: Páginas 0, 1, ...	:

0000	USO DO SISTEMA	
	2K	
07FF		
0800	RAM (6K)	
1FFF		
2000	PRIMEIRA PAG.	MA
3FFF	VIDEO 8K (RAM)	
4000		
	RAM (24 K)	
9FFF		
A000	SEGUNDA PAG.	MP
BFFF	VIDEO 8K (RAM)	
C000		
C0FF	I/O	
C100		
	ROM	
FFFF		

As indicações no mapa correspondem ao nome das variáveis que contém os endereços (pointers) destas áreas. Tais variáveis são definidas e usadas na listagem da ROM. Será dado aqui o endereço destas variáveis e de algumas outras importantes para a compreensão do programa.

- Notas:
1. Todos os valores estão em hexadecimal.
  2. Os endereços ocupam 2 bytes, com o byte de baixa ordem antes do byte de alta ordem.
  3. O valor decimal destas variáveis pode ser obtido através do comando PRINT PEEK (...) + 256 \* PEEK (... + 1).

TXTTAB	67,68	Início das linhas do programa; normalmente começa com 800.
VARTAB	69,6A	Início da área de variáveis simples, pointers a strings e a funções. Deslocado pelo comando LOHEM.
ARYTAB	6B,6C	Início de arrays de variáveis e strings.
STREND	6D,6E	Fim da área de arrays de variáveis e strings.
FRETOP	6F,70	Fim da área livre, fim da área de strings.
MENSIZ	73,74	Fim da área usada pelo BASIC, início da área de strings.
CURLIN	75,76	Número da linha que está sendo executada (linha corrente).
OLDLIN	77,78	Número da linha interrompida por CTRL-C, STOP ou END.
OLDTXT	79,7A	Endereço do último byte da última linha executada.
DATLIN	7B,7C	Número da linha da qual os dados DATA estão sendo obtidos.
DATPTR	7D,7E	Endereço do próximo byte de DATA a ler.
INPPTR	7F,80	Aponta ao buffer de teclado durante INPUT ou ao comando DATA corrente durante READ.
PRGEND	AF,80	Fim das linhas do programa.

### III - A OPERAÇÃO DO INTERPRETADOR

#### 1. A ENTRADA DE LINHAS E SUA ANÁLISE

A entrada de linhas de programa obedece à seguinte seqüência:

1. Obter a próxima linha do teclado.
2. Trocar palavras reservadas por "tokens".
3. Saltar para passo 6, se for um comando de sistema (RUN ou linha sem número).
4. Guardar comando na área de programa (TXTTAB).
5. Voltar a 1.
6. Interpretar e executar.

A cada palavra-chave corresponde um "token". O token é determinado pela rotina de análise de linha PARSE. É o valor da posição da palavra-chave na tabela de palavras-chave TOKTAB mais \$80.

Exemplos: DIM (palavra) → \$80 + \$6 = \$86 (token)  
 HPL0T (palavra) → \$80 + \$13 = \$93 (token)

As linhas "tokenizadas" são guardadas na área que começa com TXTTAB. Cada linha está precedida de um pointer à linha seguinte e do seu número de linha, em binário. Cada linha é terminada com um byte 00.



001	002	003	004	005	006	007	008
08	08	64	00	8A	41	00	

008 ENDEREÇO DO PROXIMO  
 100 EM HEXA  
 TOKEN PRINT  
 LETRA "A"  
 FIM DE LINHA

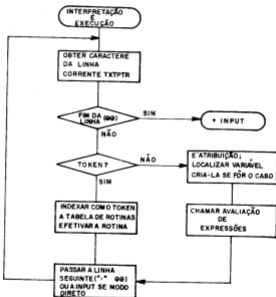
EXEMPLO DA LINHA 100PRINT A



## 2. A INTERPRETAÇÃO E EXECUÇÃO

A interpretação e início de execução de cada linha começa em EXECUTE. O diagrama abaixo mostra a análise do início da linha até a chamada da rotina de execução. Cada rotina de execução se encarrega de analisar o restante da linha.

No modo programado, esta rotina é executada continuamente para as linhas que estão na memória. Em modo imediato, o BASIC volta a aguardar a entrada de informações pelo teclado após esta rotina.



### 3. AS ROTINAS DE EXECUÇÃO DE COMANDOS

Cada rotina de execução de um comando particular analisa o restante da linha e executa o comando (INPUT, GOSUB, PRINT etc.).

Particularmente, o comando de atribuição (LET) analisa a expressão aritmética que deve ser atribuída à variável (FRNEVL).

### 4. AS ROTINAS ARITMÉTICAS E FUNÇÕES

Estas rotinas tratam números e strings. Os números são tratados sempre no formato interno de ponto flutuante. Observar que existem pseudo-acumuladores especiais para trabalhar com números ponto flutuante (FAC e ARG, TEMP1, TEMP2, TEMP3).

#### 4.1. Formatos de Variáveis

##### 4.1.1. Variáveis reais

Exemplo: A = 40000

BYTE	DESCRIÇÃO	HEX	DEC	ASC
0	Nome da 1o byte Caractere Alfabético (pos.)	41	65	A
1	Variável 2o byte Caractere Alfanumérico (pos.)	00	0	0
2	Expoente (M.s. byte) byte mais significativo	190	144	-
3	(M.s. byte) byte mais significativo	1C	28	-
4	Mantissa	40	64	-
5		00	0	-
6	(m.s. byte) byte menos significativo	00	0	-

Nome da variável = 1o byte deve ser um caractere alfabético

2o byte pode ser um caractere alfabético ou numérico e sua existência é facultativa

Obs.: Sendo variável real, o bit 7 do primeiro e do segundo byte deve ser 0, isto é, positivo.

O nome da variável pode conter mais do que dois caracteres, mas apenas o primeiro e o segundo são significativos, por exemplo, AB e AB30S são ambas a mesma variável.

Expoente = dado no 3o byte da variável, com valor máximo de ±37

Obs.: O bit 7 deste byte dá o sinal do expoente: + (se = 0) ou - (se = 1).

Mantissa = valor da variável

No exemplo, A = 40.000  
nome da variável      valor da variável

#### 4.1.2. Variáveis inteiras

Exemplo: BX = 32767

BYTE	DESCRIÇÃO	HEX	DEC	ASC
0	Nome da	1o byte	Caractere Alfabético (neg.)	C2   194   B
1	Variável da	2o byte	Caractere Alfanumérico (neg.)	80   128   0
2	Valor da	byte mais significativo		7F   127   -
3	Variável da	byte menos significativo		FF   255   -
4		não usado		00   0   -
5		não usado		00   0   -
6		não usado		00   0   -

Nome da variável = 1o byte deve ser uma letra

2o byte pode ser uma letra ou um número e pode não existir

Obs.: Sendo variável inteira, o bit 7 do primeiro e do segundo byte deve ser 1, ou seja, negativo.

Valor da variável = dado pelo 3o e 4o bytes e sendo, no máximo, igual a 7FFFH (32767 em decimal)

No exemplo, BX = 32767

nome da variável      valor da variável

Obs.: O byte mais significativo contém em seu bit mais significativo o sinal do valor: 0 (positivo) ou 1 (negativo).

### 4.1.3. Variável pointer a string

Exemplo: CDS = "HELLO"

BYTE	DESCRIÇÃO	HEX	DEC	ASC
0	Nome da	10	17	C
1	Variável	20	32	D
2	Tamanho do String	05	5	-
3	Endereço do	byte menos significativo do endereço	FB	251
4	Parâmetro do	byte mais significativo do endereço	BF	143
5		não usado	00	0
6		não usado	00	0

Nome da variável = 10 byte deve ser um caractere alfabético

20 byte pode ser qualquer caractere alfanumérico e pode inexistir

Obs.: Sendo variável pointer a string, o bit 7 do primeiro byte será 0 (positivo), e o do segundo byte, 1 (negativo).

Tamanho do string = dá o tamanho da variável

No exemplo, "HELLO" tem cinco letras, portanto, seu valor será 5:

```
CDS = "HELLO"
      12345
```

```
nome da      tamanho
variável     do string
```

Endereço do parâmetro = dá o endereço de memória onde se encontra o parâmetro

No exemplo, é BFFB

Obs.: Os strings são guardados na área FRETOP, que é alocada do fim para o começo. A área dos strings livres é recuperada, quando necessário, pela rotina GARBAG, que recoloca os strings em uso de modo a eliminar espaços vagos.

#### 4.1.4. Variáveis funções (Descritor)

Exemplo: DEF FN EF(GH) = XY

BYTE	DESCRIÇÃO	HEX	DEC	ASC
0	Nome 1o byte Caractere Alfabético (neg.) da	C5	197	E
1	Variável 2o byte Caractere Alfanumérico (pos.)	46	70	F
2	Endereço byte menos significativo do endereço da	0E	14	-
3	Definição byte mais significativo do endereço	08	8	-
4	Endereço byte menos significativo do endereço do	1B	27	-
5	Parâmetro byte mais significativo do endereço	08	8	-
6	Representação em ASCII do carac. seguinte "="	5B	91	X

Nome da variável = 1o byte deve ser sempre uma letra

2o byte pode ser uma letra ou um número

Obs.: Sendo variável função, o bit 7 do primeiro byte será 1 (negativo), e o do segundo byte, 0 (positivo).

Endereço da definição = dá o endereço onde está a definição

No exemplo, é FN

Endereço do parâmetro = dá o endereço do parâmetro da definição

No exemplo, é (GH)

Representação em ASCII = dá o próximo caractere após o sinal de = (igual) em ASCII

No exemplo, é X

DEF	FN	EF	(GH)	=	XY
	tipo de função	nome da variável	parâmetro		valor

#### 4.1.5. Variáveis arrays

Exemplo: DIM JK% (2,3)

BYTE:	DESCRIÇÃO	HEX:	DEC:	ASC:
0	Nome 1º byte Caractere Alfabético (neg.) da	CA	202	J
1	Variável 2º byte Caractere Alfanumérico (neg.)	CB	203	K
2	Deslocamento byte menos sign. do deslocamento da próxima	21	33	-
3	Variável byte mais signif. do deslocamento	00	0	-
4	Índice de Dimensões	02	2	-
5	Tamanho byte mais signif. da última dimensão da última	00	0	-
6	Dimensão byte menos sign. da última dimensão	04	4	-
7	Tamanho byte mais signif. da primeira dimensão da primeira	00	0	-
8	Dimensão byte menos signif. da primeira dim.	03	3	-

Nome da variável = 1º byte deve ser sempre uma letra

2º byte poderá ser uma letra ou um número

Obs.: Sendo variáveis arrays, o bit 7 do primeiro e do segundo bytes será 1 (negativo).

Deslocamento da próxima variável = dado pelo 3º e 4º bytes

Obs.: No exemplo, a matriz de 3X4 dá doze posições: (0,0); (1,0); (2,0); ... ; (2,3). Como cada posição ocupa 2 bytes, 12 (posições) X 2 = 24 bytes. Os 24 bytes + 9 bytes da variável são iguais a 33 bytes (= 21 em hexadecimal).

Número de dimensões = número de linhas e de colunas

No exemplo, são duas (2 e 3)

Tamanho da última dimensão = 4 (0, 1, 2, 3), no exemplo

Tamanho da primeira dimensão = 3 (0, 1, 2), no exemplo

## 5. AS ROTINAS DE I/O

As rotinas de I/O são rotinas básicas pelas quais se processa a comunicação usuário-computador. Homem e máquina têm como canais de expressão, respectivamente, teclado e vídeo.

### 5.1. Rotina de Varredura do Teclado

A rotina SCAN1, alocada em \$F043, encarrega-se de fazer a varredura do teclado, verificando se há teclas acionadas.

Sendo detectada tecla(s) pressionada(s), SCAN1 verifica a posição física da matriz e, então, codifica em ASCII. O código é obtido na tabela TABASC, alocada em \$F59D.

A tabela TABASC é dividida em grupos de quatro elementos. O primeiro elemento corresponde à tecla; o segundo, à tecla em conjunto com CONTROL; o terceiro, à tecla em conjunto com SHIFT; e o quarto, à tecla em conjunto com SHIFT e CONTROL. Para cada tecla pressionada, haverá a impressão de um destes elementos no vídeo ou, então, algum flag de controle será alterado.

### 5.2. Rotinas de Interpretação do Buffer IN

Nas operações de entrada de dados, cada código ASCII referente à digitação é armazenado no buffer IN, até que a tecla return seja detectada. Pressionada a tecla return, o conteúdo de IN será avaliado como uma linha de informação do operador para a máquina.

Como o buffer é limitado (IN = \$0200 a \$02FF), havendo transposição do número de informações numa linha, IN reinicializa, perdendo toda a informação anterior.

- Notas:
1. O TK 2000 identifica três tipos de caracteres: os alfanuméricos, os gráficos e os tokens (palavras).
  2. Todos os códigos ASCII serão interpretados como alfanuméricos ou como um comando de controle.
  3. A interpretação de IN inicia-se com a rotina ZXY, alocada em \$F55D.
  4. Para a interpretação de IN há outro buffer, auxiliar: ING, localizado na página 6, \$0600.

Através do auxílio de ING, as rotinas primeiramente interpretam o código contido em IN. Em seguida, devolve a IN o código já interpretado, ou seja, pronto para ser trabalhado pelas outras rotinas do sistema. Isto é feito do seguinte modo:

**5.2.1.** Detectado um CONTROL-B (ASCII = #5B2), o bit menos significativo de \$0452 (GRAFFLO) será invertido, sinalizando-se a presença de um caractere gráfico.

Sendo o próximo código contido em IN um caractere ASCII, este entrará normalmente como um caractere alfanumérico.

Sendo um ASCII, antes que seja colocado em ING, um caractere de controle será posto junto (#graf = #5F2), e ao código ASCII será somado #540, determinando se tratar de um caractere gráfico.

Assim será até que novo CONTROL-B seja dado "desligando" a sinalização de caracteres gráficos.

Nota: Para cada caractere gráfico são usados dois bytes.

**5.2.2.** Não havendo um CONTROL-B, e sendo encontrado um código ASCII, a rotina irá procurar o "token" na tabela TOKTABL, alocada em \$C3B0. Então, os códigos ASCII correspondentes ao token serão colocados em ING, em código ASCII, até que a palavra toda seja formada.

### 5.3. Saída de Dados pelo Vídeo

A saída de dados pelo vídeo pode se dar por três formas:

- . Alta resolução
- . Baixa resolução
- . Texto (alfanumérico e gráfico)

A área de vídeo do TK 2000 tem duas páginas fixas de 8 Kbytes cada uma, seja para alta resolução, baixa resolução ou texto. A figura a seguir mostra uma página de vídeo:



00  
 01  
 02  
 03  
 04  
 05  
 06  
 07  
 08  
 09  
 10  
 11  
 12  
 13  
 14  
 15  
 16  
 17  
 18  
 19  
 20  
 21  
 22  
 23  
 24  
 25  
 26  
 27  
 28  
 29  
 30  
 31  
 32  
 33  
 34  
 35  
 36  
 37  
 38  
 39

\$ 2000	0192	
\$ 2010	0320	
\$ 2100	0440	
\$ 2100	0570	
\$ 2200	0704	
\$ 2200	0832	
\$ 2300	0960	
\$ 2300	1088	
\$ 2020	0222	
\$ 2040	0360	
\$ 2120	0480	
\$ 2140	0610	
\$ 2220	0744	
\$ 2240	0872	
\$ 2320	1000	
\$ 2340	1120	
\$ 2050	0272	
\$ 2020	0400	
\$ 2150	0520	
\$ 2100	0650	
\$ 2250	0784	
\$ 2200	0912	
\$ 2350	1040	
\$ 2320	1160	

EM CADA CADA XAI

	0	\$ 00
	1024	\$1024
	2048	\$2048
	3072	\$3072
	4096	\$4096
	5120	\$5120
	6144	\$6144
	7168	\$7168

A forma mais elementar do sistema é a alta resolução, onde há 192 linhas horizontais. Na baixa resolução, os blocos são criados através de software, de quatro em quatro linhas de alta resolução. E, no modo texto, os blocos de texto (linhas) são formados de oito em oito, das linhas de alta resolução.

Os modos de alta e de baixa resolução podem ser bem entendidos na própria listagem da ROM. Neste subitem pretende-se mostrar como opera o modo texto, que é o encarregado de viabilizar a comunicação do usuário com o computador.

A principal rotina de saída de vídeo é COUT, situada em \$FDED.

Conforme o tipo de dados - caracteres alfanuméricos, caracteres gráficos ou "tokens" - haverá uma saída de dados distinta. O software se incumbirá de designar a rotina apropriada para cada tipo. Tal procedimento é descrito a seguir:

5.3.1. Inicialmente o flag GRAFFLG irá sinalizar ou não a presença de um caractere gráfico, operando-se, então, respectivamente, um desvio para a rotina de gráficos ou para o flag VCTB. VCTB vai indicar se anteriormente registrou-se ou não CONTROL-B.

5.3.2. Não havendo CONTROL-B, e sendo detectado um "token", este receberá o tratamento da rotina ISCBAS (\$F769) que o transaudará em caracteres alfanuméricos.

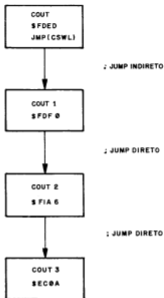
5.3.3. Havendo um CONTROL-B, será realizado um exame para detectar a presença de #graf (#\$F2).

Sendo positiva a presença de #graf, a rotina setará GRAFFLG para que o próximo caractere seja tratado como um caractere gráfico.

Este processo se interromperá somente quando for detectado outro CONTROL-B, "desligando" o modo gráfico.

5.3.4. Estando os caracteres separados em todas as suas formas, a rotina IDFFCOD (\$FBFD) desempenhará a ação das teclas de funções de vídeo. E, ainda, a rotina NXTPOS1 (\$FBF0) buscará um bloco de oito bytes para cada caractere, em GERCHR (\$F200) ou em GERGRF, conforme os caracteres sejam alfanuméricos ou gráficos, posicionando-os na página de vídeo referente à posição onde cada caractere é desejado.

Os fluxogramas abaixo permitem uma visualização das rotinas descritas neste subitem:



Seqüência de prioridade das rotinas de saída de dados



#### 6. OS UTILITÁRIOS

As rotinas utilitárias tratam comandos como CLEAR, DELETE, NEW, RUN, LIST etc.

São comandos executados diretamente. Após sua execução, o BASIC volta a solicitar a introdução de dados pelo teclado.

#### IV - PONTOS DE ENTRADA DE ROTINAS

Destaca-se aqui os principais pontos de entrada:

CHRGET	B1	carrega A de TXTPTR, incrementando-o.
CHRGOT	B7	carrega A de TXTPTR.
LINGET	CD31	converte um número de linha de TXTPTR a LINUM.
GTBYT	DA12	avalia a fórmula em TXTPTR, deixando o resultado em FAC e passando a CONINT.
PLOTFNS	E540	obtem 2 coordenadas baixa resolução de TXTPTR.
MFNS	EA3A	obtem 2 coordenadas alta resolução de TXTPTR.
FMULT	DC99	move o número de Y,A a ARG e passa a FMULTT.
FMULTT	DC9C	multiplica FAC por ARG em FAC.
FDIV	DD80	move o número de Y,A a ARG e passa a FDIVT.
FDIVT	DD83	divide ARG por FAC, resultado em FAC.
FADD	DADB	move o número de Y,A a ARG e passa a FADDT.
FADDT	DADB	soma FAC com ARG em FAC.
FSUB	DAC1	move o número de Y,A a ARG e passa a FSUBT.
FSUBT	DAC4	subtrai FAC de ARG, resultado em FAC.
FPWRT	E1B1	eleva ARG à potência FAC.
SNG	DEAA	chama SIGN e "flutua" o resultado em FAC.
ABS	DEC9	valor absoluto de FAC.
INT	DF3D	maior valor inteiro de FAC. Usa GINT e "flutua" o resultado.
SQR	E1A7	raiz quadrada de FAC.
LOG	DC5B	logaritmo base e de FAC.
EXP	E223	eleva e à potência FAC.
RND	E2CB	pseudonôm

TAN	E354	tangente de FAC.
ATN	E38B	arco tangente de FAC.
MOVFM	DE13	move de Y,A a FAC.
MOV2F	DE38	compacta FAC e move-o a TEMP2.
MOV1F	DE3B	compacta FAC e move-o a TEMP1.
MOVML	DE3D	compacta FAC e move-o à área apontada por X.
MOVHF	DE45	compacta FAC e move-o a Y,X.
MOVFA	DE6D	move ARG a FAC.
MOVAF	DE7D	move FAC a ARG.
CONUPK	DCFD	carrega ARG de Y,A.
SIGN	DE9C	coloca em A 1, 0 ou FF, conforme FAC seja positivo, zero ou negativo.
FOUT	E04E	monta em FBUFFER um string equivalente ao conteúdo de FAC. Destroi FAC. Imprime o string com STROUT.
FCOMP	DECC	compara FAC com o número compactado em Y,A. Coloca em A 1, 0 ou FF, conforme Y,A seja (<, = ou >) que FAC.
NEGOP	E1EA	faz $FAC = - FAC$ .
FADDH	DABA	faz $FAC = FAC + 1/2$ .
DIV10	DD6F	divide FAC por 10.
MUL10	DD53	multiplica FAC por 10.
SNGFLT	D61B	"flutua" o inteiro em Y.
GIVAYF	D60C	"flutua" o inteiro com sinal em A,Y.
FLOAT	DEAD	"flutua" o inteiro com sinal em A.
CONINT	DA15	converte FAC em número de um só byte em X e FACLO.
AYINT	D426	se FAC menor que 32767 e maior que - 32767, chama GINT.
GINT	DF0C	função maior inteiro.
GETADR	DA6C	converte o número de FAC em número binário de 2 bytes em LINNUM.

GETNUM	FFAA	obtém um número de 2 bytes em LINNUM de TXTPTR, passa uma vírgula e obtém um número de 1 byte em X.
COMBYTE	DA66	passa uma vírgula e obtém 1 byte em X.
FRMEVL	D8A3	avalia em FAC a fórmula de TXTPTR.
FRMNUM	D8BF	avalia em FAC a fórmula de TXTPTR. Verifica se é um número.
FIN	DF64	entrada de número ponto flutuante em FAC.
CAT	D8B1	concatena 2 strings.
STRINI	D6EF	obtém espaço para um string e monta seu descritor.
COPY	CDDC	libera o string apontado por Y,A e move-o à memória apontada por FORPNT.
MOVINS	DBEE	move o string cujo descritor é apontado por STRNG1 à memória apontada por FRESPA.
MOVSTR	DBFC	move o string apontado por Y,A à memória apontada por FRESPA.
STRLIT	D701	põe uma aspa em ENDCHR e CHARAC.
STRLT2	D707	monta um descritor para o literal de Y,A.
GETSPA	D76C	obtém espaço para string. Pode forçar uma recuperação de espaço livre.
FRETMP	D91E	libera um string temporário.
FRETHS	D94F	libera um descritor de string.
INLIN	C81B	recebe uma linha de texto.
INCHR	C86B	obtém um caractere do teclado.
STROUT	CE5F	mostra no vídeo o string de Y,A.
STRPRT	CE62	mostra o string com o descritor dado.
OUTDO	CEB1	mostra o caractere de A.
CRDO	CE20	realiza um CR (carriage return).
INPRT	E033	mostra "IN" e o número de linha de CURLIN.
LINPRT	E03E	mostra o número de 2 bytes de X,A.
PRNTFAC	E04B	mostra o valor de FAC. Destroi FAC.



PTRGET	D303	obtem um nome de variável e localiza-o na memória. TXTPTR aponta ao primeiro caractere do nome. Calcula o endereço da variável em VARPNT. Se não encontrar a variável, ela é criada.
GETARYPT	EB59	o mesmo que PTRGET para array. Não cria variável se não a encontrar.
FNDLIN	C931	procura no programa a linha de LINNUM. Coloca seu endereço em LOWTR.
DATA	CCBA	desloca TXTPTR para o fim da linha corrente.
DATAN	CCC8	calcula em Y o deslocamento de TXTPTR ao próximo fim de linha "r" ou "0".
REMN	CCC8	calcula em Y o deslocamento de TXTPTR ao próximo fim de linha "0".
ADDON	CCBD	soma Y a TXTPTR.
SCRATCH	C962	efetua o comando NEW.
CLEARC	C983	efetua o comando CLEAR.
STKINI	C99A	limpa a pilha.
RESTORE	CB69	coloca em DATPTR, pointer de DATA, o endereço de início do programa.
STXPT	C9AE	coloca em TXTPTR o endereço de início do programa.
BLTU	C6B2	move um bloco de bytes para a frente na memória, para criar espaço.
REASON	C6D2	verifica se há espaço na memória.
GARBAG	D79E	recupera espaço livre, eliminando strings não utilizados.
CONT	CB8B	move OLDTXT e OLDLIN a TXTPTR e CURLIN.
NEWSTT	CAEC	executa uma nova linha do programa.
RUN	CC37	executa o programa da memória.
GOTO	CC63	atualiza TXTPTR a partir de LINGET e FNDLIN.
LET	CD6B	avaliação da atribuição em TXTPTR. Usa CHRGET para obter o endereço da variável, avalia a fórmula e guarda o resultado.
HRR2	E748	inicializa e limpa a página 2 de alta resolução.

HGR	E74F	inicializa e limpa a página 1 de alta resolução.
BKGND	E775	enche a tela alta resolução com a última cor usada.
HPOSN	E790	posiciona o cursor alta resolução. A página é determinada por HPAG.
HPL0T	EA7E	"plots" um ponto na posição do cursor.
HLIN	E586	traça uma linha do último ponto ao ponto dado pelos registradores X,A (horizontal) e Y (vertical).
HFIN	E946	converte a posição do cursor alta resolução a coordenadas x,y. Usada após SHAPE para se localizar.
DRAW	EAE9	desenha a forma apontada por Y,X usando a cor HCOLOR. A = fator de rotação.
XDRAM	EAEF	desenha a forma apontada por Y,X invertendo a cor dos pontos existentes. A = fator de rotação.
SETCOL	F864	faz a cor de alta resolução igual a X.
SHLOAD	EAF5	carrega a tabela de formas de fita em MENSIZ e prepara o pointer em \$EB.
SAVE	CB05	grava o programa da memória na fita.
LOAD	CBEE	carrega programa a partir de fita.
ERROR	C701	imprime mensagem de erro ou desvia a rotina de erro caso ONERR ativo.
HANDLERR	E64A	prepara variáveis em função do erro cujo código está em X.
RESUME	E679	volta após erro, restaura variáveis.
ISCNTC	CB7B	verifica se foi teclado CTRL-C. Se foi, chama BREAK.
CHKNUM	D092	verifica se FAC é numérico.
CHKSTR	D094	verifica se FAC é string.
CHKVAL	D095	verifica se o resultado da última operação em FAC foi string ou número, para conferir compatibilidade de tipos de variáveis.

ISLETC	D397	verifica se A contém uma letra ASCII. C é ligado se A contém uma letra.
PARCHK	D1DA	verifica a existência de "((", avalia a fórmula e verifica a existência de "))".
CHKCLS	D1E0	verifica a existência de ")" em TXTPTR.
CHKOPN	D1E3	verifica a existência de "(" em TXTPTR.
CHKCOM	D1E6	verifica a existência de "," em TXTPTR.
SYNCHR	D1EB	verifica se o caractere em TXTPTR é igual ao caractere em A.

## V - AS TABELAS

Encontram-se na ROM as seguintes tabelas:

CMDTABL	C2D6	endereços das rotinas de execução de cada comando e função.
MATHBL	C392	endereços das rotinas de operações aritméticas, precedidos por um valor hexadecimal que corresponde a sua prioridade numa expressão.
TOKTABL	C380	tabela de nomes dos comandos para cálculo do seu "token".
MENSERRO	C54F	tabela de mensagens de erro.
GERCHR	F200	tabela de geração de caracteres de texto.
GERGRF	F400	tabela de geração de caracteres gráficos.
TABASC	F59D	códigos ASCII referentes ao teclado.
CHRTBL	FFCA	funções do monitor.
FMT1	F962	
FMT2	F9A6	
CHAR1	F9B4	tabelas referentes aos comandos do miniassembler, disassembler.
CHAR2	F9BA	
MNEML	F9C0	
MNEMR	FA00	



LISTAGEM DA ROM



```

1 *****
2 *
3 *      ROM   TX2000   *
4 *
5 *      Listagem comentada
6 *
7 *      Def
8 *
9 *      $C100 a $FFFF
10 *
11 *
12 *
13 *****
14 *
15 *      Mini-assembler
16 *
17 *      Comando SOUND,POL
18 *
19 *      Entrada parsing
20 *
21 *      Rotinas de endereçamento
22 *
23 *      FOR-NEXT loops,etc
24 *
25 *****
26
27 *Localizacao de pag. zero
28
29 D0ARRH  EQU  $00      ;Preparado por cold start
30 LOCD   EQU  $00      ;mas nao usado por outros
31 LOC1   EQU  $01      ;LOCKS,localices de memoria
32 LOC2   EQU  $02      ;usadas pelo MONITOR
33 LOC3   EQU  $03
34 GOSTROUT EQU  $03
35 LOCA   EQU  $04
36 R3L   EQU  $06      ;R3L-W,pseudo registradores
37 R3H   EQU  $07      ;de 16 bits
38 R4L   EQU  $08
39 R4H   EQU  $09
40 USR   EQU  $0A
41 CHARAC EQU  $0B      ;Usados para strings
42 ENDCR  EQU  $0E
43 MJDOSH EQU  $0F      ;Usado em rotinas de
44 PTRR   EQU  $0F      ;manipulacao de arrays
45 DTRFLG EQU  $10
46 VALTYP EQU  $11      ;$FF para string,0 se numerico
47 INTFLG EQU  $12      ;idem para variavel inteira
48 DATAFLG EQU  $13      ;Usado no PARSE
49 GDRFLG EQU  DATAFLG  ;Usado em GDRASC
50 SUBFLG EQU  $14      ;Flag de subscritos
51 INPUTFLG EQU  $15      ;$40GET, $SPREAD
52 CPTRASC EQU  $16      ;Recebe CPRTYP em FRNEVL
53 SIGMFLG EQU  $16      ;Indica sinal para TAN
54 SHAFEL EQU  $1A      ;Ponteiros usados pelas
55 SHAFEH EQU  $1B      ;rotinas de HI-RES
56 HCOLORI EQU  $1C      ;Usado pela rotina HI-RES
57 COUNTH EQU  $1D      ;Usado pela rotina HI-RES
58 WMDLFT EQU  $20      ;Ponteiros de formatacao
59 WMDNTH EQU  $21      ;de quadro de video

```



60	MDTOP	EMU	\$22	
61	MDBTM	EMU	\$23	
62	CH	EMU	\$24	;Coordenadas de cursor
63	CV	EMU	\$25	
64	DRAGL	EMU	\$26	;Ponteiros de linha
65	DRASH	EMU	\$27	;da rotina PLOT
66	BASL	EMU	\$28	;Ponteiros de linha
67	BASH	EMU	\$29	de texto
68	BASZL	EMU	\$2A	;Usados durante scrolling
69	BASZH	EMU	\$2B	
70	LMEM	EMU	\$2C	;Usado pelo disassembler
71	HZ	EMU	\$2C	;Usado por DRAW e HLINE
72	VZ	EMU	\$2D	;Usado por DRAW e VLINE
73	RMEM	EMU	\$2D	;Usado pelo disassembler
74	FORMAT	EMU	\$2E	;Mini-assembler disassembler
75	CHSUM	EMU	\$2E	;Ponteiros das rotinas
76	LASTIN	EMU	\$2F	;de cassette
77	LENGTH	EMU	\$2F	;Usado pelo disassembler
78	MMASK	EMU	\$30	;Mascara da rotina HI-RES
79	COLOR	EMU	\$30	;Contem a cor a ser plotada
80	MODE	EMU	\$31	;Usado pelo monitor
81	INFLG	EMU	\$32	;Formato normal ou inverso
82	PROMPT	EMU	\$33	;Contem o caracter de pronto
83	YSAV	EMU	\$34	;Salva conteudo de Y
84	ENDMASK	EMU	\$35	
85	YSAV1	EMU	\$35	
86	CSWL	EMU	\$36	;Ponteiro da rotina de OUT
87	ISWL	EMU	\$38	;Ponteiro da rotina de IN
88	PCL	EMU	\$3A	;Programa counter REG.
89	PCH	EMU	\$3B	
90	A1L	EMU	\$3C	;A1L-R, registradores de
91	A1H	EMU	\$3D	uso geral
92	A2L	EMU	\$3E	
93	A2H	EMU	\$3F	
94	A3L	EMU	\$40	
95	A3H	EMU	\$41	
96	A4L	EMU	\$42	
97	A4H	EMU	\$43	
98	A5L	EMU	\$44	
99	ACC	EMU	\$45	;Salva acumulador
100	A6L	EMU	\$46	
101	PAVL	EMU	\$47	;Ponteiro do endereco
102	PDVH	EMU	\$48	da pagina de video
103	VCTR	EMU	\$49	;FLAG de caracter grafico
104	RNDL	EMU	\$4E	
105	RNDH	EMU	\$4F	
106	LTRM	EMU	\$50	;Uso geral, numero de linha
107	TEMP1	EMU	\$52	;Ponteiros temporario
108	LASTPT	EMU	\$53	
109	TEMP2	EMU	\$55	
110	INDEX	EMU	\$5E	
111	DEST	EMU	\$60	;Parametros de operacoes
112	RESULT	EMU	\$62	matematicas
113	XTTAB	EMU	\$67	;De \$67 ate \$80, ver
114	BASLOC	EMU	\$67	II-0 USO DA RAM
115	VRTAB	EMU	\$69	
116	VRTAB	EMU	\$68	

117	STREND	EBU	\$60	
118	FRETOP	EBU	\$6F	
119	FRESPEC	EBU	\$71	
120	MENSIZ	EBU	\$73	
121	CURLIN	EBU	\$75	
122	OLDLIN	EBU	\$77	
123	OLDTEXT	EBU	\$79	
124	DATLIN	EBU	\$7B	
125	DATPTR	EBU	\$7D	
126	IMPTR	EBU	\$7F	
127	VARNAM	EBU	\$81	;\$\$\$;It--_real++;junct-->
128	VARPNT	EBU	\$83	;Ponteiro de variavel
129	FORPNT	EBU	\$85	;Uso geral
130	TOPSV	EBU	\$87	;Usado por INPUT
131	LASTOP	EBU	\$87	;Ind.de trab.usado em FRMVEL
132	CPRITP	EBU	\$89	;Ind.)=f em FRMVEL
133	FRCNAM	EBU	\$8A	
134	YENP3	EBU	\$8A	;REG. de ponto flutuante
135	DISCTR	EBU	\$8C	
136	OSCLEM	EBU	\$8F	;Usado em GARBAG
137	_JPADRS	EBU	\$90	;Ponteiro de instrucao JMP
138	ALGNLEN	EBU	\$91	;Usado em GARBAG
139	EXTRASV	EBU	\$92	;Mais precisao para F.P.
140	TEMP1	EBU	\$93	;Areas de trabalho para FAC
141	HIGHDS	EBU	\$94	;Usado em GARBAG
142	ARYPNT	EBU	\$94	
143	HIGHTR	EBU	\$95	;Usado para transferir bloco
144	TEMP2	EBU	\$98	;REG. de ponto flutuante
145	INDX	EBU	\$99	;Usado pelas rotinas de arrays
146	TIMEXP	EBU	\$99	;Usado em FIN(EVAL)
147	EXPON	EBU	\$9A	;Usado em FIN(EVAL)
148	DPFLG	EBU	\$9B	;Indica ponto decimal em FIN
149	LONTR	EBU	\$9B	;Uso geral
150	EXPSGN	EBU	\$9C	;Indica sinal do expoente em FIN
151	FAC	EBU	\$9D	;Area ponto flutuante principal
152	OSCTMP	EBU	\$9D	
153	VPNT	EBU	\$A0	;Pointer a variavel temporaria
154	FACSGN	EBU	FAC+5	;Sinal descompactado
155	SERLEN	EBU	\$A3	;Tamanho da serie - 1
156	FFGEN	EBU	\$A4	;Uso geral em FP
157	ARS	EBU	\$A5	;Acumulador pto flut.secund.
158	ARSGN	EBU	ARS+5	
159	SIGNCPR	EBU	\$A8	;Ind.sinal de oper.em rot. F.P.
160	STRM01	EBU	\$A8	;Utilizado para strings
161	EXTRAFAC	EBU	\$AC	;Precisao F.P.
162	SERPNT	EBU	\$AD	;Pont.aos dados da serie em f.p.
163	STRM02	EBU	\$AD	;Utilizado para strings
164	PROGNO	EBU	\$AF	;Ponteiro de fim de programa
165	BASELOC1	EBU	\$B0	;Endereco do interp. BASIC
166	CHRGET	EBU	\$B1	;Busca de caracter
167	CHG01	EBU	\$B7	
168	TXTPTR	EBU	\$B8	;Ponta proximo caractoken
169	RNDSEED	EBU	\$C9	;Numero randomico F.P
170	DDL	EBU	\$D0	;500-50F, ponteiros de
171	DDH	EBU	\$D1	tratamento nas rotinas de erro
172	DI	EBU	\$D2	
173	DDMT	EBU	\$D3	

174	EL	EW	S04	
175	DH	EW	S05	
176	LOCK	EW	S06	;Trava acesso pelo usuario,
177	ERRFLG	EW	S08	se negativo
178	ERRLIN	EW	S0A	
179	ERRPAG	EW	S0C	
180	ERRNUM	EW	S0E	
181	ERRSTX	EW	S0F	
182	XOL	EW	SE0	;Coordenada X em HI-RES
183	XOH	EW	SE1	
184	YD	EW	SE2	;Coordenada Y em HI-RES
185	DELH	EW	SE3	;Ponteiro de deletacao de linha
186	NDLORZ	EW	SE4	;Usado em HI-RES
187	HDX	EW	SE5	
188	HPAG	EW	SE6	;#linha a pagina HI-RES
189	SCALEZ	EW	SE7	;Fator de escala em HI-RES
190	SHAPEMT	EW	SEB	
191	CALCOUNT	EW	SEA	
192	YDMTC	EW	SEF	
193	FIRST	EW	SF0	;Primeira coordenada de PLOT
194	SPEEDZ	EW	SF1	;Velocidade de Output
195	TRCFLO	EW	SF2	
196	DRMASK	EW	SF3	;Mascara para controle de OUT
197	TXTPSU	EW	SF4	
198	CURLSV	EW	SF6	
199	RENSTK	EW	SF8	
200	ROTZ	EW	SF9	
201	COOL	EW	SF8	;#no geral do monitor
202	COOH	EW	SFC	
203	RTMP1	EW	SFD	;RES. temporario do monitor
204				
205	STACK	EW	S100	;Pagina 1
206				
207	IN	EW	S0200	;Pagina 2, buffer de teclado
208				
209	BRKVL	EW	S03F0	;Pagina 3, vetor do BREAK
210	SOFTVL	EW	S03F2	;Vetor de entrada do BASIC
211	SOFTVH	EW	S03F3	
212	PRREDUP	EW	S03F4	
213	AMPFR	EW	S03F5	;Comparacao do RESET
214	TK2000	EW	S03F8	
215	USRADR	EW	S03F8	
216	NMI	EW	S03F8	
217	IRNLDC	EW	S03FE	
218				
219	PGA	EW	S0400	;Pagina4
220	TSUTIL	EW	S0435	;Teste de utilitarios
221	FLERAM	EW	S0440	;Teste de memoria
222	TEST41	EW	S0441	
223	TEST42	EW	S0442	
224	TEST50	EW	S0450	
225	TEST51	EW	S0451	
226	DRAFFLE	EW	S0452	;Indica se graficos
227	TEST55	EW	S0453	
228	BUF	EW	S0460	
229	PGR1	EW	S0468	
230	PGR2	EW	S046A	

228	TEST70	EMU	\$0470	
232	TEST71	EMU	\$0471	
233	TEST72	EMU	\$0472	
234	TEST73	EMU	\$0473	
235	TEST74	EMU	\$0474	
236	FINLIN	EMU	\$0482	;fim de linha
237	TEST86	EMU	\$0486	
238	TEST88	EMU	\$0488	
239	XSAV	EMU	\$048E	;Salva REG. X
240				
241	IMG	EMU	\$0600	;Pagina 6, buffer auxiliar
242				
243	AREG	EMU	\$07F0	;Pagina 7
244	XREG	EMU	\$07F1	
245	YREG	EMU	\$07F2	
246	STATUS	EMU	\$07F3	
247	SPNT	EMU	\$07F4	
248	R14K	EMU	\$07FF	
249				
250	PRG1	EMU	\$0800	;Pagina 8
251	PRG3	EMU	\$0803	
252	PRG6	EMU	\$0806	
253	PRG9	EMU	\$0809	
254	PRG0C	EMU	\$080C	
255	PRG0F	EMU	\$080F	
256	PRG12	EMU	\$0812	
257	PRG15	EMU	\$0815	
258	PRG18	EMU	\$0818	
259				
260	ESPR14K	EMU	\$47FF	
261	R48K	EMU	\$67FF	
262	CART0	EMU	\$9000	
263	CART3	EMU	\$9003	
264	CART6	EMU	\$9006	
265	CART9	EMU	\$9009	
266	HDRFL01	EMU	\$9854	
267	HDRFL02	EMU	\$9855	
268	TSCART	EMU	\$9FF0	;Testa cartucho
269	INITCART	EMU	\$9FF1	;Inicia cartucho
270	KBOUT	EMU	\$C000	;Saida de dados pelo teclado
271	KBTPEN	EMU	\$C010	;Entrada de dados pelo teclado
272	TAPEOUT	EMU	\$C020	;Saida de dados para gravador
273	SPFR	EMU	\$C030	;Saida de som
274	CCOL08	EMU	\$C050	;Imagem colorida
275	10PEB	EMU	\$C051	;Imagem preto e branco
276	NOTAOFF	EMU	\$C052	;Comando NOTUR
277	01SPAG1	EMU	\$C054	;Primeira pagina de video
278	01SPAG2	EMU	\$C055	;Segunda pagina de video
279	PR1STEN	EMU	\$C059	;Printer strobe high
280	BTCKT0	EMU	\$C05E	;Bit control em nivel 0
281	BTCKT1	EMU	\$C05F	;Bit control em nivel 1
282	POLO	EMU	\$C064	;Sinal de Padie
283	INITPOL	EMU	\$C070	;Inicializacao
284	PRF1	EMU	\$C080	
285	PRF2	EMU	\$C088	
286	DSK	EMU	\$C100	
287	IMPR	EMU	\$C200	

```

288 ENDWIMP EQU $007A ;Endereco nao importante
289
290 *Tk2000soft tokens:
291
292 for EQU $81
293 data EQU $83
294 bell EQU $87
295 pop EQU $A1
296 goto EQU $A8
297 gosub EQU $B0
298 ree EQU $B2
299 print EQU $B4
300 tab EQU $C0
301 to EQU $C1
302 fn EQU $C2
303 spc EQU $C3
304 then EQU $C4
305 at EQU $C5
306 not EQU $C6
307 step EQU $C7
308 plus EQU $C8
309 minus EQU $C9
310 equal EQU $D0
311 sgn EQU $D2
312 scrn EQU $D7
313 leftstr EQU $D8
314 graf EQU $F2
315
316 PUT PARTEIA,01
317 ORG $C100
318
319 *MINI-ASSEMBLER
320
C100: E9 81 15 RELAT SRC MSR1 ;Calcula deslocamento relativo
C102: 4A 16 LSR
C103: 00 14 17 BNE RELERR
C105: A4 3F 18 LDY A2H ;Endereco destino
C107: A6 3E 19 LOX A2L
C109: 00 01 110 BNE RELAT1
C10A: 00 111 DEY
C10C: CA 112 RELAT1 DEX
C10D: 8A 113 TXA
C10E: 18 114 CLC
C10F: E5 3A 115 SRC PCL
C111: 85 3E 116 STA A2L ;Armazena o deslocamento
C113: 10 01 117 BPL RELAT2
C115: C8 118 IMY
C116: 98 119 RELAT2 TYA
C117: E5 38 120 SRC PCH
C119: 00 48 121 RELERR BNE INDEERR1 ;Erro,excede +129,-126 BYTES
C11B: A4 2F 122 TWDNST LDY LENGTH ;0=1,1=2,2=3 (BYTES)
C11D: 89 30 00 123 BUSBYT LDH: A2H,Y
C120: 91 3A 124 FORWARDT STA (PCL),Y ;Armaz.no end.indic.por PC+LENGTH
C122: 88 125 DEY
C123: 10 F8 126 BPL BUSBYT
C125: 20 23 FC 127 JSR UP ;Sube cursor de 2 posicoes
C126: 20 23 FC 128 JSR UP

```

C120:	20 00 FB 129	JSR	INSTRDP	;Disassembla instrucao no video
C12E:	20 53 FY 130	JSR	PCADJ	;Ajasta novo valor do PC
C131:	04 38 131	STY	PCB	
C133:	05 3A 132	STA	PCB	
C135:	4C 95 C1 133	JMP	CONTASS	;Continua assemblando
C13B:	20 00 EE 134	WPCON	JSR	TOGUB1 ;Verifica comandos (MA,MP,RA,WA)
C13B:	A4 34 135	LDT	YSAV	;Busca posicao do buffer IN
C13D:	20 AA FF 136	ONTEMO	JSR	GETNUM ;Obtem endereco hexadecimal
C140:	04 34 137	STY	YSAV	;Salva posicao do buffer IN
C142:	A0 17 138	LDT	MS17	;Ultima posicao de CRTBL
C144:	08 139	PROCCOM	DEY	
C145:	30 48 140	BRI	ASS	;Mao e zenhas, reinicia em ASS
C147:	09 CA FF 141	CHP	CRTBL,Y	;E alguma comando de CRTBL ?
C14A:	00 FB 142	BNE	PROCCOM	;Mao, proximo comando ate o fim
C14C:	C0 15 143	CPT	MS15	;E RETURN ?
C14E:	00 E8 144	BNE	WPCON	
C150:	A5 31 145	LDA	MODE	
C152:	A0 00 146	LDT	MS00	
C154:	C6 34 147	DEC	YSAV	;Volta posicao no buffer IN
C156:	20 00 FE 148	JSR	RL1	;Subst. ou impr. conteudo de mem.
C159:	4C 95 C1 149	JMP	CONTASS	;Busca nova linha p/ assemblar
C15E:	A5 30 150	PROCESS	LDA	ALH ;Assembla OPC00
C15E:	20 91 FB 151	JSR	INSD03	
C161:	AA 152	TAX		
C162:	00 00 FA 153	LDA	MENR,X	
C165:	C5 42 154	CHP	AAL	
C167:	00 13 155	BNE	PROCESS1	
C169:	00 CD FF 156	LDA	MENR,X	
C16C:	C5 43 157	CHP	AAN	
C1AE:	00 DC 158	BNE	PROCESS1	
C170:	A5 44 159	LDA	ASL	
C172:	A4 2E 160	LDT	FORMAT	
C174:	C0 90 161	CPT	MS90	;E enderecamento relativo ?
C176:	F0 00 162	BEQ	RELAT	;Sim, RELAT
C17B:	C5 2E 163	CHP	FORMAT	
C17A:	F0 9F 164	BEQ	TMDMST	
C17C:	C4 30 165	PROCESS1	DEC	ALH
C17E:	00 DC 166	BNE	PROCESS	
C180:	E6 44 167	DEC	ASL	
C182:	C6 35 168	DEC	ENDRNGR	
C184:	F0 D6 169	BEQ	PROCESS	
C186:	A4 34 170	INDERR1	LDT	YSAV ;Recupera posicao do buffer
C188:	90 171	INDERR	TIA	
C189:	AA 172	TAX		
C18A:	20 4A FF 173	JSR	PBBL2	;Imprime X espacos
C18D:	A5 DC 174	LDA	***	
C18F:	20 ED FD 175	JSR	CGUT	;Imprime ***
C192:	20 3A FF 176	ASS	JSR	BELL ;Pto de entrada do miniassembler
C195:	A9 A1 177	CONTASS	LDA	B"1"
C197:	05 33 178	STA	PROMPT	
C199:	20 47 FD 179	JSR	GETLAZ	;Obtem linha de entrada de dados
C19C:	20 E9 EE 180	JSR	ZN00	;Zera MODE
C19F:	A0 00 02 181	LDA	IN	;Le inicio do buffer IN
C1A2:	C9 A0 182	CHP	B" "	;E linha para disassemblar ?
C1A4:	F0 13 183	BEQ	ASSEM	;Sim
C1A6:	C8 184	INY		
C1A7:	C9 A4 185	CHP	B"5"	

C1A9: FD 92	186	BER	ONTEND	;Sim
C1AB: 88	187	DET		
C1AC: 20 AA FF	188	JSR	GETNUM	;Mtem endereço hexa em A2
C1AF: C9 93	189	CMF	#503	;Compara com "3"
C1B1: 00 05	190	INVALID	BNE INDEER	;Indica posicao do erro
C1B3: 0A	191	TXA		
C1B4: FD 02	192	BEQ	INDEER	
C1B6: 20 78 FE	193	JSR	A1FCPL	;Transf.o endereço de A1 para PC
C1B9: A9 03	194	ASSEMB	LDA #503	;avalia OPC00
C1BB: 85 30	195		STA A1H	
C1BD: 20 34 C2	196	ASSEMB1	JSR BUSESP	;Ignora espacos
C1CD: 0A	197	ASL		
C1C1: E9 BE	198	SBC	#58E	
C1C3: C9 C2	199	CMF	#5C2	;OPC00 valido ?
C1C5: 90 C1	1100	BCC	INDEER	;Nao erro
C1C7: 0A	1101	ASL		
C1C9: 0A	1102	ASL		
C1C9: A2 04	1103	LDX	#5D4	
C1CB: 0A	1104	ASSEMB2	ASL	;Mecanico em A4
C1CC: 26 42	1105	ROL	AAL	
C1CE: 26 43	1106	ROL	AHH	
C1D0: CA	1107	DEX		
C1D1: 10 FB	1108	RPL	ASSEMB2	
C1D3: C6 30	1109	DEC	A1H	
C1D5: FD F4	1110	BEQ	ASSEMB2	
C1D7: 10 F4	1111	RPL	ASSEMB1	
C1D9: A2 05	1112	LDX	#5D5	;Ultima posicao de CHAR1,CHAR2
C1DB: 20 34 C2	1113	COMTCH	JSR BUSESP	;Ignora espacos no buffer
C1DE: 04 04	1114	TIPEND	STI YSAV	;Salva Y
C1E0: 00 84 F9	1115	CMF	CHAR1,X	;Verifica forma de enderecamento
C1E3: 00 13	1116	BNE	TIPEND1	
C1E5: 20 34 C2	1117	JSR	BUSESP	;Ignora espacos no buffer
C1E8: 00 8A F9	1118	CMF	CHAR2,X	
C1EB: FD 00	1119	BER	INEDIAT?	;Se CHAR1 e CHAR2
C1ED: 80 8A F9	1120	LDA	CHAR2,X	
C1F0: FD 07	1121	BER	TIPEND2	
C1F2: C9 A4	1122	CMF	#5"X"	;É "X" ?
C1F4: FD 03	1123	BER	TIPEND2	
C1F6: A4 34	1124	LDI	YSAV	
C1F8: 10	1125	TIPEND1	CLC	
C1F9: 88	1126	TIPEND2	DEY	
C1FA: 26 44	1127	INEDIAT?	ROL ASL	
C1FC: ED 03	1128	CPY	#503	;CHAR1=8 , CHAR2=5
C1FE: 00 00	1129	BNE	INEDIAT	;Nao e imediato?
C200: 20 AA FF	1130	JSR	GETNUM	
C203: A5 3F	1131	LDA	A2H	
C205: FD 01	1132	BER	ZPGH	;Se pagina zero
C207: E8	1133	TRX		;Se absoluto
C208: 86 35	1134	ZPGH	STX ENDMASK	
C20A: A2 03	1135	LDX	#503	
C20C: 88	1136	DEY		
C20D: 86 30	1137	INEDIAT	STX A1H	
C20F: CA	1138	DEX		
C210: 10 C9	1139	RPL	COMTCH	
C212: A5 44	1140	LDA	ASL	
C214: 0A	1141	ASL		
C215: 0A	1142	ASL		

C254:	85 35	1143	ORA	ENDMAGK	
C258:	C7 28	1144	CMP	#520	
C26A:	80 06	1145	BCS	FINFEND	;Se >=20
C26C:	A6 35	1146	LDX	ENDMAGK	
C26E:	F0 02	1147	BEQ	FINFEND	;Se =0
C268:	09 80	1148	ORA	#580	;Se > 0 e < 20, ORAB60
C222:	85 44	1149	FINFEND	STA	ACL
C224:	84 34	1150	STY	YSAV	;Salva posicao do buffer IN
C226:	89 00 02	1151	LDA	IN,Y	
C229:	C9 88	1152	CMP	#;"	;E ";" ?
C228:	F0 04	1153	BEQ	JPROCESS	;Sim
C228:	C7 80	1154	CMP	#80	;E RETURN ?
C22F:	00 80	1155	BNE	INVALID	;Nao, linha invalida
C231:	4C 5C C1	1156	JPROCESS	JMP	PROCESS
C234:	89 00 02	1157	BUSESP	LDA	IN,T
C237:	C8	1158	INY		;Assebla linha
C238:	C9 A0	1159	CMP	#"	;E "espaco" ?
C23A:	F0 FB	1160	BEQ	BUSESP	;Sim, entao novo caracter
		1161			
C23C:	60	1162	RTS		;Nao, retorne
C238:	FF FF FF	1163	HEX	FFFFFF	
		1164			
C240:	20 12 DA	1165	SOUND	JSR	GETBYT
C243:	86 40	1166	STX	ACL	;Obtem a nota
C245:	A9 2C	1167	LDA	#,"	;Guarda em ACL
C247:	20 EB D1	1168	JSR	SYNCRB	;Compara com ","
C24A:	20 12 DA	1169	JSR	GETBYT	;Obtem a duracao da nota
C248:	86 41	1170	STX	ACH	;Guarda em ACH
C24F:	40 30 C0	1171	TONAL	LDA	SPKR
C252:	88	1172	DURACAO	BEY	;Prod. um "click" no alto-falante
C253:	00 05	1173	BNE	FREQ	
C255:	CE 41 00	1174	DEC:	ACH	
C258:	F0 09	1175	BEQ	FIMNOTA	;Acabou a nota ?
C25A:	CA	1176	FREQ	DEX	
C25B:	00 F5	1177	BNE	DURACAO	
C250:	AE 40 00	1178	LDR:	ACL	
C260:	4C 4F C2	1179	JMP	TONAL	
C263:	20 87 00	1180	FIMNOTA	JSR	CHRGOT
C266:	C9 C1	1181	CMP	#0	;Tem mais nota ?
C268:	F0 01	1182	BEQ	PROXNOT	
C26A:	60	1183	RTS		
C26F:	20 81 00	1184	PROXNOT	JSR	CHRGOT
C26E:	4C 40 C2	1185	JMP	SOUND	
		1186			
		1187			*Rotina reservada de paddle, para uma possivel interface
		1188			na ser desenvolvida
		1189			*PDL0=5C06A, PDL1=5C065, PDL2=5C066, PDL3=5C067
		1190			
C271:	20 15 DA	1191	PDL	JSR	CONINT
C274:	AD 70 C0	1192	LDA	INCPDL	;Inicia paddle
C277:	A0 00	1193	LDY	#800	
C279:	EA	1194	WOP		;Cria strazo
C27A:	EA	1195	WOP		
C27B:	80 64 C0	1196	SENEG	LDA	POLS,X
C27E:	10 04	1197	BPL	FINTENP	;Negativo ate terminar o tempo
C280:	C8	1198	INY		
C281:	80 FB	1199	BNE	SENEG	



```

C283: 88      1200      DEY
C284: 4C 18 06 1201  FINTDFP .JMP $GMFLT
      1202
C287: 80      1203      DFB $80
C288: 80      1204      DFB $80
C289: 80      1205      DFB $80
C28A: 82      1206      DFB $82
C28B: C8      1207      DFB $C8
C28C: 04      1208      DFB $04
C28D: 0A      1209      DFB $0A
C28E: 20      1210      DFB $20
C28F: F5      1211      DFB $F5
C290: F8      1212      DFB $F8
C291: 20      1213      DFB $20
C292: 73      1214      DFB $73
C293: FC      1215      DFB $FC
C294: A5      1216      DFB $A5
C295: 08      1217      DFB $08
C296: 95      1218      DFB $95
C297: 90      1219      DFB $90
C298: A5      1220      DFB $A5
C299: 09      1221      DFB $09
C29A: 85      1222      DFB $85
C29B: 81      1223      DFB $81
C29C: 18      1224      DFB $18
C29D: A5      1225      DFB $A5
C29E: 84      1226      DFB $84
C29F: 0A      1227      DFB $0A
C2A0: 0A      1228      DFB $0A
C2A1: A5      1229      DFB $A5
C2A2: 81      1230      DFB $81
C2A3: 85      1231      DFB $85
C2A4: 81      1232      DFB $81
C2A5: 60      1233      DFB $60
C2A6: FF      1234      DFB $FF
C2A7: FF      1235      DFB $FF
      1236
C2A8: A5 47 1237  TESTCAR LDA P0VL ;Testa accessorios
C2A9: F0 03 1238      BEQ WOLAH
C2AC: 4C 00 08 1239      .JMP PRG1
C2AF: A0 35 04 1240  WOLAH LDA TSUTDL
C2B2: C9 02 1241      CMP #502
C2B4: 30 08 1242      BRT F1NCAR ;hao tes ?
C2B6: C9 02 1243      CMP #502
C2B8: F0 0F 1244      BEQ CMT
C2BA: C9 03 1245      CMP #503
C2BC: 00 04 1246      BNE SET4
C2BE: 80 46 00 1247  SETFLG STA# A46
C2C1: A0 1248  F1NCAR RTS
C2C2: C9 04 1249  SET4  CMP #504
C2C4: 00 03 1250      BNE CMT
C2C6: 4C 8E C2 1251      .JMP SETFLG
C2C9: 4C 04 00 1252  CMT .JMP (LOCK)
      1253
C2CC: FF FF FF 1254      HEX FFFFFFFF
C2CF: FF
      1255

```

C200:	4C	42	E4	1256	BASIC	JMP	COLIST	↳Pontos de entrada do BASIC
				1257				
C203:	4C	28	C7	1258	BASIC2	JMP	RESTART	
				1259		FUT	PARTE18_D1	
				11				
C204:	94	08	12		CMDTABL	DA	END-1	
C206:	7F	0A	13			DA	FOR-1	
C20A:	20	00	14			DA	NEXT-1	
C20C:	89	0C	15			DA	DATA-1	
C20E:	06	0E	16			DA	INPUT-1	
C2E0:	91	E6	17			DA	DEL-1	
C2E2:	F8	02	18			DA	DIR-1	
C2E4:	06	0F	19			DA	READ-1	
C2E6:	F8	E6	110			DA	GR-1	
C2E8:	02	E7	111			DA	TEXT-1	
C2EA:	FF	00	112			DA	DSK-1	
C2EC:	91	C1	113			DA	ASS-1	
C2EE:	35	E5	114			DA	CALL-1	
C2F0:	78	E5	115			DA	PLAT-1	
C2F2:	85	E5	116			DA	HLIN-1	
C2F4:	94	E5	117			DA	VLEN-1	
C2F6:	47	E7	118			DA	NRG2-1	
C2F8:	5A	E7	119			DA	HGR-1	
C2FA:	69	EA	120			DA	HCLOR-1	
C2FC:	70	EA	121			DA	HPL0T-1	
C2FE:	E8	EA	122			DA	DRAM-1	
C300:	EE	EA	123			DA	XDRAM-1	
C302:	66	E8	124			DA	HTAB-1	
C304:	57	FC	125			DA	HOME-1	
C306:	A0	EA	126			DA	R0T-1	
C308:	A6	EA	127			DA	SCALE-1	
C30A:	F4	EA	128			DA	SHL0A0-1	
C30C:	03	E5	129			DA	TRACE-1	
C30E:	05	E5	130			DA	WTRACE-1	
C310:	09	E5	131			DA	NORMAL-1	
C312:	0C	E5	132			DA	INVERSE-1	
C314:	3F	C2	133			DA	SOUND-1	
C316:	A2	E5	134			DA	COLOR-1	
C318:	8F	CC	135			DA	POP-1	
C31A:	8C	E5	136			DA	VTAB-1	
C31C:	E6	E5	137			DA	KITER-1	
C31E:	06	E6	138			DA	LOWEN-1	
C320:	28	E6	139			DA	OWNER-1	
C322:	78	E6	140			DA	RESUME-1	
C324:	28	E7	141			DA	RECALL-1	
C326:	0E	E7	142			DA	STORE-1	
C328:	C8	E5	143			DA	SPEED-1	
C32A:	6A	CD	144			DA	LET-1	
C32C:	62	CC	145			DA	OTO-1	
C32E:	36	CC	146			DA	RUN-1	
C330:	E0	CC	147			DA	IF-1	
C332:	68	CB	148			DA	RESTORE-1	
C334:	F4	03	149			DA	AMPER-1	
C336:	45	CC	150			DA	GOSUB-1	
C338:	8F	CC	151			DA	POP-1	↳Ponto de entrada de RETURN
C33A:	00	CD	152			DA	REN-1	
C33C:	92	CB	153			DA	STOP-1	

C33E: 10 C0	154	DA	ORG070-1	
C340: 99 DA	155	DA	WAIT0-1	
C342: 91 E8	156	DA	RLAND-1	
C344: C3 E8	157	DA	NSMC-1	
C346: 2C 04	158	DA	DEF-1	
C348: 94 0A	159	DA	POKE-1	
C34A: F9 C0	160	DA	PRINT-1	
C34C: 8A C8	161	DA	COMT-1	
C34E: 88 C9	162	DA	LIST-1	
C350: 80 C9	163	DA	CLEAR-1	
C352: C4 CE	164	DA	GET-1	
C354: 5F C9	165	DA	HEX-1	
C356: AA DE	166	UMFNC	DA	SEN
C358: 36 DF	167	DA	INT	
C35A: C9 0E	168	DA	ABS	
C35C: 0A 00	169	DA	USR	
C35E: F8 05	170	DA	FRE	
C360: 01 C7	171	DA	ERROR	; ponto de entrada da rotina SCRM
C362: 70 C2	172	DA	POL-1	
C364: 19 04	173	DA	POS	
C366: A7 E1	174	DA	SMB	
C368: C8 E2	175	DA	RND	
C36A: 58 0C	176	DA	LOG	
C36C: 23 E2	177	DA	EXP	
C36E: 04 E3	178	DA	COS	
C370: 08 E3	179	DA	SIN	
C372: 54 E3	180	DA	TAN	
C374: 88 E3	181	DA	ATN	
C376: 7E 0A	182	DA	PEEK	
C378: F0 09	183	DA	LEN	
C37A: DF 04	184	DA	STR	
C37C: 21 04	185	DA	VAL	
C37E: FF 09	186	DA	ASC	
C380: 60 09	187	DA	CHRSTR	
C382: 74 09	188	DA	LEFTSTR	
C384: A0 09	189	DA	RIGHTSTR	
C386: A8 09	190	DA	INTSTR	
C388: 60 FF	191	DA	NONZ-1	; (LN)
C38A: 7C EB	192	DA	NOT00-1	
C38C: F7 03	193	DA	TK2000-1	
C38E: F7 EB	194	DA	HF-1	
C390: FF EB	195	DA	NA-1	
	196			
	197			* Os 05 hex sao usados no teste de prioridade:
	198			
C392: 79	199	MATHBL	HEX	79
C393: 0A DA	1100	DA	FADDT-1	; +
C395: 79	1101	HEX	79	
C396: C3 DA	1102	DA	FSUBT-1	; -
C398: 78	1103	HEX	78	
C399: 98 DC	1104	DA	FMULTT-1	; *
C39B: 78	1105	HEX	78	
C39C: 82 00	1106	DA	FDIVT-1	; /
C39E: 70	1107	HEX	70	
C39F: 80 E1	1108	DA	FMINT-1	; *
C3A1: 50	1109	HEX	50	
C3A2: 7C 02	1110	DA	AND-1	; AND

C8A4: 46	1111	HEX 46	
C8A5: 76 02	1112	DA 06-1	; 06
C8A7: 7F	1113	MDUS	HEX 7F
C8A8: E9 E1	1114	DA NEGOP-1	;Nenos unario
C8AA: 7F	1115	UMOT	HEX 7F
C8AB: 8F 01	1116	DA E800P-1	;NOT unario
C8AD: 64	1117	PLUS	HEX 64
C8AE: 8C 02	1118	DA POSOP-1	;Usado por (<=)
	1119		
C8B0: 45 4E C4	1120	TOKTABL	DCI 'END' ;Tabela de "tokens"(palavras)
C8B3: 46 4F 02	1121		DCI 'FOR'
C8B4: 4E 45 58	1122		DCI 'NEXT'
C8B9: 04			
C8BA: 44 41 54	1123		DCI 'DATA'
C8BD: C1			
C8BE: 49 4E 50	1124		DCI 'INPUT'
C8C1: 55 04			
C8C3: 44 45 CC	1125		DCI 'DEL'
C8C4: 44 49 CD	1126		DCI 'GEM'
C8C9: 52 45 41	1127		DCI 'READ'
C8CC: C4			
C8CD: 47 02	1128		DCI 'SR'
C8CF: 54 45 58	1129		DCI 'TEXT'
C8D2: 04			
C8E3: 44 53 C8	1130		DCI 'DSK'
C8E4: 41 53 D3	1131		DCI 'ASS'
C8E9: 43 41 4C	1132		DCI 'CALL'
C8EC: CC			
C8E8: 50 4C 4F	1133		DCI 'PLOT'
C8ED: 04			
C8E1: 48 4C 49	1134		DCI 'NLM'
C8E4: CE			
C8E5: 56 4C 49	1135		DCI 'NLM'
C8EB: CE			
C8E9: 48 47 52	1136		DCI 'NR2'
C8EC: 82			
C8E3: 48 47 02	1137		DCI 'NR'
C8FD: 48 43 4F	1138		DCI 'NCOLOR'
C8F3: 4C 4F 02			
C8F4: 48 50 4C	1139		DCI 'MPLOT'
C8F9: 4F 04			
C8FB: 44 52 41	1140		DCI 'DRAW'
C8FE: 07			
C8FF: 58 44 52	1141		DCI 'DRAW'
C402: 41 07			
C404: 48 54 41	1142		DCI 'RTAB'
C407: C2			
C408: 48 4F 48	1143		DCI 'HOME'
C40B: C5			
C40C: 52 4F 54	1144		DCI 'ROT-'
C40F: 80			
C410: 53 43 41	1145		DCI 'SCALE-'
C413: 4C 45 80			
C414: 53 48 4C	1146		DCI 'SLOWD'
C419: 4F 41 C4			
C41C: 54 52 41	1147		DCI 'TRACE'
C41F: 43 C5			

C421: 4E 4F 54 1148	DCI 'NOTFACE'
C424: 52 41 43 C5	
C428: 4E 4F 52 1149	DCI 'NORMAL'
C428: 40 41 CC	
C42E: 49 4E 56 1150	DCI 'INVERSE'
C431: 45 52 53 C5	
C435: 53 4F 55 1151	DCI 'SOUND'
C438: 4E C4	
C43A: 43 4F 4C 1152	DCI 'COLOR'
C43B: 4F 02	
C43F: 50 4F 00 1153	DCI 'POP'
C442: 56 54 41 1154	DCI 'VTAB'
C445: C2	
C446: 48 49 40 1155	DCI 'NIMEN'
C449: 45 40 8A	
C44C: 4C 4F 40 1156	DCI 'LWENH'
C44F: 45 40 8A	
C452: 4F 4E 45 1157	DCI 'OWEBB'
C455: 52 02	
C457: 52 45 53 1158	DCI 'RESUME'
C45A: 55 40 C5	
C45D: 52 45 43 1159	DCI 'RECALL'
C460: 41 4C CC	
C463: 53 54 4F 1160	DCI 'STORE'
C466: 52 C5	
C468: 53 50 45 1161	DCI 'SPEED='
C46B: 45 44 80	
C46E: 4C 45 04 1162	DCI 'LET'
C471: 47 4F 54 1163	DCI 'GOTO'
C474: C2	
C475: 52 55 CE 1164	DCI 'RUM'
C478: 49 C6 1165	DCI 'IF'
C47A: 52 45 53 1166	DCI 'RESTORE'
C47D: 54 4F 52 C5	
C481: A6 1167	ASC '8'
C482: 47 4F 53 1168	DCI 'DOORH'
C485: 55 C2	
C487: 52 45 54 1169	DCI 'RETURN'
C48A: 55 52 CE	
C48B: 52 45 C0 1170	DCI 'REN'
C490: 53 54 4F 1171	DCI 'STOP'
C493: 00	
C494: 4F CE 1172	DCI 'WH'
C496: 57 41 49 1173	DCI 'WAIT'
C499: 04	
C49A: 4C 4F 41 1174	DCI 'LOAD'
C49D: C4	
C49E: 53 41 56 1175	DCI 'SAVE'
C4A1: C5	
C4A2: 44 45 C6 1176	DCI 'DEF'
C4A5: 50 4F 48 1177	DCI 'PAGE'
C4A8: C5	
C4A9: 50 52 49 1178	DCI 'PRINT'
C4AC: 4E 04	
C4AE: 43 4F 4E 1179	DCI 'COMT'
C4B1: 04	
C4B2: 4C 49 53 1180	DCI 'LIST'

C4B5: 04	
C4B6: 43 4C 45 1181	DCI 'CLEAR'
C4B9: 41 02	
C4BB: 47 45 04 1182	DCI 'GET'
C4BE: 4E 45 07 1183	DCI 'MEM'
C4C1: 54 41 42 1184	DCI 'TABL'
C4C4: A8	
C4C5: 54 CF 1185	DCI 'TO'
C4C7: 46 CE 1186	DCI 'FN'
C4C9: 53 58 43 1187	DCI 'SPEC'
C4CC: A8	
C4CD: 54 48 45 1188	DCI 'THEN'
C4D0: CE	
C4D1: 41 04 1189	DCI 'AT'
C4D3: 4E 4F 04 1190	DCI 'WOT'
C4D6: 53 54 45 1191	DCI 'STEP'
C4D9: 00	
C4DA: A8 1192	ASC "A"
C4DB: A0 1193	ASC "B"
C4DC: AA 1194	ASC "C"
C4DD: AF 1195	ASC "D"
C4DE: DE 1196	ASC "E"
C4DF: 41 4E C4 1197	DCI 'AND'
C4E2: 4F 02 1198	DCI 'OR'
C4E4: BE 1199	ASC "J"
C4E5: B0 1200	ASC "K"
C4E6: BC 1201	ASC "L"
C4E7: 53 47 CE 1202	DCI 'SEM'
C4EA: 4F 4E 04 1203	DCI 'DIT'
C4ED: 41 42 00 1204	DCI 'MS'
C4FD: 53 53 02 1205	DCI 'USR'
C4F3: 46 52 C5 1206	DCI 'PRE'
C4F6: 53 43 52 1207	DCI 'SCRN'
C4F9: 4E A8	
C4FB: 58 44 CC 1208	DCI 'PDL'
C4FE: 58 4F 03 1209	DCI 'POS'
C501: 53 51 02 1210	DCI 'SAR'
C504: 52 4E C4 1211	DCI 'RND'
C507: 4C 4F C7 1212	DCI 'LOG'
C50A: 45 58 00 1213	DCI 'EXP'
C50D: 43 4F 03 1214	DCI 'COS'
C510: 53 4F CE 1215	DCI 'SIN'
C513: 54 41 CE 1216	DCI 'TAB'
C516: 41 54 CE 1217	DCI 'ATR'
C519: 58 45 45 1218	DCI 'PEEK'
C51C: 08	
C51D: 4C 45 CE 1219	DCI 'LEW'
C520: 53 54 52 1220	DCI 'STRS'
C523: A4	
C524: 56 41 CC 1221	DCI 'VAL'
C527: 41 53 C3 1222	DCI 'ASC'
C52A: 43 48 52 1223	DCI 'DMS'
C52D: A4	
C52E: 4C 45 46 1224	DCI 'LEFTS'
C531: 54 A4	
C534: 52 4F 47 1225	DCI 'RIGHTS'
C536: 48 54 A4	

```

CS3F: 40 49 44 1226 DCI 'MIDS'
CS3C: AA
CS3D: 4C CD 1227 DCI 'LN'
CS3F: 40 4F 54 1228 DCI 'MOTOR'
CS42: 4F D2
CS44: 54 48 32 1229 DCI 'TK2000'
CS47: 30 30 80
CS4A: 40 E0 1230 DCI 'MP'
CS4C: 40 C1 1231 DCI 'MA'
CS4E: 00 1232 HEX 00
      368 PUT PARTE1C_01
      31
      32 NENSERRO
CS4F: 4E 45 58 13 MXwFOR DCI 'NEXT SEN FOR'
CS52: 54 20 53 45 40 20 46 4F
CS5A: 02
CS5B: 53 49 4E 34 SINTERR0 DCI 'SINTAXE'
CS5E: 54 41 58 45 A0
CS63: 52 45 54 15 RTMw0S0 DCI 'RETURN SEN GOSUB'
CS64: 53 52 4E 20 53 45 40 20
CS6E: 47 4F 53 55 C2
CS73: 4E 41 4F 16 MhaDATA DCI 'MAO HA MAIS DATA'
CS74: 20 48 41 20 40 41 49 53
CS7E: 20 44 41 54 C1
CS80: 54 41 4C 37 WLTILES DCI 'WLMR ILEGAL'
CS84: 4F 52 20 49 4C 45 47 41
CS8E: CC
CS8F: 45 53 54 38 ESTOR DCI 'ESTOURO'
CS92: 4F 53 52 CF
CS94: 45 58 43 39 ExcHEN DCI 'EXCEDE MEMORIA'
CS99: 45 44 45 20 40 45 40 4F
CSA1: 52 49 C1
CSA4: 43 4F 40 310 COMwDEF DCI 'COMANDO NAO DEFINIDO'
CSA7: 41 4E 44 4F 20 4E 41 4F
CSAF: 20 44 45 46 49 4E 49 44
CSB7: CF
CSB8: 49 4E 44 311 INDILES DCI 'INDICE ILEGAL'
CSB8: 49 43 45 20 49 4C 45 47
CSC0: 41 CC
CSC5: 52 45 27 312 RediwRR DCI 'RE'DIR' DE ARRWA'
CSC8: 44 49 40 27 20 44 45 20
CS00: 41 52 52 41 D9
CS05: 44 49 56 313 DTVw0Z0 DCI 'DIVISAO POR ZERO'
CS08: 49 53 41 4F 20 50 4F 52
CS0B: 20 58 45 52 CF
CS0E: 43 4F 40 314 COMwLES DCI 'COMANDO ILEGAL'
CS0B: 41 4E 44 4F 20 49 4C 45
CSF0: 47 41 CC
CSF3: 49 4E 43 315 INCOMP DCI 'INCOMPATIVEL'
CSFA: 4F 40 50 41 54 49 56 45
CSFE: CC
CSFF: 45 58 43 316 EXCSTR0 DCI 'EXCEDE STRING'
CA02: 45 44 45 20 53 54 52 49
CA0A: 4E C7
CA0C: 46 4F 52 317 FORw0wCX DCI 'FORMULA NUNTO COMPLEXA'
CA0F: 40 55 4C 41 20 40 55 49
CA17: 54 4F 20 43 4F 40 50 4C

```

```

C63F: 45 58 C1
C640: 49 40 50 118 IMPOSSL DCI 'IMPOSSIVEL'
C641: 4F 53 53 49 56 45 CC
C642: 46 55 4E 119 FUNDEF DCI 'FUNCAO NAO DEFINIDA'
C643: 43 41 4F 20 4E 41 4F 20
C644: 44 45 46 49 4E 49 44 C1
      120
C645: 20 23 45 121 ERRIN ASC ' NERR0'0700
C646: 52 52 4F 07 00
C647: 20 45 40 122 DMMSG ASC ' EH '00
C648: 20 00
C649: 00 123 BREAKTH HEX 00
C64A: 42 52 45 124 ASC 'BREAK'0700
C64B: 41 48 07 00
      125
C654: 8A 126 GTFORPMT TSI ;Procurar na pilha
C655: EB 127 INX
C656: EB 128 INX
C657: EB 129 INX
C658: EB 130 INX
C659: 80 01 01 131 FMEFOR LDA STACK+1,X
C65C: C9 01 132 CMP #for
C65E: D0 21 133 BNE RET1
C660: A5 86 134 LDA FORPMT+1
C662: D0 0A 135 BNE SAMEFOR? ;Se variavel especificada
C664: 80 02 01 136 LDA STACK+2,X ;Obter pointer a variavel FOR
C667: 85 85 137 STA FORPMT
C669: 80 03 01 138 LDA STACK+3,X
C66C: 85 86 139 STA FORPMT+1
C66E: D0 03 01 140 SAMEFOR? CMP STACK+3,X ;Comp enderecos de variavel FOR
C671: D0 07 141 BNE XOFOR ;Diferentes,desviar
C673: A5 85 142 LDA FORPMT
C675: D0 02 01 143 CMP STACK+2,X
C678: F0 07 144 BE# RET1
C67A: 8A 145 XOFOR TXA ;FOR incorreto
C67B: 10 146 CLC ;Preparar para pesq.o proximo
C67C: 6F 12 147 ADC #512
C67E: AA 148 TAX
C67F: D0 08 149 BNE FMEFOR
C681: 60 150 RET1 RTS
      151
C682: 20 02 C6 152 BLTU JSR REASON ;na espaco?
C683: 85 60 153 STA STREND ;file da area de array=(A,Y)
C687: 84 6E 154 STY STREND+1
      155
      156 * Preparar para mover de LOWTR ate HIGHTR-1
      157 * para a area a partir do byte seguinte a HIGHDS
      158
C689: 38 159 BLTU# SEC
C68A: A5 96 160 LDA HIGHTR
C68C: E5 98 161 SBC LOWTR
C68E: 85 5E 162 STA INDEX
C690: A6 163 TAY
C691: A5 97 164 LDA HIGHTR+1
C693: E5 9C 165 SBC LOWTR+1
C695: AA 166 TAX
C696: EB 167 INX

```



C697: 90	168	TYA	
C698: F0 23	169	BEQ NOPAG	;Se nao pag parcial,desviar
C69A: A5 96	170	LDA HIGHTR	;Prey.para mover pag.parcial
C69C: 38	171	SEC	;antes,para maximizar velocid.
C69D: E5 5E	172	SBC INDEX	
C69F: 85 96	173	STA HIGHTR	
C6A1: 80 03	174	BCS SETEND	
C6A3: C6 97	175	DEC HIGHTR+1	
C6A5: 38	176	SEC	
C6A6: A5 94	177	SETEND LDA HIGHDS	
C6A8: E5 5E	178	SBC INDEX	
C6AA: 85 94	179	STA HIGHDS	
C6AC: 80 08	180	BCS NOBYT	
C6AE: C6 95	181	DEC HIGHDS+1	
C6B0: 90 04	182	BCC NOBYT	
C6B2: 81 96	183	INBYT LDA (HIGHTR),Y	
C6B4: 91 94	184	STA (HIGHDS),Y	
C6B6: 88	185	NOBYT DEY	
C6B7: 00 F9	186	BNE INBYT	
C6B9: 81 96	187	LDA (HIGHTR),Y	
C6BB: 91 94	188	STA (HIGHDS),Y	
C6BD: C6 97	189	NOPAG DEC HIGHTR+1	
C6BF: C6 95	190	DEC HIGHDS+1	
C6C1: CA	191	DEX	;Outra pagina a mover?
C6C2: 00 F2	192	BNE NOBYT	
C6C4: 60	193	RTS	
	194		
	195	*Teste de espaco na pilha usado por FOR, BOSUB, FRMVEL	
	196		
C6C5: 0A	197	CHKMEM ASL	;Extra com A+1,3 ou 9
C6CA: 6F 36	198	ADC #536	
C6CB: 80 35	199	BCS HENERR	
C6CA: 85 5E	1100	STA INDEX	
C6CC: BA	1101	TSX	
C6CD: E4 5E	1102	CPX INDEX	
C6CF: 90 2E	1103	BCC HENERR	
C6D1: 60	1104	RTS	
	1105		
C6D2: C4 70	1106	BEASGN CPY FRETOP+1	;Verifica se A,Y < FRETOP
C6D4: 90 28	1107	BCC RET2	;Sim,retorna
C6DA: 00 04	1108	BNE RS1	;No,limpa area
C6DB: C5 6F	1109	CMP FRETOP	
C6DA: 90 22	1110	BCC RET2	
C6DC: 48	1111	RS1 PHA	;Guardar A,Y e TEMP1 e TEMP2
C6DD: A2 09	1112	LDX #FAC-TEMP1-1	
C6DF: 98	1113	TYA	
C6E1: 48	1114	RS2 PHA	
C6E1: 85 93	1115	LDA TEMP1,X	
C6E3: CA	1116	DEX	
C6E4: 10 FA	1117	BPL RS2	
C6EA: 20 9E 07	1118	JSR GARRNG	;Reorganizar a area
C6E9: A2 F7	1119	LDX #TEMP1-FAC+1	
C6EB: 68	1120	RS3 PLA	;Restaurar TEMP1,TEMP2 e A,Y
C6EC: 95 90	1121	STA FAC,X	
C6ED: EB	1122	INX	
C6EF: 30 FA	1123	BNI RS3	
C6F1: 68	1124	PLA	

```

CAF2: A0 1125 TAY
CAF3: A8 1126 PLA ;Agora tem espaço?
CAF4: C4 70 1127 CPY FRETOP+1
CAF6: 90 06 1128 BCC RET2 ;Sim,retornar
CAF8: 00 05 1129 BNE MEMERR ;Nao,erro falta de memoria
CAF9: C5 AF 1130 CMP FRETOP
CAF0: 80 01 1131 BCS MEMERR
CAF5: 60 1132 RET2 RTS
1133
CAF7: A2 47 1134 MEMERR LDX MEMMEN-MEMERR0
C701: 24 06 1135 ERROR BIT ERRFLG ;MEMRR ativo?
C703: 10 03 1136 BPL DOERRMSG ;Nao,desviar
C705: 4C 4A E6 1137 JMP HANDLERR
1138
C706: 20 20 CE 1139 DOERRMSG JSR CR00
C708: 20 7F CE 1140 JSR OUTQUES
C70E: 80 4F C5 1141 ERLUP LDA MEMERR0,X
C711: 48 1142 PNA
C712: 20 81 CE 1143 JSR OUTDO
C713: E8 1144 INX
C716: 68 1145 PLA
C717: 10 F5 1146 BPL ERLUP
C719: 20 9A C9 1147 JSR STXINI
C71C: A9 3F 1148 LDA MEMRRIN
C71E: A0 C6 1149 LOY MEMRRIN
C720: 20 5F CE 1150 PRINTM? JSR STROUT
C723: A4 76 1151 LOY CURLIN+1 ;modo direto
C725: C8 1152 INY
C726: F0 03 1153 BEB RESTART ;Sim, desviar
C728: 20 33 E0 1154 JSR INPRT
1155
C729: 20 20 CE 1156 RESTART JSR CR00
C72E: A2 BE 1157 LDX #FF
C730: 20 10 C8 1158 JSR INLIM2 ;Obter input direto
C733: 86 80 1159 STX TXTPTR ;apontar ao buffer de input
C735: 84 89 1160 STY TXTPTR+1
C737: 46 08 1161 LSR ERRFLG ;ligar MEMRR
C739: 20 81 00 1162 JSR CHRGET
C73C: A8 1163 TAX
C73D: F0 EC 1164 BEB RESTART ;Se nao ha input
C73F: A2 FF 1165 LDX #FF ;ligar indicador de modo direto
C741: 86 76 1166 STX CURLIN+1 ;byte de alta ordem de CURLIN
C743: 90 06 1167 BCC KOLIN ;desv.se houver suaver.de linha
C745: 20 6E C8 1168 JSR GETIN ;Case contrario,analizar
C748: 4C 1F C8 1169 JMP TRACE? e executar
C74B: A6 AF 1170 KOLIN LDX PRGEND
C74D: 86 69 1171 STX VARTAB
C74F: A6 80 1172 LDX PRGEND+1
C751: 86 6A 1173 STX VARTAB+1
C753: 20 31 C0 1174 JSR LINDET ;Obter numero da linha
C756: 20 6E C8 1175 JSR GETIN e analisar input
C759: 84 0F 1176 STY PTR ;guardar indice no buffer input
C75B: 20 31 C9 1177 JSR FNDLIN ;ha alguma linha?
C75E: 90 44 1178 BCC NOLIN? ;Nao,desviar
C760: A0 01 1179 LOY #01 ;Sim,apagar a linha
C762: 81 98 1180 LDA (LONTR),Y ;Obter o apontador
C764: 85 5F 1181 STA INDEXT+1

```

C766:	AS 69	1182	LDA	VARTAB	
C768:	BS 5E	1183	STA	INDEX	
C76A:	AS 9C	1184	LDA	LOWTR+1	
C76C:	BS 61	1185	STA	DEST+1	
C76E:	AS 98	1186	LDA	LOWTR	
C770:	88	1187	DEY		
C771:	F1 98	1188	SBC	(LOWTR),Y	;linka-apontador
C773:	18	1189	CLC		
C774:	AS 69	1190	ADC	VARTAB	
C776:	BS 69	1191	STA	VARTAB	;Novo fim de programa
C778:	BS 60	1192	STA	DEST	
C77A:	AS 6A	1193	LDA	VARTAB+1	
C77C:	69 FF	1194	ADC	BSFF	
C77E:	BS 6A	1195	STA	VARTAB+1	
C780:	ES 9C	1196	SBC	LOWTR+1	
C782:	AA	1197	TAX		
C783:	30	1198	SEC		
C784:	AS 98	1199	LDA	LOWTR	
C786:	ES 69	1200	SBC	VARTAB	
C788:	AB	1201	TAY		;Indice para mover pag parcial
C789:	80 03	1202	BCS	ML1	
C78B:	EB	1203	INX		
C78C:	C6 61	1204	DEC	DEST+1	
C78E:	18	1205	ML1	CLC	
C78F:	AS 5E	1206	ADC	INDEX	
C791:	90 03	1207	BCC	MOVW	
C793:	C6 5F	1208	DEC	INDEX+1	
C795:	18	1209	CLC		
C796:	81 5E	1210	MOVW	LDA (INDEX),Y	;Mover resto do programa
C798:	91 60	1211	STA	(DEST),Y	;ao espaco da linha apagada
C79A:	CB	1212	INY		
C79B:	90 F9	1213	BNE	MOVW	
C79D:	E6 5F	1214	INC	INDEX+1	
C79F:	E6 61	1215	INC	DEST+1	
C7A1:	CA	1216	DEX		;Mover outra pagina?
C7A2:	D0 F2	1217	BNE	MOVW	
C7A4:	A0 00 02	1218	MEMLN?	LDA IN	;So numero de linha?
C7A7:	FD 38	1219	BEQ	LIMSET	;Sim,saltar a LIMSET
C7A9:	AS 73	1220	LDA	MEMSIZ	
C7AB:	A4 74	1221	LDY	MEMSIZ+1	
C7AD:	BS 6F	1222	STA	FRETOP	
C7AF:	84 70	1223	STY	FRETOP+1	
C7B1:	AS 69	1224	LDA	VARTAB	;Preparar movimento de memoria
C7B3:	BS 96	1225	STA	HIGHTR	para inserir nova linha
C7B5:	AS 0F	1226	ADC	PATR	
C7B7:	BS 94	1227	STA	HIGHDS	
C7B9:	AA 6A	1228	LDY	VARTAB+1	
C7BB:	84 97	1229	STY	HIGHTR+1	
C7BD:	90 01	1230	BCC	WAPRG	
C7BF:	CB	1231	INY		
C7C0:	84 95	1232	WAPRG	STY HIGHDS+1	
C7C2:	20 82 C6	1233	JSR	BLTU	;Realizar o movimento
C7C5:	AS 50	1234	LDA	LIMNUM	
C7C7:	AA 51	1235	LDY	LIMNUM+1	
C7C9:	80 FE 01	1236	STA	IN-2	
C7CB:	8C FF 01	1237	STY	IN-1	
C7CD:	AS 60	1238	LDA	STREND	

```

C701: A4 6E 1239          LDY STREND+1
C703: 85 69 1240          STA VARTAB
C705: 84 6A 1241          STY VARTAB+1
C707: A4 6F 1242          LDY PTR
C709: 89 F8 01 1243      INSRTLIN LDA IN-5,Y ;Inserir nova linha
C70C: 88 1244          DEY
C70D: 91 98 1245          STA (LOWTR),Y
C70F: D0 F8 1246          BNE INSRTLIN
C7E1: 20 7C C9 1247      LINKSET JSR SETPTRS ;Note que LINKSET pode ser
C7E4: A5 67 1248          LDA TXTTAB ;chamado por 0(RTR)
C7E6: A4 68 1249          LDY TXTTAB+1
C7E8: 85 5E 1250          STA INDEX
C7EA: 84 5F 1251          STY INDEX+1
C7EC: 18 1252          CLC
C7ED: A0 01 1253      MOLDNK LDY #501
C7EF: 81 5E 1254          LDA (INDEX),Y
C7F1: 00 08 1255          BNE PUTLINK
C7F3: A5 69 1256          LDA VARTAB
C7F5: 85 6F 1257          STA PRGEND
C7F7: A5 6A 1258          LDA VARTAB+1
C7F9: 85 80 1259          STA PRGEND+1
C7FB: 4C 28 C7 1260      JMP RESTART
C7FE: A0 04 1261      PUTLINK LDY #504 ;Preparar apontadores
C800: C8 1262      FINEOL INY
C801: 81 5E 1263          LDA (INDEX),Y
C803: D0 F8 1264          BNE FINEOL
C805: C8 1265          INY
C806: 98 1266          TYA
C807: 65 5E 1267          ADC INDEX
C809: A6 1268          TAX
C80A: A0 00 1269          LDY #500
C80C: 91 5E 1270          STA (INDEX),Y
C80E: A5 5F 1271          LDA INDEX+1
C810: 69 00 1272          ADC #500
C812: C8 1273          INY
C813: 91 5E 1274          STA (INDEX),Y
C815: 86 5E 1275          STX INDEX
C817: 85 5F 1276          STA INDEX+1
C819: 90 02 1277          BCC MOLDNK
1278
C818: A2 80 1279      INLINK LDY #500
C81A: 86 33 1280      INLINK STX PROMPT
C81F: A0 35 04 1281          LDA TSUTIL
C822: D0 04 1282          BNE INLINK
C824: 20 6A F0 1283      INLINK JSR GETLN
C827: 4C 48 C8 1284      JMP INLINK ;Continua em INLINK
C82A: A0 F0 9F 1285      INLINK LDA TSCART
C82B: C9 01 1286          CMP #501
C82F: D0 F3 1287          BNE INLINK
C831: A2 00 1288          LDY #500
C832: 8E 35 04 1289          STX TSUTIL
C836: A9 52 1290          LDA #'R'
C838: 90 00 02 1291          STA IN,X
C83B: E8 1292          INX
C83C: A9 55 1293          LDA #'U'
C83E: 90 00 02 1294          STA IN,X
C841: E8 1295          INX

```

```

CB42: A9 4E 1296 LDA #'N'
CB44: 90 00 02 1297 STA IN,X
CB47: EB 1298 INX
CB48: ED EF 1299 INLDNS CPX #DEF ;Terminar linha com DEF
CB4A: 90 02 1300 BCC GOB0FFS ;Colocar marc.de fim de linha
CB4C: A2 EF 1301 LDX #DEF
CB4E: A9 80 1302 GOB0FFS LDA #500
CB50: 90 00 02 1303 STA IN,X
CB53: BA 1304 TXA
CB54: F0 08 1305 BEQ NOT
CB56: ED FF 01 1306 STRIP LDA IN-1,X ;Converter em ASCII
CB59: 29 7F 1307 AND #57F
CB5B: 90 FF 01 1308 STA IN-1,X
CB5E: CA 1309 DEX
CB5F: 00 F5 1310 BNE STRIP
CB61: A9 00 1311 MOI LDA #500
CB63: A2 FF 1312 LDX #5FF
CB65: A0 01 1313 LDY #501
CB67: 60 1314 RTS
          319 PUT PARTE10,01
          31
CB68: 20 0C F0 32 ENCHR JSR IMO
CB6B: 29 7F 33 AND #57F
CB6D: 60 34 RTS
          35
CB6E: A6 88 36 GETIN LDX TXTPTR
CB70: CA 37 DEX
CB71: A0 04 38 LDY #504
CB73: 84 13 39 STY DATAFLG
CB75: 24 06 310 BIT LOCK ;Programa protegido?
CB77: 10 08 311 BPL PARSE ;Sim,ignorar input
CB79: 68 312 PLA ;e rodar o programa novamente
CB7A: 68 313 PLA
CB7B: 20 7C C9 314 JSR SETPTRS
CB7E: 4C EC CA 315 JMP NEMSTT
          316
CB81: EB 317 PARSE INX
CB82: 80 90 02 318 KROCH LDA IN,X
CB85: 24 13 319 BIT DATAFLG
CB87: 70 04 320 BVS SE ;Desviar se instrucao DATA
CB89: C9 20 321 CMP #' '
CB8B: F0 F4 322 BEQ PARSE
CB8D: 08 323 SE PHP
CB8E: 85 0E 324 STA ENCHR
CB90: 28 325 PLP
CB91: C9 22 326 CMP #'"'
CB93: F0 74 327 BEQ SHIN
CB95: 70 40 328 BVS PUTIN ;Desviar se instrucao DATA
CB97: C9 3F 329 CMP #'?'
CB99: 00 04 330 BNE TOK?
CB9B: A9 BA 331 LDA @print
CB9D: 00 45 332 BNE PUTIN ;Sempre
CB9F: C9 30 333 TOK? CMP #'0'
CBAB: 90 04 334 BCC ISTOP?
CBAD: C9 3C 335 CMP #'('
CBAG: 90 30 336 BCC PUTIN
CBAB: 84 AD 337 ISTOP? STY STMSG2

```

CB4F: A9 8D	138		LDA	WTOKTABL-5100	
CB4E: 85 9D	139		STA	FAC	
CB4D: A9 C2	140		LDA	WTOKTABL-5100	
CB4F: 85 9E	141		STA	FAC+1	
CB81: A0 00	142		LDY	#500	
CB83: 84 0F	143		STY	PNTR	;Guarda token corrente 580
CB85: 88	144		DEY		
CB86: 86 88	145		STX	TXTPTR	
CB88: CA	146		DEX		
CB8F: C8	147	HY	INY		
CB8A: 00 02	148		BNE	HX	
CB8C: E6 9E	149		INC	FAC+1	
CB8E: E8	150	KX	INX		
CB8F: 80 00 02	151	LIN	LDA	IN,X	
CB82: C9 20	152		CMR	#' '	;Saltar espaços
CB8A: F0 FB	153		BEQ	HX	
CB84: 38	154		SEC		
CB87: F1 9D	155		SBC	(FAC),Y	;Bate com palav.chave corrente?
CB89: F0 EE	156		BEQ	NY	;Sim, proximo caracter
CB8B: C9 80	157		CMR	#80	;Bate com ult.carac.da palavra-
	158	*			;chave
CB8D: 00 41	159		BNE	SKIPTOK	;Nao, passar ao prox.token
CB8F: 05 0F	160		ORA	PNTR	;Obter o token
CB81: C9 C5	161		CMR	#at	
CB83: 00 00	162		BNE	PUTTOK	
CB85: 80 01 02	163		LDA	IN+1,X	
CB88: C9 4E	164		CMR	#'N'	;Prioridade a ATN
CB8A: F0 34	165		BEQ	SKIPTOK	
CB8C: C9 4F	166		CMR	#'0'	;Prioridade a T0
CB8E: F0 30	167		BEQ	SKIPTOK	
CB8D: A9 C5	168		LDA	#at	
CB82: 44 A0	169	PUTTOK	LDY	STRNG2	
CB84: E8	170	PUTIN	INX		
CB85: C8	171		INY		
CB86: 99 FB 01	172		STA	IN-5,Y	
CB89: 89 FB 01	173		LDA	IN-5,Y	
CB8C: F8 39	174		BEQ	DOME	
CB8E: 38	175		SEC		
CB8F: C9 3A	176		SBC	#'+'	
CB81: F8 04	177		BEQ	SSF	;Desligar DATAFLG
CB83: C9 49	178		CMR	#ata-'#'	no fim da instrucao
CB85: 00 02	179		BNE	BEV?	
CB87: 05 13	180	SSF	STA	DATAFLG	
CB89: 38	181	RDV?	SEC		
CB8A: E9 78	182		SBC	#res-'#'	
CB8C: 00 84	183		BNE	KXCHR	
CB8E: 05 0E	184		STA	ENDCHR	;Limpar flag de literais
C900: 80 00 02	185	SHFTIN	LDA	IN,X	
C903: F0 0F	186		BEQ	PUTIN	
C905: C5 0E	187		CMR	ENDCHR	
C907: F0 08	188		BEQ	PUTIN	
C909: C8	189	SHIN	INY		
C90A: 99 FB 01	190		STA	IN-5,Y	
C90B: E8	191		DEX		
C90E: 00 F0	192		BNE	SHFTIN	;Loop ate tera.os literais
C910: A6 88	193	SKIPTOK	LDX	TXTPTR	
C912: E6 0F	194		INC	PNTR	;Proximo TOKEN

```

C914: 81 90 195 SK2 LDA (FAC),Y ;Saltar palav.chave corrente
C916: C8 196 IMY
C917: 00 02 197 BNE FLU7
C919: E6 9E 198 IMC FAC+1
C91B: 0A 199 FLU7 ASL
C91C: 90 F6 1100 BCC SK2 ;Loop ate saltar a palav+chav.
C91E: 81 90 1101 LDA (FAC),Y
C920: 00 90 1102 BNE LIN ;Loop ate fim de tabela
1103 * ;palavra-chave
C922: 80 00 02 1104 LDA IN,X ;Nao e palavra chave
C925: 10 88 1105 BPL PUTT0C ;Sempre
C927: 99 FD 01 1106 DOME STA IN-3,Y ;EOL se modo direto
C92A: C6 89 1107 DEC TXTPTR+1 ;Apontar TXTPTR a IN-1
C92C: A9 FF 1108 LDA #5FF
C92E: 85 88 1109 STA TXTPTR
C930: 60 1110 RTS
1111
1112 *Procurar no programa linha cujo nam esta agora em LINNAM.
1113 *Ao sair, carry ligado se encontrada, deslig caso contrar.
1114 *LOWTR aponta para a linha se foi encontrada, e aponta
1115 *para a proxima caso contrario
1116
C931: A5 67 1117 FNDLIN LDA TXTTAB ;Comeco da pesquisa
C933: A6 68 1118 LDX TXTTAB+1 no inicio do programa
C935: A0 01 1119 FL1 LDY #501 ;Comeco da pesquisa em A,X
C937: 85 98 1120 STA LOWTR
C939: 86 9C 1121 STX LOWTR+1
C93B: 81 98 1122 LDA (LOWTR),Y ;Obter apontador
C93D: FD 1F 1123 BEQ NOSUCH ;Desviar se fim de programa
C93F: C8 1124 IMY
C940: C8 1125 IMY
C941: A5 51 1126 LDA LINNAM+1
C943: 01 98 1127 CMP (LOWTR),Y ;Comp.nam.da linha (part.alta)
C945: 90 18 1128 BCC RET3 ;Nao, encontrado
C947: FD 03 1129 BEQ FL2
C949: 88 1130 DEY
C94A: 00 09 1131 BNE GETLINK ;Obter proxima linha
C94C: A5 50 1132 FL2 LDA LINNAM
C94E: 88 1133 DEY
C94F: 01 98 1134 CMP (LOWTR),Y ;Numero de linha (baixa ordem)
C951: 90 0C 1135 BCC RET3 ;Passou da linha,nao encontrado
C953: FD 0A 1136 BEQ RET3 ;Encontrado
C955: 88 1137 GETLINK DEY
C956: 81 98 1138 LDA (LOWTR),Y ;Obter proximo apontador
C958: AA 1139 TAX ;(Alta ordem)
C959: 88 1140 DEY
C95A: 81 98 1141 LDA (LOWTR),Y ;Parte baixa ordem
C95C: 80 07 1142 BCS FL1 ;Sempre
C95E: 18 1143 NOSUCH CLC
C95F: 60 1144 RET3 RTS
1145
C960: 00 FD 1146 NEW BNE RET3 ;Desviar se erro de sintaxe
C962: A9 00 1147 SCRTOH LDA #500
C964: 85 06 1148 STA LOCK ;Permitir comandos de usuario
C966: A8 1149 TAX
C967: 91 67 1150 STA (TXTTAB),Y
C969: C8 1151 IMY

```

```

C96A: 91 67 3152 STA (TXTTAB),Y
C96C: A5 67 3153 LDA TXTTAB
C96E: 69 02 3154 ADC #502 ;Se indeterminado,carry
C970: 85 69 3155 STA WARTAB
C972: 85 AF 3156 STA PAGEHD
C974: A5 68 3157 LDA TXTTAB+1
C976: 69 00 3158 ADC #500
C978: 85 6A 3159 STA WARTAB+1
C97A: 85 80 3160 STA PAGEHD+1
C97C: 20 AE C9 3161 SETPTRS JSR STXTPT
C97F: A9 00 3162 LDA #500
C981: 00 2A 3163 CLEAR BNE RET4
C983: A5 73 3164 CLEAR LDA MEMSZ
C985: A4 74 3165 LDY MEMSZ+1
C987: 85 6F 3166 STA FRETOP
C989: 84 70 3167 STY FRETOP+1
C98B: A5 69 3168 LDA WARTAB
C98D: A4 6A 3169 LDY WARTAB+1
C98F: 85 68 3170 STA ARYTAB
C991: 84 6C 3171 STY ARYTAB+1
C993: 85 60 3172 STA STREHD
C995: 84 6E 3173 STY STREHD+1
C997: 20 69 C8 3174 JSR RESTORE
C99A: A2 55 3175 STKINI LDR #55
C99C: 86 52 3176 STX TEMPT
C99E: 68 3177 PLA
C9A0: A8 3178 TAY
C9A2: 68 3179 PLA
C9A4: A2 F8 3180 LDX #5F8 ;Guardar topo da pilha para
C9A6: 9A 3181 TXS ;apontador e numero da linha
C9A8: A8 3182 PHA
C9AA: 98 3183 TYA
C9AC: A8 3184 PHA
C9AE: A9 00 3185 LDA #500
C9B0: 85 7A 3186 STA OLDTEXT+1 ;CONT normalmente
C9B2: 85 14 3187 STA SUBFLG
C9B4: 60 3188 RET4 RTS
3189
C9AE: 18 3190 STXTPT CLC
C9AF: A5 67 3191 LDA TXTTAB
C9B1: 69 FF 3192 ADC #5FF
C9B3: 85 88 3193 STA TXPTR
C9B5: A5 68 3194 LDA TXTTAB+1
C9B7: 69 FF 3195 ADC #5FF
C9B9: 85 89 3196 STA TXPTR+1
C9BB: 60 3197 RETST RTS
3198
C9BC: 90 DA 3199 LIST BCC STRTRNG ;Ma numero de linha?
C9BE: F0 08 3200 BEQ STRTRNG ;Nao
C9C0: C9 C9 3201 CNP #minus ;Sim, intervalo de LIST com 0
C9C2: F0 04 3202 BEQ STRTRNG
C9C4: C9 2C 3203 CNP #', '
C9C6: 80 E5 3204 BNE RET4
C9C8: 20 31 CD 3205 STRTRNG JSR LINGET ;LIMMUP=comeco do intervalo
C9CA: 20 31 C9 3206 JSR FNDLIN ;LONTR aponta a primeira linha
C9CC: 20 B7 00 3207 JSR CHNGT ;Intervalo especificado?
C9CE: F0 10 3208 BEQ HATWST ;Nao, desviar

```



CY03: C9 C9	1209	CMP	#minus	
CY05: F0 04	1210	BEQ	ENDRNG	
CY07: C9 2C	1211	CMP	#','	
CY09: 00 84	1212	BNE	RET3	
CY0B: 20 81 00	1213	ENDRNG	JSR	CHRGET ;Atualizar TXTPTR
CY0E: 20 31 C0	1214	JSR	LINGET	;LIMM=fim do intervalo
CY11: 00 CA	1215	BNE	RET4	;Desviar se erro de sintaxe
CY13: A8	1216	WDLNLT	PLA	;furar da pilha end.de volta
CY14: 68	1217		PLA	
CY15: A5 50	1218	LDA	LIMM#	
CY17: 05 51	1219	ORA	LIMM#+1	
CY19: 00 04	1220	BNE	NULST	
CY1B: A9 FF	1221	LDA	#5FF	;Intervalo maximo
CY1D: 85 50	1222	STA	LIMM#	
CY1F: 85 51	1223	STA	LIMM#+1	
CY21: A0 01	1224	NULST	LDY	#501
CY23: 81 98	1225	LDA	(LOWTR),Y	;Byte alta orden do apontador
CY25: F0 44	1226	BEQ	LSTED	
CY27: 20 70 C0	1227	JSR	ISCONT	;testar CONTROL-C
CY2A: 20 20 C0	1228	JSR	CR00	
CY2D: C8	1229	INY		
CY2E: 81 98	1230	LDA	(LOWTR),Y	;Obter numero da linha
CA00: AA	1231	TAX		
CA01: C8	1232	INY		
CA02: 81 98	1233	LDA	(LOWTR),Y	
CA04: C5 51	1234	CMP	LIMM#+1	
CA06: 00 04	1235	BNE	LST0?	
CA08: E4 50	1236	CPX	LIMM#	
CA0A: F0 02	1237	BEQ	LSTLIM	
CA0C: 80 20	1238	LST0?	BCL	LSTED
CA0E: 84 05	1239	LSTLIM	STY	FORPNT
CA10: 20 3E E0	1240	JSR	LIMPRT	;Imprime X,A
CA13: A9 20	1241	LDA	#' '	
CA15: A4 05	1242	LSTLOOP	LDY	FORPNT
CA17: 29 7F	1243	AND	#57F	
CA19: 20 81 CE	1244	SENDCH	JSR	OUT00
CA1C: A5 24	1245	SENDCH	LDA	CH
CA1E: C9 26	1246	CMP	#526	;Se passou de 26, CR
CA20: 90 07	1247	BCC	NCR	
CA22: 20 20 CE	1248	JSR	CR00	
CA25: A9 05	1249	LDA	#505	;E 5 espacos
CA27: 85 24	1250	STA	CH	
CA29: C8	1251	NCR	INY	
CA2A: 81 98	1252	LDA	(LOWTR),Y	
CA2C: 00 10	1253	BNE	TORCH?	
CA2E: AA	1254	TAX		;Ao fim da linha,
CA2F: 81 98	1255	LDA	(LOWTR),Y	obter apontador
CA31: AA	1256	TAX		
CA32: C8	1257	INY		
CA33: 81 98	1258	LDA	(LOWTR),Y	
CA35: 86 98	1259	STX	LOWTR	;Apontar a proxima linha
CA37: 85 9C	1260	STA	LOWTR+1	
CA39: 00 84	1261	BNE	NULST	
CA3B: A9 00	1262	LSTED	LDA	#500 ;CR e fim
CA3D: 20 81 CE	1263	JSR	OUT00	
CA40: 4C EC CA	1264	JMP	NEWSIT	
	1265			

```

CA43: 08      1266  GETCHR  INY          ;Obter caracter da tabela
CA44: 00 02   1267          BNE  GC
CA46: E6 9E   1268          LMC  FAC+1
CA4B: B1 90   1269  GC     LDA  (FAC),Y
CA4A: 60      1270          RTS
                320
CA4E: 10 CC   11     TOKEN? BPL  SENDCHR ;Desviar se nao for token
CA4D: 4C 38 F7 12          JMP  BUSTOKEN
CA5D: 38      13     COMTTOK SEC
CA51: E9 7F   14          SBC  #57F    ;Fazer pointer a tabela
CA53: AA      15          TAX
CA54: 04 85   16          STY  FORPNT ;Guardar pointer a linha
CA56: A0 80   17          LDY  #10KTABL-$100
CA5B: B4 90   18          STY  FAC    ;Fazer FAC apontar a tabela
CA5A: A0 C2   19          LDY  #10KTABL-$100
CA5C: B4 9E   110         STY  FAC+1
CA5E: A0 FF   111         LDY  #57F
CA6D: CA      112  SKPTX  DEX          ;Contar tokens ate X
CA61: F0 07   113         BEQ  PRITOK
CA63: 20 43 CA 114  TOKLP  JSR  GETCHR
CA66: 10 F8   115         BPL  TOKLP
CA6B: 30 F6   116         BMT  SKPTX
CA6A: A9 20   117  PRITOK LDA  #' '    ;Token encontrado,mandar espaco
CA6C: 20 B1 CE 118         JSR  OUTDO
CA6F: 20 43 CA 119  TOKLUP JSR  GETCHR ;E token
CA72: 30 05   120         BHI  TOKDONE
CA74: 20 B1 CE 121         JSR  OUTDO
CA77: 00 F6   122         BNE  TOKLUP
CA79: 20 B1 CE 123  TOKDONE JSR  OUTDO ;Enviar ultimo carac.do token
CA7C: A9 20   124         LDA  #' '    ;Mandar espaco final
CA7E: 00 95   125         BNE  LISTLOOP ;Voltar a linha
                126
                127  #FOR coloca os seguintes 18 bytes na pilha:
                128  # TEXTPR
                129  # Numero da linha
                130  # Valor da variavel FOR (ponto flutuante 5 bytes)
                131  # Sinal do STEP
                132  # Valor do STEP (5 bytes)
                133  # FORPNT (pointer a variavel)
                134  # "Token" de FOR
                135
CA8D: A9 80   136  FOR     LDA  #580
CA82: 85 14   137          STA  SUBPLG ;Indices nao sao permitidos
CA84: 20 48 CD 138          JSR  LET
CA87: 20 54 CA 139  JSR  GETFORPNT ;Variavel FOR esta ativa?
CA8A: 00 05   140          BNE  FOR2   ;Desviar se nao
CA8C: 0A      141          TRA
CA8B: A9 0F   142          ADC  #50F  ;Sim,cancele-a,e as seguintes
CA8F: AA      143          TAX
CA9D: 9A      144          TYS
CA91: 68      145  FOR2   PLA
CA92: 68      146          PLA
CA93: A9 09   147          LDA  #50F
CA95: 20 C5 CA 148  JSR  CHECKEN ;Verf.se stack pointer >= 540
CA98: 20 C8 CC 149  JSR  DATAN  ;Apontar ao comando seguinte
CA9B: 18      150          CLC
CA9C: 9B      151          TYA

```

```

CA00: 65 88 152      ADC  TXTPTR
CA01: 48             153      PNA
CA02: A5 89 154      LDA  TXTPTR+1
CA03: 69 00 155      ADC  #500
CA04: 48             156      PNA
CA05: A5 76 157      LDA  CURLIN+1
CA06: 48             158      PNA
CA07: A5 75 159      LDA  CURLIN
CA08: 48             160      PNA
CA09: A9 C1 161      LDA  #0
CA10: 20 E8 01 162    JSR  SYMCHR
CA11: 20 92 00 163    JSR  CHRWR
CA12: 20 BF 00 164    JSR  FRMWR
CA13: A5 A2 165      LDA  FACSEM
CA14: 09 7F 166      ORA  #57F
CA15: 25 9E 167      AND  FAC+1
CA16: 85 9E 168      STA  FAC+1
CA17: A9 C9 169      LDA  #STEP
CA18: A0 CA 170      LDY  #1/STEP
CA19: 85 5E 171      STA  INDEX
CA20: 84 5F 172      STY  INDEX+1
CA21: 4C 40 01 173    JMP  PUSHFAC
CA22: A9 2D 174      LDA  #0
CA23: A0 3C 175      LDY  #0
CA24: 20 13 0E 176    JSR  N4F4
CA25: 20 87 00 177    JSR  CHRDOT
CA26: C9 C7 178      CMP  #step
CA27: 00 06 179      BNE  ONESTEP
CA28: 20 81 00 180    JSR  CHRGET
CA29: 20 BF 00 181    JSR  FRMWR
CA30: 20 9C 0E 182    JSR  SIGH
CA31: 20 30 01 183    JSR  PSFACX
CA32: A5 86 184      LDA  FORPNT+1
CA33: 48             185      PNA
CA34: A5 85 186      LDA  FORPNT
CA35: 48             187      PNA
CA36: A9 81 188      LDA  #for
CA37: 48             189      PNA
CA38: BA 190      NEWSTT  TSX
CA39: 86 F8 191      STX  RENSTX
CA40: 20 78 08 192    JSR  ISCNTC
CA41: A5 88 193      LDA  TXTPTR
CA42: A4 89 194      LDY  TXTPTR+1
CA43: A6 76 195      LDX  CURLIN+1
CA44: E8 196      INX
CA45: F0 04 197      BEQ  DIRCT
CA46: 85 79 198      STA  OLDTEXT
CA47: 84 7A 199      STY  OLDTEXT+1
CA48: A0 00 1100   DIRCT  LDY  #500
CA49: 81 80 1101      LDA  (TXTPTR),Y
CA50: 00 50 1102      BNE  COLON?
CA51: A0 02 1103      LDY  #02
CA52: 81 88 1104      LDA  (TXTPTR),Y
CA53: 18 1105      CLC
CA54: F0 34 1106      BEQ  GOEND
CA55: C8 1107      SBY
CA56: 81 88 1108      LDA  (TXTPTR),Y

```

;Preparar para retorno a STEP  
;Retorno a STEP  
;STEP default = 1  
;Foi especificado um step,  
;Note-o  
;Executar novo comando  
;Modo direto  
;Desviar se sim  
;Guardar TXTPTR se estiver no  
programa para CONT eventual  
;Fim da linha?  
;Não, '?' ?  
;Sim, link 0?  
;Fim se linha 0

CB0F: 05 75	3109	STA	CURLIN	;Nao, e fim,
CB11: 08	3110	IMY		;guardar o numero da linha
CB12: 01 00	3111	LDA	(TXTPTR),Y	
CB14: 05 76	3112	STA	CURLIN+1	
CB16: 98	3113	TYA		
CB17: 45 88	3114	ADC	TXTPTR	;Posic. pointer ao texto TXTPTR
CB19: 05 88	3115	STA	TXTPTR	
CB18: 90 02	3116	BCC	TRACE?	
CB10: E6 89	3117	IMC	TXTPTR+1	
CB1F: 24 F2	3118	TRACE?	BIT	TRCFLG ;Foi pedido trace?
CB21: 10 14	3119	BPL	EXECUTE	;Nao,desviar
CB23: A6 76	3120	LDX	CURLIN+1	
CB25: E8	3121	INX		
CB26: F0 0F	3122	BEQ	EXECUTE	;Saltar se comando direto
CB28: A9 23	3123	LDA	#"B"	;desprie "B"
CB2A: 20 81 CE	3124	JSR	OUTDO	
CB20: A6 75	3125	LDX	CURLIN	
CB2F: A5 76	3126	LDA	CURLIN+1	
CB31: 20 3E E0	3127	JSR	LINPRT	;E o numero
CB34: 20 7C CE	3128	JSR	OUTSP	
CB37: 20 81 00	3129	EXECUTE	JSR	CHRGET ;Obter primeiro caract. do comando
CB36: 20 42 CB	3130	JSR	GOOND	;Comeca a processar a linha
CB30: 4C EC CA	3131	JMP	HEMSTT	;Retornar a outro comando
	3132			
CB40: F0 60	3133	GOEND	BEQ	END4
CB42: F0 33	3134	GOOND	BEQ	RETS
CB44: E9 80	3135	GOOND2	SBC	#580 ;Token?
CB46: 90 17	3136	BCC	WOTOK	;Desviar se nao for
CB4B: C9 40	3137	CMF	#540	;Token tipo rotina?
CB4A: 90 06	3138	BCC	GOOND3	;Erro de sintaxe se nao for
CB4C: C9 68	3139	CMF	#568	
CB4E: 90 16	3140	BCC	JSY	
CB50: E9 12	3141	SBC	#512	
CB52: 0A	3142	GOOND3	ASL	;Obter endereco da rotina
CB53: A8	3143	TAY		
CB54: 89 07 C2	3144	LDA	CNDTABL+1,Y	
CB57: 48	3145	PHA		;E coloca-la no stack
CB58: 89 06 C2	3146	LDA	CNDTABL,Y	
CB5B: 48	3147	PHA		
CB5C: 4C 81 00	3148	JMP	CHRGET	;Obter proximo caracter
CB5F: 4C 68 C0	3149	WOTOK	JMP	LET ;Atribuiçao a variav.(A=8)
CB62: C9 3A	3150	COLON?	CMF	#'+'
CB64: F0 89	3151	BEQ	TRACE?	
CB66: 4C F1 D1	3152	JSY	JMP	SYNERR
	3153			
CB69: 38	3154	RESTORE	SEC	
CB6A: A5 67	3155	LDA	TXTAB0	
CB6C: E9 01	3156	SBC	#501	
CB6E: A4 68	3157	LDY	TXTAB0+1	
CB70: 80 01	3158	BCC	SETDA	
CB72: 88	3159	DEY		
CB73: 85 70	3160	SETDA	STA	DATPTR
CB75: 84 7E	3161	STY	DATPTR+1	
CB77: 60	3162	RETS	RTS	
	3163			
CB78: 84 0F	3164	ISCONT	STY	YCNTC
CB7A: 20 43 F0	3165	JSR	SCAN1	

```

CB7D: 44 E7 1166      LDY  YCNTC
CB7F: EA 1167      MFP
CB80: C9 83 1168      CMP  #583
CB82: F0 01 1169      BEQ  GK
CB84: 60 1170      RTS
CB85: 20 68 C8 1171  GK     JSR  IMCHR
CB8B: A2 FF 1172  ERFLG?  LDX  #5FF      ;Foi tentado CONTROL-C
CB8A: 24 08 1173      BIT  ERFLG
CB8C: 10 03 1174      BPL  CTRC?
CB8E: 4C 4A E8 1175      JMP  HANDLERR
CB91: C9 83 1176      CMP  #503
      1177
CB93: 80 01 1178  STOP  BCS  END2
      1179
CB95: 18 1180  END  CLC
CB96: 00 3C 1181  END2  BNE  RET6
CB98: A5 88 1182      LDA  TXTPTR
CB9A: A4 89 1183      LDY  TXTPTR+1
CB9C: A6 76 1184      LDX  CURLIN+1      ;Modo direto?
CB9E: E8 1185      INX
CB9F: F0 0C 1186      BEQ  END3      ;Sim,desviar
CBA1: 85 79 1187      STA  OLDTEXT
CBA3: 84 7A 1188      STY  OLDTEXT+1
CBA5: A5 75 1189      LDA  CURLIN
CBA7: A4 76 1190      LDY  CURLIN+1
CBA9: 85 77 1191      STA  OLDLIN
CBAB: 84 78 1192      STY  OLDLIN+1
CBAD: 68 1193  END3  PLA
CBAE: 68 1194      PLA
CBAF: A9 4C 1195  END4  LDA  #BREAKIN
CBB1: A0 C4 1196      LDY  #BREAKIN
CBB3: 90 03 1197      BEQ  GOSTANT
CBB5: 4C 20 C7 1198      JMP  PRANTIN?
CBB8: 4C 28 C7 1199  GOSTANT  JMP  RESTART
      321      PUT  PARTE1F,01
      31
CBBB: 00 17 32  COWT  BNE  RET6
CBBD: A2 03 33      LDX  #IMPOSSL-MENSERR0
CBBF: A4 7A 34      LDY  OLDTEXT+1
CBC1: 00 03 35      BNE  COW
CBC3: 4C 01 C7 36      JMP  ERROR
CBC6: A5 79 37  COW     LDA  OLDTEXT
CBC8: 85 88 38      STA  TXTPTR
CBCA: 84 89 39      STY  TXTPTR+1
CBCC: A5 77 310     LDA  OLDLIN
CBCE: A4 78 311     LDY  OLDLIN+1
CB00: 85 75 312     STA  CURLIN
CB02: 84 76 313     STY  CURLIN+1
CB04: 60 314     RET6  RTS
      315
CB05: 38 316     SWE  SEC
CB06: A5 AF 317     LDA  PROGEND
CB08: E5 67 318     SRC  TXTTAB
CB0A: 85 50 319     STA  LINNUM
CB0C: A5 80 320     LDA  PROGEND+1
CB0E: E5 68 321     SRC  TXTTAB+1
CB10: 85 51 322     STA  LINNUM+1

```

```

CBE2: 20 15 CC 120      JSR  VARTIO
CBE5: 20 CD FE 124      JSR  WRITE
CBEB: 20 26 CC 125      JSR  PROGIO
CBEB: 4C CD FE 126      JMP  WRITE
CBEE: 20 15 CC 127      LONG JSR  VARTIO
CBF1: 20 FD FE 128      JSR  READM
CBF4: 1B      129      CLC
CBF5: A5 67 130      LDA  TTTTAB
CBF7: 65 50 131      ADC  LINNUM
CBF9: 85 69 132      STA  VARTAB
CBFB: A5 68 133      LDA  TTTTAB+1
CBFD: 65 51 134      ADC  LINNUM+1
CBFF: 85 6A 135      STA  VARTAB+1
CC01: A5 52 136      LDA  TEMPPT
CC03: 85 06 137      STA  LOCK
CC05: 20 26 CC 138      JSR  PROGIO
CC08: 20 FD FE 139      JSR  READM
CC0B: 24 06 140      BIT  LOCK      ;Se negativo ler cassete
CC0C: 10 03 141      BPL  JUNK
CC0F: 4C 7C C9 142      JMP  SETPTRS
CC12: 4C E1 C7 143      JUNK JMP  LINKSET
      144
CC15: A9 50 145      VARTIO LDA  MLINNUM
CC17: A0 00 146      LDY  #000
CC19: 85 3C 147      STA  AIL
CC1B: 84 30 148      STY  A1H
CC1D: A9 52 149      LDA  BTEMPPT
CC1F: 85 3E 150      STA  AZL
CC21: 84 3F 151      STY  A2H
CC23: 84 06 152      STY  LOCK
CC25: 60 153      RTS
      154
CC26: A5 67 155      PROGIO LDA  TTTTAB
CC28: A4 68 156      LDY  TTTTAB+1
CC2A: 85 3C 157      STA  AIL
CC2C: 84 30 158      STY  A1H
CC2E: A5 69 159      LDA  VARTAB
CC30: A4 6A 160      LDY  VARTAB+1
CC32: 85 3E 161      STA  AZL
CC34: 84 3F 162      STY  A2H
CC36: 60 163      RTS
      164
CC37: 08 165      RUM  PHP
CC3B: C6 76 166      DEC  CUBLINE+1
CC3A: 28 167      PLP
CC3B: D0 03 168      BNE  RUMLINE      ;Desviar se ha numero de linha
CC3D: 4C 7C C9 169      JMP  SETPTRS      ;Especificar comeco do program
CC40: 20 83 C9 170      RUMLINE JSR  CLEARC      ;Limpar variaveis
CC43: 4C 5A CC 171      JMP  BOLINE      ;Desviar a linha especificada
      172
      *#0SUB deixa na pilha:
      * Endereco de volta (NEWSTT)
      * TXTPTR
      * Numero da linha
      * "Tokens" do #0SUB
      173
CC46: A9 03 179      #0SUB LDA  #000

```

```

CC40: 20 C5 C6 180      JSR  CHMEN      ;Pointer da pilha e !=536?
CC41: A5 B9 181      LDA  TXTPTR+1
CC42: 48 182      PMA
CC4E: A5 B8 183      LDA  TXTPTR
CC50: 48 184      PMA
CC51: A5 76 185      LDA  CURLIN+1
CC52: 48 186      PMA
CC54: A5 75 187      LDA  CURLIN
CC56: 48 188      PMA
CC57: A9 80 189      LDA  @gsubs
CC59: 48 190      PMA
CC5A: 20 B7 00 191     GOLTME JSR  CHGOT
CC5D: 20 A3 CC 192     JSR  GOTO
CC60: 4C EC DA 193     JMP  MENSTT
      194
CC63: 20 31 C0 195     GOTO  JSR  LIMGET      ;Obter linha do GOTO
CC64: 20 C8 CC 196     JSR  RENW        ;Y Apontando ao fim da linha
CC69: A5 76 197      LDA  CURLIN+1    ;Pag.corrente ( pagina do GOTO
CC6B: C5 51 198      CMP  LIMNUM+1
CC6D: 80 08 199      BCS  G01        ;Mao, pesq. do comeco do progr.
CC6F: 98 1100      TYA             ;Sim, pesq. da proxima linha
CC70: 28 1101      SEC
CC71: A5 B8 1102     ABC  TXTPTR
CC73: A6 B9 1103     LDX  TXTPTR+1
CC75: 98 07 1104     BCC  G02
CC77: E8 1105     INX
CC78: 80 04 1106     BCS  G02
CC7A: A5 67 1107     LDA  TXTTAB      ;Obter comeco do programa
CC7C: A6 68 1108     LDX  TXTTAB+1
CC7E: 20 35 C9 1109     G02  JSR  FL1        ;Procurar linha do GOTO
CC81: 90 1E 1110     BCC  UNDERR     ;Erro! linha nao encontrada
CC83: A5 98 1111     LDA  LOWTR      ;TXTPTR apontando a linha GOTO
CC85: E9 01 1112     SBC  #501
CC87: 85 B8 1113     STA  TXTPTR
CC89: A5 9C 1114     LDA  LOWTR+1
CC8B: E9 00 1115     SBC  #500
CC8D: 85 B9 1116     STA  TXTPTR+1
CC8F: 60 1117     RET7  RTS        ;retornar a MENSTT ou GOSUB
      1118
CC90: 00 F0 1119     POP  @ME  RET7
CC92: A9 FF 1120     LDA  #FFF
CC94: 85 B5 1121     STA  FORPNT
CC96: 20 54 C6 1122     JSR  BTFORPNT
CC99: 9A 1123     TXS
CC9A: C9 80 1124     CMP  @gsubs     ;E ultimo GOSUB?
CC9C: F0 08 1125     BEQ  RETURN
CC9E: A2 14 1126     LDX  #BT#GOSB-MENSERR0
CCA0: 2C 1127     HEX  2C
CCA1: A2 55 1128     UNDERR LDX  #C3#DEF-MENSERR0
CCA3: 4C 01 C7 1129     JMP  ERROR
      1130
CCA6: 4C F1 D1 1131     GSYMER JMP  SYMER
      1132
CCA9: 68 1133     RETURN  PLA
CCAM: 68 1134     PLA
CCAB: C0 42 1135     CPY  @esp+2
CCAD: F0 38 1136     BEB  PULL3     ;Desviar se POP

```

CCAF: 85 75	1137	STA	CURLIN	;Obter numero da linha	
CCB1: 68	1138	PLA			
CCB2: 85 76	1139	STA	CURLIN+1		
CCB4: 68	1140	PLA			
CCB5: 85 88	1141	STA	TXTPTR		
CCB7: 68	1142	PLA			
CCB8: 85 89	1143	STA	TXTPTR+1		
CCBA: 20 C8 CC	1144	DATA	JSR	DATAN	;Passar ao proximo comando
CCB4: 98	1145	ADDDN	TYA		
CCB5: 18	1146		CLC		
CCB7: 45 88	1147	ADC	TXTPTR		
CCC1: 85 88	1148	STA	TXTPTR		
CCC3: 90 02	1149	BCC	RETB		
CCC5: E6 89	1150	INC	TXTPTR+1		
CCC7: 60	1151	RETB	RTS		
	1152				
CCCB: A2 3A	1153	DATAN	LDX	B'''	;fazer Y=deslocamento
CCCA: 2C	1154		HEX	2C	ate fim de linha ou ":'"
CCCB: A2 00	1155	RETN	LDX	#500	
CCCD: 86 00	1156		STX	CHNAC	
CCCF: A0 00	1157		LDY	#500	
CCD1: 84 0E	1158		STY	ENDCHR	
CCD3: A5 0E	1159	RMI	LDA	ENDCHR	;Para contar pares de apostrofe
CCD5: A6 00	1160		LDX	CHNAC	
CCD7: 85 00	1161		STA	CHNAC	
CCD9: 86 0E	1162		STX	ENDCHR	
CCDB: 81 88	1163	RMC	LDA	(TXTPTR),Y	
CCDD: F0 08	1164		BEQ	RETB	;Se fim da linha,
CCDF: C5 0E	1165		CMP	ENDCHR	sair com Y=deslocamento
CCD1: F0 E4	1166		BEQ	RETB	
CCD3: C8	1167		IMY		
CCD4: C9 22	1168		CMP	B''''	
CCD6: 00 F3	1169		BNE	RMC	;Se nao for apostrofe,continuar
CCDB: F0 E9	1170		BEQ	RMI	;Trocar par e continuar
	1171				
CCDA: 68	1172	PULL3	PLA		
CCDB: 68	1173		PLA		
CCDC: 68	1174		PLA		
CCDE: 60	1175		RTS		
	1176				
CCDE: 20 A3 00	1177	IF	JSR	FRNEVL	
CCF1: 20 87 00	1178		JSR	CHGOT	
CCF4: C9 A8	1179		CMP	#got	
CCF4: F0 05	1180		BEQ	TRUE?	
CCF8: AF CA	1181		LDA	Rthen	
CCFA: 20 E8 01	1182		JSR	SYNCH	
CCFB: A5 90	1183	TRUE?	LDA	FAC	;Condicao verdadeira ou falsa?
CCFF: 00 05	1184		BNE	IFTRUE	;Se verdadeira,desviar
CCD1: 20 C8 CC	1185	RETN	JSR	RETN	;Saltar resto da linha
CCD4: F0 87	1186		BEQ	ADDDN	;Desviar sempre
	1187				
CCD6: 20 87 00	1188	IFTRUE	JSR	CHGOT	;Comando ou numero?
CCD9: 00 03	1189		BCC	JGCHD	;Caso comando
CCD8: 4C 63 CC	1190		JMP	GOTO	;Caso numero
CCDE: 4C 42 CB	1191	JGCHD	JMP	GCHD	;Executar o comando
	1192				
CCD1: 20 12 DA	1193	ONGOTO	JSR	GETBYT	;Obter numero em FAC+4



```

C014: 40      1194      PWA
C015: C9 00   1195      CNP #yosub
C017: F0 04   1196      BEQ OMCNT
C019: C9 A8   1197      GOT0?  CNP #yoto
C01B: 00 89   1198      BNE OSYNER
C01D: C6 A1   1199      OMCNT  DEC FAC+4      ;Chegou?
C01F: 00 04   1200      BNE KOKUM      ;Nao,continuar pesquisando
C021: 68      1201      PLA          ;Sim, obter comando
C022: 4C 44 C8 1202      JMP GOCK02     e executar
C025: 20 B1 00 1203      KOKUM  JSR CHRGET
C02B: 20 31 C0 1204      JSR LINGET
C02B: C9 2C   1205      CNP '#,'
C02B: F0 EE   1206      BEQ OMCNT
C02F: 68      1207      PLA          ;Nao encontrado,ignorar comando
C030: 40      1208      RET9  RTS
                                322      PUT PARTE16,01
                                31
C031: A2 00   12      LINGET  LDX #500      ;de numero ASCII
C033: 86 50   13      STX LINMM      a endereco HEX em LINMM
C035: 86 51   14      STX LINMM+1
C037: 80 F7   15      ASCHEX  BCS RET9      ;Fim ao encontrar primeiro
C039: E9 2F   16      SEC '#0'-1     caracter nao numerico
C03B: 85 00   17      STA CHARAC
C03B: A5 51   18      LDA LINMM+1
C03F: 85 5E   19      STA INDEX
C041: C9 19   110      CNP #5FA/10   ;Numero de linha muito grande?
C043: 80 04   111      BCS GOT0?     ;Sim,erro
C045: A5 50   112      LDA LINMM
C047: 0A      113      ASL
C04B: 26 5E   114      ROL INDEX
C04A: 0A      115      ASL
C04B: 26 5E   116      ROL INDEX
C04D: 65 50   117      ADC LINMM
C04F: 85 50   118      STA LINMM
C051: A5 5E   119      LDA INDEX
C053: 65 51   120      ADC LINMM+1
C055: 85 51   121      STA LINMM+1
C057: 06 50   122      ASL LINMM     ;Valor anterior *10
C059: 26 51   123      ROL LINMM+1
C05B: A5 50   124      LDA LINMM
C05D: 65 00   125      ADC CHARAC
C05F: 85 50   126      STA LINMM
C061: 90 02   127      BCC K0016
C063: E6 51   128      INC LINMM+1   ;Somado ao novo algarismo
C065: 20 B1 00 129      K0016  JSR CHRGET
C06B: 4C 37 C0 130      JMP ASCHEX
                                131
C06B: 20 03 00 132      LET  JSR PTRGET
C06E: 85 05   133      STA FORPNT
C070: 84 86   134      STY FORPNT+1
C072: A9 00   135      LDA #equal
C074: 20 E8 01 136      JSR SYNCR
C077: A5 12   137      LDA WALTYP+1
C079: 48      138      PWA
C07A: A5 11   139      LDA WALTYP
C07C: 48      140      PWA
C07D: 20 A3 00 141      JSR FINEVL

```

CD00:	68	142	PLA		
CD01:	2A	143	RDL	;Carry=sinal VMLTP	
CD02:	20	95	05	144 JSR CHEVAL	
CD03:	00	18	145 BNE LETSTR	;Se string de caracteres	
CD07:	68	146	PLA		
CD08:	10	12	147 LET2 BPL LETREAL		
CD0A:	20	8C	DE	148 JSR RND8	;Variavel inteira
CD0D:	20	26	D4	149 JSR AYINT	
CD10:	A0	00	150 LDY #00		
CD12:	A5	A0	151 LDA VPNT		
CD14:	91	85	152 STA (FORPNT),Y		
CD16:	C8	153 IMY			
CD17:	A5	A1	154 LDA VPNT+1		
CD19:	91	85	155 STA (FORPNT),Y		
CD1E:	60	156	RTS		
CD1C:	4C	41	DE	157 LETREAL JMP SETFOR	
CD1F:	68	158	LETSTR LETSTR PLA		
CD40:	A0	02	159 PUTSTR LDY #02		
CD42:	E1	A0	160 LDA (VPNT),Y		
CD44:	C5	70	161 CNP FRETOP+1		
CD46:	90	17	162 BCC COPSTR	;Nao esta na area de strings	
CD48:	00	07	163 BNE DESC?		
CD4A:	88	164	DEY		
CD4B:	E1	A0	165 LDA (VPNT),Y		
CD4D:	C5	6F	166 CNP FRETOP		
CD4F:	90	0E	167 BCC COPSTR		
CD81:	A4	A1	168 DESC? LDY VPNT+1	;Tem descritor?	
CD83:	C4	6A	169 CPY VARTAB+1		
CD85:	90	08	170 BCC COPSTR	;Sim,copiar	
CD87:	00	00	171 BNE MEMDESC		
CD89:	A5	A0	172 LDA VPNT		
CD8B:	C5	69	173 CNP VARTAB		
CD8D:	80	07	174 BCS MEMDESC		
CD8F:	A5	A0	175 COPSTR LDY VPNT	;Copiar somente o descritor	
CD91:	A4	A1	176 LDY VPNT+1		
CD93:	4C	9C	CD	177 JMP COPY	
CD94:	A0	00	178 MEMDESC LDY #00	;Criar novo descritor	
CD96:	E1	A0	179 LDA (VPNT),Y		
CD9A:	20	EF	84	180 JSR STR1M1	
CD9C:	A5	9C	181 LDA DSCPTR		
CD9F:	A4	80	182 LDY DSCPTR+1		
CD01:	85	A0	183 STA STRM1		
CD03:	B4	AC	184 STY STRM1+1		
CD05:	20	EE	06	185 JSR MOVING	
CD08:	A9	90	186 LDA #FAC		
CD0A:	A0	00	187 LDY #00		
CD0C:	85	9C	188 COPY STA DSCPTR		
CD0E:	B4	80	189 STY DSCPTR+1		
CD0D:	20	4F	D9	190 JSR FRETMS	
CD21:	A0	00	191 LDY #00		
CD23:	E1	9C	192 LDA (DSCPTR),Y		
CD27:	91	85	193 STA (FORPNT),Y		
CD29:	C8	194	IMY		
CD2A:	E1	9C	195 LDA (DSCPTR),Y		
CD2C:	91	85	196 STA (FORPNT),Y		
CD2E:	C8	197	IMY		
CD2F:	E1	9C	198 LDA (DSCPTR),Y		

```

CDF1: 91 85 199 STA (FOMPWT),Y
CDF3: 60 1100 RTS
      1101
CDF4: 20 62 CE 1102 PRSTRNG JSR STRPRT
CDF7: 20 87 00 1103 JSR CHRGET
CDFA: F0 24 1104 PRINT BEB CR00 ;Desviar se fim do comando
CDFC: F0 2F 1105 PRINT2 BEB RET10
CDFE: C9 C0 1106 CMP #tab
CE00: F0 3F 1107 BEB TABHERE
CE02: C9 C3 1108 CMP "#spc
CE04: 18 1109 CLC
CE05: F0 34 1110 BEB TABHERE
CE07: C9 2C 1111 CMP #' '
CE09: 18 1112 CLC
CEDA: F0 1C 1113 BEB TAB
CEDC: C9 38 1114 CMP #' ;'
CE0E: F0 44 1115 BEB NEXTCHR
CE10: 20 A3 00 1116 JSR FRMVAL ;Analisar formula
CE13: 24 11 1117 BIT VALTYP
CE15: 30 00 1118 BNE PRSTRNG ;Desviar se string
CEL7: 20 4E ED 1119 JSR FOUT ;Conv.numero em FAC a string
CE1A: 20 01 D7 1120 JSR STRLIT ;Criar descritor temporario
CE10: 4C F4 CD 1121 JNP PRSTRNG ;Apagar
      1122
CE20: A9 00 1123 CR00 LDA #500
CE22: 20 81 CE 1124 JSR OUTDO
CE25: A9 FF 1125 NEGATE EOR #FF
CE27: 60 1126 RET10 RTS
      1127
CE28: A5 24 1128 TAB LDA CH
CE2A: C9 18 1129 CMP #518
CE2C: 90 05 1130 BCC WCOLN
CE2E: 20 20 CE 1131 JSR CR00
CE31: D0 21 1132 BNE NEXTCHR ;Sempre
CE33: 69 10 1133 WCOLN ADC #510
CE35: 29 F0 1134 AND #5FD ;Tab 16,32
CE37: 85 24 1135 STA CH
CE39: 90 19 1136 BCC NEXTCHR ;Sempre
      1137
CE38: 08 1138 TABHERE PHP ;Restaurar casos SPC ou TAB
CE3C: 20 0F DA 1139 JSR GETBYTC
CE3F: C9 29 1140 CMP #' '
CE41: F0 03 1141 BEQ SPC?
CE43: 4C F1 D1 1142 JNP SYNERR
CE46: 28 1143 SPC? PLP
CE47: 90 07 1144 BCC TABIT ;Desviar se SPC
CE49: CA 1145 DEX
CE4A: 8A 1146 TXA
CE4B: E5 24 1147 SBC CH ;Calc.num.de espacos a mandar
CE4D: 90 05 1148 BCC NEXTCHR ;Caso negativo
CE4F: AA 1149 TAX
CE50: E8 1150 TABIT INX
CE51: CA 1151 WSPC DEX
CE52: D0 06 1152 BNE D0SPC
CE54: 20 81 00 1153 NEXTCHR JSR CHRGET ;Testar fim de comando
CE57: 4C FC CD 1154 JNP PRINT2
CE5A: 20 7C CE 1155 D0SPC JSR OUTSP

```

```

CE50: 00 F2 1156 BNE KOSPC ;Sempre
      1157
CESF: 20 01 07 1158 STROUT JSR STRLIT ;Imprimir string em (A,Y)
CE62: 20 1A 09 1159 STRPRT JSR FREFAC ;Obter pointer ao string
CE65: AA 1160 TAX ;Comprimento
CE66: A0 00 1161 LDY #500
CE68: EB 1162 INX
CE69: CA 1163 MOVAR DEY
CE6A: FD 88 1164 BEB RET10 ;Sair se fim de string
CE6C: 81 5E 1165 LDA (INDEX),Y
CE6E: 20 81 CE 1166 JSR OUTDO
CE71: C8 1167 IMY
CE72: C9 00 1168 CMP #500
CE74: 00 F3 1169 BNE MOVAR
CE76: 20 25 CE 1170 JSR NEGATE
CE79: 4C 69 CE 1171 JMP MOVAR
      1172
CE7C: A9 20 1173 OUTSP LDA #' '
CE7E: 2C 1174 HEX 2C
CE7F: A9 3F 1175 OUTRUES LDA #'?'
CE81: 09 80 1176 OUTDO ORA #500
CE83: C9 A0 1177 CMP #'*' ;Caracter de controle?
CE85: 90 02 1178 BCC SEMO ;Sim,desviar
CE87: 05 F3 1179 ORA ORMASK ;Converter ou nao mudar
CE89: 20 ED FD 1180 SEND JSR COUT
CE8C: 29 7F 1181 AND #57F
CE8E: 40 1182 PMA
CE8F: A5 F1 1183 LDA SPEEDZ
CE91: 20 AB FC 1184 JSR WAIT
CE94: 68 1185 PLA
CE95: 60 1186 RTS
      1187
CE96: A5 15 1188 INPUTERR LDA INPUTFLG
CE98: FD 12 1189 BEB RESPERR ;Se INPUT
CE9A: 30 04 1190 BNE READERR ;Se READ
CE9C: A0 FF 1191 LDY #FFF ;Se GET
CE9E: 00 04 1192 BNE ERLIN
CEA0: A5 78 1193 READERR LDA DATLIN
CEA2: A4 7C 1194 LDY DATLIN+1
CEA4: 85 75 1195 ERLIN STA CURLIN
CEA6: 84 74 1196 STY CURLIN+1
CEA8: 4C F1 01 1197 JMP SYNERR
CEAB: 68 1198 IMPERR PLA
CEAC: 2A 08 1199 RESPERR BIT ERRFLG
CEAE: 10 05 1200 BPL DONEENT
CEB0: A2 FE 1201 LDY #5FE ;Resposta errada
CEB2: 4C A4 E6 1202 JMP HMMLEERR
CEB5: A9 17 1203 DONEENT LDA #REDIG
CEB7: A0 00 1204 LDY #HREDIG
CEB9: 20 5F CE 1205 JSR STROUT
CEBC: A5 79 1206 LDA OLDTEXT
CEBE: A4 7A 1207 LDY OLDTEXT+1
CEC0: 85 88 1208 STA TXTPTR
CEC2: 84 89 1209 STY TXTPTR+1
CEC4: 60 1210 RTS
      1211
CEC5: 20 20 04 1212 GET JSR ERRDIR

```

```

CECB: A2 01 1213      LDX #IN+1      ;Similar input
CECA: A0 02 1214      LDY #1IN+1
CECC: A9 00 1215      LDA #500
CECE: 80 01 02 1216      STA IN+1
CED1: A9 40 1217      LDA #540      ;Ligar INPUTFLG
CED3: 20 10 0F 1218      JSR #ACINMP
CED6: 40      1219      RTS
      1220
CED7: C9 22 1221      INPUT CMP #'*'      ;Testar se veio string de
CED9: 00 0E 1222      BNE #OUT      ;pedido de input
CEDB: 20 A9 01 1223      JSR STRTXT
CEDC: A9 38 1224      LDA #';'
CEED: 20 EB 01 1225      JSR SYNCNR
CEEC: 20 62 CE 1226      JSR STPRPT
CEED4: 4C EC CE 1227      JMP DIR?
CEEY: 20 7F CE 1228      #OUT JSR OUTWUES      ;Nao veio string, imprimir??
CEEC: 20 20 D6 1229      DIR? JSR ERRODR
CEEF: A9 2C 1230      LDA #'.'
CEF1: 80 FF 01 1231      STA IN-1
CEF4: 20 18 C9 1232      JSR INLIN
CEF7: A0 00 02 1233      LDA IN
CEFA: C9 03 1234      CMP #503      ;CONTROL-C?
CEFC: 90 10 1235      BNE ZF
CEFE: 4C 80 C8 1236      JMP ERFLG?
      1237
CF01: 20 7F CE 1238      #XIN JSR OUTWUES
CF04: 4C 18 C8 1239      JMP INLIN
CF07: A6 70 1240      READ LDX #DATPTR
CF09: A4 7E 1241      LDY #DATPTR+1
CF0B: A9 98 1242      LDA #598
CF0D: 2C 1243      HEX 2C
CF0E: A9 00 1244      ZF LDA #500
CF10: 85 15 1245      #ACINMP STA INPUTFLG
CF12: B6 7F 1246      STX #MPTR
CF14: 84 80 1247      STY #MPTR+1
CF16: 20 03 03 1248      #XINP JSR PTRGET
CF19: 85 85 1249      STA #FORPNT
CF1B: 84 86 1250      STY #FORPNT+1
CF1D: A5 88 1251      LDA #TXPTR
CF1F: A4 89 1252      LDY #TXPTR+1
CF21: 85 87 1253      STA #TSPV
CF23: 84 88 1254      STY #TSPV+1
CF25: A6 7F 1255      LDX #MPTR
CF27: A4 80 1256      LDY #MPTR+1
CF29: 86 88 1257      STX #TXPTR
CF2B: 84 89 1258      STY #TXPTR+1
CF2D: 20 87 00 1259      JSR #CRG00T
CF30: 00 21 1260      BNE #INSTART
CF32: 24 15 1261      BIT #INPUTFLG
CF34: 90 11 1262      BVC #SND?      ;Desviar se nao for GET
CF36: 20 18 F0 1263      JSR #MA
CF39: 20 02 F8 1264      JSR #ND
CF3C: 29 7F 1265      AND #57F
CF3E: 80 00 02 1266      STA IN
CF41: A2 FF 1267      LDX #IN-1
CF43: A0 01 1268      LDY #1IN-1
CF45: 00 00 1269      BNE #STOP

```

		1270			
0F40:	30 7F	1271	SNOW?	BMI	FIMONTA
0F41:	20 7F CE	1272		JSR	OUTAKUS
0F4C:	20 01 CF	1273		JSR	WSTM
0F4F:	86 88	1274	STXP	STX	TXTPTR
0F51:	84 89	1275		STY	TXTPTR+1
0F53:	20 81 00	1276	INSTART	JSR	CHRGET
0F54:	24 11	1277		BIT	VALTYP
0F58:	10 31	1278		RPL	MURIN
0F5A:	24 13	1279		BIT	IMPFLG
0F5C:	50 09	1280		BVC	PUTCHR
0F5E:	E8	1281		IMX	
0F5F:	86 88	1282		STX	TXTPTR
0F61:	AF 00	1283		LDA	#500
0F63:	85 00	1284		STA	CHARAC
0F65:	F0 0C	1285		BEQ	PEMCHR
0F67:	85 00	1286	PUTCHR	STA	CHARAC
0F69:	C9 22	1287		CMF	#'"'
0F6B:	F0 07	1288		BEQ	PECHR
0F6D:	AF 3A	1289		LDA	#':'
0F6F:	85 00	1290		STA	CHARAC
0F71:	AF 2C	1291		LDA	#',"
0F73:	18	1292	PEMCHR	CLC	
0F74:	85 0E	1293	PECHR	STA	ENDCHR
0F76:	A5 88	1294		LDA	TXTPTR
0F78:	A4 89	1295		LDB	TXTPTR+1
0F7A:	AF 00	1296		ABC	#500
0F7C:	90 01	1297		BCC	SEP
0F7E:	C8	1298		ENI	
0F7F:	20 07 07	1299	SKP	JSR	STRLT2
0F82:	20 57 0A	1300		JSR	PSINT
0F85:	20 40 C0	1301		JSR	PUTSTR
0F88:	4C 9A CF	1302		JMP	WX
0F8B:	48	1303	MURIN	PHA	
0F8C:	AD 00 02	1304		LDA	IN
0F8F:	F0 30	1305		BEQ	IMPFIN
0F91:	68	1306	DATIN	PLA	
0F92:	20 44 0F	1307		JSR	FIN
0F95:	A5 12	1308		LDA	INTFLG
0F97:	20 88 C0	1309		JSR	LET2
0F9A:	20 87 00	1310	WX	JSR	CHRGOT
0F9D:	F0 07	1311		BEQ	SMPWT
0F9F:	C9 2C	1312		CMF	#',"
0FA1:	F0 03	1313		BEQ	SMPWT
0FA3:	4C 96 CE	1314		JMP	IMPTEER
		1315			
0FA6:	A5 88	1316	SMPWT	LDA	TXTPTR
0FA8:	A4 89	1317		LDB	TXTPTR+1
0FAA:	85 7F	1318		STA	IMPTR
0FAC:	84 80	1319		STY	IMPTR+1
0FAE:	A5 87	1320		LDA	TXPSV
0FB0:	A4 88	1321		LDB	TXPSV+1
0FB2:	85 88	1322		STA	TXTPTR
0FB4:	84 89	1323		STY	TXTPTR+1
0FB6:	20 87 00	1324		JSR	CHRGOT
0FB9:	F0 33	1325		BEQ	IMPDSNE
0FB8:	20 EA 01	1326		JSR	CHRCOR

;Desviar se nao for GET

;Saltar apostrofes se houver

;Vez de DATA?

;Sim,desviar

;Colocar numero FF em TXTPWT

;Colocar em VAR1

;Desviar se input terminado?

; ",no input?

;Erro mesao

;Se o comand.nao for executado,  
o progr.deve ter uma virgula

```

CFBE: 4C 16 CF 1327      JMP  KXIMP      ;Obter proximo input
CFCL: 45 15 1328  INPFIN  LDA  INPUTFLG
CFCC: 00 CC 1329      BNE  DATIN
CFCS: 4C AB CE 1330      JMP  IMPERR
1331
CFCB: 20 CB CC 1332  FINDATA JSR  DATAN      ;Obter desloc.do prox.comando
CFCA: 00 1333      INY
CFCC: AA 1334      TAX
CFCD: 00 12 1335      BNE  NOS       ;Desviar se ":"
CFCE: A2 24 1336      LDX  @#aDATA+MEMSERRO
CFD1: 00 1337      INY
CFD2: 81 80 1338      LDA  (TXTPTR),Y ;fim de programa?
CFD4: F0 5F 1339      BEQ  BERR      ;Sisterra
CFD6: 00 1340      INY
CFD7: 81 80 1341      LDA  (TXTPTR),Y ;Obter prox.numero de linha
CFD9: 85 7B 1342      STA  DATLIN
CFDB: 00 1343      INY
CFDC: 81 80 1344      LDA  (TXTPTR),Y
CFDE: 00 1345      INY
CFDF: 85 7C 1346      STA  DATLIN+1
CFE1: 81 80 1347  NOS    LDA  (TXTPTR),Y ;Obter prio. token do comando
CFE3: AA 1348      TAX
CFE4: 20 80 CC 1349      JSR  ADOON     ;Atualizar TXTPTR
CFE7: ED 83 1350      CPX  #data
CFE9: 00 00 1351      BNE  FINDATA   ;Loop ate encontrar DATA
CFEB: 4C 53 CF 1352  INPDOOM     LDA  IMPTR      ;Token DATA encontrado
CFEE: A5 7F 1353      LDX  IMPTR+1   ;Nao precisa mais de input
CFE0: A4 80 1354      LDY  IMPTR+1
CFE2: A6 15 1355      LDX  INPUTFLG
CFE4: 10 03 1356      BPL  HTD
CFE6: 4C 73 CB 1357      JMP  SETDA     ;Se veio de DATA
CFE9: A0 00 1358  HTD    LDY  #000
CFEB: 81 7F 1359      LDA  (IMPTR),Y ;input a mais?
CFED: F0 07 1360      BEQ  RET11
CFEF: A9 07 1361      LDA  #EXIG    ;Sem erro
0001: A0 00 1362      LDY  #EXIG
0003: 4C 5F CE 1363      JMP  STROUT
0006: 60 1364  RET11  RTS
1365      PUT  PARTE1H,01
1366
0007: 45 58 54 12  EXIG  ASC  'EXTRA (IGNORADO)'0000
000A: 52 41 20 49 47 4E 4F 52
0012: 41 44 4F 00 00
1367
0017: 52 45 44 14  REDIG  ASC  'REDIGITE'0000
001A: 49 47 49 54 45 00 00
1368
0021: 00 04 16  NEXT  BNE  VARXKT   ;Desviar se variavel
0023: A0 00 17      LDY  #000     especificada
0025: F0 03 18      BEQ  SKPV
0027: 20 03 03 19  VARXKT JSR  PTRGET   ;Achar pointer a variavel
002A: 85 85 110  SKPV  STA  FORPMT
002C: 04 86 111      STY  FORPMT+1
002E: 20 5A C6 112      JSR  GETFORPMT ;Achar dados do FOR
0031: F0 04 113      BEQ  GOTFOR   na pilha
0033: A2 00 114      LDX  #KSEFOR+MEMSERRO
0035: F0 69 115  GERR  BEQ  JERROR   ;Sempre

```

```

0027: 9A    116  G0TFOR  TXS          ;pontar a pilha
0028: E9    117          INX
0029: E9    118          INX
003A: E9    119          INX
003B: E9    120          INX
003C: BA    121          TXA          ;byte de baixa orden
003D: E9    122          INX          ;do valor de STEP
003E: E9    123          INX
003F: E9    124          INX
0040: E9    125          INX
0041: E9    126          INX
0042: E9    127          INX
0043: 86 60  128          STX DEST      ;byte de baixa orden
0045: A0 01  129          LDY #001     do endereco da variavel FOR
0047: 20 13 DE 130        JSR R0A#R      ;STEP em FAC
004A: BA    131          TSX
004B: 80 09 01 132        LDA STACK+9,X
004E: 85 A2  133          STA FACSW
0050: A5 85  134          LDA FORPMT
0052: A4 86  135          LDY FORPMT+1
0054: 20 08 DA 136        JSR FADD      ;Somar ao valor de FOR
0057: 20 41 DE 137        JSR SETFOR   ;Devolver novo valor
005A: A0 01  138          LDY #001
005C: 20 CE DE 139        JSR FCOMP2   ;Comparar com valor final
005F: BA    140          TSX
0060: 38    141          SEC
0061: FD 09 01 142        SBC STACK+9,X
0064: F0 17  143          BEQ ENDFOR   ;Desviar se FOR terminado
0066: 80 0F 01 144          LDA STACK+9,X ;Nao, preparar numero
0069: 85 75  145          STA CURLIN   da linha de FOR
006B: 80 10 01 146          LDA STACK+$10,X
006E: 85 76  147          STA CURLIN+1
0070: 80 12 01 148          LDA STACK+$12,X;E posicionar TXTPTR logo apos
0073: 85 88  149          STA TXTPTR   instrcao FOR
0075: 80 11 01 150          LDA STACK+$11,X
0078: 85 89  151          STA TXTPTR+1
007A: 4C EC DA 152        G0NEMST     JMP NEMSTT
007D: BA    153          ENDFOR      TXA
007E: 69 11  154          ADC #61      ;Ligar CARRY
0080: A6    155          TAX          ;Cancelar FOR sabindo
0081: 9A    156          TXS          ;a pilha de $12
0082: 20 B7 00 157        JSR CHRGOT
0085: C9 2C  158          CMP #' '     ;Outra variavel no NEMST?
0087: D0 F1  159          BNE G0NEMST
0089: 20 81 00 160        JSR CHRGET
008C: 20 27 00 161        JSR WRRGT    ;Mao retorna
162 *****
163 * *
164 * Avaliacao de formulas *
165 * *
166 * Localizacao dos pointers *
167 * *
168 * E tratamentos de strings *
169 * *
170 *****
008F: 20 A3 00 171        FRMUM     JSR FRMVL
0092: 18    172        CHKUM     CLC

```



```

0093: 24 373 HEX 24
0094: 38 374 CNKSTR SEC
0095: 24 11 375 CHKVAL BIT VALTYP
0097: 30 03 376 BND CV2
0099: 80 03 377 BCS KISHTCH
009E: 60 378 RETL2 RTS
009C: 80 FD 379 CV2 BCS RETL2
009E: A2 64 380 KISHTCH LDI RINDCMP-NEMSEERR0
00A0: 4C 01 C7 381 JERROR JMP ERROR
382
383 #Rotina principal de avaliacao de formulas
384 #Na entrada TXTPTR aponta ao primeiro caracter da formula
385
00A3: A6 88 386 FRMENV LDX TXTPTR
00A5: 00 02 387 BNE N01
00A7: C6 89 388 DEC TXTPTR+1
00A9: C6 88 389 N01 DEC TXTPTR
00AB: A2 00 390 LDI #500
00AD: 24 391 DFB $24 ;Prioridade inicial
00AE: 48 392 FEVLOOP PHA ;Por na pilha ultimo CPRTYP
00AF: BA 393 TXA
00B0: 48 394 PHA ;E prioridade
00B1: A9 01 395 LDA #501
00B3: 20 C5 C6 396 JSR CHMEM ;Testar a pilha != $28
00B6: 20 88 01 397 JSR GETVAL ;Obter valor do descritor
00B9: A9 00 398 LDA #500 de string em TXTPTR
00BB: 85 89 399 STA CPRTYP
00BD: 20 87 00 399B FRMENV2 JSR CHRGDT
00CD: 38 399C CPROM SEC
00C1: E9 CF 399D SBC #5CF ;Token )
00C3: 90 17 399E BCC CHKTYP
00C5: C9 03 399F CMP #500 you =, (
00C7: 80 13 3999 BCS CHKTYP ;Nao, desviar
00C9: C9 01 399A CMP #501
00CB: 2A 399B BDL
00CC: 49 01 399C EOR #501
00CE: 45 89 399D EOR CPRTYP ;Ligar bits de CPRTYP=000000 )=(
00D0: C3 89 399E CMP CPRTYP
00D2: 90 61 399F BCC SHTXERR
00D4: 85 89 3999 STA CPRTYP
00D6: 20 81 00 3999B JSR CHGET ;Outro operador?
00D9: 4C C0 00 3999C JMP CPROM ;Testar de novo (,+,)
00DC: A6 89 3999D CHKTYP LDX CPRTYP
00DE: D0 2C 3999E BNE COMPARE ;Desviar se (,+,)
00E0: 80 78 3999F BCS NOTMATH ;Desviar se proximo token ) *(
00E2: 69 07 39999 ADC #5CF+plus
00E4: 90 77 3999A BCC NOTMATH ;Desviar se proximo token ( "+
00E6: 65 11 3999B ADC VALTYP ;"+ e ultimo resultado string?
00E8: D0 80 3999C BNE ARITH ;Nao, desviar
00EA: 4C 81 D8 3999D JMP CAT ;Sim concatenar
00ED: 69 FF 3999E ARITH ADC #5FF ;A contem deslocamento do "+"
00EF: 85 5E 3999F STA INDEX
00F1: DA 39999 ABL
00F2: 65 5E 3999A ADC INDEX ;Vezes 3
00F4: A8 3999B TAY
00F5: 68 3999C PREFTEST PLA ;Obter ultima prioridade
00F6: 09 92 C3 3999D CMP #MATHBL,Y ;Comparar com priorid. corrente

```

```

DOFF: 80 47 >130      BCS DOWTH ;Se priorid. corrente, executar
DOFB: 20 92 D0 >131   JSR CHKNUM ;0 ultimo result. foi numero?
DOFE: 48              >132  XNOP  PNA
DOFF: 20 25 D1 >133   SAMP  JSR PSHMAD ;Guardar operacao na pilha
D102: 68              >134   PLA
D103: A4 87 >135     LDY  LASTOP
D105: 10 17 >136     BPL  PREFNC ;Desviar se ha mais formula
D107: A6              >137   TAX
D108: F0 56 >138     BEB  GOEX ;Sem oper. matem. na forma. sair
D10A: 00 5F >139     BNE  DOWATH ;Executar ultima operacao
>140
D10C: 46 11 >141     COMPARE LSR  WALTYP ;Permitir comparacao de string
D10E: 0A >142       TXA
D10F: 2A >143       ROL
D110: A6 88 >144     LDY  TXTPTR ;Fazer CPRTYP=0000 => C com C-
D112: D0 02 >145     BNE  MD2 ;carry do ultimo
D114: C6 8F >146     DEC  TXTPTR+1 teste de caracteres
D116: C6 88 >147     MD2   DEC  TXTPTR
D118: A0 18 >148     LDY  #PLUS-MATHBL ;Forcar uso de POSOP para
D11A: 05 89 >149     STA  CPRTYP ; as 3 comparacoes
D11C: 00 07 >150     BNE  PREFTEST ;Sempre
>151
D11E: D9 92 C3 >152   PREFNC CMP  MATHBL,Y
D121: 80 48 >153     BCS  DOWATH ;Executar se prioridade correta
D123: 90 D9 >154     BCC  XNOP ;Obter mais formula
>155
D125: 8F 94 C3 >156   PSHMAD LDA MATHBL+2,Y
D128: 48 >157       PNA ;Colocar endereco da rotina
D129: 8F 93 C3 >158   LDA MATHBL+1,Y ;matematica na pilha
D12C: 48 >159       PNA
D12D: 20 38 D1 >160   JSR  PSHF ;Retornar na JMP(INDEX)
D130: A5 89 >161     LDA  CPRTYP
D132: 4C AE D0 >162   JMP  FEVL00P
>163
D135: 4C F1 D1 >164   SWTXERR JMP  SYMERR
>165
D138: A5 A2 >166     PSHF  LDA  FACSGN ;Obter FACSGN para coloca-lo
D13A: BE 92 C3 >167   LDY  MATHBL,Y ;na pilha
>168
>169 #Preparar INDEX para return e colocar FAC na pilha
>170 # A contem FACSGN ou -1,0,1 se vier de STEP
>171
D130: A8 >172     PSHFAC  TAY ;Chamada por STEP
D13C: 68 >173     PLA ;Tirar da pilha ender. de volta
D13F: 85 5E >174     STA  INDEX ;Coloca-lo em INDEX
D141: EA 5E >175     JMC  INDEX ;Esta rotina supoe que o ender.
D143: 68 >176     PLA ;de volta nao esta num limite
D144: 05 5F >177     STA  INDEX+1 ;de pagina
D146: 98 >178     TYA
D147: 48 >179     PNA ;Colocar FACSGN na pilha
D148: 20 9C DE >180   PUSHFAC JSR  INDB ;chamado por FOR
D148: A5 A1 >181     LDA  FAC+4
D148: 48 >182     PNA
D14E: A5 A0 >183     LDA  FAC+3
D150: 48 >184     PNA
D151: A5 9F >185     LDA  FAC+2
D153: 48 >186     PNA

```

```

0154: A5 9E 1187 LDA FAC+1
0156: 48 1188 PMA
0157: A5 9D 1189 LDA FAC
0159: 48 1190 PMA
015A: AC SE 00 1191 JMP (INDEX) ;Equivalente a RTS
                                324 PUT PARTEII_01
                                31
015D: A0 FF 32 NOTNATH LDT #FFF ;Preparar saída
015F: 48 33 PLA
0160: F0 23 34 GOEX BEB EXIT ;Sem funcao antes a exec., sair
0162: C9 64 35 DINTH CMP #64 ;Era (+)?
0164: F0 03 36 BEB DINTH ;Sim, permitir compar. de string
0166: 20 92 00 37 JSR DINTHM
0169: 84 87 38 DINTH STY LASTOP
                                39
                                310 *Tirar numero ponto flutuante da pilha,
                                311 *colocar em ARG e ir a rotina automatica
                                312 *via RTS (o endereço foi colocado na pilha):
                                313 *(Observar que as rotinas ( = ) todas vao a PDSMP).
                                314
016B: 68 115 DONATH PLA
016C: 4A 116 LSR ;Restaurar carry
016D: 85 16 117 STA CP#WSK ;00000)=(
016F: 48 118 PLA
0170: 85 A5 119 STA ARG
0172: 68 120 PLA
0173: 85 A6 121 STA ARG+1
0175: 68 122 PLA
0176: 85 A7 123 STA ARG+2
0178: 68 124 PLA
0179: 85 A8 125 STA ARG+3
017B: 68 126 PLA
017C: 85 A9 127 STA ARG+4
017E: 68 128 PLA
017F: 85 AA 129 STA ARG+5
0181: 45 A2 130 EOR FACSDH
0183: 85 AB 131 STA S#MCP#
0185: A5 9D 132 EXIT LDA FAC ;Desviar a rotina com status
0187: 60 133 RTS ;posicionado por FAC
                                134
                                325 *Obter valor da variavel, funcao ou numero apos TXTP#,
                                326 *ou apontar ao descritor de string se for string
                                327 *e coloca-lo em FAC. Esta rotina tambem avalia expressoes
                                328 *entre parenteses por chamada recursiva a FRNEVL.
                                329 *E a rotina central de FRNEVL.
                                330
018B: A9 00 341 GETVAL LDA #000
018A: 85 11 342 STA VALTTP
018C: 20 B1 00 343 SKIP JSR CHRGET
018F: 80 03 344 BCS V#?
0191: 4C 64 DF 345 NUMBER JMP FTH ;Numerico
0194: 20 97 03 346 V#? JSR ISLET# ;E variavel?
0197: 80 64 347 BCS V#L
0199: C9 2E 348 CMP #'.'
019B: F0 F4 349 BEB NUMBER
019D: C9 C9 350 CMP #lines
019F: F0 55 351 BEB #N#

```

```

00A1: C9 C8 152      CMP #plus
00A3: F0 E7 153      BEB SKIP
00A5: C9 22 154      CMP #""
00A7: 00 0F 155      BNE NOT?
00A9: A5 88 156      STRTXT LDA TXTPTR ;String explicitos construir
00AB: A4 89 157      LDY TXTPTR+1 ;descriptores
00AD: A9 00 158      ADC #500
00AF: 90 01 159      BCC ST1
00B1: C8 160      IMY
00B2: 20 01 07 161     ST1 JSR STRLIT
00B5: 4C 57 0A 162     JMP POINT ;Obter pointer a descrit ou FAC
163
00B8: C9 C6 164      NOT?  CMP #not
00BA: 00 10 165      BNE FN?
00BC: A0 18 166      LDY #UNOT-MATHTEL
00BE: 00 38 167      BNE ENUL ;Sempre
168
00C0: A5 90 169      EMNOP LDA FAC ;Esta rotina so pode
00C2: 00 03 170      BNE NOT? ;ser chamada por NOT atraves
00C4: A0 01 171      LDY #501 ;do desvio anterior a ENUL
00C6: 2C 172      HEX 2C
00C7: A0 00 173      NOT? LDY #500
00C9: 4C 18 0A 174     JMP SEMFLT
175
00CC: C9 C2 176      FN?  CMP #fn
00CE: 00 03 177      BNE SGN?
00D0: 4C 6E 0A 178     JMP FUNCT
179
00D3: C9 02 180      SGN? CMP #sgn
00D5: 90 03 181      BCC PARCHK
00D7: 4C 34 02 182     JMP UNARY
183
00DA: 20 E3 01 184     PARCHK JSR CHKOPN ;Ha ou "(" ou TXTPTR?
00DB: 20 A3 00 185     JSR FRANEL ;Sim, avaliar e testar ")"
00ED: A9 29 186      CHKCLS LDA #'')'
00EF: 2C 187      HEX 2C
00E3: A9 28 188      CHKOPN LDA #'('
00E5: 2C 189      HEX 2C
00E6: A9 2C 190      CHRCOM LDA #', ' ;',', ou TXTPTR?
00E8: A0 00 191      SYNCHR LDY #500
00EA: D1 88 192      CMP (TXTPTR),Y
00EC: 00 03 193      BNE SYNERR
00EE: 4C B1 00 194     JMP CHRGET ;Se OK, obter prox caract e ret
195
00F1: A2 0C 196      SYNERR LDY #SINTERR-NESENERR
00F3: 4C D1 C7 197     JMP ERROR
198
00F6: A0 15 199      NIH  LDY #NUMS-MATHTEL
00F8: 68 100      ENUL  PLA
00F9: 68 101      PLA
00FA: 4C FF 00 102     JMP SAVOP
103
00FD: 20 03 03 104     VARL  JSR PTRGET
0208: 05 A0 105      STA UPNT
0202: 84 A1 106      STY UPNT+1
0204: A6 11 107      LDY VALTYP ;E string?
0206: F0 05 108      BEB URI ;Nao, desviar

```

0208:	A2 00	>109	LOX	MS00	
020A:	B6 AC	>110	STX	EXTRAFAC	
020C:	60	>111	RTS		
		>112			
0200:	A6 12	>113	WR1	LOX	INTFLG ;Variavel inteira ?
020F:	10 00	>114	BPL	VR2	;Nao, desviar
0211:	A0 00	>115	LOY	MS00	
0213:	B1 A0	>116	LDA	(VPNT),Y	;Obter byte de alta ordem
0215:	AA	>117	TAX		
0216:	C0	>118	INY		
0217:	B1 A0	>119	LDA	(VPNT),Y	;Obter byte de baixa ordem
0219:	AB	>120	TAY		
021A:	BA	>121	TXA		
0218:	4C 0C 06	>122	JMP	OSWAYF	;Transformar em ponto flutuante
021E:	4C 13 0E	>123	WR2	JMP	NOVFN ;Novo (A,Y) a FAC
		>124			
0221:	20 B1 00	>125	SCREEN	JSR	CHRGET
0224:	20 A0 E3	>126	JSR	PLOTFRM	
0227:	0A	>127	TXA		
0228:	A4 F0	>128	LOY	FIRST	
022A:	20 69 F0	>129	JSR	SCRN	
022D:	AB	>130	TAY		
022E:	20 18 D6	>131	JSR	SGHFLT	
0231:	4C E0 D1	>132	JMP	CHKCLS	
		>133			
		>134			#Processar operadores unarios (fascos)
		>135			
0234:	C9 07	>136	UNARY	OMP	Msca ;Nao unario, especial
0236:	F0 E9	>137	BEQ	SCREEN	
0238:	0A	>138	ASL		
0239:	40	>139	PHA		
023A:	AA	>140	TAX		
023B:	20 B1 00	>141	JSR	CHRGET	
023E:	ED CF	>142	CPX	!leftstr#2-1	
0240:	90 20	>143	BCC	NOTinstr	;Desv se nao for oper de string
0242:	20 E3 D1	>144	JSR	CHROPN	;Testar "("
0245:	20 A3 00	>145	JSR	FRNEVL	;Processar concatenacao,...
0248:	20 E6 D1	>146	JSR	CHRDOM	
024B:	20 94 00	>147	JSR	CHRSTR	;Certificar que e string
024E:	68	>148	PLA		
024F:	AA	>149	TAX		;Obter pointer a rotina
0250:	A5 A1	>150	LDA	VPNT+1	
0252:	40	>151	PHA		
0253:	A5 A0	>152	LDA	VPNT	
0255:	40	>153	PHA		
0256:	BA	>154	TXA		
0257:	40	>155	PHA		;Coloca-lo de volta na pilha
0258:	20 12 0A	>156	JSR	GETBYT	;Obter primeiro parametro em X
025B:	40	>157	PLA		
025C:	AB	>158	TAY		;Y apontando a rotina
025D:	0A	>159	TXA		
025E:	40	>160	PHA		;Por primeiro paramet na pilha
025F:	4C 67 02	>161	JMP	GOROUT	;Exec rotina de string requer
		>162			
0262:	20 0A D1	>163	NOTinstr	JSR	PANCHK ;Testar sintaxe e avaliar arg.
0265:	40	>164	PLA		;Obter token #2
0266:	40	>165	TAY		

```

0267: 8F 82 C2 1166 GOROUT LDA UNFAC-S44,Y;S44 = sign2
026A: 85 91 1167 STA JHPADR5+1
026C: 8F 83 C2 1168 LDA UNFAC-S43,Y
026F: 85 92 1169 STA JHPADR5+2
0271: 2D 90 00 1170 JSR JHPADR5 ;Mas retorna para LEFTS, KIDS,
0274: 4C 92 00 1171 JMP CHKMLM RIGHTS
1172
0277: A5 A5 1173 OR LDA ARG
0279: 05 90 1174 ORA FAC
027B: 00 08 1175 BNE TRUE
027D: A5 A5 1176 AND LDA ARG
027F: F0 04 1177 BEQ FALSE
0281: A5 90 1178 LDA FAC
0283: 00 03 1179 BNE TRUE
0285: A0 00 1180 FALSE LDY #500
0287: 2C 1181 HEX 2C
0289: A0 01 1182 TRUE LDY #501
028A: 4C 18 D6 1183 JMP SCNFLT
325 PUT PARTE1,J,01
11
12 *Rotina coeas as comparacoes (,=,)
13
0290: 20 95 00 14 P000P JSR CHKVAL
0293: 80 13 15 BCS STORCP ;Desviar se strings
0295: A5 AA 16 LDA ARGSSN ;Se ARGSSN positivo, eliminar
0298: 09 7F 17 ORA #57F bit de alta ordem de ARG1
029A: 25 A6 18 AND ARG+1
029C: 85 A6 19 STA ARG+1
029E: A9 A5 110 LDA ARG
029F: A0 00 111 LDY #500
02A1: 20 CC 0E 112 JSR FC00P ;Retornar A= -1,0,1 conforme
02A3: AA 113 TAX ;ARG (,=,) FAC
02A5: 4C D8 02 114 JMP MUNCMP
02A8: A9 00 115 STORCP LDA #500
02AA: 85 11 116 STA VALTYP
02AC: C6 8F 117 DEC CNTTYP
02AE: 20 1A 0F 118 JSR FREFAC
02B0: 85 90 119 STA FAC ;Tamanho do string
02B2: 86 9C 120 STY FAC+1
02B4: A5 A6 121 LDA ARG+2
02B6: A4 A9 122 LDY ARG+4
02B8: 20 1E 0F 124 JSR FRETYP
02BA: 86 A8 125 STX ARG+2
02BC: 84 A9 126 STY ARG+4
02BE: A6 127 TAX ;Tamanho do string ARG
02C0: 38 128 SEC
02C2: E5 90 129 SBC FAC ;Fazer X= menor tamanho
02C4: F0 08 130 BEQ SFS
02C6: A9 01 131 LDA #501
02C8: 90 04 132 BCC SFS
02CA: A6 90 133 LDY FAC
02CC: A9 FF 134 LDA #5FF
02CE: 85 A2 135 SFS STA FACSSN ;Indicar o mais curto
02CF: A0 FF 136 LDY #5FF
02D1: E8 137 INX
02D3: C8 138 CNLOOP INY

```

0203:	CA	139		DEX	
0204:	00 07	140		BNE DOCHP	
0204:	A6 A2	141		LDX FACSEM	;Se iguais ate agora, decidir
0204:	30 0F	142	MUNCHP	BRI CWPONE	pele tamanho
0204:	18	143		CLC	
0208:	90 0C	144		BCC CWPONE	
0208:	81 A8	145	DOCHP	LDA (ARG+3),Y	
020F:	01 9E	146		CMP (FAC+1),Y	
02E1:	F0 0F	147		BEQ CNFLOOP	
02E3:	A2 FF	148		LDX #SFF	
02E3:	80 02	149		BCC CWPONE	
02E7:	A2 01	150		LDX #S01	
02E9:	EB	151	CWPONE	INX	;Converter FF,0,1 & 1,2,4
02EA:	8A	152		TXA	
02EB:	2A	153		ROL	
02EC:	25 14	154		AND CPMASK	;00000 >=1
02EE:	F0 02	155		BEQ JF	;Se nao bate, falso
02F0:	A9 01	156		LDA #S01	;Pelo menos um bate, verdadeiro
02F2:	4C A0 0E	157	JF	JMP FLOWT	
02F5:	60	158		RTS	
		159			
02FA:	20 E6 01	160	KIDDIN	JSR CHRCOM	
02F9:	AA	161	DIM	TAX	
02FA:	20 08 03	162		JSR PTRGET2	;Criar e zerar arrays
02FD:	20 07 00	163		JSR CHRGOT	
0300:	00 F4	164		BNE KIDDIN	
0302:	60	165		RTS	
		166			
0303:	A2 00	167	PTRGET	LDX #S00	
0305:	20 07 00	168		JSR CHRGOT	;Obter nome de variavel
0308:	86 10	169	PTRGET2	STX INTFLG	;X e nome da var.se vem de DIM
030A:	85 81	170	PTRGET3	STA VARNAM	;Ponto de estrada de FAC
030C:	20 87 00	171		JSR CHRGOT	
030F:	20 97 03	172		JSR ISLETC	;E uma letra?
0312:	80 03	173		BCC NAMOK	;Sim, desviar
0314:	4C F1 01	174	BADNAM	JMP SYERR	;Nao, erro
0317:	A2 00	175	NAMOK	LDX #S00	
0319:	86 11	176		STX VALTYP	
0318:	86 12	177		STX INTFLG	
0318:	4C 21 03	178		JMP MORNAM	
0320:	00	179		BRK	
0321:	20 81 00	180	MORNAM	JSR CHRGET	;Segundo caract do nome da var.
0324:	90 05	181		BCC GTLT	;Desviar se numerico
0326:	20 97 03	182		JSR ISLETC	;E alfa?
0329:	90 08	183		BCC STRNG?	;Nao, desviar
032B:	AA	184	GTLT	TAX	;Guard seq. caract do nome em X
032C:	20 81 00	185	BYPASS	JSR CHRGET	;Achar end do nome da variavel
032F:	90 F8	186		BCC BYPASS	;Loop se numerico
0331:	20 97 03	187		JSR ISLETC	
0334:	80 F6	188		BCC BYPASS	;Ou alfa
0336:	C9 24	189	STRNG?	CMP #'S'	;Ligar indicadores de tipo
0338:	D0 06	190		BNE INTVAR?	de variaveis
033A:	A9 FF	191		LDA #SFF	
033C:	85 11	192		STA VALTYP	;Indicar string
033E:	00 10	193		BNE KIN	;Sempre
0340:	C9 25	194	INTVAR?	CMP #'Z'	
0342:	D0 13	195		BNE SCDOCH	

0344:	85 14	196	LDW	SUBFLG	;Variavel inteira permitida?
0346:	30 CC	197	BNI	VARNAM	;Nao, erro
0348:	89 80	198	LDW	MS00	
034A:	85 12	199	STA	INTFLG	;Indicar inteiro
034C:	05 81	1100	ORA	VARNAM	
034E:	85 81	1101	STA	VARNAM	;Ligar bit de alta ordens do
0350:	8A	1102	MDN	TXA	;primeiro caracter do nome
0351:	09 80	1103	ORA	MS00	;Ligar bit de alta ordens do
0353:	8A	1104	TAX		;segundo caracter do nome
0354:	20 81 00	1105	JSR	CHGET	
0357:	86 82	1106	SCDCH	STX VARNAM+1	;Segundo caracter do nome
0359:	38	1107	SEC		;de variavel
035A:	05 14	1108	ORA	SUBFLG	;Indices permitidos e array?
035C:	E9 28	1109	SBC	#'C'	
035E:	00 03	1110	BNE	ESB	;Nao, desviar
0360:	4C 38 04	1111	JARY	JMP ARRAY	
0363:	24 14	1112	BSB	BIT SUBFLG	
0365:	30 02	1113	BNI	USEARCH	;Desv se vem de FOR, DEF ou FN
0367:	70 F7	1114	BVS	JART	;Desv se chamada por GETARYPT
0369:	89 00	1115	VSEARCH	LDW MS00	
036B:	85 14	1116	STA	SUBFLG	
036D:	85 69	1117	LDW	VARTAB	;Inicializar pointer a variavel
036F:	A6 6A	1118	LDX	VARTAB+1	
0371:	80 00	1119	LDY	MS00	
0373:	86 9C	1120	MXVAR	STX LOWTR+1	
0375:	85 98	1121	MVL	STA LOWTR	
0377:	E4 6C	1122	CPX	ARTTAB+1	;Fim das variaveis simples?
0379:	80 04	1123	BNE	NV2	;Nao, continuar
037B:	C5 68	1124	CMP	ARTTAB	
037D:	F0 22	1125	BEQ	NOTFMD	;Sim, criar uma?
037F:	85 81	1126	NV2	LDW VARNAM	
0381:	01 98	1127	CMP	(LOWTR),Y	
0383:	00 08	1128	BNE	KOPTR	;Desviar se nao for esta
0385:	85 82	1129	LDW	VARNAM+1	
0387:	C8	1130	JNY		
038B:	01 98	1131	CMP	(LOWTR),Y	
038A:	F0 6C	1132	BEQ	SETVPNT	;Desviar se encontrada
038C:	88	1133	DEY		
038D:	18	1134	KOPTR	CLC	
038E:	85 98	1135	LDW	LOWTR	
0390:	69 07	1136	ADC	MS07	
0392:	90 E1	1137	BCC	MVL	
0394:	EB	1138	INX		
0395:	00 0C	1139	BNE	MXVAR	;Sempre
		1140			
		1141			*Verificar se e letra de A a Z. Ligar carry se for.
		1142			*desligar caso contrario
		1143			
0397:	C9 41	1144	ISLETC	CMP #'A'	
0399:	90 05	1145	BCC	RTHL	
039B:	E9 58	1146	SBC	#'Z'+1	
039D:	38	1147	SEC		
039E:	E9 85	1148	SBC	MS100-'Z'+51	;Obter A original
03A0:	60	1149	RTHL	RTS	
		1150			
03A1:	68	1151	NOTFMD	PLA	;Obter baixa ord do end de cham
03A2:	48	1152	PHA		;Voltar a pilha



```

03A3: C9 FC 1153      CMP #5FC      ;Chamado por VARL?
03A5: 00 0F 1154      BNE MEMVAR   ;Nao, desviar
03A7: 8A          ISX
03A8: 80 02 01 1156      LDA STACK+2,X ;Alter alta ord do endereco de
03AB: C9 01 1157      CMP #)VARL+2  chamada de VARL?
03AD: 00 07 1158      BNE MEMVAR   ;Nao, desviar
03AF: A9 84 1159      LDA #)WOBRK   ;Se nao e atribuicao, eliminar
03B1: A0 03 1160      LDY #)WOBRK   ; endereco das variaveis de
03B3: 60 1161      RTS          ; maneira que volte valor 0
03B4: 00 1162      TWORRE BRK
03B5: 00 1163      BRK
1164
1165      ;Mover arrays para dar espaco a novas variaveis
1166
03B6: A5 68 1167      MEMVAR LDA ARYTAB
03B8: A4 6C 1168      LDY ARYTAB+1
03BA: 05 98 1169      STA LOWTR
03BC: 84 9C 1170      STY LOWTR+1
03BE: A5 60 1171      LDA STREND
03C0: A4 6E 1172      LDY STREND+1
03C2: 85 96 1173      STA HIGHTR
03C4: 84 97 1174      STY HIGHTR+1
03C6: 1E 1175      CLC
03C7: 6F 07 1176      ADC #507      ;Preparar para mover 7 bytes
03C9: 90 01 1177      BCC MM
03CB: C8 1178      INY
03CC: 85 94 1179      MAI STA HIGH8G   ;Preparar endereco de destino
03CE: 84 95 1180      STY HIGH8G+1
03D0: 20 82 CA 1181      JSR BLTU      ;Mover
03D3: A5 94 1182      LDA HIGH8D
03D5: A4 95 1183      LDY HIGH8D+1
03D7: C8 1184      INY          ;BLTU posiciona este valor
03D8: 05 68 1185      STA ARYTAB    com 1 a menos
03DA: 84 6C 1186      STY ARYTAB+1
03DC: A0 00 1187      LDY #500
03DE: A5 81 1188      LDA VARMMN
03E0: 91 98 1189      STA (LOWTR),Y ;Guardar nome de variavel
03E2: C8 1190      INY
03E3: A5 82 1191      LDA VARMMN+1
03E5: 91 98 1192      STA (LOWTR),Y
03E7: A9 00 1193      LDA #500     ;Colocar valor em 0
03E9: C8 1194      INY
03EA: 91 98 1195      STA (LOWTR),Y
03EC: C8 1196      INY
03ED: 91 98 1197      STA (LOWTR),Y
03EF: C8 1198      INY
03F0: 91 98 1199      STA (LOWTR),Y
03F2: C8 1200      INY
03F3: 91 98 1201      STA (LOWTR),Y
03F5: C8 1202      INY
03F6: 91 98 1203      STA (LOWTR),Y
03F8: A5 98 1204      SETWPT LDA LOWTR
03FA: 1E 1205      CLC
03FB: 6F 02 1206      ADC #502
03FD: A4 9C 1207      LDY LOWTR+1
03FF: 90 01 1208      BCC SUP
0401: C8 1209      INY

```

```

0402: 85 83 1210 SWP STA VARPNT ;Apont ao prim byte do valor
0404: 84 84 1211 STY VARPNT+1
0406: 60 1212 RTS
326 PUT PARTE11,D1
11
0407: A5 0F 12 GETARY LDA WURD1H ;Obter numero de dimensoes
0409: 0A 13 GETARY2 ASL ;vezes 2
040A: 6F 05 14 ADC WWS5 ;+5(acao, desloc., num.de dia.)
040C: A5 98 15 ADC LONTR ;Somar ao pointer da variavel
040E: A4 9C 16 LDT LONTR+1
0410: 90 01 17 BCC ED
0412: 08 18 INT
0413: 85 94 19 80 STA ARYPNT ;Agora apontar ao primeiro
0415: 84 95 110 STY ARYPNT+1 ;descriptor no array
0417: 60 111 RTS
112
0418: 90 80 00 113 NEGMMM HEX 90800000 ;-32768
0418: 00 114
041C: 20 81 00 115 MAKINT JSR CHRGET
041F: 20 8F 00 116 JSR FRMMUN
0422: A5 A2 117 MKINT LDA FACSGR ;Erro se negativo
0424: 30 00 118 BRT K11
0426: A5 90 119 AYINT LDA FAC
042B: C9 90 120 CMP WSP0 ;Abs(2*15)?
042A: 90 09 121 BCC K12 ;Sim, desviar
042C: A9 18 122 LDA WHEGMM ;"-2*15?
042E: A0 04 123 LDT WHEGMM
0430: 20 0C DE 124 JSR FCONP
0433: 00 7E 125 K11 BNE IGERR ;Nao, erro
0435: 4C 0C DF 126 K12 JNP 81NT
127
128 ;Rotina para localizar um elemento de um array
129 ;ou criar um array
130
0438: A5 14 131 ARRAY LDA SUBFLG ;Indices dados?
043A: 00 47 132 BNE FNDARY ;Nao, desviar
043C: A5 10 133 LDA DFNFLG
043E: 05 12 134 ORA INTFLG ;Ligar bit de alta ordem se I
0440: 48 135 PMA
0441: A5 11 136 LDA WLTYP
0443: 48 137 PMA
0444: A0 00 138 LDT WSD0
0446: 98 139 NXTDEN TYA
0447: 48 140 PMA
0448: A5 82 141 LDA WVENMH+1
044A: 48 142 PMA
044B: A5 81 143 LDA WVENMH
044D: 48 144 PMA
044E: 20 1C 04 145 JSR MAKINT
0451: 68 146 PLA
0452: 85 81 147 STA WVENMH
0454: 68 148 PLA
0455: 85 82 149 STA WVENMH+1
0457: 68 150 PLA
0458: 48 151 TAY
0459: 8A 152 TSX

```

```

045A: 80 02 01 153 LDA STACK+2,X ;Obter VALTYP e INTFLE
045B: 48 154 PHA ;e duplica-los
045E: 80 01 01 155 LDA STACK+1,X
0461: 48 156 PHA
0462: A5 A0 157 LDA FAC+3 ;Obter indices e coloca-los na
0464: 90 02 01 158 STA STACK+2,X pilha so lag de VALTYP e INTFLE
0467: A5 A1 159 LDA FAC+4
0469: 90 01 01 160 STA STACK+1,X
046C: C8 161 INT
046D: 20 87 00 162 JSR CHRGOT
0470: C9 2C 163 CMP #' '
0472: F0 02 164 BEQ NXTDIN ;Loop ate colocar todos os
0474: 84 0F 165 STY MURDIN indices na pilha
0476: 20 E0 01 166 JSR CHKCLS
0479: 68 167 PLA
047A: 85 11 168 STA VALTYP ;Obter VALTYP e INTFLE
047C: 68 169 PLA
047D: 85 12 170 STA VALTYP+1
047F: 29 7F 171 AND #57 ;Mascarar bit de INTFLE
0481: 85 10 172 STA DINFLE obtendo DINFLE
0483: A6 68 173 FMDARY LDX ARYTAB
0485: A5 6C 174 LDA ARYTAB+1
0487: 86 98 175 ARY_LOOP STX LOWTR
0489: 85 9C 176 STA LOWTR+1
048B: C5 6E 177 CMP STREND+1
048D: 00 04 178 BNE ARYMAN?
048F: E4 60 179 CPX STREND
0491: F0 3F 180 BEQ NOTFOUND
0493: A0 00 181 ARYMAN? LDY #50
0495: 81 98 182 LDA (LOWTR),Y ;Obter nome do array
0497: C8 183 IMY
0499: C5 81 184 CMP VARMAN ;E o procurado?
049A: D0 06 185 BNE NOARY ;Nao, desviar
049C: A5 82 186 LDA VARMAN+1
049E: D1 98 187 CMP (LOWTR),Y
04A0: F0 16 188 BEQ ARYFOUND
04A2: C8 189 MURRY IMY
04A3: 81 98 190 LDA (LOWTR),Y
04A5: 18 191 CLC
04A6: A5 98 192 ADC LOWTR
04A8: AA 193 TAX
04A9: C8 194 INT
04AA: 81 98 195 LDA (LOWTR),Y
04AC: A5 9C 196 ADC LOWTR+1
04AE: 90 07 197 BCC ARY_LOOP
04B0: A2 69 198 SUBERR LDX #WIDLES-HEXERR0
04B2: 2C 199 HEX 2C
04B3: A2 34 100 IWERR LDX #WIDLES-HEXERR0
04B5: 4C 01 C7 101 JER JMP ERROR
;102
04B8: A2 76 103 ARYFOUND LDX #REDIMAR-HEXERR0
04BA: A5 10 104 LDA DINFLE
04BC: D0 F7 105 BNE JER
04BE: A5 14 106 LDA SUBFLE
04C0: F0 02 107 BEB CHRDDN
04C2: 38 108 SEC ;Necessario para STORE
04C3: 60 109 RTS ;Saida caso veio de RETARYPT

```

```

1110
04C4: 20 07 04 1111 CHRDIN JSR GETARY
04C7: A5 0F 1112 LDA MUMDIN ;Obter numer de dimensoes especific
04C9: A0 04 1113 LDY #504
04CB: 01 08 1114 CMP (LOWTR),Y ;Igual ao numero atual?
04CD: 00 E1 1115 BNE SUBERR ;Nao, erro
04CF: 4C A5 05 1116 JMP ENDELEM ;Pesquisar elemento especific
1117
04D2: A5 14 1118 NOTFOUND LDA SUBFLG ;Veio de GETARYPT?
04D4: FD 05 1119 BEB MAKARY ;Nao, criar novo array
04D6: A2 24 1120 LDX #MhADATA+PENSEIRO
04D8: 4C 01 C7 1121 JMP ERROR ;Sim, erro
1122
04DB: 20 07 04 1123 MAKARY JSR GETARY
04DE: 20 02 C6 1124 JSR REASON
04E1: A9 00 1125 LDA #500
04E3: A8 1126 TAY
04E4: 85 A6 1127 STA STRING2+1
04E6: A2 05 1128 LDX #505
04E8: A5 81 1129 LDA VARNAM
04EA: 91 08 1130 STA (LOWTR),Y
04EC: 10 01 1131 BPL MDIT
04EE: CA 1132 DEX ;Array de inteiro
04EF: CB 1133 KINT DRY
04F0: A5 82 1134 LDA VARNAM+1
04F2: 91 08 1135 STA (LOWTR),Y
04F4: 10 02 1136 BPL RAR ;Desviar se array de reais
04F6: CA 1137 DEX
04F7: CA 1138 DEX
04F9: 86 A0 1139 RAR STX STRING2 ;i=5,3,2 conforme for real
04FA: A5 0F 1140 LDA MUMDIN string, inteiro
04FC: C8 1141 INY ;Saltar desloc ao proximo array
04FE: C8 1142 INY ;(que sera posicao mais tarde)
04FF: 91 08 1143 INY
0501: 91 08 1144 STA (LOWTR),Y
0503: A2 08 1145 SANDIN LDX #508 ;Dimensao padrao mais 1
0505: A9 00 1146 LDA #500
0507: 24 10 1147 BIT DIMFLG ;Array dimensionado?
0509: 68 1148 BVC DFLTDIN ;Nao, desviar
050A: 18 1149 PLA ;Obter dimensao especific em A,X
050B: 69 01 1150 CLC
050D: AA 1151 ADC #501
050E: 68 1152 TAX
050F: 69 00 1153 PLA
0511: C8 1154 DFLTDIN ADC #500
0512: 91 08 1155 INY ;Construir tabela de dimensoes
0514: C8 1156 STA (LOWTR),Y
0515: BA 1157 INY
0516: 91 08 1158 TAX
0518: 20 C7 05 1159 STA (LOWTR),Y
051B: 86 A0 1160 JSR RALT
051D: 85 A6 1161 STX STRING2
051F: A4 5E 1162 STA STRING2+1
0521: C6 0F 1163 LDY INDEX ;Obter Y guardado por RALT
0523: 00 C0 1164 DEC MUMDIN ;Contar dimensoes
0525: A5 05 1165 BNE SANDIN ;Loop ate terminar
0527: 45 05 1166 ADC #PTINT+1 ;Montar ao fim do array

```

0527:	80 50	1167	BCC	BNE	
0529:	85 95	1168	STA	ARYPMT+1	
0528:	A8	1169	TAY		
052C:	8A	1170	TXA		
052D:	85 94	1171	ADC	ARYPMT	
052F:	90 03	1172	BCC	ZARY	
0531:	C8	1173	IMY		
0532:	F0 52	1174	BEQ	BNE	
0534:	20 02 C6	1175	ZARY	JSR	REASON ;testar se ha espaco e
0537:	85 60	1176	STA	STREND	zerar o array
0539:	84 6E	1177	STY	STREND+1	
053B:	A9 00	1178	LDA	#500	
053D:	E6 AE	1179	INC	STRNG2+1	
053F:	A4 60	1180	LDY	STRNG2	
0541:	F0 05	1181	BEQ	KOPS	
0543:	88	1182	ZLUP	DEY	
0544:	91 94	1183	STA	(ARYPMT),Y	
0546:	00 FB	1184	BNE	ZLUP	
0548:	C6 95	1185	KOPD	DEC	ARYPMT+1 ;Apostar a proxima pagina
054A:	C6 AE	1186	DEC	STRNG2+1	
054C:	00 F5	1187	BNE	ZLUP	;Loop ate terminar
054E:	E6 95	1188	INC	ARYPMT+1	
0550:	38	1189	SEC		
0551:	A5 60	1190	LDA	STREND	;Computar deslocamento do
0553:	E5 98	1191	SBC	LOWTR	proximo array
0555:	A0 02	1192	LDY	#502	
0557:	91 98	1193	STA	(LOWTR),Y	;Coloca-lo apes o nome
0559:	A5 6E	1194	LDA	STREND+1	
055B:	C8	1195	IMY		
055C:	E5 9C	1196	SBC	LOWTR+1	
055E:	91 98	1197	STA	(LOWTR),Y	
0560:	A5 10	1198	LDA	DINFLG	;De DIN?
0562:	00 62	1199	BNE	RTN2	;Sim, desviar
		327	PUT	PANTELA,D1	
0564:	C8	11	IMY		
0565:	81 98	12	FMOLEN	LDA	(LOWTR),Y ;Achar elemento especificado
0567:	85 0F	13	STA	MINDIN	do array a partir de indice
0569:	A9 00	14	LDA	#500	colocado na pilha por NEXTDIN
056B:	85 A0	15	STA	STRNG2	
056D:	85 AE	16	DINLUP	STA	STRNG2+1
056F:	C8	17	IMY		
0570:	68	18	PLA		
0571:	AA	19	TAX		
0572:	85 A0	110	STA	FAC+3	;Obter indice e testa-lo
0574:	68	111	PLA		;contra diressao
0575:	85 A1	112	STA	FAC+4	
0577:	D1 98	113	CMP	(LOWTR),Y	
0579:	90 0E	114	BCC	DINOK	
057B:	00 06	115	BNE	OSE	
057D:	C8	116	IMY		
057E:	84	117	TXA		
057F:	D1 98	118	CMP	(LOWTR),Y	
0581:	90 07	119	BCC	DINOK2	
0583:	4C 80 04	120	OSE	JMP	SUBERR
		121			
0586:	4C FF C6	122	BNE	JMP	MENERR
		123			

```

0589: C8 124 D1M0K IMY
058A: A5 AE 125 D1M0K2 LDA STRNG2+1 ;Primeira vez?
058C: 05 AD 126 ORA STRNG2
058E: 18 127 CLC
058F: F0 DA 128 BEB K0DM ;Sim, desviar
0591: 20 C7 05 129 JSR MULT ;Calcular produto das dimensoes
0594: 8A 130 TRM
0595: A5 AD 131 ADC FAC+3
0597: AA 132 TAX
0598: 98 133 TYA
0599: A4 SE 134 LDY INDEX ;Obter Y guardado por MULT
059B: A5 A1 135 K0DM ADC FAC+4 ;Proximo DIM
059D: 86 AD 136 STX STRNG2
059F: C6 DF 137 DEC M0DM0N
05A1: D8 CA 138 BNE D1M0LP ;Loop ate terminarem os indices
05A3: 85 AE 139 STA STRNG2+1
05A5: A2 05 140 LDX #S0S
05A7: A5 B1 141 LDA V0RMM0N
05A9: 10 D1 142 BPL N0M0TA ;Desviar se nao for inteiro
05AB: CA 143 DEX
05AC: A5 B2 144 N0M0TA LDA V0RMM0N+1
05AE: 10 D2 145 BPL R0M0Y ;Desviar se real
05B0: CA 146 DEX
05B1: CA 147 DEX
05B2: 86 64 148 R0RY STX RESULT+2
05B4: A9 00 149 LDA #S00
05B6: 20 D0 05 150 JSR M0L ;Multipl prod das dimens pelo
05B9: 8A 151 TRM ; tamanho de cada elemento
05BA: A5 94 152 ADC A0YPNT ;Somar endereco do array para
05BC: 85 B3 153 STA V0RFPNT obter endereco final
05BE: 98 154 TRM
05BF: A5 95 155 ADC A0YPNT+1
05C1: 85 B4 156 STA V0RFPNT+1
05C3: A8 157 TAX
05C4: A5 B3 158 LDA V0RFPNT
05C6: A0 159 RTM2 RTS
160
161 ;Multiplicacao de 16 bits, ponto fixo, de (LOWTR),Y
162 ;com STRNG2 com produto em A,X.
163 ;Usada pelas rotinas de indexacao de arrays
164
05C7: 84 SE 165 MULT STY INDEX ;Guardar Y para recup apos RTS
05C9: 81 98 166 LDA (LOWTR),Y
05CB: 85 64 167 STA RESULT+2
05CD: 08 168 DEY
05CE: 81 98 169 LDA (LOWTR),Y
05D0: 85 65 170 M0L STA RESULT+3
05D2: A9 10 171 LDA #S10 ;Indice para multipl 16 bits
05D4: 85 99 172 STA I0M0X
05D6: A2 D0 173 LDX #S00
05D8: A0 D0 174 LDY #S00
05DA: 8A 175 M02 TRM ;Deslocar X,Y a esquerda de 1 bit
05DB: DA 176 ASL
05DC: AA 177 TAX
05DE: 98 178 TYA
05DF: 2A 179 ROL
05E0: A8 180 TAY

```

05E0: 80 A4	181		BCS OWE	;Erro se produto > 16 bits
05E2: D6 A0	182		ASL STRING2	;Deslocar para fora o bit
05E4: 26 AE	183		RRL STRING2+1	de alta ordem do multiplicador
05E6: 90 08	184		BCC NU3	;Desviar se bit= 0
05E8: 18	185		CLC	
05EA: 8A	186		TXA	
05EC: 65 64	187		ADC RESULT+2	;Soma outro multiplicador a X,Y
05EE: AA	188		TAX	
05F0: 98	189		TXA	
05F2: 65 65	190		ADC RESULT+3	
05F4: A8	191		TAX	
05F6: 80 92	192		BCS OWE	;Erro se produto > 16 bits
05F8: C6 99	193	NU3	DEC IM0X	
05FA: 00 E3	194		BNE NU2	;Loop ate terminar
05FC: 60	195		RTS	
	196			
05FE: A5 11	197	FRE	LDA VALTYP	
0600: FD 03	198		BEQ FRE2	
0602: 20 1A 09	199		JSR FREFAC	
0604: 20 9E 07	100	FRE2	JSR GARBAG	
0606: 38	101		SEC	
0608: A5 6F	102		LDA FRETOP	
060A: E5 60	103		SBC STREND	
060C: A8	104		TAX	
060E: A5 70	105		LDA FRETOP+1	
0610: E5 6E	106		SBC STREND+1	
0612: A2 00	107	GIWYF	LDX #500	;Transformar em ponto flutuante
	108	*		so inteiro com sinal de A,Y
0614: 86 11	109		STX VALTYP	;Indicar que e numero
0616: 85 9E	110		STA FAC+1	
0618: 84 9F	111		STY FAC+2	
061A: A2 90	112		LDX #590	;Rodar DP 16 bits a direita
061C: 4C 85 DE	113		JMP FLO1	
061E: A4 24	114	POS	LDY CH	
0620: A9 00	115	SONFLT	LDA #500	
0622: 38	116		SEC	
0624: FD EC	117		BEQ GIWYF	
0626: A6 76	118	ERRDR	LDX CURLTN+1	
0628: E8	119		JMX	
062A: 00 A1	120		BNE RTM2	;Retornar se modo indireto
062C: A2 96	121		LDX WCONILES-MENSERR0	
062E: 2C	122		HEX 2C	
0630: A2 80	123	UNDFWC	LDX WFUNDEF-MENSERR0	
0632: 4C 01 C7	124		JMP ERRDR	
	125			
0634: 20 58 D6	126	DEF	JSR FWC?	;Preparar variavel nome da fascao
0636: 20 20 D6	127		JSR ERRDR	
0638: 20 E3 D1	128		JSR CHKSPH	
063A: A9 80	129		LDA #580	
063C: 85 14	130		STA SUBFLS	;Nao permitir variaveis inteir.
063E: 20 03 03	131		JSR PTRGET	;Obter pointer aos argumentos
0640: 20 92 00	132		JSR CHONUM	
0642: 20 ED 01	133		JSR CHKCLS	
0644: A9 00	134		LDA #null	
0646: 20 E8 01	135		JSR STMCNR	
0648: 40	136		PHA	
064A: A5 0A	137		LDA VARPNT+1	;Primeiro caracter apos "="

0648:	48	1138	PHA	
064C:	A5 B3	1139	LDA	VARPNT
064E:	48	1140	PHA	
064F:	A5 B7	1141	LDA	TXTPTR+1
0651:	48	1142	PHA	
0652:	A5 B8	1143	LDA	TXTPTR
0654:	48	1144	PHA	
0655:	20 BA CC	1145	JSR	DATA ;Saltar a proxima instrucao
0659:	4C C7 D6	1146	JMP	FNCDATA ;Preparar pointers ao "valor"
		1147		
		1148		no "nome" de uma funcao e uma variavel simples
		1149		*cujo nome tem a forma (seg,pos), ou "valor" contem
		1150		* Pointer a defn
		1151		* Pointer a variavel argumento
		1152		* Primeiro caracter da definicao
		1153		
065B:	A9 C2	1154	FNC?	LDA #Fn
065D:	20 EB D1	1155	JSR	SYNCR
0660:	09 B0	1156	ORA	#S0
0662:	05 14	1157	STA	SUBFLD ;Indicar variavel simples e lig
0664:	20 DA D3	1158	JSR	PTRGET3 ;bit de alta ordem do primeiro
		1159	*	;caracter do nome
0667:	05 BA	1160	STA	FNCNAM ;Guardar o pointer
0669:	04 B8	1161	STY	FNCNAM+1
066B:	4C 92 D0	1162	JMP	CHKRUM
		1163		
066E:	20 58 D6	1164	FUNCT	JSR FNC? ;Obter pointer ao nome da func.
0671:	A5 B8	1165	LDA	FNCNAM+1
0673:	48	1166	PHA	
0674:	A5 BA	1167	LDA	FNCNAM
0676:	48	1168	PHA	
0677:	20 DA D1	1169	JSR	PARCK ;Avaliar argumento (em FAC)
067A:	20 92 D0	1170	JSR	CHKRUM
067D:	68	1171	PLA	
067E:	05 BA	1172	STA	FNCNAM
0680:	68	1173	PLA	
0681:	05 B8	1174	STA	FNCNAM+1
0683:	A0 D2	1175	LDY	#S02
0685:	01 BA	1176	LDA	(FNCNAM),Y ;Obter pointer ao argumento
0687:	05 B3	1177	STA	VARPNT
0689:	AA	1178	TAX	
068A:	C8	1179	INY	
068B:	01 BA	1180	LDA	(FNCNAM),Y
068D:	F0 99	1181	BEQ	UNDFNC ;Se byte de alta ordem 0,
068F:	05 B4	1182	STA	VARPNT+1 ;nao estava definido
0691:	C8	1183	INY	
0692:	01 B7	1184	SAVOLD	LDA (VARPNT),Y ;Salvar valor da variavel
0694:	48	1185	PHA	;argumento
0695:	08	1186	DEY	
0696:	10 FA	1187	BPL	SAVOLD
0698:	44 B4	1188	LDY	VARPNT+1 ;Apontar ao valor do argumento
069A:	20 45 DE	1189	JSR	MOVWF ;FAC -> (VARPNT)
069D:	A5 B9	1190	LDA	TXTPTR+1 ;Guardar posicao
069F:	48	1191	PHA	
06A0:	A5 B8	1192	LDA	TXTPTR
06A2:	48	1193	PHA	
06A3:	01 BA	1194	LDA	(FNCNAM),Y ;Y= 0



```

06A5: 85 88 1195 STA TXTPTR ;Ajustar a definicao da funcao
06A7: C8 1196 INY
06A8: 81 8A 1197 LDA (FNCNAM),Y
06AA: 85 89 1198 STA TXTPTR+1
06AC: A5 8A 1199 LDA WRPNT+1
06AE: 48 1200 PLA
06AF: A5 83 1201 LDA WRPNT
06B1: 48 1202 PLA
06B2: 20 8F 00 1203 JSR FRNDUM ;Avaliar a funcao
06B5: 60 1204 PLA
06B6: 85 8A 1205 STA FNCNAM
06B8: 68 1206 PLA
06B9: 85 88 1207 STA FNCNAM+1
06BB: 20 87 00 1208 JSR CHR50T ;Ter que ser fim de instrucao
06BE: F0 03 1209 BEQ GETOLO
06C0: 4C F1 01 1210 JMP SYNERR
1211 PUT PARTE1H.01
1212
1213
06C3: 68 12 GETOLO PLA ;Obter posicao do programa
06C4: 85 88 13 STA TXTPTR
06C6: 68 14 PLA
06C7: 85 89 15 STA TXTPTR+1
06C9: A0 00 16 FNCDATA LDY #500 ;Obter valor da variavel argum
06CB: 68 17 PLA
06CC: 91 8A 18 STA (FNCNAM),Y
06CE: 68 19 PLA
06CF: C8 110 INY
06D0: 91 8A 111 STA (FNCNAM),Y
06D2: 68 112 PLA
06D3: C8 113 INY
06D4: 91 8A 114 STA (FNCNAM),Y
06D6: 68 115 PLA
06D7: C8 116 INY
06D8: 91 8A 117 STA (FNCNAM),Y
06DA: 68 118 PLA
06DB: C8 119 INY
06DC: 91 8A 120 STA (FNCNAM),Y
06DE: 60 121 RTS
122
06DF: 20 92 00 123 STR JSR CHRMUM ;Certificar que e um numero
06E2: A0 00 124 LDY #500
06E4: 20 50 E0 125 JSR FACSTRNG ;Converter a string na pilha
06E7: 68 126 PLA
06E8: 68 127 PLA
06E9: A9 FF 128 LDA #5FF ;Acentar a pilha -1 para forcar
06EB: A0 00 129 LDY #500 movimento do string
06ED: F0 12 130 BEQ STRLIT ;Criar descritor e mover string
131
132 *Criar descritor de string
133
06EF: A6 A0 134 STRINI LDX FAC+3
06F1: A4 A1 135 LDY FAC+4
06F3: 86 BC 136 STX DISPTR
06F5: 84 8D 137 STY DISPTR+1
06F7: 20 AC 07 138 STKSPA JSR GETSPA ;A ter o tamanho
06FA: 86 9E 139 STX FAC+1 ;Guardar descritor em FAC
06FC: 84 9F 140 STY FAC+2

```

06FE: 85 90	141		STA	FAC	
0700: 40	142		RTS		
	143				
0701: A2 22	144	STRLIT	LDR	#'''	
0703: 86 80	145		STX	CHARAC	;Frio delimitadores de literal
0705: 84 0E	146		STX	ENDCHR	
0707: 85 A8	147	STRLT2	STA	STRING1	
0709: 84 AC	148		STY	STRING1+1	
070B: 85 9E	149		STA	FAC+1	;Para descritor
070D: 84 9F	150		STY	FAC+2	
070F: A0 FF	151		LDR	MSFF	
0711: C8	152	FEND	IRI		;Achar fim do string
0712: 81 A8	153		LDA	(STRING1),Y	
0714: F0 0C	154		BEQ	ZEND	
0716: C5 06	155		CMR	CHARAC	
0718: F0 04	156		BEQ	MSFF?	
071A: C5 0E	157		CMR	ENDCHR	
071C: 00 F3	158		BNE	FEND	
071E: C9 22	159	MSFF?	CMR	#'''	
0720: F0 01	160		BEQ	AZ	
0722: 18	161	ZEND	CLC		
0723: 84 98	162	AZ	STY	FAC	;Tamanho no descr temporario
0725: 98	163		TYA		
0726: 65 A8	164		ADC	STRING1	
0728: 85 A0	165		STA	STRING2	;Apontar ao fim do string
072A: A6 AC	166		LDR	STRING1+1	
072C: 90 01	167		BCC	FE1	
072E: E8	168		INX		
072F: 86 A6	169	FE1	STX	STRING2+1	
0731: A5 AC	170		LDA	STRING1+1	
0733: F0 04	171		BEQ	FE2	;String e movida, se esta em
0735: C9 02	172		CMR	MSFF	MSFF ou no buffer de entrada
0737: 00 08	173		BNE	PUTHEW	;Caso contr., so prep. descritor
0739: 98	174	FE2	TYA		;Obter tamanho em A
073A: 20 EF 84	175		JSR	SIRIMI	;Dar espaco para string
073D: A6 A8	176		LDR	STRING1	
073F: A4 AC	177		LDR	STRING1+1	
0741: 20 FC 88	178		JSR	MOVSTR	;E move-lo
0744: A6 52	179	PUTHEW	LDR	TEMPPT	
0746: E0 5E	180		CPX	#TEMPST+9	;Descritores temporarios demais?
0748: 00 05	181		BNE	PUTEMP	
074A: A2 80	182		LDR	#FORNULC=HEXERR0	
074C: 4C 01 C7	183	JERR	JMP	ERR0M	
	184				
074F: A5 90	185	PUTEMP	LDA	FAC	;Montar descritor temporario
0751: 95 00	186		STA	LOC0,X	
0753: A5 9E	187		LDA	FAC+1	
0755: 95 01	188		STA	LOC1,X	
0757: A5 9F	189		LDA	FAC+2	
0759: 95 02	190		STA	LOC2,X	
075B: A0 00	191		LDR	MS00	
075D: 86 A0	192		STX	FAC+3	
075F: 84 A1	193		STY	FAC+4	
0761: 88	194		DEY		
0762: 84 11	195		STY	WALTYP	;Indicar string
0764: 86 53	196		STX	LASTPT	;Apontar ao proximo descritor
0766: E8	197		INX		

```

0767: E8 398 INK
0768: E8 399 INK
0769: B6 52 1000 STX TEMPT
0768: 60 1001 RTS
1002
1003 *Criar espaço para string, tamanho em A
1004
076C: 46 13 1005 GETSPA LSR GARFLD ;Permitir reorganizacao da area
076E: 48 1006 GETSPC PHA
076F: 49 FF 1007 EOR #5FF
0771: 38 1008 SEC
0772: 65 6F 1009 ADC FRETOP ;Subtrair tamanho de FRETOP
0774: A4 70 1010 LDY FRETOP+1
0776: 80 01 1011 BCS CY
0778: 88 1012 DEY
0779: C4 6E 1013 CY CPY STREND+1
0778: 90 11 1014 BCC FULL ;Desviar se nao houver espaço
0770: 00 04 1015 BNE GOTSPA
077F: C5 60 1016 CMP STREND
0781: 90 06 1017 BCC FULL
0783: 85 6F 1018 GOTSPA STA FRETOP
0785: 84 70 1019 STY FRETOP+1
0787: 85 71 1020 STA FRESPC
0789: 84 72 1021 STY FRESPC+1
078B: AA 1022 TAX
078C: 68 1023 PLA
078D: 60 1024 RTS
1025
078E: A2 40 1026 FULL LDX #540
0790: A5 13 1027 LDA GARFLD ;Ja houve reorganizacao?
0792: 30 88 1028 BHI JERR ;Sim, erro
0794: 20 9E 07 1029 JSR GARBAG
0797: A9 80 1030 LDA #5B0 ;Indicar reorganizacao feita
0799: 85 13 1031 STA GARFLD
079B: 68 1032 PLA
079C: 00 00 1033 BNE GETSPC
079E: A6 73 1034 GARBAG LDX MEMSIZ ;Reorganizar a partir do comeco
07A0: A5 74 1035 LDA MEMSIZ+1
07A2: 86 4F 1036 FNDVAR STX FRETOP ;Uma passada por todas as
07A4: 85 70 1037 STA FRETOP+1 variav para cada string ativo
07A6: A0 00 1038 LDY #500
07A8: 84 88 1039 STY FNCHNH+1 ;Ind.que nao houve ainda reorg.
1040
1041 *Apontar LOWTR ao fim da area de string
1042
07AA: A5 60 1043 LDA STREND
07AC: A6 6E 1044 LDX STREND+1
07AE: 85 98 1045 STA LOWTR
07B0: 86 9C 1046 STX LOWTR+1
07B2: A9 25 1047 LDA #TEMPST ;Apontar a descritores de
07B4: A2 00 1048 LDX #TEMPST strings temporarios
07B6: 85 5E 1049 STA INDEX
07B8: 86 5F 1050 STX INDEX+1
07BA: C5 52 1051 TMR CMP TEMPT ;fim dos temporarios?
07BC: F0 05 1052 BEB SWMS ;Sim, ir a variaveis simples
07BE: 20 30 08 1053 JSR DVAR ;Tratar os temporario
07C1: F0 F7 1054 BEB TMR ;Semre

```

```

>155
07C3: A9 07 >156 SWAB LDA #507
07C5: B5 0F >157 STA DSCLN
07C7: A5 69 >158 LDA WRTAB
07C9: A6 6A >159 LDX WRTAB+1
07CB: B5 5E >160 STA INDEX
07CD: B6 5F >161 STX INDEX+1
07CF: E4 6C >162 SWAB CPX ARYTAB+1 ;Fim de variaveis simples?
07D1: 00 04 >163 BNE SWABO ;Nao, continuar
07D3: C5 68 >164 CMP ARYTAB
07D5: F0 05 >165 BEB ARYTAB ;Sim, tratar arrays
07D7: 20 30 08 >166 SWABO JSR DWABS ;Tratar variaveis simples
07D9: F0 F3 >167 BEQ SWAB ;Sempre
>168
07DC: B5 94 >169 ARYTAB STA ARYPNT
07DE: B6 95 >170 STX ARYPNT+1
07E0: A9 03 >171 LDA #503
07E2: B5 0F >172 STA DSCLN
07E4: A5 94 >173 ARYTAB2 LDA ARYPNT
07E6: A6 95 >174 LDX ARYPNT+1
07E8: E4 6E >175 ARYTAB3 CPX STREND+1 ;Fim de arrays?
07EA: 00 07 >176 BNE ARYVGO ;Nao, tratar um array
07EC: C5 60 >177 CMP STREND
07EE: 00 03 >178 BNE ARYVGO
07F0: 4C 7C 00 >179 JMP GRPAB ;Todas as variaveis testadas,
>180 * ; mover a de cima
329
07F3: B5 5E >180 PUT PARTE10,D1
07F5: B6 5F >181 ARYVGO STA INDEX
07F7: A0 03 >182 STX INDEX+1
07F9: B1 5E >183 LDY #500
07FB: AA >184 LDA (INDEX),Y ;Obter nome do array
07FD: C8 >185 TAX
07FF: B1 5E >186 LDY (INDEX),Y
0801: 08 >187 PHP ;Guardar seu tipo
0803: C8 >188 INY
0805: B1 5E >189 LDA (INDEX),Y ;Obter desloc do proximo array
0807: A5 94 >190 ADC ARYPNT ;Computar enderecos
0809: B5 94 >191 STA ARYPNT ;E apontar a eles
080B: C8 >192 INY
080D: B1 5E >193 LDA (INDEX),Y
080F: A5 95 >194 ADC ARYPNT+1
0811: B5 95 >195 STA ARYPNT+1
0813: 20 >196 PLS
0815: 13 03 >197 BPL ARYTAB2 ;Desviar se nao for string
0817: 0A >198 TXA
0819: 30 90 >199 BHI ARYTAB2 ;Desviar se nao for string
081B: C8 >200 INY
081D: B1 5E >201 LDA (INDEX),Y ;Obter numero de dimensoes
081F: A0 03 >202 LDY #500
0821: DA >203 ASL
0823: A9 05 >204 ADC #505
0825: A5 5E >205 ADC INDEX ;Apontar ao primeiro elemento
0827: B1 5E >206 STA INDEX do array
0829: 90 02 >207 BCC BPL1
082B: E4 5F >208 INX INDEX+1
082D: A6 5F >209 BPL1 LDX INDEX+1

```

0826:	E4	95	131	ARYSTR	CPX	ARYFMT+1	;Fim do array?
0828:	00	04	132		BNE	GOGO	;Mas, tratar proximo elemento
082A:	CC	94	133		CMF	ARYFMT	
082C:	F0	8A	134		BEQ	ARYFAG	;Sim, proximo array
082E:	20	30	08	GOGO	JSR	DVAR	
0831:	F0	F3	136		BEQ	ARYSTR	;Sempre
			137				
0833:	81	5E	138	DVAR5	LDA	(INDEX),T	;Variavel list ou def de funcao?
0835:	30	25	139		BMI	DVAR5	;Sim, saltar
0837:	C8		140		INX		
083B:	81	5E	141		LDA	(INDEX),Y	;Variavel string?
083A:	10	30	142		BPL	DVAR5	;Mas, saltar
083C:	C8		143		INX		
083D:	81	5E	144	DVAR	LDA	(INDEX),T	;Obter tamanho
083F:	F0	2B	145		BEQ	DVAR5	;Ignorar se tamanho 0
0841:	C8		146		INX		
0842:	81	5E	147		LDA	(INDEX),Y	;Obter endereco do string
0844:	AA		148		TAX		
0845:	C8		149		INX		
0846:	F1	5E	150		LDA	(INDEX),Y	
0848:	C5	70	151		CMF	FRETOP+1	
084A:	90	04	152		BCC	DV1	
084C:	00	1E	153		BNE	DVAR5	
084E:	E4	6F	154		CPX	FRETOP	
0850:	80	1A	155		BCS	DVAR5	;Saltar se ja reorganizado
0852:	C5	9C	156	DV1	CMF	LOWTR+1	;Achado pto acima do ult.string?
0854:	90	16	157		BCC	DVAR5	;Mas, saltar
0856:	00	04	158		BNE	DV2	;Sim, apontar a este ponto
0858:	E4	9B	159		CPX	LOWTR	
085A:	90	10	160		BCC	DVAR5	
085C:	86	9B	161	DV2	STX	LOWTR	
085E:	85	9C	162		STA	LOWTR+1	
0860:	A5	5E	163		LDA	INDEX	
0862:	A6	5F	164		LDX	INDEX+1	
0864:	85	8A	165		STA	FRCMM	
0866:	86	8B	166		STX	FRCMM+1	
0868:	A5	8F	167		LDA	DSLEN	
086A:	85	91	168		STA	BLENGTH	
086C:	A5	8F	169	DVAR5	LDA	DSLEN	;Preparar para proxima variavel
086E:	18		170		CLC		
086F:	A5	5E	171		ADC	INDEX	
0871:	85	5E	172		STA	INDEX	
0873:	90	02	173		BCC	VOOME	
0875:	E6	5F	174		INX	INDEX+1	
0877:	A6	5F	175	VOOME	LDX	INDEX+1	
0879:	A0	00	176		LDY	#500	
087B:	A0		177		RTS		
			178				
			179				#Feita a passagen pelas variaveis, mover agora o ultimo
			180				#string ao topo e voltar para tratar outro.
			181				
087C:	A6	8B	182	GRBPAS	LDX	FRCMM+1	;Reorganizacao feita?
087E:	F0	F7	183		BEQ	VOOME	;Sim, retornar
0880:	A5	91	184		LDA	BLENGTH	
0882:	29	04	185		AND	#504	;4 se simples, 0 caso contrario
0884:	AA		186		LSR		
0885:	AB		187		TAY		

```

DBB4: 85 91 188 STA BLENGTH ;2 se simples, 0 caso contrario
DBB5: 81 8A 189 LDA (FMCNAM),Y
DBB6: 65 98 190 ADC LOWTR
DBB7: 85 96 191 STA HIGHTR
DBB8: 45 9C 192 LDA LOWTR+1
DBB9: 69 00 193 ADC #00
DBB2: 85 97 194 STA HIGHTR+1
DBBA: 45 6F 195 LDA FRETOP
DBB3: 46 7D 196 LOX FRETOP+1
DBB8: 85 94 197 STA HIGHDS
DBBA: 86 95 198 STX HIGHDS+1
DBBC: 20 89 C6 199 JSR BLTUD ;Mover string para cima e
DBBF: 44 91 1100 LDY BLENGTH ;fixar seu descritor
DBA1: CB 1101 INY
DBA2: 45 94 1102 LDA HIGHDS
DBA4: 91 8A 1103 STA (FMCNAM),Y
DBA6: AA 1104 TAX
DBA7: E6 95 1105 INC HIGHDS+1
DBA9: 45 95 1106 LDA HIGHDS+1
DBAB: CB 1107 INY
DBAC: 91 8A 1108 STA (FMCNAM),Y ;X,A aponta agora ao string avc.
DBAE: 4C A2 07 1109 JMP FNDVAR ;Procurar outro para deslocar
1110
DBB1: 45 A1 1111 CAT LDA FAC+4 ;Salvar prim. pointer do descr.
DBB3: 4B 1112 PHA
DBB4: 45 A0 1113 LDA FAC+3
DBB6: 4B 1114 PHA
DBB7: 20 88 01 1115 JSR GETVAL
DBBA: 20 94 00 1116 JSR CHKSTR ;Obt.point.ao descr.do segundo
DBBD: 4B 1117 PLA ;Recuperar primeiro
DBBE: 85 AB 1118 STA STRNG1 ;pointer do descritor
DBCD: 6B 1119 PLA
DBCE: 85 AC 1120 STA STRNG1+1
DBCF: A0 00 1121 LDY #00
DBD5: 81 AB 1122 LDA (STRNG1),Y ;Somar tamanhos
DBD7: 6B 1123 CLC
DBDB: 71 A0 1124 ADC (FAC+3),Y
DBDC: 90 05 1125 BCC NTL ;OK se < $100
DBDE: A2 80 1126 LOX NEXTSTRG-NEWSEKRO
DBDF: 4C 01 C7 1127 JMP ERRM
1128
DBD1: 20 EF 06 1129 NTL JSR STRINT ;Obt.espaco p/ concaten. string
DBD4: 20 EE 08 1130 JSR MOVDS ;Mover primeiro string
DBD7: 45 BC 1131 LDA DSCPTR ;Liberar o segundo
DBD9: A4 80 1132 LDY DSCPTR+1
DBDB: 20 1E 09 1133 JSR FRETMP
DBDE: 20 00 09 1134 JSR MOVSTR ;Mover segundo string
DBE1: 45 AB 1135 LDA STRNG1 ;Liberar o primeiro
DBE3: A4 AC 1136 LDY STRNG1+1
DBE5: 20 1E 09 1137 JSR FRETMP
DBE8: 20 44 07 1138 JSR PUTHEM ;Preparar o descritor
DBEB: 4C 80 00 1139 JMP FRHEVL2 ;Mais formula
1140
DBEE: A0 00 1141 MOVDS LDY #00 ;Mover string cujo descritor
DBFD: 81 AB 1142 LDA (STRNG1),Y ;esta em (STRNG1) para (FRESPC)
DBF2: 4B 1143 PHA ;tamanho
DBF3: CB 1144 INY

```

08F4: 81 A8	>145	LDA	(STRNG1),Y	
08F6: AA	>146	TAX		;For pointer ao string em X,Y
08F7: CB	>147	INY		
08F8: 81 A8	>148	LDA	(STRNG1),Y	
08FA: A8	>149	TAY		
08FB: 68	>150	PLA		;Obter tamanho
08FC: 86 5E	>151	MOVSTR	STX INDEX	;Mover string a X,Y
08FE: 84 5F	>152	STY	INDEX+1	;Em INDEX para (FRESPC)
0900: A8	>153	MOVSTR	TAY	;Tamanho em A)
0901: FD 0A	>154	BEQ	HW3	
0903: 48	>155	PHA		
0904: 88	>156	HW2	DEY	
0905: 81 5E	>157	LDA	(INDEX),Y	
0907: 91 71	>158	STA	(FRESPC),Y	
0909: 98	>159	TYA		
090A: D0 FB	>160	BNE	HW2	
090C: 68	>161	PLA		
090D: 18	>162	HW3	CLC	
090E: 65 71	>163	ADC	FRESPC	
0910: 85 71	>164	STA	FRESPC	
0912: 90 02	>165	BCC	BPL2	
0914: E6 72	>166	INC	FRESPC+1	
0916: 60	>167	BPL2	RTS	
	>168			
0917: 20 94 00	>169	FRESTR	JSR CHRST2	;Último resultado e string?
091A: 65 A0	>170	FREFAC	LDA FAC+3	;Obter pointer ao descritor
091C: A4 A1	>171	LDY	FAC+4	
091E: 85 5E	>172	FRETHP	STA INDEX	;Liberar descritor temporario
0920: 84 5F	>173	STY	INDEX+1	cujo pointer esta em (A,T)
0922: 20 4F D9	>174	JSR	FRETHS	;Libera descritor se temporario
0925: 08	>175	PHP		;Guardar se ultimo liberado
0926: A0 00	>176	LDY	HW00	
0928: 81 5E	>177	LDA	(INDEX),Y	
092A: 48	>178	PHA		;Colocar tamanho na pilha
092B: CB	>179	INY		
092C: 81 5E	>180	LDA	(INDEX),Y	
092E: AA	>181	TAX		;Obter pointer ao string em X,Y
092F: CB	>182	INY		
0930: 81 5E	>183	LDA	(INDEX),Y	
0932: A8	>184	TAY		
0933: 68	>185	PLA		;Obter tamanho e status
0934: 28	>186	PLP		
0935: 00 13	>187	BNE	MB	;Desv. se nao for tempor liberado
0937: CA 70	>188	CPY	FRETOP+1	;E o string mais baixo
0939: 00 0F	>189	BNE	MB	da memoria?
	>190	PUT	PARTE1P,01	
0938: E4 6F	>191	CPX	FRETOP	
093A: D0 08	>192	BNE	MB	;Mas, desviar
093F: 48	>193	PHA		
0940: 18	>194	CLC		;Sim, apagar o string
0941: 65 6F	>195	ADC	FRETOP	
0943: 85 6F	>196	STA	FRETOP	
0945: 90 02	>197	BCC	BPL3	
0947: E6 70	>198	INC	FRETOP+1	
0949: 68	>199	BPL3	PLA	
094A: 86 5E	>200	MB	STX INDEX	;X,Y tem endereco do string
094C: 84 5F	>201	STY	INDEX+1	e Z o tamanho

094E: 60	112		RTS		
	113				
094F: 04 54	114	FRETRG	CPY	LASTPT+1	;Liberar descritor temporario
0951: 00 0C	115		BME	RTNG	
0953: 05 53	116		CHP	LASTPT	
0955: 00 08	117		BME	RTNG	
0957: 05 52	118		STA	TEMPPT	
0959: 09 03	119		SEC	MS03	
095B: 05 53	120		STA	LASTPT	
095D: 40 00	121		LDY	MS00	
095F: 60	122	RTNG	RTS		
	123				
0960: 20 15 DA	124	CHRSTR	JSR	CONGHT	;Converter a byte em X
0963: 0A	125		TXA		
0964: 4B	126		PHA		;Guardar
0965: A9 01	127		LDA	MS01	;Obter espaco para string
0967: 20 F7 06	128		JSR	STRSPA	tamanho i
096A: 68	129		PLA		;Restaurar numero
096B: A0 00	130		LDY	MS00	;Colocar no string
096D: 91 9E	131		STA	(FAC+1),Y	
096F: 68	132		PLA		
0970: 68	133		PLA		
0971: 4C 44 07	134		JMP	PUTMEM	
	135				
0974: 20 03 D9	136	LEFTSTR	JSR	INSTRNG	;Obter parametro I
0977: 01 8C	137		CMP	(DSCPTR),Y	;Menor que tamanho
0979: 9B	138		TYA		;=0(indice no comeco do string)
097A: 90 04	139	INS1	BCC	INS2	;Desviar se parametro I > tamanho
097C: 81 8C	140		LDA	(DSCPTR),Y	;Obter tamanho
097E: AA	141		TAX		
097F: 9B	142		TYA		
0980: 4B	143	INS2	PHA		;Guardar indice comeco do string
0981: 8A	144	INS3	TXA		
0982: 4B	145	INS4	PHA		;Guardar novo tamanho
0983: 20 F7 06	146		JSR	STRSPA	;Obter espaco para o string
0986: A5 8C	147		LDA	DSCPTR	
098B: A4 80	148		LDY	DSCPTR+1	
098A: 20 1E D9	149		JSR	FRETRG	;Liberar descritor temporario
098D: 68	150		PLA		;Obter tamanho
098E: A8	151		TAY		
098F: 68	152		PLA		;Obter deslocamento do string
0990: 18	153		CLC		;E endereco ao pointer
0991: 65 5E	154		ADC	INDEX	
0993: 85 5E	155		STA	INDEX	
0995: 90 02	156		BCC	BPL4	
0997: E6 5F	157		TXC	INDEX+1	
0999: 9B	158	BPL4	TYA		;Obter tamanho
099A: 20 00 D9	159		JSR	MOVSTR	;Colocar string na area de strings
099D: 4C 44 07	160		JMP	PUTMEM	
	161				
09A0: 20 03 D9	162	RIGHTSTR	JSR	INSTRNG	;Obter parametro I
09A3: 18	163		CLC		
09A4: F1 8C	164		SEC	(DSCPTR),Y	;Tamanho -1
09A6: 49 FF	165		EDR	MSFF	;Tamanho -1
09A8: 4C 7A D9	166		JMP	INS1	
	167				
09AB: A9 FF	168	HIDSTR	LDA	MSFF	;Preparar area grande para



```

D9AD: 85 A1 >69 STA FAC+4 ; eventual segundo parametro
D9AF: 20 B7 00 >70 JSR CHRGIT
D9B2: C9 29 >71 CMP #' ) ; Tem segundo parametro?
D9B4: F0 D4 >72 BEQ IMS5 ; Nao, desviar
D9B6: 20 E4 D1 >73 JSR CHKCOM
D9B9: 20 L2 DA >74 JSR GETBYT ; obter seg. parametro em FAC+4
D9BC: 20 D3 D9 >75 INWS JSR INSTRNG ; obter primeiro parametro
D9BF: CA >76 DEX
D9C0: 0A >77 TXA
D9C1: 48 >78 PHA ; Coloc na pilha o desloc espec
D9C2: 18 >79 CLC
D9C3: A2 00 >80 LDX #500
D9C5: F1 BC >81 SBC (DSCPTR),Y ; -Tamanho original -1
D9C7: 80 B8 >82 BCS IMS3 ; Desv se desloc ) velho tamanho
D9C9: 49 FF >83 EOR #FF ; tamanho do resto
D9CB: C5 A1 >84 CMP FAC+4 ; ( Tamanho especificado?
D9CD: 90 B3 >85 BCC IMS4 ; Sim, desviar
D9CF: A5 A1 >86 LDA FAC+4 ; obter tamanho especificado
D9D1: 80 AF >87 BCS IMS4 ; Sempre
>88
>89 *Rotina comaa a LEFTS, MID$ e RIGHTS para testar *)
>90 *Tirar da pilha o endereco de volta, obter o pointer
>91 *de descricao e obter o primeiro parametro da instrucão
>92
D9D3: 20 E0 D1 >93 INSTRNG JSR CHKCLS
D9D4: 68 >94 PLA ; Tirar da pilha endereco
D9D7: 68 >95 TAY ; de volta e guardar
D9D8: 68 >96 PLA
D9D9: 85 91 >97 STA BLENGTH
D9DB: 68 >98 PLA ; Tirar da pilha endereco de
D9DC: 68 >99 PLA ; volta anterior (de GOROUT)
D9DD: 68 >100 PLA ; obter primeiro parametro
D9DE: AA >101 TAX
D9DF: 68 >102 PLA
D9E0: 85 BC >103 STA DSCPTR
D9E2: 68 >104 PLA
D9E3: 85 80 >105 STA DSCPTR+1
D9E5: A5 91 >106 LDA BLENGTH
D9E7: 40 >107 PHA ; Por de volta na pilha
D9E8: 90 >108 TYA ; endereco de volta
D9E9: 40 >109 PHA
D9EA: A0 00 >110 LDY #500
D9EC: 0A >111 TXA ; Transf primeiro parametro a A
D9ED: F0 10 >112 BEQ GO10 ; Erro se 0
D9EF: 60 >113 RTS
>114
D9F0: 20 F4 D9 >115 LEH JSR GETSTR
D9F2: 4C 18 04 >116 JMP SGNFLT
>117
D9F4: 20 17 D9 >118 GETSTR JSR FRESTR
D9F7: A2 00 >119 LDX #500
D9F8: 06 11 >120 STX VALTYP
D9FA: 40 >121 TAY ; Contes tamanho
D9FE: 60 >122 RTS
>123
D9FF: 20 F4 D9 >124 ASG JSR GETSTR ; obter pointer string em INDEX
D902: F0 00 >125 BEQ GO10 ; Erro se tamanho 0

```

DA04:	A0 D0	1126		LDY	#500	
DA06:	81 5E	1127		LDA	(INDEX),Y	;Obter primeiro caracter
DA08:	A8	1128		TAY		;do string
DA09:	4C 18 06	1129		JMP	SENFALT	;transformar em fluante
DA0C:	4C 83 04	1130	GO16	JMP	DBERR	
		1131				
DA0F:	20 81 00	1132	GTBYTC	JSR	CHRGET	
DA12:	20 8F 00	1133	GETBYT	JSR	FRNUM	
DA15:	20 22 D4	1134	CONENT	JSR	MCINT	
DA18:	A6 A0	1135		LDX	FAC+3	;(-256 ?
DA1A:	00 F0	1136		BNE	GO19	;Nao, erro
DA1C:	A6 A1	1137		LDX	FAC+4	
DA1E:	4C 87 00	1138		JMP	CHRGET	
		1139				
DA21:	20 F4 09	1140	VAL	JSR	GETSTR	;Obter pointer string em INDEX
DA24:	80 03	1141		BNE	VL2	
DA26:	4C 68 08	1142		JMP	ZEROFAC	;Retornar 0 se tamanho=0
DA29:	A6 B8	1143	VL2	LDX	TXPTR	
DA2B:	A4 B9	1144		LDY	TXPTR+1	
DA2D:	86 A0	1145		STX	STRNG2	
DA2F:	84 AE	1146		STY	STRNG2+1	
DA31:	A6 5E	1147		LDX	INDEX	
DA33:	86 B8	1148		STX	TXPTR	;Aposar TXPTR ao comeco
DA35:	18	1149		CLC		;do string
DA36:	A5 5E	1150		ADC	INDEX	;Somar tamanho
DA38:	85 60	1151		STA	DEST	;Aposar DEST ao fim do string + 1
DA3A:	A6 5F	1152		LDX	INDEX+1	
DA3C:	86 B9	1153		STX	TXPTR+1	
DA3E:	90 01	1154		RCC	VL3	
DA40:	E8	1155		INX		
DA41:	86 61	1156	VL3	STX	DEST+1	
DA43:	A0 00	1157		LDY	#500	
DA45:	81 60	1158		LDA	(DEST),Y	;Obter byte seguinte ao string
DA47:	48	1159		PHA		;Guarda-lo
DA48:	A9 00	1160		LDA	#500	
DA4A:	91 60	1161		STA	(DEST),Y	;Colocar 0 no lugar
DA4C:	20 87 00	1162		JSR	CHRDOT	
DA4F:	20 64 BF	1163		JSR	FIN	;Avaliar string
DA52:	68	1164		PLA		
DA53:	A0 00	1165		LDY	#500	
DA55:	91 60	1166		STA	(DEST),Y	;Trocar byte final
DA57:	A6 A0	1167	POINT	LDX	STRNG2	
DA59:	A4 AE	1168		LDY	STRNG2+1	
DA5B:	86 B8	1169		STX	TXPTR	
DA5D:	84 B9	1170		STY	TXPTR+1	
DA5F:	60	1171		RTS		
		1172				
DA60:	20 8F 00	1173	GTNUM	JSR	FRNUM	;Avaliar sintaxe: 2bytes, 1byte
DA63:	20 6C DA	1174		JSR	GETADR	;2bytes -> LNUM
DA66:	20 E6 01	1175	CONBYTE	JSR	CHKCON	;1byte -> X
DA69:	4C 12 DA	1176		JMP	GETBYT	
		1177				
DA6C:	A5 90	1178	GETADR	LDA	FAC	;FAC (= \$FFFF?)
DA6E:	C9 91	1179		CPY	#591	
DA70:	80 9A	1180		BCS	GO16	;Nao, erro
DA72:	20 0C 0F	1181		JSR	QINT	;Converter a inteiro

```

0A75: A5 A0 1182 LDA FAC*3 ;E aover
0A77: A4 A1 1183 LDY FAC*4
0A79: 84 50 1184 STY LINUM ;A LINUM
0A7B: 85 51 1185 STA LINUM+1
0A7D: 60 1186 RTS
                                331 PUT PARTE10,D1
                                31
0A7E: A5 50 12 PEEK LDA LINUM ;Proteger LINUM
0A80: 48 13 PNA
0A81: A5 51 14 LDA LINUM+1
0A83: 48 15 PNA
0A84: 20 6C DA 16 JSR GETADR
0A87: A0 00 17 LDY #00
0A89: 81 50 18 LDA (LINUM),Y ;Fazer o PEEK
0A8B: A8 19 TAY
0A8C: 68 110 PLA ;Obter LINUM
0A8D: 85 51 111 STA LINUM+1
0A8F: 68 112 PLA
0A90: 85 50 113 STA LINUM
0A92: 4C 18 D6 114 JNP SGNFLT ;Flutuar Y
                                115
0A95: 20 60 DA 116 POKE JSR STMUR ;Obter byte para POKE em Z
0A9B: 8A 117 TXA ;E endereco em LINUM
0A9F: A0 00 118 LDY #00
0A9B: 91 50 119 STA (LINUM),Y
0A9D: 60 120 RTS
                                121
0A9E: 20 60 DA 122 WAITB JSR STMUR ;Obter endereco em LINUM
0AA1: 86 85 123 STX FORPNT ;E mascara especific em FORPNT
0AA3: A2 90 124 LDX #00
0AA5: 20 87 00 125 JSR CHRDOT ;Especificado byte de inversao?
0AA8: FD 03 126 BEB W2 ;Nao, desviar
0AAA: 20 66 DA 127 JSR CONEYTE ;Obte-lo
0AAD: 86 86 128 W2 STX FORPNT+1 ;Preparar byte de inversao
0AAF: A0 00 129 LDY #00
0AB1: 81 50 130 W3 LDA (LINUM),Y ;Obter byte no endereco
0AB3: 45 86 131 EOR FORPNT+1 ;Inverter
0AB5: 25 85 132 AND FORPNT ;Mascarar
0AB7: FD F8 133 BEB W3 ;Loop ate diferente de 0
0AB9: 60 134 RTM RTS
                                135
                                136 *****
                                137 * *
                                138 * Rotinas de ponto *
                                139 * *
                                140 * flutuante *
                                141 * *
                                142 *****
                                143
0ABA: A9 7E 144 FADDH LDA #HALF ;FAC + 1/2 -> FAC
0ABC: A0 E1 145 LDY #1HALF
0ABE: 4C D8 DA 146 JMP FADD
                                147
0AC1: 20 FD DC 148 FSUB JSR CONEYK ;Carregar ARG com (A,Y)
0AC4: A5 A2 149 FSUBT LDA FACSSM ;ARG - FAC -> FAC
0AC6: 49 FF 150 EOR #FFF
0AC8: 85 A2 151 STA FACSSM

```

DA0A: 45 AA	Y52		EOR	ABSGN	
DA0C: 85 AB	Y53		STA	SGMCPB	
DA0E: 45 9B	Y54		LDA	FAC	
DA0B: 4C 0B DA	Y55		JMP	FADDT	
	Y56				
DA03: 20 0A DC	Y57	ADD	JSR	SHIFT	;Deslocar 1 byte
DA06: 90 3C	Y58		BCC	N	;Sempre
DA0B: 20 F0 DC	Y59	FADD	JSR	CONMPK	; (A,T) em ARG
DA0B: 00 03	Y60	FADDT	BNE	AD1	;ARG + FAC -> FAC
DA0D: 4C 60 DC	Y61		JMP	NOVFA	;Se FAC= 0, so mover ARG
DA0E: A6 AC	Y62	AD1	LDX	EXTRAFAC	;Byte a mais para precisao
DA0E: B6 92	Y63		STX	EXTRASV	; em todas as rotinas FP
DA0E: A2 A5	Y64		LDX	MARG	;Preparar para deslocar ARG
DA0E: A5 A5	Y65		LDA	ARG	
DA0E: AB	Y66	AD2	TAY		
DA0E: F0 CE	Y67		BEQ	RTH	;Se ARG= 0, sair
DA0E: 3B	Y68		SEC		
DA0E: E5 90	Y69		SEC	FAC	;Obter diferenca entre expoente
DA0E: F0 24	Y70		BEQ	ADS	;Somar se mesmo expoente
DA0F: 90 12	Y71		BCC	AD0	
DA0F: 84 90	Y72		STY	FAC	;Troca
DA0F: 84 AA	Y73		LDY	ABSGN	
DA0F: 84 A2	Y74		STY	FACSGN	
DA0F: 49 FF	Y75		EOR	MSFF	
DA0F: 69 00	Y76		ADC	MS00	
DA0F: A0 00	Y77		LDY	MS00	
DA0F: 84 92	Y78		STY	EXTRASV	
DA0D: A2 90	Y79		LDX	MFAC	;Preparar para deslocar FAC
DA0D: 00 04	Y80		BNE	AD4	
DA0D: A0 00	Y81	AD3	LDY	MS00	
DA0D: 84 AC	Y82		STY	EXTRAFAC	
DA0D: C9 F9	Y83	AD4	CMP	MSF9	;Deslocar de quantos bits?
DA0D: 30 C7	Y84		BNE	AD0	;Desviar se mais que 7
DA0D: A6	Y85		TAY		;Indice ao num de deslocamento
DA0D: A5 AC	Y86		LDA	EXTRAFAC	
DA0D: 56 01	Y87		LSR	LOC1,X	
DA0D: 20 21 DC	Y88		JSR	SHFTB	;Deslocar
DA0D: 24 AB	Y89	AD5	BIT	SGMCPB	;Mesmo sinal?
DA0D: 10 57	Y90		BPL	ADMAN	;Sim, somar as mantissas
DA0D: A0 90	Y91		LDY	MFAC	
DA0D: ED A5	Y92		CPX	MARG	;Qual esta ajustada?
DA0D: F0 02	Y93		BEQ	SUBMAN	;Se ARG, fazer FAC+ARG
DA0D: A0 A5	Y94		LDY	MARG	;Se FAC, fazer ARG-FAC
DA0D: 3B	Y95	SUBMAN	SEC		
DA0D: 49 FF	Y96		EOR	MSFF	
DA0D: A5 92	Y97		ADC	EXTRASV	
DA0D: B5 AC	Y98		STA	EXTRAFAC	
DA0D: B9 04 00	Y99		LDA	LOC4,Y	
DA0D: F5 04	Y100		SBC	LOC4,X	
DA0D: B5 A1	Y101		STA	FAC+4	
DA0D: B9 03 00	Y102		LDA	LOC3,Y	
DA0D: F5 03	Y103		SBC	LOC3,X	
DA0D: B5 A0	Y104		STA	FAC+3	
DA0D: B9 02 00	Y105		LDA	LOC2,Y	
DA0D: F5 02	Y106		SBC	LOC2,X	
DA0D: B5 9F	Y107		STA	FAC+2	
DA0D: B9 01 00	Y108		LDA	LOC1,Y	

```

083F: F5 01 >109 SBC L0CL,X
0841: 85 9E >110 STA FAC+1
0843: 80 03 >111 SGMIF RCS SIGNDF ;Desviar se diferenca positiva
0845: 20 88 08 >112 JSR NEGDFAC
0848: A0 00 >113 SGMIF LDY #500 ;Deslocar digito significativo
084A: 98 >114 TYA ;contando deslocamento em A
084B: 18 >115 CLC
084C: A6 9E >116 FLOOP LDX FAC+1
084E: 00 46 >117 BNE FR2 ;Repetir ate diferente de 0
0850: A6 9F >118 LDX FAC+2
0852: B6 9E >119 STX FAC+1
0854: A6 A0 >120 LDX FAC+3
0856: B6 9F >121 STX FAC+2
0858: A6 A1 >122 LDX FAC+4
085A: B6 A0 >123 STX FAC+3
085C: A6 AC >124 LDX EXTRAFAC
085E: B6 A1 >125 STX FAC+4
0860: 84 AC >126 STY EXTRAFAC ;Zerar byte a mais
0862: 69 08 >127 ADC #508 ;Contar os 8 bits
0864: C9 20 >128 CMP #044 ;Feito 4 vezes?
0866: 00 E4 >129 BNE FLOOP ;Nao, loop
0868: A9 00 >130 ZEROFAC LDA #500
086A: 85 9D >131 AtoFAC STA FAC
086C: 85 A2 >132 AtoFACS STA FACSGM
086E: 6D >133 RTS
>134
086F: 65 92 >135 ADMAH ADC EXTRASV ;Somar mantissas
0871: 85 AC >136 STA EXTRAFAC
0873: A5 A1 >137 LDA FAC+4
0875: 65 A9 >138 ADC ARG+4
0877: 85 A1 >139 STA FAC+4
0879: A5 A0 >140 LDA FAC+3
087B: 65 A8 >141 ADC ARG+3
087D: 85 A0 >142 STA FAC+3
087F: A5 9F >143 LDA FAC+2
0881: 65 A7 >144 ADC ARG+2
0883: 85 9F >145 STA FAC+2
0885: A5 9E >146 LDA FAC+1
0887: 65 A6 >147 ADC ARG+1
0889: 85 9E >148 STA FAC+1
088B: 4C A7 06 >149 JMP FR3
>150
088E: 69 01 >151 FR1 ADC #501 ;Contar bits deslocados
0890: 06 AC >152 ASL EXTRAFAC
0892: 26 A1 >153 ROL FAC+4
0894: 26 A0 >154 ROL FAC+3
0896: 26 9F >155 ROL FAC+2
0898: 26 9E >156 ROL FAC+1
089A: 10 F2 >157 FR2 BPL FR1 ;Repetir ate FAC+1 negativo
089C: 38 >158 SEC
089E: E5 9D >159 SBC FAC ;"Fixar" expoente
089F: 80 C7 >160 BCS ZEROFAC
08A1: 49 FF >161 EOR #5FF
08A3: 69 01 >162 ADC #501
08A5: 85 9D >163 STA FAC ;Carry e desligado aqui
08A7: 9D 0E >164 FR3 BCC RTMS
08A9: E6 9D >165 FROUND INC FAC

```

```

DBAB: F0 A2 1166      BEB OVERFLOW
DBAD: 66 9E 1167      B0R FAC+1
DBAF: 66 9F 1168      B0R FAC+2
DBB1: 66 A0 1169      B0R FAC+3
DBB3: 66 A1 1170      B0R FAC+4
DBB5: 66 AC 1171      B0R EXTRAFAC
DBB7: 60 1172      RTMS RTS
                1173
DBBB: A5 A2 1174      NEGFAO LDA FACSSM ;Fazer complemento a um
DBBA: 49 FF 1175      EOR MSFF
DBBC: 85 A2 1176      STA FACSSM
DBBE: A5 9E 1177      NEG2 LDA FAC+1
DBCD: 49 FF 1178      EOR MSFF
DBCE: 85 9E 1179      STA FAC+1
DBC4: A5 9F 1180      LDA FAC+2
DBC6: 49 FF 1181      EOR MSFF
DBCB: 85 9F 1182      STA FAC+2
DBCA: A5 A0 1183      LDA FAC+3
DBCC: 49 FF 1184      EOR MSFF
DBCE: 85 A0 1185      STA FAC+3
DBD0: A5 A1 1186      LDA FAC+4
DBD2: 49 FF 1187      EOR MSFF
DBD4: 85 A1 1188      STA FAC+4
DBD6: A5 AC 1189      LDA EXTRAFAC
DBD8: 49 FF 1190      EOR MSFF
DBDA: 85 AC 1191      STA EXTRAFAC
DBDC: E6 AC 1192      IMC EXTRAFAC ;Somar bit para obter o
DBDE: 00 DE 1193      BNE RTM6 ;complemento a 2
DBE0: E6 A1 1194      PLUSSEPS IMC FAC+4 ;Somar carry de EXTRA
DBE2: 00 DA 1195      BNE RTM6
DBE4: E6 A0 1196      IMC FAC+3
DBE6: 00 DA 1197      BNE RTM6
DBE8: E6 9F 1198      IMC FAC+2
DBEA: 00 D2 1199      BNE RTM6
DBEC: E6 9E 1200      IMC FAC+1
DBEE: 60 1201      RTM6 RTS
                1202
DBEF: A2 40 1203      OVERFLOW LDX NEXTON-NONZERRO
DBF1: 4C 01 C7 1204      JNF ERROR
                332      PUT PARTEIX,01
                11
DBF4: A2 61 12      SHFTRES LDX WRESALT-1 ;Entrada de FUALT
DBF6: 84 D4 13      XISFT LDY LOCA,X ;Carry deve ter sido ligado
DBF8: 84 AC 14      STY EXTRAFAC
DBFA: 84 03 15      LDY LOCS,X
DBFC: 84 D4 16      STY LOCA,X
DBFE: 84 02 17      LDY LOCS,X
DC00: 84 03 18      STY LOCS,X
DC02: 84 D1 19      LDY LOCS,X
DC04: 84 02 110     STY LOCS,X
DC06: A4 A4 111     LDY FPSEM ;SFF se vem de 4DNT p/ sua. neg
DC08: 84 01 112     STY LOCS,X ;Caso contrario 0
DC0A: 69 08 113     SHIFT ADC #508 ;Desloc 1,X a dir $100-h bits
DC0C: 30 E9 114     BRL XISFT ;fazer deslocamento de byte
DC0E: F0 E6 115     BEB XISFT ;se esta no intervalo
DC10: E9 08 116     SBC #508
DC12: A8 117      TAY ;Contar para deslocamento

```

DC13:	A5 AC	118		LDA	EXTRAFAC	Final de bit
DC15:	80 14	119		BCS	SH3	;Sair se nao precisa de desloc
DC17:	16 01	120	SHL	AGL	LOC1,X	;Deslocar somente os 7 bits
DC19:	90 02	121		BCC	SH2	de baixa ordem de 1,X
DC18:	F6 01	122		INC	LOC1,X	;Forcar proxima instrucao
DC10:	76 01	123	SH2	ROR	LOC1,X	a ligar carry
DC17:	76 01	124		ROR	LOC1,X	
DC21:	76 02	125	SHFR	ROR	LOC2,X	
DC23:	76 03	126		ROR	LOC3,X	
DC25:	76 04	127		ROR	LOC4,X	
DC27:	6A	128		ROR		
DC28:	08	129		JMY		
DC29:	00 EC	130		RNE	SH1	
DC28:	18	131	SH3	CLC		
DC2C:	60	132		RTS		
		133				
DC2D:	81 00 00	134	HJM	HEX	8100000000	
DC30:	00 00					
DC32:	03	135	LOGSER	HEX	03	;Indice do numero de coeficientes
DC33:	7F 5E 56	136		HEX	7F5E56879	
DC36:	08 79					
DC38:	80 13 98	137		HEX	801398064	
DC38:	08 64					
DC3D:	80 76 38	138		HEX	8076389D6	
DC40:	93 16					
DC42:	82 38 AA	139		HEX	8238AA3820	
DC45:	38 20					
DC47:	80 25 04	140	SHRacio	HEX	803504F334	;SHR(1/2)
DC4A:	F3 34					
DC4C:	81 25 04	141	SHRdois	HEX	813504F334	;SHR(2)
DC4F:	F3 34					
DC51:	80 20 00	142	SHRacio	HEX	8080000000	;1/2
DC54:	00 00					
DC56:	80 31 72	143	LOGdois	HEX	80317217F8	;LN(2)
DC59:	17 F8					
		144				
DC5E:	20 9C DE	145	LOG	JSR	SIGN	;Logaritmo natural de FAC
DC5E:	F0 02	146		BEQ	G18	;Argumento deve ser > 0
DC6D:	10 03	147		BPL	LGC	
DC62:	4C 83 04	148	G18	JMP	16ER8	
DC65:	A5 9D	149	L62	LDA	FAC	;Guardar expoente - SH0
DC67:	E9 7F	150		SBC	#57F	;Limpar carry
DC69:	48	151		PHA		
DC6A:	A9 80	152		LDA	#80	;Normalizar entre .5 e 1
DC6C:	85 9D	153		STA	FAC	
DC6E:	A9 47	154		LDA	#SHRacio	
DC70:	A0 0C	155		LDY	#1SHRdois	
DC72:	20 08 DA	156		JSR	FAD0	;Calc na serie de pot. impares
DC75:	A9 4C	157		LDA	#SHRdois	de (SHR(2)X-1)/(SHR(2)X+1)
DC77:	A0 0C	158		LDY	#1SHRdois	
DC79:	20 80 00	159		JSR	F01V	
DC7C:	A9 20	160		LDA	#HJM	
DC7E:	A0 0C	161		LDY	#HJM	
DC80:	20 C1 04	162		JSR	FSUB	
DC83:	A9 32	163		LDA	#LOGSER	
DC85:	A0 0C	164		LDY	#1LOGSER	
DC87:	20 76 E2	165		JSR	000SER	;Calcular LOG +.5 base 2

DC8A: 8F 51	166	LDA	MEMACIO	
DC8C: 80 0C	167	LDI	0	MEMACIO
DC8E: 20 08 DA	168	JSR	FAD0	
DC91: 68	169	PLA		
DC92: 20 EF DF	170	JSR	ADMACC	;Somar expoente original - 500
DC95: 8F 56	171	LDA	MLG60is	;Obtido LOG base 2, converter a
DC97: 80 0C	172	LDI	0	MLG60is
DC99: 20 FD DC	173	FMULT	JSR COMUPK	;base e multiplic. por LOG(2)
DC9C: 80 03	174	FMULT	BNE FNU	;(A,Y) em ARG
DC9E: 4C FC DC	175	JMP	RTK7	;FAC * ARG -> FAC
	176			
DCA1: 20 28 00	177	FNU	JSR ADEXP	
DCA4: 8F 00	178	LDA	MS20	;Inicializar produto
DCA6: 85 62	179	STA	RESULT	
DCA8: 85 63	180	STA	RESULT+1	
DCAA: 85 64	181	STA	RESULT+2	
DCAC: 85 65	182	STA	RESULT+3	
DCAE: 85 AC	183	LDA	EXTRAFAC	;Multiplicar digitos de FAC por
DCB0: 20 CA DC	184	JSR	FH1	ARG e somar a RESULT
DCB3: 85 A1	185	LDA	FAC+4	
DCB5: 20 CA DC	186	JSR	FH1	
DCB8: 85 A0	187	LDA	FAC+3	
DCBA: 20 CA DC	188	JSR	FH1	
DCBD: 85 9F	189	LDA	FAC+2	
DCBF: 20 CA DC	190	JSR	FH1	
DCC2: 85 9E	191	LDA	FAC+1	
DCC4: 20 CF DC	192	JSR	FH2	
DCC7: 4C 03 DE	193	JMP	RESIFAC	;Nov RESULT,a FAC e normalizar
	194			
	195			*Rotina para multiplicar A por ARG e somar a RESULT
	196			
DCCA: 80 03	197	FH1	BNE FH2	;Fazer mult.de 8 bits se nao 0
	198			
DCCC: 4C F4 08	199	JMP	SHFTRES	;Desloc.o prod.de um byt.p/accel.
DCCF: 8A	1100	FH2	LSR	;Desloc.p/fara bit de baixa ord.
DCD0: 09 80	1101	ORA	MS80	;Prep para contar 8 bits
DCD2: 80	1102	FH3	TAY	;Guardar
DCD3: 90 19	1103	BCC	FH4	;Desviar se bit de baixa ord 0
DCD5: 18	1104	CLC		;Multi bit por ARG em RESULT
DCD6: 85 65	1105	LDA	RESULT+3	
DCD8: 85 66	1106	ADC	ARG+4	
DCDA: 85 65	1107	STA	RESULT+3	
DCDC: 85 64	1108	LDA	RESULT+2	
DCDE: 85 68	1109	ADC	ARG+3	
DCED: 85 64	1110	STA	RESULT+2	
DCE2: 85 63	1111	LDA	RESULT+1	
DCE4: 85 67	1112	ADC	ARG+2	
DCE6: 85 63	1113	STA	RESULT+1	
DCE8: 85 62	1114	LDA	RESULT	
DCEA: 85 66	1115	ADC	ARG+1	
DCEC: 85 62	1116	STA	RESULT	
DCEE: 66 62	1117	FH4	ROR	RESULT ;Deslocar o produto de 1 bit
DCFD: 66 63	1118	ROR	RESULT+1	
DCF2: 66 64	1119	ROR	RESULT+2	
DCF4: 66 65	1120	ROR	RESULT+3	
DCF6: 66 AC	1121	ROR	EXTRAFAC	
DCFB: 98	1122	TYA		;Obter atualizado



```

BCF9: 4A      1123      LSR          ;Deslocar para fora prox bit
BCFA: 00 D6   1124      RNE FAC     ;Loop 8 vezes (via ORA #580)
BCFC: 60      1125      RTN7      RTS
          1126
          1127 ;Descompactar numero em (A,T) e mover a ARG
          1128
DCFD: 85 5E   1129      COMPMX    STA INDEX
BCFF: 84 5F   1130      STY INDEX+1
0001: A0 04   1131      LDY #04
0003: 81 5E   1132      LDA (INDEX),Y
0005: 85 A9   1133      STA ARG+4
0007: 88     1134      DEY
0008: 81 5E   1135      LDA (INDEX),Y
000A: 85 A8   1136      STA ARG+3
000C: 88     1137      DEY
000D: 81 5E   1138      LDA (INDEX),Y
000F: 85 A7   1139      STA ARG+2
0011: 88     1140      DEY
0012: 81 5E   1141      LDA (INDEX),Y
0014: 85 AA   1142      STA ARG580 ;Guardar sinal
0016: 4C A2   1143      EOR FAC580
0018: 85 AB   1144      STA SGMCP8 ;Preparar comparacao de sinais
001A: A5 AA   1145      LDA ARG580 ;Obter bit mais significativo
001C: 09 80   1146      ORA #80    ;Preparar primeiro bit
001E: 85 A6   1147      STA ARG+1 ;Guardar bit mais significativo
0020: 88     1148      DEY
0021: 81 5E   1149      LDA (INDEX),Y
0023: 85 A5   1150      STA ARG   ;Guardar expoente
0025: A5 98   1151      LDA FAC   para registro de status
0027: 60      1152      RTS
          1153
0028: A5 A5   1154      ADEXP    LDA ARG
002A: F0 1F   1155      ADEX2    BEB ZERO
002C: 18     1156      CLC
002D: 65 90   1157      ADC FAC
002F: 90 04   1158      BCC ADEX3 ;Desviar se nao houver overflow
0031: 30 10   1159      BMI J0V
0033: 18     1160      CLC      ;OK, +580 nao vai dar overflow
0034: 2C     1161      HEX 2C
0035: 10 14   1162      ADEX3    BPL ZERO ;Se ainda positivo, underflow
0037: A9 80   1163      ADC #80  ;Corrigir para deslocamento 800
0039: 85 90   1164      STA FAC
003B: D0 03   1165      BNE ADEX4
003D: 4C 6C 08 1166      JMP AtoFACS
0040: A5 AB   1167      ADEX4    LDA SGMCP8
0042: 85 A2   1168      STA FAC580
0044: 60      1169      RTS
          1170
0045: A5 A2   1171      OUTOFRMS LDA FAC580
0047: A9 FF   1172      EOR #FF
0049: 30 05   1173      BRL J0V  ;Erro se numero positivo
004B: 68     1174      ZERO    PLA
004C: 68     1175      PLA
004D: 4C 68 08 1176      JMP ZEROFAC ;Retornar 0 se numero negativo
0050: 4C EF 06 1177      J0V     JMP OVERFLOW
          1178
          1179 ;Retina para multiplicar FAC por 10

```

```

>180
0053: 20 70 DE 1181 MUL10 JSR MOVWF ;Copiar FAC em ARG
0056: 4A 1182 TAR ;A contem FAC
0057: F0 50 1183 BEQ RTHM ;Sair se FAC = 0
0059: 18 1184 CLC
005A: 69 02 1185 ADC #502 ;Simular #4
005C: 80 F2 1186 SCS JOV
005E: A2 00 1187 LDX #500 ;Indica soma de operando
0060: 86 AB 1188 STX SGNCFR deacao sinal
0062: 20 E8 DA 1189 JSR A02 ;FAC #4 + ARG -> ARG
0065: E6 90 1190 INC FAC ;= multiplicacao por 2
0067: F0 E7 1191 BEQ JOV
0069: 60 1192 RTHM RTS
>193
006A: 84 20 00 1194 NUM10 HEX 8420080000
006D: 00 00
>195
1196 #Rotina para dividir ABS(FAC) por 10
1197
006F: 20 70 DE 1198 DIV10 JSR MOVWF ;Copiar FAC em ARG
0072: A9 6A 1199 LDA #NUM10 ;Preparar para por 10 em FAC
0074: A0 00 1200 LDY #NUM10
0076: A2 00 1201 LDX #500
0078: 86 AB 1202 DIV STX SGNCFR
007A: 20 13 DE 1203 JSR MOVWF ;Colocar (A,Y) em FAC
007D: 4C 83 00 1204 JMP F0E7T ;Dividir ARG por FAC
333 PUT PARTE15,D1
31
0080: 20 F0 DC 12 FDIV JSR COMPK ;(A,Y) -> ARG
0083: F0 76 13 FDIVT BEQ DIV2 ;ARG/FAC -> FAC
0085: 20 8C DE 14 JSR RMOB
0088: A9 00 15 LDA #500
008A: 38 16 SEC
008B: C5 90 17 SBC FAC
008D: 85 90 18 STA FAC
008F: 20 28 00 19 JSR ADEXP ;Obter expoente de ARG/(2*FAC)
0092: E6 90 110 INC FAC ;*2
0094: F0 8A 111 BEQ JOV
0096: A2 FC 112 LDX #4 ;Indice do loop
0098: A9 01 113 LDA #501 ;Contador de bits e quoc. parc.
009A: 84 A6 114 FDL LDY ARG+1 ;ARG >= FAC?
009C: C4 9E 115 CPT FAC+1
009E: 00 0A 116 BNE F02
00A0: 84 A7 117 LDY ARG+2
00A2: C4 9F 118 FDB CPT FAC+2
00A4: 00 0A 119 BNE F02
00A6: A4 AB 120 LDY ARG+3
00A8: C4 A0 121 CPT FAC+3
00AA: 00 0A 122 BNE F02
00AC: A4 A9 123 LDY ARG+4
00AE: C4 A1 124 CPT FAC+4
00B0: 08 125 F02 PHP ;Sim, ligar carry
00B1: 2A 126 RDL ;Subir contador de bits e rodar
127 * ;bit quociente
00B2: 90 09 128 BCC F03 ;Saltar ate completar os 8 bits
00B4: E8 129 INX ;Subir indice do loop
00B5: 95 65 130 STA RESULT+3,X ;Guardar um byte do quociente

```

```

0007: FD 32 131      BEB FD6      ;Desviar se ultimo
0009: 10 34 132      BPL FD7      ;Saida final quando Z=1
000B: A9 01 133      LDA #501     ;Desligar contador de bits
000D: 28 134          F03  PLP      ;ARG != FAC?
000E: 80 0E 135      D0E: 80 0E 135      BCS F05      ;Subtrai divisor se for o caso
000D: 06 A9 136      F04  ASL ARG+4 ;Destocar ARG de um bit
00C2: 26 A8 137      B0L ARG+3
00C4: 26 A7 138      B0L ARG+2
00C6: 26 A6 139      B0L ARG+1
00CB: 80 E6 140      BCS F02      ;Desv se novo ARG da overflow
00CA: 30 CE 141      BRT F01      ;Testar se e possivel a divisao
00CC: 10 E2 142      BPL F02      ;Nao e necessario comparacao
00CE: A8 143          F05  TAY      ;Proteger quociente parcial
00CF: A5 A9 144      LDA ARG+4
00D1: E5 A1 145      SBC FAC+4
00D3: 85 A9 146      STA ARG+4
00D5: A5 A8 147      LDA ARG+3
00D7: E5 A0 148      SBC FAC+3
00D9: 85 A8 149      STA ARG+3
00DB: A5 A7 150      LDA ARG+2
00DD: E5 9F 151      SBC FAC+2
00DF: 85 A7 152      STA ARG+2
00E1: A5 A6 153      LDA ARG+1
00E3: E5 9E 154      SBC FAC+1
00E5: 85 A6 155      STA ARG+1
00E7: 98 156      TYA
00E8: 4C C0 D0 157      JMP F04
00EB: A9 A0 158          F06  LDA #540     ;Prep cont de bits para ultimo
00ED: 00 CE 159          BNE F03      ;Sempre
160
00EF: 0A 161          F07  ASL
00FD: 0A 162          ASL
00F1: 0A 163          ASL
00F2: 0A 164          ASL
00F3: 0A 165          ASL
00F4: 0A 166          ASL
00F5: 85 AC 167      STA EXTRAFAC ;0% ult 2 bits vao a EXTRAFAC
00F7: 20 168          PLP
00F8: 4C D0 DE 169      JMP RESIFAC
00F8: A2 B6 170          DIV2  LDX #D100z00-HEXSEKRO
00FD: 4C D1 C7 171      JMP ERROR
00D0: A5 62 172      RESIFAC LDA RESULT
00D2: 85 9E 173      STA FAC+1
00D4: A5 63 174      LDA RESULT+1
00D6: 85 9F 175      STA FAC+2
00D8: A5 64 176      LDA RESULT+2
00DA: 85 A0 177      STA FAC+3
00DC: A5 65 178      LDA RESULT+3
00DE: 85 A1 179      STA FAC+4
0E10: 4C A8 D8 180      JMP SIGMIF
181
182      ;Rotina para obter numero ponto flutuante compactado
183      ;New (A,T), descompactar e mover a FAC
184
DE13: 85 SE 185      MOVFN STA INDEX
DE15: 84 SF 186      STY INDEX+1
DE17: A0 D4 187      LDY #504

```

DE19: 81 5E	188	LDA	(INDEX),Y	
DE18: 85 A1	189	STA	FAC+4	
DE1D: 88	190	DEY		
DE1E: 81 5E	191	LDA	(INDEX),Y	
DE20: 85 A0	192	STA	FAC+3	
DE22: 88	193	DEY		
DE23: 81 5E	194	LDA	(INDEX),Y	
DE25: 85 9F	195	STA	FAC+2	
DE27: 88	196	DEY		
DE28: 81 5E	197	LDA	(INDEX),Y	
DE2A: 85 A2	198	STA	FACSGN	;Descompactar
DE2C: 09 80	199	ORA	#90	
DE2E: 85 9E	100	STA	FAC+1	
DE30: 88	101	DEY		
DE31: 81 5E	102	LDA	(INDEX),Y	
DE33: 85 90	103	STA	FAC	
DE35: 84 AC	104	STY	EXTRAFAC	;Y= 0
DE37: 60	105	RTS		;Status de acordo com FAC
	106			
DE38: A2 98	107	MOV2F	LDX #TEMP2	;Compactar FAC em TEMP2
DE3A: 2C	108	HEX	2C	
DE3B: A2 93	109	MOV1F	LDX #TEMP1	;Compactar FAC em TEMP1
DE3D: A0 80	110	MOVML	LDY #90	;Byte de alta ordem do endereço
DE3F: F0 04	111	BER	MOVNF	de destino = 0
DE41: A6 85	112	SETFOR	LDX FORPNT	;Chamado por LET e NEXT
DE43: A4 86	113		LDY FORPNT+1	
DE45: 20 8C DE	114	MOVNF	JSR RNDN	;Compactar FAC na memória (X,Y)
DE48: 84 5E	115	STX	INDEX	
DE4A: 84 5F	116	STY	INDEX+1	
DE4C: A0 04	117	LDY	#04	
DE4E: A5 A1	118	LDA	FAC+4	
DE50: 91 5E	119	STA	(INDEX),Y	
DE52: 88	120	DEY		
DE53: A5 A0	121	LDA	FAC+3	
DE55: 91 5E	122	STA	(INDEX),Y	
DE57: 88	123	DEY		
DE58: A5 9F	124	LDA	FAC+2	
DE5A: 91 5E	125	STA	(INDEX),Y	
DE5C: 88	126	DEY		
DE5D: A5 A2	127	LDA	FACSGN	
DE5F: 09 7F	128	ORA	#57	
DE61: 25 9E	129	AND	FAC+1	
DE63: 91 5E	130	STA	(INDEX),Y	
DE65: 88	131	DEY		
DE66: A5 90	132	LDA	FAC	
DE68: 91 5E	133	STA	(INDEX),Y	
DE6A: 84 AC	134	STY	EXTRAFAC	;Y=0
DE6C: 60	135	RTS		
	136			
DE6D: A5 A4	137	MOVFA	LDA ARGSGN	;Mover ARG a FAC
DE6F: 85 A2	138	NFA	STA FACSGN	;Entrada de FPWRT para
DE71: A2 05	139		LDX #505	fazer ABS antes
DE73: 85 A4	140	NFA2	LDA ARG-1,X	
DE75: 95 9C	141		STA FAC-1,X	
DE77: 0A	142	DEX		
DE78: 80 F9	143	BNE	NFA2	
DE7A: 86 AC	144	STX	EXTRAFAC	

DE7C:	60	1145		RTS	
		1146			
DE7D:	20 9C DE	1147	NOHF	JSR RND0	;Arredondar, depois
DE8D:	A2 D6	1148	NF	LDX #504	; mover FAC a ARS
DE82:	85 9C	1149	NF2	LDA FAC-1,X	; inclusive sinal
DE84:	95 A4	1150		STA ARS-1,X	
DE86:	CA	1151		DEX	
DE87:	D0 F9	1152	8ME	NF2	
DE89:	86 AC	1153		STX EXTRAFAC	
DE8B:	60	1154	RTW	RTS	
		1155			
		1156			*Rotina de uso geral para arredondar FAC usando os
		1157			bits mais significativos de EXTRAFAC
		1158			
DE8C:	A5 90	1159	RND0	LDA FAC	;Evitar se numero=0
DE8E:	F0 F8	1160	BEQ	RTW	
DE9D:	D6 AC	1161	ASL	EXTRAFAC	;se EXTRAFAC negativo
DE92:	90 F7	1162	BCC	RTW	;Soar, um bit
DE94:	20 E0 D8	1163	ROUND	JSR PLUSEPS	ao numero em FAC
DE97:	D0 F2	1164	8ME	RTW	
DE99:	4C A9 D8	1165	JMP	ROUND	;Arredondar se expoente afetado
		1166			
DE9C:	A5 90	1167	SIGN	LDA FAC	;testar sinal de FAC e
DE9E:	F0 09	1168	BEQ	RTM10	retornar -1,0,1 em A
DEA0:	A5 A2	1169	SIGN1	LDA FACSGM	de acordo com o resultado
DEA2:	2A	1170	SIGN2	ROL	
DEA3:	A9 FF	1171		LDA #MFT	
DEA5:	80 02	1172	BCS	RTM10	
DEA7:	A9 01	1173		LDA #S01	
DEA9:	60	1174	RTM10	RTS	
		1175			
DEAA:	20 9C DE	1176	SGM	JSR SIGN	;Converter FAC a -1,0,1
DEAD:	85 9E	1177	FLDNT	STA FAC+1	;Converter a flutuante
DEAF:	A9 00	1178		LDA #S00	conteudo de A
DEB1:	85 9F	1179		STA FAC+2	
DEB3:	A2 88	1180		LDX #S88	;DF a direita 8 bits
DEB5:	A5 9E	1181	FL01	LDA FAC+1	;Entrada de SIGNIF para
DEB7:	A9 FF	1182		GOR #MFT	conv. a f.int.de 2 bytes c/sgn.
DEB9:	2A	1183		ROL	;Ligar carry se numero positivo
DEBA:	A9 00	1184	FLO2	LDA #S00	;Entr de LIMPRT para converter
DEBC:	85 A1	1185		STA FAC+4	a flutuante int de 2 bytes
DEBE:	85 A0	1186		STA FAC+3	sem sinal
DECD:	86 90	1187		STX FAC	;preparar expoente
DECE:	85 AC	1188		STA EXTRAFAC	;limpar byte extra
DEC4:	85 A2	1189		STA FACSGM	;Faze-lo positivo
DEC6:	4C 43 08	1190		JMP SGN0F	;ajust sinal e bit mais signif
		1191			
DEC9:	46 A2	1192	ABS	LSR FACSGM	;mudar sinal para positivo
DECB:	60	1193		RTS	
		334		PUT	PARTE1T,01
		11			
		12			*Rotina para comparar FAC com numero compactado em (A,T)
		13			
DECC:	85 60	14	FCOMP	STA DEST	
DECE:	84 61	15	FCOMP2	STY DEST+1	;Entrada de NEXT
DEDD:	40 00	16		LDY #S00	
DEDE:	81 60	17		LDA (DEST),Y	

DE04:	C8	18		IMY	
DE05:	AA	19		TAX	
DE06:	FD C4	210		REG SIGM	;Desviar se (A,Y) for zero
DE08:	B1 60	211		LDA (DEST),Y	
DE0A:	45 A2	212		EDR FAC5M	
DE0C:	30 C2	213		BNT SIGM1	;Desviar se sinais diferentes
DE0E:	E4 90	214		CPX FAC	
DE10:	00 21	215		BNE FC1	;Desviar se expoentes diferentes
DE12:	B1 60	216		LDA (DEST),Y	;Descompactar e comparar
DE14:	09 80	217		ORA #5B0	
DE16:	C5 9E	218		CMP FAC+1	
DE18:	00 19	219		BNE FC1	
DE1A:	C8	220		IMY	
DE1B:	B1 60	221		LDA (DEST),Y	
DE1D:	C5 9F	222		CMP FAC+2	
DE1F:	00 12	223		BNE FC1	
DEF1:	-C8	224		IMY	
DEF2:	B1 60	225		LDA (DEST),Y	
DEF4:	C5 A0	226		CMP FAC+3	
DEF6:	00 08	227		BNE FC1	
DEF8:	C0	228		INT	
DEF9:	A9 7F	229		LDA #57F	;Usar bit extra de FAC para
DEFB:	C5 AC	230		CMP EXTRA/FAC	determinar carry para a
DEFD:	B1 60	231		LDA (DEST),Y	ultima comparacao
DEFF:	E5 A1	232		SEC FAC+4	
DF01:	FD 28	233		REG RTN11	;Saida se numeros iguais
DF03:	A5 A2	234	FC1	LDA FAC5M	
DF05:	90 02	235		BCC FC2	;Desviar se (A,Y) ( FAC em
DF07:	49 FF	236		EDR #5FF	valor absoluto
DF09:	4C 0E	237	FC2	JMP SIGM2	
		238			
		239			*** Saída de FCOMP, A=-1,0,-1 conforme (A,Y) (,=) FAC
		240			
DF0C:	A5 90	241	Q1M1	LDA FAC	;Converter FAC a sua parte int
DF0E:	FD 4A	242		REG ZFAC	
DF10:	38	243		SEC	;Super FAC ) 2^31
DF11:	E9 A0	244		SBC #5A0	;Resultado em
DF13:	24 A2	245		BIT FAC5M	FAC+1 a FAC+4
DF15:	10 09	246		BPL Q11	;(57E-5A2)
DF17:	AA	247		TAX	
DF18:	A9 FF	248		LDA #5FF	
DF1A:	B5 A4	249		STA FPGEN	
DF1C:	20 BE 00	250		JSR NEG2	
DF1F:	8A	251		TXA	
DF20:	A2 90	252	Q11	LDX #FAC	
DF22:	C9 F9	253		CMP #5F9	;Mais de 7 bits a deslocar?
DF24:	10 06	254		BPL Q12	;Sao, desviar
DF26:	20 0A 0C	255		JSR SHFT	;Sim, realizar desloc do byte
DF29:	04 A4	256		STY FPGEN	;Y=0
DF2B:	60	257	RTN11	RTS	
		258			
DF2C:	A8	259	Q12	TAY	;Numero de bits a deslocar
DF2D:	A5 A2	260		LDA FAC5M	
DF2F:	29 80	261		AND #5B0	;Obter sinal
DF31:	46 9E	262		LSR FAC+1	
DF33:	05 9E	263		ORA FAC+1	
DF35:	B5 9E	264		STA FAC+1	;Restabelecer sinal

DF37: 20 21 0C	165	JSR	SHTFR	;Deslocar
DF38: 84 44	166	STY	FPSEN	;Y=0
DF3C: 60	167	RTS		
	168			
DF3D: 45 98	169	INT	LDA	FAC
DF3F: C9 A0	170	CMR	#5A0	;12°30?
DF41: 80 28	171	BCC	RTN12	;Não, sair
DF43: 20 0C 0F	172	JSR	QINT	
DF46: 84 AC	173	STY	EXTRAFAC	;Y=0
DF48: 45 A2	174	LDA	FAC5GH	
DF4A: 84 A2	175	STY	FAC5GH	
DF4C: 49 80	176	EOB	#5B0	;Testar sinal
DF4E: 2A	177	ROL		;Guardar como status do carry
DF4F: A9 A0	178	LDA	#5A0	;Prep esp inicial de 2°31
DF51: 85 90	179	STA	FAC	
DF53: 45 A1	180	LDA	FAC+4	;Guardar digito mais signif
DF55: 85 80	181	STA	CHARAC	;Para EXP e para teste de
DF57: 4C 43 08	182	JMP	SGNIF	paridade em FPMBT
	183			
DF5A: 85 9E	184	ZFAC	STA	FAC+5
DF5C: 85 9F	185	STA	FAC+2	;Rotina INT precisa de todos
DF5E: 85 A0	186	STA	FAC+3	os bytes = 0
DF60: 85 A1	187	STA	FAC+4	
DF62: AB	188	TAY		;E Y=0
DF63: 60	189	RTM12	RTS	
	190			
	191			*Avaliar numero ponto flutuante em TXPTR
	192			
DF64: A0 00	193	FIN	LDY	#500
DF66: A2 0A	194	LDX	#50A	;Zerar de TMEXP a SERLEN
DF68: 94 99	195	FIN2	STY	TMEXP,X
DF6A: CA	196	DEX		;199 - 5A3
DF6B: 10 FB	197	BPL	FIN2	
DF6D: 90 0F	198	BCC	NO3DIGIT	
DF6F: C9 2D	199	CMR	#'-'	
DF71: D0 04	1100	BNE	FIN3	
DF73: 86 A3	1101	STX	SERLEN	;Indicar numero negativo se "-"
DF75: F0 04	1102	BEQ	EWAL	
DF77: C9 2B	1103	FIN3	CMR	#'+'
DF79: D0 05	1104	BNE	CHKDP	
DF7B: 28 B1 00	1105	EWAL	JSR	CHRGET
DF7E: 90 58	1106	NO3DIGIT	BCC	INSRT010
DF80: C9 2E	1107	CHKDP	CMR	#','
DF82: F0 2E	1108	BEQ	SETDP	
DF84: C9 45	1109	CMR	#'E'	
DF86: D0 30	1110	BNE	ADJEXP	
DF88: 2D B1 00	1111	JSR	CHRGET	;Obter expoente
DF8B: 90 17	1112	BCC	GOGEX	
DF8D: C9 C9	1113	CMR	#minus	;Expoente negativo?
DF8F: F0 0E	1114	BEQ	SETSGN	;Sim, indicar
DF91: C9 28	1115	CMR	#'-'	;Não pode estar em
DF93: F0 0A	1116	BEQ	SETSGN	forma de TOKEN
DF95: C9 C8	1117	CMR	#plus	;Semelhante para '+'
DF97: F0 08	1118	BEQ	DFD16	
DF99: C9 28	1119	CMR	#'+'	
DF9B: F0 04	1120	BEQ	DFD16	
DF9D: D0 87	1121	BNE	SGMCHK	;Numero completo

```

1122
DFYF: 66 9C 1123 SETSN R08 EXPSEM ;Indicar expoente negativo
DFAI: 20 B1 00 1124 DPTD3 JSR CHRSET ;Obter prox algar do expoente
DFAA: 90 5C 1125 SGGEX BCC GETEXP ;Numero, desviar
DFA6: 24 9C 1126 SGMCHK BIT EXPSEM
DFAB: 10 DE 1127 BPL ADJEXP
DFAA: A9 00 1128 LDA #S00
DFAC: 38 1129 SEC ;Tornar expoente negativo
DFAD: E5 9A 1130 SBC EXPON
DFAF: 4C BA DF 1131 JMP AEX
DFB2: 66 98 1132 SETDP R08 DPFLG
DFB4: 24 98 1133 BIT DPFLG
DFB6: 50 C3 1134 BVC EVAL ;Desviar se primeiro ".",
1135
DFB8: A5 9A 1136 ADJEXP LDA EXPON ;Ajustar expoente e sair
DFAA: 38 1137 AEX SEC
DFBB: E5 99 1138 SBC THPEXP
DFB0: 85 9A 1139 STA EXPON
DFBF: F0 12 1140 BEQ EVDOME
DFC1: 10 09 1141 BPL DPNIGHT
DFC3: 20 6F D0 1142 DPLEFT JSR DIV10
DFC6: E6 9A 1143 INC EXPON
DFC8: D0 F9 1144 BNE DPLEFT
DFCA: FD 07 1145 BEQ EVDOME
DFCC: 20 53 D0 1146 DPNIGHT JSR MULT0
DFCF: C6 9A 1147 DEC EXPON
DFD1: D0 F9 1148 BNE DPNIGHT
DFD3: A5 A3 1149 EVDOME LDA SERLEN ;Negativo?
DFD5: 30 D1 1150 BHI EVD
DFD7: 60 1151 RTS
1152
DFD8: 4C EA E1 1153 EVD JMP WEDOP
1154
DFD8: 48 1155 INSRD10 PMA ;Guardar digito
DFDC: 24 98 1156 BIT DPFLG ;Apareceu ponto decimal?
DFDE: 10 02 1157 BPL NDP ;Nao, desviar
DFE0: E6 99 1158 INC THPEXP ;Sim, ajustar
DFE2: 20 53 D0 1159 NDP JSR MULT0 ;Passar por cima do pto decimal
DFE4: 68 1160 PLA ;Ajustar digito a esquerda
DFE6: 38 1161 SEC ;do ponto decimal
DFE7: E9 30 1162 SBC #'0' ;Mascara
DFE9: 20 EF DF 1163 JSR ADDACC
DFEC: 4C 78 DF 1164 JMP EVAL ;Loop ate terminar
1165
1166 ;Rotina para adicionar A a FAC
1167
DFE7: 48 1168 ADDACC PMA
DFFD: 20 7D DE 1169 JSR H0AUF ;Copiar FAC em A06
DFF3: 68 1170 PLA
DFF4: 20 AD DE 1171 JSR FLOAT
DFF7: A5 AA 1172 LDA AREGSN
DFF9: 45 A2 1173 EOR FACSEM
DFFB: 85 A8 1174 STA SGMCFR
DFFD: A6 90 1175 LOX FAC ;Mo sinal se FAC = 0
DFFF: 4C D8 DA 1176 JMP FADDT
1177
E002: A5 9A 1178 GETEXP LDA EXPON ;Novo expoente } ???

```



```

E004: C9 0A 1179      CMP  #50A
E006: 90 09 1180      BEC  #006      ;Nao, desviar
E008: A9 64 1181      LDA  #64A     ;Grande demais
E00A: 24 9C 1182      BIT  EXP5M    ;Expoente negativo?
E00C: 30 11 1183      BMI  STEX    ;Sim, obtenhamos 0
E00E: 4C EF 08 1184    CMP  OVERFLOW ;Nao, overflow
E011: 0A          1185  #006  ASL          ;Expoente velho vezes 10
E012: 0A          1186      ASL
E013: 18          1187      CLC
E014: 65 9A     1188      ADC  EXP5M
E016: 0A          1189      ASL
E017: 18          1190      CLC
E018: A0 00     1191      LDY  #500
E01A: 71 88     1192      ABC  (TXTPTR),Y ;Somar novo algoritmo
E01C: 38          1193      SEC
E01D: E9 30     1194      SBC  #'0'      ;Compensar ASCII
E01F: 85 9A     1195  STEX  STA  EXP5M
E021: 4C A1 0F 1196    JMP  OPD10
      1197
      335      PUT  PARTE10,01
      11
E024: 98 3E 0C 12  CEM#cdc  HEX  983EBC1FFD ;99.999.999,9
E027: 1F FD
E029: 9E 6E 68 13  BIL#e0H  HEX  9E6E6827FD ;999.999.999
E02C: 27 FD
E02E: 9E 6E 68 14  BIL#w0  HEX  9E6E682900 ;1.000.000.000
E031: 28 00
E033: A9 47 15  IMPRT  LDA  #1M#00      ;Imprimir "IM"
E035: A0 C4 16      LDY  #1M#00
E037: 20 48 ED 17      JSR  PRSTR
E03A: A5 76 18      LDA  CURLIN+1
E03C: A6 75 19      LDY  CURLIN
E03E: 85 9E 110  LIMPRT STA  FAC+1      ;Imprimir A,X em decimal
E040: 86 9F 111  STX  FAC+2
E042: A2 90 112  LOX  #5F0
E044: 38 113      SEC
E045: 20 8A DE 114  JSR  FLO2
E048: 20 4E ED 115  PRMTFAC JSR  FOUT      ;Imprimir num pto flut em FAC
E04B: 4C 5F CE 116  PRSTR  JMP  STROUT   ;Imprimir string de A,X
      117
      11B  *Converte FAC a string em STACK e aponta A,Y a ele
      119
E04E: A0 01 120  FOUT  LDY  #501
      121
      122  *Entrada de STRS coloca string em SFF (Y=0) de maneira
      123  *a forçar movimento da string a area de string
      124
E050: A9 20 125  FACSTRNG LDA  #'-'
E052: 88 126      DEY
E053: 24 A2 127      BIT  FAC5M
E055: 10 04 128      BPL  SFS0
E057: C8 129      IMY
E05B: 99 FF 00 130    STA  STACK-1,Y
E05B: 85 A2 131  SFS6  STA  FAC5M      ;Valor absoluto
E05B: 04 A0 132      STY  STRNG2
E05F: C8 133      IMY
E060: A9 30 134      LDA  #530

```

```

ED62: A6 9D 135      LDX FAC          ;Numero = 07
ED64: 00 03 136      BNE NOTZE
ED66: 4C 71 E1 137      JMP WNDUP        ;Sim, terminar
ED68: A9 00 138      NOTZE LDA #000
ED6E: E0 80 139      CPX #500        ;Numero != 1?
ED6D: F0 02 140      BEQ #0
ED6F: 80 09 141      BCS STE         ;Sim, desviar
ED71: A9 2E 142      #0 LDA #0LHAB
ED73: A0 E0 143      LDY #0LHAB
ED75: 20 99 0C 144      JSR FMULT       ;Mover o decimal e tornar o
ED7B: A9 F7 145      LDA #57         ;o pto fixo para acelerar
ED7A: 85 99 146      STE
ED7C: A9 29 147      CMB#L LDA #0L#L#L
ED7E: A0 E0 148      LDY #0L#L#L
ED80: 20 CC 0E 149      JSR FCOMP       ;Normalizar entre 100.000.000 e
ED83: F0 1E 150      BEQ INTPART     999.999.999
ED85: 10 12 151      BPL J010
ED87: A9 24 152      CMB#H LDA #0C#H#0C
ED89: A0 E0 153      LDY #0C#H#0C
ED8B: 20 CC 0E 154      JSR FCOMP
ED8E: F0 02 155      BEQ J#10
ED90: 10 0E 156      BPL ROUN
ED92: 20 53 00 157      J#10 JSR #0L#10 ;Desv. se estiver no intervalo
ED95: C6 99 158      DEC T#EXP
ED97: 00 EE 159      BNE C#F#H
ED99: 20 6F 00 160      J#10 JSR #0V#10
ED9C: E6 99 161      JMC T#EXP
ED9E: 00 0C 162      BNE C#F#H
EDA0: 20 BA 0A 163      ROUN JSR #0C#H ;Arredondar
EDA3: 20 0C 0F 164      INTPART JSR #0T ;Converter forma normal a int
EDA6: A2 01 165      LDX #01 ;Pointer 0F
EDA8: A5 99 166      LDA T#EXP
EDA9: 18 167      CLC
EDAB: 69 0A 168      AOC #0A ;Testar se numero ( ,01
EDAD: 30 09 169      B#1 DPLOC ;Desv se neces exp negativo
EDAf: C9 08 170      CMP #08 ;Test se numero > 999.999.999
EDB1: 80 06 171      BCS DPL ;Desv. se neces exp positivo
EDB3: 6F 9F 172      AOC #57 ;Subtrair 1
EDB5: AA 173      TAX ;Apontar a DP
EDB6: A9 02 174      LDA #02
EDB8: 38 175      DPLOC SEC ;Calcular expoente correto
EDB9: E9 02 176      DPL SBC #02
EDBB: 85 9A 177      STA EXP#0 ;0 se nao ha expoente
EDBD: 86 99 178      STX T#EXP ;Numero de digitos antes de 0F
EDBF: 8A 179      T#A
EDC0: F0 02 180      BEQ PUTDP
EDC2: 10 13 181      BPL #K#STR ;Desviar se nao comeca com 0F
EDC4: A4 A0 182      PUTDP LDY #STR#02
EDC6: A9 2E 183      LDA #",."
EDC8: CB 184      D#Y
EDC9: 99 FF 00 185      STA STACK-1,T
EDCC: 8A 186      T#A
EDCD: F0 06 187      BEQ SVY
EDCF: A9 30 188      LDA # "0"
EDD1: CB 189      D#Y
EDD2: 99 FF 00 190      STA STACK-1,T
EDD5: 84 A0 191      SVY STY #STR#02

```

E007:	A0 00	192	PARSTR	LDI	#500	;Zerar numero enquanto se
E009:	A2 80	193		LDX	#580	cria string
E00B:	A5 A1	194	MSLUP	LDA	FAC+4	
E00D:	18	195		CLC		
E00E:	79 86 E1	196		ADC	DECTBL+3,Y	
E0E1:	85 A1	197		STA	FAC+4	
E0E3:	A5 A0	198		LDA	FAC+3	
E0E5:	79 85 E1	199		ADC	DECTBL+2,Y	
E0E8:	85 A0	1100		STA	FAC+3	
E0EA:	A5 9F	1101		LDA	FAC+2	
E0EC:	79 84 E1	1102		ADC	DECTBL+1,Y	
E0EF:	85 9F	1103		STA	FAC+2	
E0F1:	A5 9E	1104		LDA	FAC+1	
E0F3:	79 83 E1	1105		ADC	DECTBL,Y	
E0F6:	85 9E	1106		STA	FAC+1	
E0F8:	EB	1107		INX		;Contar em X
E0F9:	80 04	1108		BCS	PARITY?	;Continuar a somar os subtrair
E0FB:	10 DC	1109		BPL	MSLUP	se numero decimal positivo e
E0FD:	30 02	1110		BMI	COUNTED	carry ligado ou numero decimal
E0FF:	30 0A	1111	PARITY?	BMI	MSLUP	negativo e carry desligado
E101:	8A	1112	COUNTED	TXA		
E102:	90 04	1113		BCC	MARKDIBIT	
E104:	49 FF	1114		EOR	#5FF	;Ajustar o contador para o
E106:	69 0A	1115		ADC	#50A	caso de numero decimal posit
E108:	69 2F	1116	MARKDIBIT	ADC	#10'-1	;Converter contador a
E10A:	CB	1117		INY		palgarismo ASCII
E10B:	CB	1118		INY		
E10C:	CB	1119		INY		
E10D:	CB	1120		INY		
E10E:	8A 03	1121		STY	VARPNT	;Guardar pointer a DECTBL
E110:	A4 A0	1122		LDI	STRNG2	;Obter pointer ao string
E112:	CB	1123		INY		
E113:	AA	1124		TAX		
E114:	29 7F	1125		AND	#57F	
E116:	99 FF 00	1126		STA	STACK-1,Y	
E119:	C6 99	1127		DEC	TMPEXP	;Deslocar ponto decimal
E11B:	00 06	1128		BNE	SAYY	
E11D:	A9 2E	1129		LDA	#','	;Inseri-lo no local adequado
E11F:	CB	1130		INY		
E120:	99 FF 00	1131		STA	STACK-1,Y	
E123:	84 A0	1132	SAYY	STY	STRNG2	;Guardar pointer do string
E125:	A4 83	1133		LDI	VARPNT	;Obter pointer DECTBL
E127:	8A	1134		TXA		
E128:	49 FF	1135		EOR	#5FF	;Trocar sinal de X
E12A:	29 80	1136		AND	#500	
E12C:	AA	1137		TAX		
E12D:	CD 24	1138		CPY	#524	
E12F:	00 AA	1139		BNE	MSLUP	;Loop ate terminar
E131:	AA A0	1140		LDI	STRNG2	
E133:	89 FF 00	1141	HWBACK	LDA	STACK-1,Y	
E136:	8B	1142		DEY		
E137:	C9 30	1143		CMP	#'0'	;Suprimir zeros finais
E139:	F8 F8	1144		BEQ	HWBACK	
E13B:	C9 2E	1145		CMP	#','	;Se terminar em 0P, recobri-lo
E13D:	F8 01	1146		BEQ	NEEDEX?	
E13F:	CB	1147		INY		
E140:	A9 2B	1148	NEEDEX?	LDA	#'+'	

```

E142: A6 9A >149      LDX EXPON
E144: F0 2E >150      BEQ MARKEND ;Desviar se nao ha expoente
E146: 10 08 >151      BPL PUTEK ;Desviar se expoente positivo
E148: AF 00 >152      LDA #000
E14A: 30 >153      SEC
E14B: E5 9A >154      SBC EXPON ;Tansa-to negativo
E14D: 4A >155      TAX
E14E: A9 20 >156      LDA #'-'
E150: 99 01 01 >157  PUTEK STA STACK+1,Y
E153: A9 45 >158      LDA #'E'
E155: 99 00 01 >159  STA STACK,Y
E158: BA >160      TRX ;Expoente em A
E159: A2 2F >161      LDX #'0'-1 ;Usar X para contar expoente
E15B: 30 >162      SEC ;em ASCII
E15C: E8 >163      WHATEX INX
E15D: E9 0A >164      SBC #504 ;Dividir por 10
E15F: B0 FB >165      BCS WHATEX
E161: 69 3A >166      ABC #'0'+10 ;Ajustar resto
E163: 99 03 01 >167  STA STACK+3,Y ;= Parte baixa do exp em ASCII
E165: BA >168      TRX ;Obter quociente
E167: 99 02 01 >169  STA STACK+2,Y ;= Parte alta do exp em ASCII
E16A: A9 00 >170      LDA #000
E16C: 99 04 01 >171  STA STACK+4,Y ;Marcar fia
E16F: F0 08 >172      BEQ PNTSTK
E171: 99 FF 00 >173  WHATEX STA STACK-1,Y
E174: A9 00 >174      MARKEND LDA #000
E176: 99 00 01 >175  STA STACK,Y
E179: A9 00 >176      PNTSTK LDA #CSTACK
E17B: A0 01 >177      LDY #1STACK
E17D: 40 >178      RTS
      306 PUT PARTEIV,01
      11
E17E: 80 00 00 >12  HALF HEX 800000000
E181: 00 00
      13
      14 #Representacao em 32 bits hexa das potencias de 10
      15
E183: FA 0A 1F >6  DECTBL HEX F0A01F00 ;-100000000
E186: 00
E187: 00 98 96 >7  HEX 00989680 ;100000000
E18A: 80
E18B: FF FD 8D >8  HEX FFFD8DC0 ;-10000000
E18E: C0
E18F: 00 01 86 >9  HEX 000186A0 ;1000000
E192: A0
E193: FF FF 80 >10  HEX FFFF80F0 ;-100000
E196: F0
E197: 00 00 03 >11  HEX 000003E8 ;1000
E19A: E8
E19B: FF FF FF >12  HEX FFFFFFF8 ;-100
E19E: 9C
E19F: 00 00 00 >13  HEX 0000000A ;10
E1A2: 0A
E1A3: FF FF FF >14  HEX FFFFFFFF ;-1
E1A6: FF
      115
      116 TEND = *

```

```

>17
E1A7: 20 70 DE 118 SBR JSR MOVWF ;Calcular como potencia de 1/2
E1AA: 89 7E 119 LDA #HALF
E1AC: A0 E1 120 LDY #HALF
E1AE: 20 13 DE 121 JSR MOVWF ;Colocar 1/2 em FAC
>22
E1B1: FD 70 123 FPMRT BEQ EXP ;#MG*FAC -> FAC
E1B3: A5 A5 124 LDA #0
E1B5: 00 03 125 BNE FW1
E1B7: 4C 6A D8 126 JMP AtoFAC ;Fazer FAC=0 se #MG=0
>27
E1BA: A2 8A 128 FW1 LDX #TEMP3
E1BC: A0 00 129 LDY #00
E1BE: 20 45 DE 130 JSR MOVWF ;Guardar em TEMP3
E1C1: A5 AA 131 LDA #ROSGM
E1C3: 10 0F 132 BPL FAC ;Desviar se argumento positivo
E1C5: 20 30 0F 133 JSR INT ;Obter parte inteira do exp
E1C8: AF 0A 134 LDA #TEMP3
E1CA: A0 00 135 LDY #00
E1CC: 20 CC DE 136 JSR FCOMP ;E potencia inteira
E1CF: 00 03 137 BNE FW2
E1D1: 98 138 TYA ;Sim, permitir argum negativo
E1D2: A4 00 139 LDY #CHARAC ;Obter paridade (de INT)
E1D4: 20 AF DE 140 FW2 JSR #FA ;Mover argumento a FAC
E1D7: 98 141 TYA ;Bit menos signif so pode ser
E1D8: 48 142 PWA ; ligado por LDY #CHARAC
E1D9: 20 58 0C 143 JSR LOG ;Obter LOG (argumento)
E1DC: A9 8A 144 LDA #TEMP3
E1DE: A0 00 145 LDY #00
E1E0: 20 99 0C 146 JSR #MULT ;Calcular exponencial#LOG (arg)
E1E3: 20 23 E2 147 JSR EXP ;Elevar a potencia e
E1E6: A8 148 PLA ;Exponente era inteiro negativo?
E1E7: 4A 149 LSR
E1E8: 90 DA 150 BCC RTN13 ;Nao, retornar
E1EA: A5 90 151 MDDP LDA FAC ;Resultado zero?
E1EC: F0 06 152 BEQ RTN13 ;Sim, retornar
E1EE: A5 A2 153 LDA #FACGM
E1F0: 4F FF 154 EOR #FFF
E1F2: 05 A2 155 STA #FACGM
E1F4: 60 156 RTN13 RTS
>57
158 #Os coeficientes dos valores indicados aqui
159 #sao ajustados para precisao
160
E1F5: 81 38 AA 161 LOGE HEX 8128AA3829 ;LOG (e) na base 2
E1F8: 38 29
E1FA: 07 162 EXPSER DF0 7 ;Indice do numero de coefic
E1FB: 71 34 58 163 HEX 7134583E56 ;(log(2)^7)/7!
E1FE: 3E 56
E200: 74 16 7E 164 HEX 74167E8318 ;(log(2)^6)/6!
E203: 83 18
E205: 77 2F EE 165 HEX 772FEE385 ;(log(2)^5)/5!
E208: E3 85
E20A: 7A 10 84 166 HEX 7A10841C2A ;(log(2)^4)/4!
E20D: 1C 2A
E20F: 7C 63 59 167 HEX 7C6359580A ;(log(2)^3)/3!
E212: 58 0A

```

```

E214: 7E 75 FD 168      HEX 7E75FDE7D6 ;(log(21^2)/2)
E217: E7 C4
E219: 80 31 72 169      HEX 8031721810 ;log(2)
E21C: 18 10
E21E: 81 00 00 170      HEX 8100000000 ;1
E221: 00 00
                                     ;71
E223: A9 F5 172      EXP LDA #LOGe ;e^FAC -> FAC
E225: A0 E1 173      LDY #1LOGe
E227: 20 99 0C 174      JSR FMULT ;Prep para calc e^(xLOG(e))
E22A: A5 AC 175      LBA EXTRAFAC
E22C: A9 50 176      ADC #50
E22E: 90 02 177      BCC X1
E230: 20 94 0E 178      JSR ROUNO
E233: 85 92 179      STA EXTRASV
E235: 20 80 0E 180      JSR #AF ;Copiar a ARG
E238: A5 90 181      LDA FAC
E23A: C9 88 182      CMP #88 ;Dentro do intervalo
E23C: 90 03 183      BCC X2 ;Nao, desviar
E23E: 20 45 00 184      JSR OUTOFRNG ;Zero ou overflow
E241: 20 30 0F 185      JSR INT ;Obter parte inteira em FAC
E244: A5 00 186      LDA #0
E246: 18 187      CLC
E247: A9 81 188      ADC #81 ;Segundo teste de intervalo
E249: F0 F3 189      BEQ OOR
E24B: 38 190      SEC
E24C: E9 01 191      SBC #01
E24E: 48 192      PNA
E24F: A2 05 193      LDY #05
E251: 85 A5 194      X3 LDA ARG,X ;Trocar ARG com FAC
E253: 84 90 195      LDY FAC,X
E255: 95 90 196      STA FAC,X
E257: 94 A5 197      STY ARG,X
E259: CA 198      DEX
E25A: 10 F5 199      BPL X3
E25C: A5 92 1100      LDA EXTRASV
E25E: 85 AC 1101      STA EXTRAFAC
E260: 20 C4 0A 1102      JSR FSUBT ;Subtrair parte inteira
E263: 20 EA E1 1103      JSR #EGOP
E266: A9 FA 1104      LDA #EXPSE
E268: A0 E1 1105      LDY #EXPSE
E26A: 20 8C E2 1106      JSR SERIES ;Usar series para a parte frac
E26D: A9 00 1107      LDA #00
E26F: 85 AB 1108      STA #GNCFR
E271: 68 1109      PLA
E272: 20 2A 00 1110      JSR #DEX2 ;Somar cop da parte inteira
E275: 60 1111      RTS
                                     ;112
E276: 85 A0 1113      OORSER STA SERPNT ;Calcular ax+bx^3+cx^5+...
E278: 84 AE 1114      STY SERPNT+1 ;Com SERPNT apontado aos
E27A: 20 38 DE 1115      JSR #MOVIF coeficientes ...c,b,a.
E27D: A9 93 1116      LDA #TEMP1 ;Numero de coefic = SERLEN+1
E27F: 20 99 DC 1117      JSR FMULT ;x^2
E282: 20 90 E2 1118      JSR #SERMADR ;fazer serie com x^2
E285: A9 93 1119      LDA #TEMP1 ;Obter x de novo
E287: A0 00 1120      LDY #00
E289: 4C 99 DC 1121      JMP FMULT ;Multiplicar pela serie e sair

```

```

>122
E28C: 85 A0 >123 SERDES STA SERPNT ;Calcular a+b*c*x^2+...
E28E: 84 AE >124 STY SERPNT+1 ;Com SERPNT apontado aos
E290: 20 38 DE >125 SERMATH JSR NOVZF coeficientes ...c,b,a.
E293: 81 AD >126 LDA (SERPNT),Y
E295: 85 A3 >127 STA SERLEN ;Colocar em SERLEN valor do
E297: A4 AD >128 LOY SERPNT inicio da tabela e apontar
E299: C8 >129 TMY ;SERPNT ao ultimo coeficiente
E29A: 98 >130 TRA ;(que e o primeiro da tabela)
E29B: D0 02 >131 BNE SS
E29D: E6 AE >132 TNC SERPNT+1
E29F: 85 AD >133 SS STA SERPNT
E2A1: A4 AE >134 LOY SERPNT+1
E2A3: 20 59 DC >135 SERLOOP JSR FMULT
E2A6: A5 AD >136 LDA SERPNT
E2A8: A4 AE >137 LOY SERPNT+1
E2AA: 18 >138 CLC
E2AB: A9 05 >139 ADC #05 ;Passar SERPNT ao prox coefic
E2AD: F0 01 >140 RCC NXTERM
E2AF: C8 >141 TMY
E2B0: 85 AD >142 NXTERM STA SERPNT
E2B2: 84 AE >143 STY SERPNT+1
E2B4: 20 08 DA >144 JSR FADD ;Somar proximo coeficiente
E2B7: A9 98 >145 LDA #TEMP2 ;Obter x de novo
E2B9: A0 00 >146 LOY #00
E2BB: C6 A3 >147 DEC SERLEN
E2BD: D0 E4 >148 BNE SERLOOP ;Loop ate terminar
E2BF: 60 >149 RTM14 RTS
>150
E2C0: 98 35 44 >151 RMDA0J1 HEX 9835447A
E2C3: 7A
E2C4: A8 28 81 >152 RMDA0J2 HEX 6828B146
E2C7: 46
>153
E2C8: 20 9C DE >154 RND JSR SIGH ;Obter sinal do argumento
E2CB: AA >155 TAX ;Testar
E2CD: 30 18 >156 BMI R01 ;Se -, usar FAC corrente
E2CE: A9 C9 >157 LDA #MAGSEED
E2D0: A0 00 >158 LOY #00 ;Mover raiz corrente da FAC
E2D2: 20 13 DE >159 JSR NOVFN
E2D5: DA >160 TRA ;Restaurar sinal
E2D6: F0 E7 >161 BEB RTM14 ;Sair se RND(0)
E2D8: A9 D0 >162 LDA #RMDA0J1 ;Juntar
E2DA: A0 E2 >163 LOY #RMDA0J1
E2DC: 20 59 DC >164 JSR FMULT
E2DF: A9 C4 >165 LDA #RMDA0J2 ;Juntar de novo
E2E1: A0 E2 >166 LOY #RMDA0J2
E2E3: 20 08 DA >167 JSR FADD
E2E6: A6 A1 >168 R01 LDY FAC+4 ;Mais
E2E8: A5 9E >169 LDA FAC+1 ;Trocar byte mais signif com
E2EA: 85 A1 >170 STA FAC+4 byte menos signif
E2EC: B6 9E >171 STX FAC+1
E2EE: A9 D0 >172 LDA #00
E2F0: 85 A2 >173 STA FACSGN ;Tomar valor absoluto
E2F2: A5 9D >174 LDA FAC
E2F4: 85 AC >175 STA EXTRAFAC ;Posicionar bit extra ao acaso
E2F6: A9 D0 >176 LDA #00 ;Ajustar para o intervalo 0-1

```

E2F8:	85	90	>177		STA	FAC	
E2FA:	20	48	08	1178	JSR	SIGNIF	;Normalizar
E2FD:	A2	C9	>179		LDX	WORDSEED	;Mover FAC a raiz RND
E2FF:	A0	00	>180		LDY	WS00	
E301:	4C	45	0E	1181	R02	JMP	NOVAF
				337		PUL	PARTE2A,02
				31			
E304:	A9	80	>2		C05	LDA	WPIdiv2
E306:	A0	E3	>3			LDY	WPIdiv2
E308:	20	08	0A	>4		JSR	FA00
				15			
E308:	20	70	0E	>6	S1M	JSR	NOVAF
E30E:	A9	85	>7			LDA	WPIveze2
E310:	A0	E3	>8			LDY	WPIveze2
E312:	A6	AA	>9			LDX	AR00GM
E314:	20	78	00	>10		JSR	01V
E317:	20	70	0E	>11		JSR	NOVAF
E31A:	20	30	0F	>12		JSR	INT
E31D:	A9	00	>13			LDA	WS00
E31F:	85	A8	>14			STA	SGMCPR
E321:	20	C4	0A	>15		JSR	FSUBT
E324:	A9	8A	>16			LDA	WQUARTER
E326:	A0	E3	>17			LDY	WQUARTER
E328:	20	C1	0A	>18		JSR	FSUB
E32B:	A5	A2	>19			LDA	FACSGM
E32D:	48	>20				PHA	
E32E:	10	00	>21			BPL	ST1
E330:	20	BA	0A	>22		JSR	FA00H
E333:	A5	A2	>23			LDA	FACSGM
E335:	30	09	>24			BMI	ST2
E337:	A5	16	>25			LDA	SIGNFLG
E339:	A9	FF	>26			EDR	WSFF
E33B:	85	16	>27			STA	SIGNFLG
E33D:	20	EA	E1	>28	ST1	JSR	NEGOP
E340:	A9	8A	>29		ST2	LDA	WQUARTER
E342:	A0	E3	>30			LDY	WQUARTER
E344:	20	08	0A	>31		JSR	FA00
E347:	68	>32				PLA	
E34B:	10	03	>33			BPL	ST3
E34A:	20	EA	E1	>34		JSR	NEGOP
E34D:	A9	8F	>35		ST3	LDA	WSINSE
E34F:	A0	E3	>36			LDY	WSINSE
E351:	4C	76	E2	>37		JMP	000SER
				338			
E354:	20	38	0E	>39	TAM	JSR	NOVAF
E357:	A9	80	>40			LDA	WS00
E359:	85	16	>41			STA	SIGNFLG
E35B:	20	08	E3	>42		JSR	SIN
E35E:	A2	8A	>43			LDX	WTEMP3
E360:	A0	00	>44			LDY	WS00
E362:	20	01	E3	>45		JSR	R02
E365:	A9	93	>46			LDA	WTEMP1
E367:	A0	00	>47			LDY	WS00
E369:	20	13	0E	>48		JSR	NOVAF
E36C:	A9	80	>49			LDA	WS00
E36E:	85	A2	>50			STA	FACSGM
F370:	A5	1A	>51			LDA	SIGNFLG



```

E372: 20 7C E3 152      JSR TAM2
E375: A9 8A 153      LDA #TAMP3      ;Obter seno
E377: A0 00 154      LDY #000
E379: 4C 80 00 155     JMP F01V      ;E dividir
                156
E37C: 48 157      TAM2 PHA
E37D: 4C 30 E3 158     JMP S11
                159
E38D: 81 49 0F 160     PDiv2  HEX 81490FD4A2
E38E: 0A A2
E38F: 83 49 0F 161     PDiv2z2 HEX 83490FD4A2
E388: 0A A2
E38A: 7F 00 00 162     QUARTER HEX 7F00000000
E38B: 00 00
                163
                164      *Coeficientes ajustados para precisao
                165
E38F: 05 166      STNSR  DFB 5      ;Indice do numero de coefic
E390: 84 EA 1A 167     HEX 84EA1A2018      ;(2pi)**11/16!
E393: 20 18
E395: 86 28 07 168     HEX 8628078F8F      ;(2pi)**9/9!
E398: F8 FB
E39A: 87 99 68 169     HEX 8799689801      ;(2pi)**7/7!
E39D: 89 01
E39F: 87 23 35 170     HEX 8723350FE1      ;(2pi)**5/5!
E3A2: 0F E1
E3A4: 86 A5 50 171     HEX 86A550E728      ;(2pi)**3/3!
E3A7: E7 28
E3A9: 83 49 0F 172     HEX 83490FD4A2      ;2pi
E3AC: 0A A2
E3AE: A6 03 C1 173     HEX A603C1C8D4
E3B1: C8 D4
E3B3: C8 05 CA 174     HEX C805CA4CECA
E3B4: CE CA
                175
E3B8: A5 A2 176      ATM  LDA FACSEN
E3BA: 48 177      PHA
E3BC: 10 03 178      BPL ATH1
E3BD: 20 EA E1 179     JSR NEGOP
E3C0: A5 9D 180      ATM1 LDA FAC
E3C2: 48 181      PHA
E3C3: C9 81 182      CMP #0B1      ;Normalizar entre 0 e 1
E3C5: 90 07 183      BCC ATN2
E3C7: A9 20 184      LDA #HIGH
E3C9: A0 0C 185      LDY #HIGH
E3CB: 20 80 00 186     JSR F01V
E3CE: A9 E8 187      ATM2 LDA #ATNSR
E3D0: A0 E3 188      LDY #1ATNSR
E3D2: 20 76 E2 189     JSR 000SER
E3D5: 68 190      PLA
E3D6: C9 81 191      CMP #0B1
E3D8: 90 07 192      BCC ATN3
E3DA: A9 80 193      LDA #P1Div2
E3DC: A0 E3 194      LDY #P1Div2
E3DE: 20 C1 DA 195     JSR FSUB      ;Compensar para normalizacao
E3E1: 68 196      PLA
E3E2: 10 03 197      BPL RTN15

```

E3E4:	4C EA E1	>98	JMP	HE99F	
E3E7:	60	>99	RTM15	RTS	
		>100			
E3EB:	08	>101	ATMSER	DFB	508 ;Indice do numero de coefic
E3E9:	76 83 83	>102	HEX	7A83838003	
E3EC:	80 08				
E3EE:	79 1E F4	>103	HEX	791EF4AF5	
E3F1:	A6 F5				
E3F3:	78 83 FC	>104	HEX	7883FC8010	
E3F6:	80 10				
E3F8:	7C 0C 1F	>105	HEX	7C0C1F67CA	
E3FB:	67 CA				
E3FD:	7C 0E 53	>106	HEX	7C0E5308C1	
E400:	C8 C1				
E402:	70 14 64	>107	HEX	701464704C	
E405:	70 4C				
E407:	70 87 EA	>108	HEX	7087EAS17A	
E40A:	51 7A				
E40C:	70 63 30	>109	HEX	706330887E	
E40F:	88 7E				
E411:	7E 92 4A	>110	HEX	7E924A993A	
E414:	99 3A				
E416:	7E 4C CC	>111	HEX	7E4CCC91C7	
E419:	91 C7				
E41B:	7F AA AA	>112	HEX	7FAAAA6A13	
E41E:	AA 13				
E420:	B1 09 00	>113	HEX	B109000000	
E423:	00 00				
		>114			
		>115	*Retina CHRGET (e raiz RND) a serem colocados em		
		>116	*MBI na pagina zero		
		>117			
E425:	E6 B8	>118	ZPSTUFF	INC	TXTPTR
E427:	D0 02	>119	BNE	N10	
E429:	E6 B9	>120	INC	TXTPTR+1	
E42B:	AD 7A 00	>121	N10	LDA	EMWIMP
E42E:	C9 3A	>122	CMF	#" "	;Ret com carry lig se nao numer
E430:	80 0A	>123	BCS	RTM16	;Ligar flag Z se ""
		>124	*		;ou fim de linha
E432:	C9 20	>125	CMF	#" "	;Saltar espacos
E434:	F0 EF	>126	BEQ	ZPSTUFF	
E436:	38	>127	SEC		
E437:	E9 30	>128	SBC	#"0"	;Desl carry se numer, ligar se
E439:	38	>129	SEC		; nao numer e conserv valor de A
E43A:	E9 D0	>130	SBC	#\$100-'0'	
E43C:	60	>131	RTM16	RTS	
		>132			
E43D:	80 4F C7	>133	HEX	804FC7525B	;Somente para numero aleatorio
E440:	52 58				
		>134			
E442:	A2 FF	>135	COLDST	LDX	#\$FF
E444:	B6 76	>136	STX	CURLIM+1	;Inicializar modo direto
E446:	A2 F8	>137	LDX	#\$FB	;e pilha
E448:	9A	>138	TXS		;os 4 bytes mais altos da pilha
E449:	89 42	>139	LDA	#COLDST	so usados para pointer e
E44B:	AD E4	>140	LDY	#COLDST	numero da linha para input
		>141	*		; de linha

E44D:	85 01	1142	STA	GOMM0H+1	
E44F:	84 02	1143	STY	GOMM0H+2	
E451:	85 04	1144	STA	GOSTROUT+1	
E453:	84 05	1145	STY	GOSTROUT+2	
E455:	20 0A ES	1146	JSR	NOERR	;Iniciar texto normal
E45B:	A9 4C	1147	LDA	RS4C	;Preparar areas de jump
E45A:	85 00	1148	STA	GOMM0H	
E45C:	85 03	1149	STA	GOSTROUT	
E45E:	85 90	1150	STA	JMPADR0	
E460:	85 0A	1151	STA	USR	;Inicializar enderecos USR
E462:	A9 83	1152	LDA	RIERR	; com rotina de erro
E464:	A0 04	1153	LDY	#10ERR	
E466:	85 08	1154	STA	USR+1	
E468:	84 0C	1155	STY	USR+2	; "Valor ilegal"
E46A:	A2 1C	1156	LDX	RS1C	
E46C:	80 24 E4	1157	LDX	ZPSTUFF-1,X	
E46F:	95 80	1158	STA	CHRGET-1,X	
E471:	86 F1	1159	STX	SPEEDZ	;Inicializar SPEED com
E473:	CA	1160	DEX		;ZSS (SPEEDZ=1)
E474:	00 F6	1161	BNE	WVZF	
E476:	86 F2	1162	STX	TRCFLG	;Ligar NOTRACE
E478:	8A	1163	TAX		
E479:	85 A4	1164	STA	FPGEN	;Contem 0 exceto na rotina INT
E47E:	85 54	1165	STA	LASTPT+1	;Sempre contem 0
E47D:	48	1166	PNA		
E47E:	A9 03	1167	LDA	RS03	;Inicializar DSLEN com valor
E480:	85 BF	1168	STA	DSLEN	esperado por GARBAG
E482:	20 20 CE	1169	JSR	CR00	
E485:	A9 01	1170	LDA	RS01	
E487:	80 FD 01	1171	STA	IN-3	;Prop pointer fantasoa a \$101
E48A:	80 FC 01	1172	STA	IN-4	
E480:	A2 55	1173	LDX	RS55	;Inicializar indice aos
E48F:	86 52	1174	STX	TEMPPT	descritores tempor. de STRINGS
E491:	A9 01	1175	LDA	RS01	;Teste de area de memoria
E493:	80 FF 07	1176	STA	R16K	
E496:	A9 02	1177	LDA	RS02	
E498:	80 FF 47	1178	STA	ESPR16K	
E49E:	C0 FF 07	1179	CMR	R16K	
E49E:	00 09	1180	BNE	T40K	
E4A0:	A9 00	1181	LDA	RS00	
E4A2:	80 40 04	1182	STA	FLGRAM	
E4A5:	A0 1F	1183	LDY	RS1F	
E4A7:	00 28	1184	BNE	STHDMN	
E4A9:	A9 03	1185	LDX	RS03	
E4AB:	80 FF 07	1186	STA	R40K	
E4AE:	40 FF 07	1187	EDR	R40K	
E4B1:	F0 0E	1188	BEQ	SET40K	
E4B3:	A9 01	1189	LDA	RS01	
E4B5:	80 40 04	1190	STA	FLGRAM	
E4B8:	A0 35 04	1191	LDA	TSUT1L	
E4BB:	00 12	1192	BNE	HISFFF	
E4BD:	A0 5F	1193	LDY	RS5F	
E4BF:	00 10	1194	BNE	STHDMN	
E4C1:	A9 02	1195	LDX	RS02	
E4C3:	80 40 04	1196	STA	FLGRAM	
E4C6:	A0 35 04	1197	LDA	TSUT1L	
E4C9:	00 04	1198	BNE	HISFFF	

E4CB:	AD 9F	1199	LDY	W59F	
E4CD:	DD 02	1200	BNE	STHMEM	
E4CF:	AD 5F	1201	HISFFF	LDY	W5FF
E4D1:	A9 FF	1202	STHMEM	LDA	W5FF
E4D3:	05 73	1203	STA	NEWSIZ	
E4D5:	04 74	1204	STY	NEWSIZ+1	
E4D7:	05 6F	1205	STA	FRETOP	
E4D9:	04 70	1206	STY	FRETOP+1	
E4DB:	05 71	1207	STA	FRESPC	
E4DD:	04 72	1208	STY	FRESPC+1	
E4DF:	A5 47	1209	LDA	PGAL	
E4E1:	00 04	1210	BNE	INI1000	
E4E3:	A2 08	1211	LDX	W500	
E4E5:	00 02	1212	BNE	INIBAS	
E4E7:	A2 10	1213	INI1000	LDX	W510
E4E9:	A0 00	1214	INIBAS	LDY	W500
E4EB:	04 67	1215	STY	BASLOC	
E4ED:	06 68	1216	STX	BASLOC+1	
E4EF:	04 06	1217	STY	LOCK	;Inicializar byte LOCK e
E4F1:	90	1218	TYA		
E4F2:	91 67	1219	STA	(BASLOC),Y	;Byte inicial de programa
E4F4:	E6 67	1220	INC	BASLOC	
E4F6:	00 02	1221	BNE	WIBASLOC	
E4F8:	E6 68	1222	INC	BASLOC+1	
E4FA:	A5 67	1223	WIBASLOC	LDA	BASLOC
E4FC:	A4 68	1224	LDY	BASLOC+1	
E4FE:	20 02 C6	1225	JSR	REASON	
E501:	20 62 C9	1226	JSR	SCOTCH	
E504:	AD 35 04	1227	LDA	TSUTIL	
E507:	F0 11	1228	BEQ	INIBAS0	
E509:	AD F0 9F	1229	LDA	TSCART	
E50C:	E9 01	1230	CMY	W501	
E50E:	DD 0A	1231	BNE	INIBAS3	
E510:	AD F1 9F	1232	LDA	INICART	
E513:	A6 F2 9F	1233	LDX	INICART+1	
E516:	05 67	1234	STA	BASLOC	
E518:	06 68	1235	STX	BASLOC+1	
E51A:	A9 03	1236	INIBAS0	LDA	W500
E51C:	05 AF	1237	STA	BASLOC1-1	
E51E:	AD 68 00	1238	LDA	BASLOC+1	
E521:	05 00	1239	STA	BASLOC1	
E523:	A9 5F	1240	LDA	WSTROUT	
		338	PUT	PARTE20_02	
		11			
		12			
		13			*****Agora recobrir areas 0-5
		14			
E525:	AD CE	14	LDY	WSTROUT	
E527:	05 04	15	STA	BOSTROUT+1	
E529:	04 05	16	STY	BOSTROUT+2	
E52B:	A9 28	17	LDA	WRESTART	
E52D:	AD C7	18	LDY	WRESTART	
E52F:	05 01	19	STA	COMANH+1	
E531:	04 02	110	STY	COMANH+2	
E533:	6C 01 00	111	JMP	(COMANH+1)	
		112			
		113			*****
		114			

```

>15 * Graficos,etc *
>16 * *
>17 *****
>18
ES3A: 20 BF 00 119 CALL JSR FRMMUN
ES3F: 20 6C 04 120 JSR GETADR
ES3C: 4C 50 00 121 JMP (LIMMUN)
122
123 *Observar que CALL nao prepara os registradores
124 *Ea sa CALL:
125 * Y tem o byte de baixa ordem do endereço CALL
126 * A tem o byte de alta ordem do endereço CALL
127 * X contem $90
>28
ES3F: 40 129 RTS
130
131 *Subrotinas baixa resolucao
132 *Subrotina para obter coordenadas *,*
133
ES40: 20 12 0A 134 PLOTFMS JSR GETBYT ;Obter primeira coordenada
ES43: E0 30 135 CPX #530 ;Testar se menor que 48
ES45: 80 13 136 BCS GOERR
ES47: 86 F0 137 STX FIRST
ES4F: AF 2C 138 LDA #", "
ES48: 20 E8 01 139 JSR SYNCHR ;Teste de sintaxe
ES4E: 20 12 0A 140 JSR GETBYT ;Obter segunda coordenada
ES51: E0 30 141 CPX #530 ;Tem que ser < 48
ES53: 80 05 142 BCS GOERR
ES55: 86 2C 143 STX H2 ;Preparar H2 e V2
ES57: 86 20 144 STX V2
ES59: 40 145 RTS
146
ES5A: 4C 83 04 147 GOERR JMP INERR
148
ES5B: 20 40 E5 149 LINCOR JSR PLOTFMS ;Obter coordenadas A,B
ES60: E4 F0 150 CPX FIRST ;B(A)?
ES62: 90 08 151 BCS AT? ;Nao, desviar
ES64: AD F0 152 LDA FIRST ;Sim, trocar A por B
ES66: 85 2C 153 STA H2
ES68: 85 20 154 STA V2
ES6A: 86 F0 155 STX FIRST
ES6C: AF C5 156 AT? LDA Nat
ES6E: 20 E8 01 157 JSR SYNCHR ;Teste de sintaxe
ES71: 20 12 0A 158 JSR GETBYT
ES74: E0 30 159 CPX #530 ;Testar se < 48
ES76: 80 E2 160 BCS GOERR
ES78: 40 161 RTS
162
ES79: 20 40 E5 163 PLOT JSR PLOTFMS
ES7C: 8A 164 TXA
ES7D: A4 F0 165 LDY FIRST
ES7F: C0 28 166 CPY #520 ;Coordenada X tem que ser < 40
ES81: 80 07 167 BCS GOERR
ES83: 4C 00 F8 168 JMP PLOT2
169
ES84: 20 50 E5 170 HLIN JSR LINCOR
ES8F: 8A 171 TXA

```

```

ESB4: A4 2C 172      LDY H2
ESB5: C0 2B 173      CPY #52B      ;Coordenada X tem que ser ( 40
ESB6: 80 CA 174      BCS GOERR
ESB7: A4 FD 175      LDY FIRST
ESB8: 4C 19 FB 176      JMP HLNE      ;Ir a rotina do monitor
                177
ESB9: 20 50 E5 178      VLDX JSR LINCOOR
ESBA: 8A 179          TXA
ESBB: AB 180          TAY
ESBC: C0 2B 181      CPY #52B      ;Coordenada X tem que ser ( 40
ESBD: 80 9C 182      BCS GOERR
ESBE: A5 FD 183      LDA FIRST
ESBF: 4C 28 FB 184      JMP VLNE      ;Ir a rotina do monitor
                185
ESB4: C9 00 186      COLIN= CMP #500
ESB5: 00 03 187      BNE EVALVAR
ESB7: 20 B1 00 188      JSR CHGET
                189
                190      #Rotinas nao graficas
                191
ESBA: 20 12 DA 192      EVALVAR JSR GETBYT
ESAB: 8A 193          TXA
ESAC: 29 0F 194      AND #50F
ESAD: AA 195          TAX
ESA5: 29 03 196      AND #503
ESA6: 00 DA 197      BNE STCOR
ESA7: 8A 198          TXA
ESA8: 4A 199          LSR
ESA9: 4A 1100         LSR
ESAA: AA 1101         TAX
ESAB: 8A 1102         STCOR TXA
ESAC: 4C 6A FB 1103      JMP SETCOL
ESA5: 20 12 DA 1104      VLAB JSR GETBYT
ESAD: CA 1105         DEX
ESA7: 8A 1106         TXA
ESA8: C9 18 1107      CMP #518      ;Intervalo correto?
ESA9: 80 94 1108      BCS GOERR
ESAA: 4C 5C FB 1109      JMP TABV      ;Usar rotina do monitor
                1110
ESA9: 20 12 DA 1111      SPEED JSR GETBYT
ESAC: 8A 1112         TXA      ;1100 - SPEED -> SPEED2
ESA5: 49 FF 1113      EOR #5FF
ESA6: AA 1114         TAX
ESA7: EB 1115         JNE
ESA8: 86 F1 1116      STX SPEED2
ESA9: 60 1117         RTS
                1118
ESDA: 38 1119         TRACE SEC
ESDB: 90 1120         HEX Y0
ESDC: 18 1121         NOTRACE CLC
ESDE: 66 F2 1122      BOR TRIFLO
ESDF: 60 1123         RTS
                1124
ESDA: AF 00 1125      NORMAL LDA #500
ESDB: 2C 1126         HEX 2C
ESDC: AF 7F 1127      INVERSE LDA #57F
ESDE: A2 00 1128      MRA LDX #500

```

```

E9E1: 85 32 1129 SI STA INWFLB
E9E3: 84 F3 1130 STX ORNBASK
E9E5: 60 1131 RTS
1132
E9E6: 60 1133 RTS
1134
E9E7: 20 BF D0 1135 HINEN JSR FRMMH ;óter HINEN especificado
E9EA: 20 4C DA 1136 JSR GETADR
E9ED: A5 50 1137 LDA LINMM
E9EF: C5 60 1138 CMP STREMO
E9F1: A5 51 1139 LDA LINMM+1
E9F3: E5 6E 1140 SBC STREMO+1
E9F5: 80 03 1141 BCS SETHI ;Acima da tabela de variaveis?
E9F7: 4C FF C6 1142 JMH JMP NEWERR ;Nao erro
1143
E9FA: A5 50 1144 SETHI LDA LINMM
E9FC: 85 73 1145 STA NEWSTZ
E9FE: 85 6F 1146 STA FRETDP
E600: A5 51 1147 LDA LINMM+1
E602: 85 74 1148 STA NEWSTZ+1
E604: 85 70 1149 STA FRETDP+1
E606: 60 1150 RTS
339 PUT PARTE2C_02
31
E607: 20 BF D0 12 LOWEN JSR FRMMH
E60A: 20 4C DA 13 JSR GETADR
E60D: A5 50 14 LDA LINMM
E60F: C5 73 15 CMP NEWSTZ
E611: A5 51 16 LDA LINMM+1
E613: E5 74 17 SBC NEWSTZ+1
E615: 80 ED 18 BCS JMH
E617: A5 50 19 LDA LINMM
E619: C5 69 110 CMP VARTAB
E61B: A5 51 111 LDA LINMM+1
E61D: E5 6A 112 SBC VARTAB+1
E61F: 90 06 113 BCC JMH
E621: A5 50 114 LDA LINMM
E623: 85 69 115 STA VARTAB
E625: A5 51 116 LDA LINMM+1
E627: 85 6A 117 STA VARTAB+1
E629: 4C 83 C9 118 JMP CLEARC ;LOWEN Tiapa variaveis
119
E62C: A9 A8 120 SMERR LDA Wpoto
E62E: 20 CB D1 121 JSR SYNCHR
E631: A5 80 122 LDA TXTPTR
E633: 85 F4 123 STA TXTPSV
E635: A5 89 124 LDA TXTPTR+1
E637: 85 F5 125 STA TXTPSV+1
E639: 38 126 SEC
E63A: 6A 08 127 FOR ERRFLG
E63C: A5 75 128 LDA CURLIN
E63E: 85 F6 129 STA CURLSV
E640: A5 76 130 LDA CURLIN+1
E642: 85 F7 131 STA CURLSV+1
E644: 20 CB CC 132 JSR REIN ;Ignorar resto da linha
E647: 4C 80 CC 133 JMP ADD3H
134

```

```

135  Rotina de tratamento de erro se OMERR GOTO estiver ativo
136
E644: 86 DE 137 HANDLER STX ERRNUM
E64C: A6 F8 138     LDX RERRSTK ;Obter pointer da pilha
139     * ;guardando em NEWSTT
E64E: 86 DF 140     STX ERRSTK ;Testar
E650: A5 75 141     LDA CURLIN
E652: 85 84 142     STA ERRLIN
E654: A5 76 143     LDA CURLIN+1
E656: 85 88 144     STA ERRLIN+1
E658: A5 79 145     LDA OLDTEXT
E65A: 85 DC 146     STA ERRPOS
E65C: A5 7A 147     LDA OLDTEXT+1
E65E: 85 D0 148     STA ERRPOS+1
E660: A5 F4 149     LDA TXTPSV
E662: 85 88 150     STA TXTPTR
E664: A5 F5 151     LDA TXTPSA+1
E666: 85 89 152     STA TXTPTR+1
E668: A5 F6 153     LDA CURLSV
E66A: 85 75 154     STA CURLIN
E66C: A5 F7 155     LDA CURLSV+1
E66E: 85 76 156     STA CURLIN+1
E670: 20 87 D0 157     JSR CHRGOT
E672: 20 63 CC 158     JSR GOTO ;Ir a linha OMERR especificada
E674: 4C EC DA 159     JMP NEWSTT
160
E679: A5 DA 161     RESUME LDA ERRLIN
E67B: 85 75 162     STA CURLIN
E67D: A5 88 163     LDA ERRLIN+1
E67F: 85 76 164     STA CURLIN+1
E681: A5 DC 165     LDA ERRPOS
E683: 85 88 166     STA TXTPTR
E685: A5 D0 167     LDA ERRPOS+1
E687: 85 89 168     STA TXTPTR+1
E689: A6 DF 169     LDX ERRSTK ;Restaurar pointer da pilha ao
E68B: 9A 170     TXS ;valor que tinha antes do erro
E68C: 4C EC CA 171     JMP NEWSTT
172
E68F: 4C F1 D1 173     JSTN JMP SYNERR
174
E692: 80 F8 175     DEL BCS JSTN ;Erro se numero nao especific
E694: A6 AF 176     LDX PRGEND
E696: 86 69 177     STX WRTAB
E698: A6 80 178     LDX PRGEND+1
E69A: 86 6A 179     STX WRTAB+1
E69C: 20 31 CD 180     JSR LDMRET ;Obter comeco do intervalo
E69E: 20 31 C9 181     JSR FMDLIN ;Achar esta linha ou a proxima
E6A2: A5 98 182     LDA LONTR
E6A4: 85 6D 183     STA DEST
E6A6: A5 9C 184     LDA LONTR+1
E6A8: 85 61 185     STA DEST+1
E6AA: A9 2C 186     LDA #','
E6AC: 20 EB D1 187     JSR SYNCHR ;Testar sintaxe
E6AE: 20 31 CD 188     JSR LDMGET ;Obter fim do intervalo (nao
189     * ;fazer nada se fim do intervalo
190     * ; nao especificado)
E6B2: E6 50 191     INC LONTR ;Apontar depois dele, +1

```



```

E6B4: 00 02 192      BNE  #11
E6B6: E6 51 193      INC  LINDUM+1
E6B8: 20 31 C9 194    #11 JSR  FNDLIN      ;#char linha seguinte
E6BA: 45 98 195      LDA  LOWTR
E6BC: C5 40 196      CMP  DEST
E6BE: 45 9C 197      LDA  LOWTR+1
E6C0: E5 41 198      SBC  DEST+1
E6C2: 80 01 199      BCS  MOVDMW      ;#over, a nao ser que o
E6C4: 40      100      RTS              ;intervalo nao seja valido
      101
E6C6: 40 00 102 MOVDMW LDY  #500      ;#ov de LOWTR ate VARTAB a DEST
E6C8: B1 98 103 MOVIT  LDA  (LOWTR),Y
E6CA: 91 40 104      STA  (DEST),Y
E6CC: EA 98 105      INC  LOWTR
E6CE: 00 02 106      BNE  #12
E6D0: E6 9C 107      INC  LOWTR+1
E6D2: EA 40 108 #12  INC  DEST
E6D4: 00 02 109      BNE  #13
E6D6: E6 41 110      INC  DEST+1
E6D8: 45 49 111 #13  LDA  VARTAB
E6DA: C5 98 112      CMP  LOWTR
E6DC: 45 4A 113      LDA  VARTAB+1
E6DE: E5 9C 114      SBC  LOWTR+1
E6E0: 80 E6 115      BCS  MOVIT
E6E2: A6 41 116      LOX  DEST+1      ;#Fazer Y,X = DEST-1
E6E4: A4 40 117      LDY  DEST
E6E6: 00 01 118      BNE  #0X
E6E8: CA      119      OX
E6E9: 88      120 #0X  DEY
E6EA: B6 4A 121      STX  VARTAB+1    ;#apontar VARTAB ao ultimo
E6EC: 84 49 122      STY  VARTAB      byte movido
E6EE: 4C E1 C7 123  JMP  LINKSET     ;#Restaurar pointers apos apagar
      124
E6F1: A9 14 125 GR1  LDA  #514
E6F3: 85 22 126      STA  WNDTOP
E6F5: 20 58 FC 127  JSR  HOME
E6F7: 40      128      RTS
E6F9: 20 58 FC 129  OR   JSR  HOME
E6FB: 20 F1 E6 130  JSR  GR1
E6FD: 40 50 C0 131  LDA  TOCBLOW
E701: 40      132      RTS
E703: A0 51 C0 133  TEXT  LDA  IOPER
E705: A9 00 134      LDA  #500
E707: 80 22 00 135  STA: WNDTOP
E709: 20 58 FC 136  JSR  HOME
E70E: 40      137      RTS
      138
      139 ;#Rotina para guardar e restaurar arrays
      140 ;#nao usadas para disco
      141
E70F: 20 59 EB 142  STONE JSR  GETARYPT
E712: A0 80 143      LDY  #500
E714: B1 98 144      LDA  (LOWTR),Y
E716: AA      145      TAX
E717: 88      146      DEY
E719: B1 98 147      LDA  (LOWTR),Y
E71A: E9 01 148      SBC  #501

```

```

E71C: 80 01 1149      BCS  SLI
E71E: CA 1150      DEX
E71F: 85 50 1151  SLI  STA  LINNUM
E721: 86 51 1152      STA  LINNUM+1
E723: 20 00 FE 1153      JSR  WRITE
E726: 20 3C EB 1154      JSR  TAPEPHT
E729: 4C 00 FE 1155      JMP  WRITE
E72C: 20 59 EB 1156  RECALL JSR  GETARYPT
E72F: 20 FD FE 1157      JSR  READR
E732: A0 02 1158      LDY  #502
E734: B1 98 1159      LDA  (LOWTR),Y
E736: C5 50 1160      CMP  LINNUM
E738: CB 1161      INY
E739: B1 98 1162      LDA  (LOWTR),Y
E73B: E5 51 1163      SBC  LINNUM+1
E73D: B0 03 1164      BCS  NOK
E73F: 4C FF C6 1165      JMP  HONEKR
E742: 20 3C EB 1166  NOK  JSR  TAPEPAT
E745: 4C FD FE 1167      JMP  READR
                                     340  PUT  PARTE20,02
                                     11
                                     12  *Rotinas de inicializacao de alta resolucao
                                     13
E748: 20 58 FC 14  HGR2  JSR  HOME
E74B: A9 17 15      LDA  #517
E74D: 85 22 16      STA  WNDTOP
E74F: 20 58 FC 17  HGR1  JSR  HOME
E752: 2C 55 98 18      BIT  HGRFLG2
E755: A0 50 C0 19      LDA  TOCOLOR
E758: 4C 64 E7 110     JMP  WTTXT
E75B: 20 58 FC 111    HGR   JSR  HOME
E75E: 20 F1 E6 112     JSR  OR1
E761: 2C 54 98 113     BIT  HGRFLG1
E764: A0 50 C0 114    WTTXT LDA  TOCOLOR
E767: A5 48 115      LDA  POAH
E769: 85 E6 116      SETNPG STA  HPAG
E76B: A0 57 98 117     LDA  #957
E76E: A0 58 98 118     LDA  #958
E771: A9 00 119      HCLR  LDA  #500 ;Preparar fundo preto
E773: 85 1C 120      STA  HCOLOR1
E775: A5 E6 121      BKGD0 LDA  HPAG ;Inicializar memoria
E777: 85 18 122      STA  SHAPR  alta resolucao
E779: A0 00 123      LDY  #500
E77B: B4 1A 124      STY  SHAPL
E77D: A5 1C 125      BKGD01 LDA  HCOLOR1
E77F: F1 1A 126      STA  (SHAPL),Y ;Fazer alta resolucao=HCOLOR1
E781: 20 FE E7 127     JSR  CSWFT2 ;Corrigir desvio de cores
E784: CB 128      INY ;Tornar duas vezes mais
E785: 00 F6 129      BNE  BKGD01 lento o clear
E787: E6 18 130      INC  SHAPR
E789: A5 18 131      LDA  SHAPR
E78B: 29 1F 132      AND  #51F ;fim?
E78D: 00 EE 133      BNE  BKGD01
E78F: 60 134      RTS
                                     135
                                     136  *Rotinas de posicionamento e desenho alta resolucao
                                     137

```

```

E790: 85 E2 138 HPOSM STA Y0 ;Entrar com 1 em A
E791: 84 E0 139 STX X0L ; e XL em X
E794: 84 E1 140 STY X0H ; e XH em Y
E79a: 48 141 FNA
E797: 29 C0 142 AND #SC0
E79F: 85 2a 143 STA GBASL ;Para coordenada Y=00ASCDEF
E79b: 4A 144 LSR ;Calcular endereço base
E79C: 4A 145 LSR ;Ea EBASL,EBASH para acessar
E79D: 05 2a 146 ORA GBASL ;memoria da tela via (GBASL),Y
E79F: 85 2a 147 STA GBASL
E7A1: 48 148 PLA
E7A2: 85 27 149 STA GBASH
E7A4: 0A 150 ASL ;Calcular
E7A5: 0A 151 ASL ;GBASH=PPPF0HCD
E7A6: 0A 152 ASL ;GBASL=EABASHCD
E7A7: 24 27 153 ROL GBASH ;com PFF=001 PARA $2000 a $3FFF
E7A9: 0A 154 ASL ;e PFF=010 para $4000 a $BFFF
E7AA: 24 27 155 ROL GBASH ;dada coordenada Y=ABCEFGH
E7AC: 0A 156 ASL
E7AD: 4a 2a 157 ROR GBASL
E7AF: 45 27 158 LDA GBASH
E7B1: 29 1F 159 AND #S1F
E7B3: 00 E6 00 160 ORA: HPAG
E7B4: 85 27 161 STA GBASH
E7B8: 8A 162 TBA ;Dividir X0 por 7 para obter
E7B9: C0 00 163 CPT #600 indice a partir de endereço
E7B8: F0 05 164 BEQ HPOSM2 base (quociente) e bit dentro
165 * ;do byte de memoria da tela
E7B0: A0 23 166 LOY #623 ;mascara especific pelo resto
E7BF: 6F 04 167 AOC #604
E7C1: C0 168 HPOSM1 IMY
E7C2: E9 07 169 HPOSM2 SEC #607
E7C4: 80 FB 170 BCS HPOSM1
E7C4: 84 E5 171 STY HMOX ;Funciona para X0 de 0 a 279
E7C8: 4A 172 TAX ;byte de baixa ordem em X
E7C9: 80 3A E0 173 LDA HSBTBL-SF9,X
E7CC: 85 30 174 STA HMASK ;De alta ordem em Y na entrada
E7CE: 90 175 TZA
E7CF: 4A 176 LSR
E7D0: 45 E4 177 LDA HCOLORZ ;Se byte impar (carry ligado)
E7D2: 85 1C 178 STA HCOLOR01 ;Rodar os bits
E7D4: 80 28 179 BCS CSHT2
E7D4: 60 180 RTS
181
E7D7: 20 90 E7 182 HPL0T0 JSR HPOSM
E7D4: 45 1C 183 LDA HCOLOR01 ;Calcular pos do bit em EBAS
E7D1: 51 26 184 EOR (GBASL),Y ;HMOX e HMASK a partir
E7D5: 25 30 185 AND HMASK ;das coordenadas Y em A
E7D3: 51 26 186 EOR (GBASL),Y ;e X em X,Y
E7E2: 91 26 187 STA (GBASL),Y ;Para qualquer bit, substituir
E7E4: 60 188 RTS ;bit correspondente de HCOLOR01
189
190 #Rotinas de deslocamento para alta resolucao
191
E7E5: 10 23 192 LFTINT BPL RIGHT ;Usar sinal para rotina
E7E7: 45 30 193 LEFT LDA HMASK ;esquerda/direita
E7E9: 4A 194 LSR ;Deslocar 7 bits de baixa ordem

```

E7EA: 80 05	195		BCS LEFT1	de HMASK de uma posicao
E7EC: 49 00	196		ORA #5C0	
E7EE: 85 30	197	LRI	STA HMASK	
E7FD: 60	198		RTS	
	199			
E7F1: 88	1100	LEFT1	DEY	;Decrementar indice horizontal
E7F2: 10 02	1101		BPL LEFT2	
E7F4: A0 27	1102		LDT #527	;Dar a volta na tela
E7F6: 49 00	1103	LEFT2	LDA #5C0	;Movo HMASK, ponto mais a
E7F8: 85 30	1104	MEMWDX	STA HMASK	direita do byte
E7FA: 84 E5	1105		STY #60X	;Atualizar indice horizontal
E7FC: A5 1C	1106	CSHFT	LDA #COLOR1	
E7FE: 0A	1107	CSHFT2	ASL	;Rodar os 7 bits de baixa ordem
E7FF: C9 CD	1108		CMF #5C0	de #COLOR1 de 1 posicao
E801: 10 06	1109		BPL RTS1	;Desviar se branco ou preto
E803: A5 1C	1110		LDA #COLOR1	
E805: 49 7F	1111		ORA #57F	
E807: 85 1C	1112		STA #COLOR1	
E809: 60	1113	RTS1	RTS	
	1114			
E80A: A5 30	1115	RIGHT	LDA HMASK	
E80C: 0A	1116		ASL	;Deslocar os 7 bits de baixa
E80D: 49 80	1117		ORA #580	ordem de HMASK de um bit
E80F: 30 00	1118		BRI LRI	
E811: A7 81	1119		LDA #581	
E813: C8	1120		BNY	;Proximo byte
E814: C0 28	1121		CPY #528	
E816: 90 E0	1122		BCC MEMWDX	
E818: A0 00	1123		LDY #500	;Dar a volta
E81A: 80 0C	1124		BCS MEMWDX	;Sempre
E81C: 18	1125	LRUD11	CLC	;Nao ha rotacao de 90 graus
E81D: A5 D1	1126	LRUD12	LDA 01H	
E81F: 2F 0A	1127		AND #50A	;Se bit2=0 nao desenhav
E821: F0 25	1128		BEQ LRUD4	
E823: A9 7F	1129		LDA #57F	
E825: 25 30	1130		AND HMASK	
E827: 31 26	1131		AND (GBASL),Y	;Ligado bit da tela?
E829: D0 19	1132		BNE LRUD3	
E82B: E6 EA	1133		INC COLCOUNT	
E82D: A9 7F	1134		LDA #57F	
E82F: 25 30	1135		AND HMASK	
E831: 10 11	1136		BPL LRUD3	;Sempre
E833: 18	1137	LRUD1	CLC	;Nao ha rotacao de 90 graus
E834: A5 D1	1138	LRUD2	LDA 01H	
E836: 2F 0A	1139		AND #50A	;Se bit2=0 nao desenhav
E838: F0 0E	1140		BEQ LRUD4	
E83A: 81 26	1141		LDA (GBASL),Y	
E83C: A5 1C	1142		ORA #COLOR1	;Posicionar bit da tela de alta
E83E: 25 30	1143		AND HMASK	resolucao com #COLOR1 correto
E840: D0 02	1144		BNE LRUD3	;Se muda o bit da tela,
E842: E6 EA	1145		INC COLCOUNT	incrementar contador
E844: 51 26	1146	LRUD3	ORA (GBASL),Y	
E846: 91 26	1147		STA (GBASL),Y	
E848: A5 D1	1148	LRUD4	LDA 01H	;Somar quadrante ao vetor espec
E84A: A5 03	1149		ADC #00H	;E mover a esquerda, direita,
E84C: 29 03	1150		AND #503	alto do baixo conforme
	1151	END	=	*-1

```

EBE4: 09 02 1152      CNP  #502
EBE5: 6A             1153      ROR
EBE6: 80 92 1154      BCS  LTRT
EBE7: 30 30 1155      UPDOWN BHI  DOWN      ;Sinal para escolher alto/baixo
EBE8: 18             1156      LPO  CLC
EBE9: A5 27 1157      LDA  #BASH      ;Calcular end base para subir
EBEA: 2C 3A E9 1158      BIT  E01C      proxima linha em (BASH) dando
EBEB: 00 22 1159      BNE  UP4      a volta em 192
EBEC: 06 26 1160      ASL  #BASH
EBED: 80 1A 1161      BCS  UP2
EBEE: 2C 40 E8 1162      BIT  E03
EBEF: F0 05 1163      BEB  UP1
EBF0: 69 1F 1164      ADC  #51F      ;*** NAPA DE BITS ***
EBF1: 38             1165      SEC
EBF2: 80 12 1166      BCS  UP3      ;Para linha= ABCDEFGH
EBF3: 69 23 1167      UP1  ADC  #523
EBF4: 48             1168      PHA
EBF5: A5 26 1169      LDA  #BASH      ;BASH= FABAB000
EBF6: 69 80 1170      ADC  #580      ;BASH= PFFFF000
EBF7: 80 02 1171      BCS  UP5
EBF8: 69 F0 1172      ADC  #5FD      ;Onde PFF= 001 para pag 1
EBF9: 85 26 1173      UP5  STA  #BASH
EBFA: 58             1174      PLA
EBFB: 80 02 1175      BCS  UP3
EBFC: 69 1F 1176      UP2  ADC  #51F
EBFD: 66 26 1177      UP3  ROR  #BASH
EBFE: 69 FC 1178      UP4  ADC  #5FC
EBFF: 85 27 1179      UPDOWN STA #BASH
EB00: 60             1180      RTS
EB01: 341           1181      PUT  PARTEZE_02
EB02: 31
EB03: 18             1182      CLC
EB04: A5 27 1183      DOWN LDA #BASH
EB05: 69 04 1184      ADC  #504      ;Calcular end base para descer
EB06: 25           1185      E04  = #-1      proxima linha, dando
EB07: 2C 3A E9 1186      BIT  E01C      a volta em 192
EB08: 00 F3 1187      BNE  UPDOWN1
EB09: 06 26 1188      ASL  #BASH
EB0A: 90 19 1189      BCC  DOWN1
EB0B: 69 E0 1190      ADC  #5E0
EB0C: 18             1191      CLC
EB0D: 2C 88 E8 1192      BIT  E04
EB0E: F0 13 1193      BEB  DOWN2
EB0F: A5 25 1194      LDA  #BASH
EB10: 69 90 1195      ADC  #550
EB11: 49 F0 1196      EOR  #5FD
EB12: F0 02 1197      BEB  DOWN3
EB13: 49 F0 1198      EOR  #5FD
EB14: 85 26 1199      DOWN STA #BASH
EB15: AC E4 00 1200      LDA  #PAE
EB16: 90 02 1201      BCC  DOWN2
EB17: 69 E0 1202      DOWN ADC #5EC
EB18: 66 26 1203      DOWN ROR #BASH
EB19: 90 00 1204      BCC  UPDOWN1
EB1A: 25
EB1B: 26      *Rotinas para desenhar linhas em alta resolucao
EB1C: 27

```

EBB1:	48	128	HLINRL	PWA	
EBB2:	49	129		LDA	W500
EBB4:	49	130		STA	XDL
EBB6:	49	131		STA	XDM
EBB8:	49	132		STA	Y0
EBBA:	48	133		PLA	
EBBB:	48	134	HGLIN	PWA	;Na entrada :
EBBC:	48	135		SEC	;EL: A
EBBD:	49	136		SBC	XDL
EBBF:	48	137		PWA	;DM: X
EBD0:	48	138		TXA	;Y : Y
EBD1:	49	139		SBC	XDM
EBD3:	49	140		STA	GRMT
EBD5:	48	141		BDS	HLIN2
EBD7:	48	142		PLA	
EBD9:	49	143		EOR	W5FF
EBDA:	49	144		ADC	W5D1
EBDC:	48	145		PWA	;0= direita (DX positivo)
EBDD:	49	146		LOA	W500
EBDF:	49	147		SBC	GRMT
EBE1:	49	148	HLIN2	STA	DM
EBE3:	49	149		STA	EH
EBE5:	48	150		PLA	;Inicializar EL,H com
EBE6:	49	151		STA	XDL
EBE8:	49	152		STA	EL
EBEA:	48	153		PLA	
EBEB:	49	154		STA	XDL
EBED:	48	155		STA	XDM
EBEF:	48	156		TYA	
EBE1:	48	157		CLC	
EBE3:	49	158		SBC	Y0
EBE5:	49	159		BCC	HLIN3
EBE7:	49	160		EOR	W5FF
EBE9:	49	161		ADC	W5FE
EBEB:	49	162	HLIN3	STA	01
EBED:	49	163		STY	Y0
EBED:	48	164		ROR	GRMT
EBEF:	48	165		SEC	
EBF0:	49	166		SBC	00L
EBF2:	48	167		TAX	
EBF3:	49	168		LDA	W5FF
EBF5:	49	169		SBC	DM
EBF7:	49	170		STA	COUNTH
EBF9:	48	171		LDY	HW0X
EBFB:	48	172		BDS	MOVEX2
EBFD:	48	173	MOVEX	AGL	
EBFE:	20	174		JSR	LFR1
E901:	48	175		SEC	
E902:	48	176		LDA	EL
E904:	48	177	MOVEX2	ADC	01
E906:	48	178		STA	EL
E908:	48	179		LDA	EH
E90A:	49	180		SBC	W500
E90C:	48	181	MOVEX	STA	EH
E90E:	48	182		LDA	(0B4SL),Y
E910:	48	183		EOR	W500
E912:	25	184		AND	HW0X

;Fazer XDL,H e YD=0  
;Para desenhar linha  
;Relativas a (DX,DY)

;Na entrada :  
;EL: A  
;DM: X  
;Y : Y

;Calcular ABS (X-XD) em (DL,H)

;X para o bit de sinal do quadr  
;0= direita (DX positivo)  
;1= esquerda (DX negativo)

;Inicializar EL,H com  
;ABS (X-XD)= DELTX

;Coloc pointer no fim da linha

;Calcular -DELTX-1 em DY onde  
;DELTY = ABS (Y-YD)

;Rodar Y ate o bit de sinal  
do quadrante  
;(0= alto, 1=baixo)

;Inicializ COUNTL,H (COUNTL= X)  
;com -|DELTX + DELTY + 1|

;Indice horizontal  
;Sewers  
;Mover na direcao X  
;Usar bit 6 de GRMT para  
posicionar esquerda/direita

;Supoe carry ligado  
;Fazer (EL,H) - DELTX = (EL,H)  
;Nota: DY e (DELTY) - 1  
;Ose1 carro se (EL,H) fica NEG.  
;= ADC W5FF

;Desenhar ponto de WCOLORS

```

E914: 51 26 185      EOR  (GBASL),Y
E916: 91 26 186      STA  (GBASL),Y
E918: EB          187      INX
E919: D0 04 188      RHE  HLIM4
E91B: E6 10 189      INC  COUNTH
E91D: F0 62 190      BEQ  RTS2
E91F: A5 03 191      LDA  MORNIT
E921: 80 DA 192      BCS  MNVEX
E923: 20 53 EB 193      JSR  UPDOWN
E926: 18          194      CLC
E927: A5 04 195      LDA  EL
E929: 65 D0 196      ADC  OXL
E92B: 85 04 197      STA  EL
E92D: A5 05 198      LDA  EH
E92F: 65 D1 199      ADC  OXH
E931: 50 09 1100     BVC  HCOUNT
1101
E933: 81          1102     MSETBL  DFB  210000001
E934: 82          1103     DFB  210000010
E935: 84          1104     DFB  210000100
E936: 88          1105     DFB  210001000
E937: 90          1106     DFB  210010000
E938: A0          1107     DFB  210100000
E939: C0          1108     DFB  211000000
1109
E93A: 1C          1110     ERIC   HEX  1C
1111
1112     *Tabela de COS (90*X/16graus)*$100 - 1
1113     *com um byte de precisao, X de 0 a 16
1114
E93B: FF FE FA 1115     COSTBL  HEX  FFFFEFAF4ECC104C5
E93E: F4 EC E1 04 C5
E940: 84 A1 8D 1116     HEX  84A18D7B514F311B
E946: 78 A1 49 31 18
E94B: FF          1117     HEX  FF
1118
1119     *Rotina que restaura coordenadas de alta resolucao
1120
E94C: A5 26 1121     HFINO  LDA  GBASL
E94E: 0A          1122     ASL
E94F: A5 27 1123     LDA  GBASH
E951: 29 03 1124     AND  #503
E953: 2A          1125     ROL
E954: 05 26 1126     ORA  GBASL
E956: 0A          1127     ASL
E957: 0A          1128     ASL
E958: 0A          1129     ASL
E959: 85 E2 1130     STA  Y0
E95B: A5 27 1131     LDA  GBASH
E95E: 4A          1132     LSR
E95F: 4A          1133     LSR
E95F: 29 07 1134     AND  #507
E961: 05 E2 1135     ORA  Y0
E963: 85 E2 1136     STA  Y0
E965: A5 E5 1137     LDA  HMOX
E967: 0A          1138     ASL
E968: 65 E5 1139     ADC  HMOX

```

;Feitos os ptos (DELTX+DELTYY) ?

;Sim, voltar

;Para testar direcao

;Se carry lig, desenhara (EL,H)

;Se desligado negativo, mover

;(EL,H) + DELTX EH (EL,H)

;Sempre

\*Tabela de COS (90\*X/16graus)\*\$100 - 1

\*com um byte de precisao, X de 0 a 16

\*Rotina que restaura coordenadas de alta resolucao

;Converter endereço base

;A coordenada Y

;Para fazer GBASL= EABAB000

GBASH= PFFFFHCD

;Coordenada Y= ABCDEFH

;(FFF= pagina de tela)

;Converter HMOX (indice do

endereço base) e HMAX

;Caascara de bit) a coord

X em (XL,H), intervalo 0-\$133

```

E96A: 0A      1140      WSL
E96B: 0A      1141      TAX
E96C: 0A      1142      DEX
E96D: A5 3D   1143      LDA  HMASK
E96F: 29 7F   1144      AND  #57F
E971: E9      1145      HFTMD1 IMX
E972: 4A      1146      LSR
E973: 00 FC   1147      BNE  HFTMD1
E975: 85 E1   1148      STA  XDH
E977: 8A      1149      TXA
E97B: 18      1150      CLC
E979: A5 E5   1151      ADC  HNDX      ;Calcular HNDX*7 + LOG(HMASK),
E978: 90 02   1152      BCC  HFTMD2      base 2
E97D: E4 E1   1153      INC  XDH
E97F: 85 E0   1154      HFTMD2 STA  XDL
E981: 60      1155      RTS
E981: 60      1156
E981: 60      1157      ;Rotinas para desenhar formas em alta resolucao
E981: 60      1158
E982: 86 1A   1159      DRAM0 STX  SHAPL      ;Pointer a forma
E984: 84 18   1160      STY  SHAPEN
E986: AA      1161      DRAM1 TAX
E987: 4A      1162      LSR
E988: 4A      1163      LSR
E989: 4A      1164      LSR
E98A: 4A      1165      LSR
E98B: 85 03   1166      STA  #DRMT      ;DRMT 0= alto, 1= direita
E98D: 8A      1167      TXA      ;2=baixo, 3=esquerda
E98E: 29 DF   1168      AND  #50F
E990: 6A      1169      TAX
E991: 8C 38 E9 1170      LDY  COSTBL,X      ;Guardar COS e SIN em DIL e DY
E994: 84 00   1171      STY  DIL
E996: 49 0F   1172      EOR  #50F
E998: AA      1173      TAX
E999: 8C 3C E9 1174      LDY  COSTBL+1,X
E99C: C8      1175      INY
E99D: 84 02   1176      STY  DY
E99F: A4 E5   1177      LDY  HNDX      ;Indice do endereco base
E9A1: A2 00   1178      LOX  #500
E9A3: 86 EA   1179      STX  COLCOUNT      ;Limpar contador de colisoes
E9A5: A1 1A   1180      LDA  (SHAPL,X)      ;Primeiro byte da def de formas
E9A7: 85 D1   1181      DRAM2 STA  DGH
E9A9: A2 00   1182      LOX  #580
E9AB: 86 04   1183      STX  EL
E9AD: 86 05   1184      STX  EH      ;EL,H para vetores alto, baixo,
E9AF: A6 E7   1185      LOX  SCALEZ      ;esquerda, direita fracionarios
E9B1: A5 04   1186      DRAM3 LDA  EL      ;Fator de escala
E9B3: 38      1187      SEC      ;Se cosseno fracionario da
E9B4: A5 00   1188      ADC  DIL      overflow, mover na direcao
E9B6: 85 04   1189      STA  EL      do vetor especificado
E9B8: 90 04   1190      BCC  DRAM4
E9BA: 20 33 E8 1191      JSR  LRUD1
E9BD: 18      1192      CLC
E9BE: A5 05   1193      DRAM4 LDA  EH      ;Se o seno fracionario da
E9C0: A5 02   1194      ADC  DY      overflow mover na direcao
E9C2: 85 05   1195      STA  EH      especificada + 90 graus
E9C4: 90 03   1196      BCC  DRAM5

```



```

E9C6: 20 34 E8 1197 JSR LRU02
E9C9: CA 1198 DRAMS DEX ;Loop cont o fator de escala
E9CA: 00 E5 1199 BNE DRAM0
E9CC: A5 D1 1200 LDA D0H
E9CE: 4A 1201 LSR ;Proximos 3 bits do vetor de
E9CF: 4A 1202 LSR ;definicao de forma
E9D0: 4A 1203 LSR
E9D1: D0 D4 1204 BNE DRAM2 ;Falta este byte
E9D3: E6 1A 1205 INC SHAPEL
E9D5: D0 D2 1206 BNE N14
E9D7: E6 18 1207 INC SHAPEL+1
E9D9: A1 1A 1208 N14 LDA (SHAPEL,X) ;Pro: byte da def de formas
E9DB: D0 DA 1209 BNE DRAM2 ;Se 0, fim
E9DD: 60 1210 RTS
                                     142 PUT PARTE2F,02
                                     11
                                     12 ;Rotina para desenhar formas (XDRAM) em alta resolucao
                                     13
E9DE: 06 1A 14 XDRAM0 STX SHAPEL ;Pointer a forma
E9E0: 04 18 15 STY SHAPEH
E9E2: AA 16 XDRAM1 TAX ;A contem ROT (0 - $3F)
E9E3: 4A 17 LSR
E9E4: 4A 18 LSR
E9E5: 4A 19 LSR
E9E6: 4A 110 LSR
E9E7: 05 D3 111 STA GDRNT ;GDRNT D=alto, Indireita
E9E9: 0A 112 TZA ; Z=baixo, 3=esquerda
E9EA: 29 0F 113 AND #50F
E9EC: 4A 114 TAX
E9ED: 0C 38 E9 115 LDY COSTEL,X ;Guardar COS e SIN em D0L e D1
E9EF: 04 D0 116 STY D0L
E9F2: 49 0F 117 EOR #50F
E9F4: AA 118 TAX
E9F5: 0C 3C E9 119 LDY COSTEL+1,X
E9F8: C8 120 JMY
E9F9: 04 D2 121 STY DY
E9FB: A4 E5 122 LDY HNDX ;Indice do endereco base
E9FD: A2 D0 123 LDX #500
E9FF: 06 EA 124 STX COLCOUNT ;Ligar contador de colisoes
EA01: A1 1A 125 LDA (SHAPEL,X) ;Primeiro byte da def de formas
EA03: 05 D1 126 XDRAM2 STA D0H
EA05: A2 80 127 LDX #500
EA07: 06 D4 128 STX EL ;EL,H para vetores alto, baixo,
EA09: 06 D5 129 STX EH ;esquerda, direita fracionarios
EA0B: A6 E7 130 LDX SCALEZ ;Fator de escala
EA0D: A5 D4 131 XDRAM3 LDA EL
EA0F: 38 132 SEC ;Se cosseno fracionario da
EA10: 65 D0 133 ADC D0L ;overflow, mover na direcao
EA12: 05 D4 134 STA EL ;do vetor especificado
EA14: 90 D4 135 BCC XDRAM4
EA16: 20 1C E8 136 JSR LRU01
EA19: 18 137 CLC
EA1A: A5 D5 138 XDRAM4 LDA EH ;Se o seno fracionario da
EA1C: 65 D2 139 ADC DY ;overflow mover na direcao
EA1E: 05 D5 140 STA EH ;especificada + 90 graus
EA20: 90 D3 141 BCC XDRAM5
EA22: 20 1D E8 142 JSR LRU02

```

```

EA25: CA 143 XORAW5 BEX ;Loop cont o fator de escala
EA26: D0 E5 144 BNE XORAW5
EA2B: A5 D1 145 LDA BSH
EA2A: A8 146 LSR ;Proximos 3 bits do vetor de
EA2B: A6 147 LSR ; definicao de forma
EA2C: A4 148 LSR
EA2D: D0 D4 149 BNE XORAW2 ;Falta este byte
EA2F: E6 1A 150 INC SHAFEL
EA31: D0 D2 151 BNE N15
EA33: E6 1B 152 INC SHAFEL+1
EA35: A1 1A 153 N15 LDA (SHAFEL,X) ;Prox. byte da def. de fornaas
EA37: D0 CA 154 BNE XORAW2 ; se 0, fim
EA39: 60 155 RTS
156
157 ;Rotinas de analise usadas pelo BASIC para
158 ;acesso em alta resolucao
159
EA3A: 20 8F D0 160 HFMS JSR FFM0H
EA3D: 20 6C D4 161 JSR SETADR
EA40: A4 51 162 LDY L1MM0H+1 ;Obter coord horizontal em X,Y
EA42: A6 50 163 LDY L1MM0H
EA44: C0 D1 164 CPY #1280 ;Testar se ( Y > 280
EA46: 90 D6 165 BCC HFMS1
EA4B: D0 1D 166 BNE GGERR
EA4A: E0 1B 167 CPX #1280
EA4C: B0 19 168 BCS GGERR
EA4E: BA 169 HFMS1 TXA ;Guardar coord horiz na pilha
EA4F: 48 170 PHA
EA50: 98 171 TYA
EA51: 48 172 PHA
EA52: A9 2C 173 LDA #', '
EA54: 20 EB D1 174 JSR SYNCHR ;Testar sintaxe
EA57: 20 12 DA 175 JSR GETBYT ;Obter coordenada vertical
EA5A: E0 C0 176 CPX #92 ;Testar se ( X > 92
EA5C: B0 D9 177 BCS GGERR
EA5E: B6 90 178 STX DSCTMP ;Guardar
EA60: 68 179 PLA ;Recuperar coord horizontal
EA61: A8 180 TAY
EA62: 68 181 PLA
EA63: A4 182 TAX
EA64: A5 9D 183 LDA DSCTMP ;E coordenada vertical
EA66: 60 184 RTS ;Voltar ao prog que chamou
185
EA67: 4C 5A E5 186 GGERR JMP GGERR ;Quantidade valida
187
EA6A: C9 D0 188 HCOLOR CMP #520
EA6C: D0 D3 189 BNE EVMAR
EA6E: 20 B1 D0 190 JSR CHGET
191
EA71: 20 12 D4 192 EVMAR JSR GETBYT
EA74: E0 D8 193 CPX #520
EA76: B0 D7 194 BCS GGERR
EA7B: B0 A4 F8 195 LDA TABCOR,X
EA7B: B5 E4 196 STA HCOLORZ
EA7D: 60 197 RTS3 RTS
198
EA7E: C9 C1 199 HPLOT CMP #0 ;Pedido desenho contínuo?

```

```

EAB0: F0 D0 1100
EAB2: 20 3A EA 1101
EAB5: 20 07 E7 1102
EAB9: 20 07 00 1103 HP2
EAB9: C9 C1 1104
EAB0: 00 EE 1105
EABF: 20 EB 01 1106 HP3
EAB2: 20 3A EA 1107
EAB5: 84 90 1108
EAB7: A8 1109
EAB9: 8A 1110
EAB9: A6 90 1111
EAB9: 20 BB EB 1112
EABE: 4C BB EA 1113
      1114
EAA1: 20 12 0A 1115 ROT
EAA4: 86 F9 1116
EAA6: 60 1117
      1118
EAA7: 20 12 0A 1119 SCALE
EAA4: 86 E7 1120
EAA6: 60 1121
      1122
EAB0: 20 12 0A 1123 DUMPNT
EAB0: A5 EB 1124
EAB2: 85 1A 1125
EAB4: A5 E9 1126
EAB6: 85 1B 1127
EAB8: 0A 1128
EAB9: A2 00 1129
EAB9: C1 1A 1130
EAB0: F0 02 1131
EABF: 80 A6 1132
EAC1: 0A 1133 DP1
EAC2: 90 03 1134
EAC4: E6 1B 1135
EAC6: 1B 1136
EAC7: A8 1137 DP2
EAC8: 81 1A 1138
EACA: A5 1A 1139
EACC: AA 1140
EACD: C8 1141
EACE: 81 1A 1142
EAD0: A5 E9 1143
EAD2: 85 1B 1144
EAD4: 86 1A 1145
EAD6: 20 07 00 1146
EAD9: C9 C5 1147
EAD8: 00 09 1148
EAD0: 20 EB 01 1149
EAD0: 20 3A EA 1150
EAE3: 20 90 E7 1151
EAE6: A5 F9 1152 DP3
EAE8: 60 1153
      1154
EAE9: 20 AD EA 1155 DRAM
EAE3: 4C 86 E9 1156
      BEB HP3 ;Sim, desviar
      JSR HFMS ;Obter coord do pto de partida
      JSR HFL0T0 ;Desenha-la, preparando coord
      JSR CHR0T
      CMP #to ;linha especificado?
      BNE RTS3 ;Nao, sair
      JSR SYNCHR
      JSR HFMS ;Obter coord do fim da linha
      STY DSCTMP ;Preparar linha
      TAY
      TXA
      LDX DSCTMP
      JSR HOLIN ;Desenhar linha
      JMP HP2 ;Loop ate nao ter mais "T0"

      JSR GETBYT
      STX #0TZ
      RTS

      JSR GETBYT
      STX SCALEZ
      RTS

      JSR GETBYT ;Numero de forma especificado
      LDA SHAPEPMT
      STA SHAPEL
      LDA SHAPEPMT+1
      STA SHAPEH
      TXA
      LDX #000
      CMP (SHAPEL,X) ;Forma definida na tabela?
      BEQ DP1
      BCS GGERR ;Nao, erro
      ASL ;Achar end da forma na tabela
      BCC DP2
      INC SHAPEH
      CLC
      TAY
      LDA (SHAPEL),Y
      ADC SHAPEL
      TAX
      THY
      LDA (SHAPEL),Y
      ADC SHAPEPMT+1
      STA SHAPEH ;Guardar endereco da forma
      STX SHAPEL
      JSR CHR0T ;Testar sintaxe
      CMP #at
      BNE DP3
      JSR SYNCHR
      JSR HFMS ;Onde desenhar a forma?
      JSR HPOSN
      LDA #0TZ
      RTS

      JSR DUMPNT
      JMP DRAM1

```

```

EAF7: 20 AD EA 1157 XDRAM JSR DMPNT
EAF2: 4C E2 EY 1158 JMP XDRAM
      1159
      1160 *Carregar tabela de formas a partir de fita
      1161
EAF5: A9 D0 1162 SHLAD LDA #500
EAF7: 85 30 1163 STA A1H
EAF9: 85 3F 1164 STA A2H
EAFB: A0 50 1165 LDY #LIMNUM
EAFD: 84 3C 1166 STY A1L
EAF7: C8 1167 INY
EB00: 84 3E 1168 STY A2L
EB02: 20 FD FE 1169 JSR READM ;Ler tamanho em LIMNUM
EB05: 18 1170 CLC
EB06: A5 73 1171 LDA #ENSIZ
EB08: AA 1172 TAX
EB09: CA 1173 OEX
EB0A: 86 3E 1174 STX A2L ;#ENSIZ-1 -> pointer ao fim
EB0C: E5 50 1175 SBC #LIMNUM ;#ENSIZ-2 tamanho -> pointer
EB0E: 48 1176 PHA ;ao começo
EB0F: A5 74 1177 LDA #ENSIZ+1
EB11: A8 1178 TAX
EB12: EB 1179 TXN
EB13: D0 D1 1180 RNE SL1
EB15: B0 1181 DEY
EB16: 84 3F 1182 SL1 STY A2H
EB18: E5 51 1183 SBC #LIMNUM+1
EB1A: C5 6E 1184 CMP #STREND+1
EB1C: 90 D2 1185 BCC SL2
EB1E: D0 D3 1186 BNE SL3 ;Pagina deve ser 1 #STREND
EB20: 4C FF C6 1187 SL2 JMP #MEMBR
EB23: 85 74 1188 SL3 STA #ENSIZ+1 ;Por em #MEMBR o pointer as
EB25: 85 70 1189 STA #FRETOP+1 formas valor da tabela de
EB27: 85 30 1190 STA A1H formas
EB29: 85 E9 1191 STA #SHAPEPNT+1
EB2B: 68 1192 PLA
EB2C: 85 EB 1193 STA #SHAPEPNT
EB2E: 85 73 1194 STA #ENSIZ
EB30: 85 6F 1195 STA #FRETOP
EB32: 85 3C 1196 STA A1L
EB34: 20 FA FC 1197 JSR #RDBEIT ;Achar aresta
EB37: A9 D3 1198 LDA #503
EB39: 4C D2 FF 1199 JMP #READ2 ;Ler tabela de formas em #A1L
      343 PUT #PARTE20,02
      11
EB3C: 18 12 TAPEPNT CLC
EB3D: A5 98 13 LDA #LOWTR
EB3F: A5 50 14 ADC #LIMNUM
EB41: 85 3E 15 STA A2L
EB43: A5 9C 16 LDA #LOWTR+1
EB45: A5 51 17 ADC #LIMNUM+1
EB47: 85 3F 18 STA A2H
EB49: A0 D4 19 LDY #504
EB4B: 81 98 110 LDA #LOWTR)_Y
EB4D: 20 D9 D4 111 JSR #GETARY2
EB50: A5 94 112 LDA #HIGH3
EB52: 85 3C 113 STA A1L

```

```

EB54: A5 95 114      LDA HIGHDS+1
EB56: B5 30 115      STA ALH
EB58: 60 116      RTS
117
118      ;Motina nao grafica
119
EB59: A9 40 120      GETAR1PT LDA #540      ;Chamada por STORE e RECALL
EB5E: B5 14 121      STA SUBFLG
EB50: 20 03 03 122     JSR PTRGET
EB40: A9 00 123      LDA #500
EB42: B5 14 124      STA SUBFLG
EB44: 4C 15 0C 125     JMP UAR710
126
127      ;Observar que se #MOLEFT nao for 0,HTAB pode
128      ;imprimir fora da tela, isto e, no programa
129
EB67: 20 12 DA 130     HTAB JSR GETBYT
EB6A: CA 131      DEX
EB6B: BA 132      TXA
EB6C: C9 20 133     HTAB1 CMP #52B
EB6E: F0 DA 134      BEQ HTAB2
EB70: E9 2B 135     SBC #52B
EB72: 4B 136      PHA
EB73: 20 20 0E 137     JSR CR00
EB76: 6B 138      PLA
EB77: 4C 6C ED 139     JMP HTAB5
EB7A: B5 24 140     HTAB2 STA CH
EB7C: 60 141      RTS
142
EB7D: 20 12 04 143     MOTOR JSR GETBYT
EB81: BA 144      TXA
EB81: C9 04 145     CMP #504      ;E water ?
EB83: F0 02 146     BEQ AQUE?    ;Sim
EB85: B0 6B 147     BCS JSWTRB
EB87: C9 02 148     AQUE? CMP #502      ;Motor A es B ?
EB89: F0 02 149     BEQ MOTRA
EB8B: 69 01 150     ADC #501      ;Se motor B soma 1+carry
EB8D: 6A 151     MOTRA TAX
EB8E: B0 52 00 152     BNE MOTAOFF,8 ;Liga/desl A ou B
EB91: 60 153      RTS
154
EB92: 20 07 00 155     NL/AD JSR CRGOT
EB95: C9 41 156     CMP #541      ;Compara proxima letra com A
EB97: 00 05 157     BNE CHKIFLT
EB99: E6 B6 158     INC TXPTR
EB9B: 4C EE C8 159     JMP LOAD
EB9E: 20 EE EB 160     CHKIFLT JSR CHKT      ;Check se a proxima letra e T
EBA1: A4 B6 161     LDY TXPTR
EBA3: 20 EF EE 162     JSR KWORO
EBA6: B4 B6 163     STY TXPTR
EBA8: 20 AE ED 164     JSR LOADT
EBAB: A0 B6 04 165     LDA PDR1
EBAE: 4C B9 04 166     LDY PDR1+1
EBB1: B5 67 167     STA BASLOC
EBB3: B4 B6 168     STY BASLOC+5 ;Indica o inicio do programa
EBB7: A0 BA 04 169     LDA PER2
EBB8: 4C BA 04 170     LDY PER2+1

```

EB8B: 85 6F	171		STA	WARTAB	;Indica o fim do programa
EB8D: 84 6A	172		STY	WARTAB+1	
EB8F: 85 AF	173		STA	BASLOC1-1	
EBC1: 84 8D	174		STY	BASLOC1	
EBC3: 6D	175		RTS		
	176				
EBC4: 2D B7 0D	177	MSAVE	JSR	CHRGOT	
EBC7: C9 41	178		CMF	#541	;Compara proxima letra com A
EBC9: 8D 05	179		BNE	CHKIFST	
EBCB: E6 8B	180		INC	TXPTR	
EBCD: 4C 05 C8	181		JMP	SAVE	;SAVE no formato APPLE
EBD0: 2D EE EB	182	CHKIFST	JSR	CHKT	;Compara se a proxima letra e T
EBD2: A5 67	183		LOA	BASLOC	
EBD5: 44 6B	184		LOY	BASLOC+1	
EBD7: 85 3C	185		STA	AIL	
EBD9: 84 3D	186		STY	ALR	
EBDB: A5 AF	187		LOA	BASLOC1-1	
EBDD: 44 8D	188		LOY	BASLOC1	
EBDF: 85 3E	189		STA	AZL	
EBE1: 84 3F	190		STY	AZH	
EBE3: 44 8B	191		LOY	TXPTR	
EBE5: 2D EF EE	192		JSR	KNWRD	
EBE8: 84 8B	193		STY	TXPTR	
EBEA: 2D 9A EC	194		JSR	SHRET	
EBED: 6D	195		RTS		
	196				
EBEE: C9 54	197	CHKT	CMF	#554	;Compara com T
EBF0: FD 03	198		BEQ	ACKINH	
EBF2: 4C F1 01	199	JSMERR	JMP	SYNERR	
EBF5: E6 8B	100	ACKINH	INC	TXPTR	;Avanca keyword input buffer
EBF7: 6D	101		RTS		
	102				
EBF8: A9 A0	103	MP	LOA	#540	;End alto da segunda pag.de video
EBFA: 85 4B	104		STA	PGW	
EBFC: A0 55 C0	105		LOA	DISPAG2	;Chaveia segunda pag. de video
EBFF: 6D	106		RTS		
	107				
EC00: A9 2D	108	MA	LOA	#520	;End alto da prim. pag. de video
EC02: 85 4B	109		STA	PGW	
EC04: A0 54 C0	110		LOA	DISPAG1	;Chaveia primeira pag. de video
EC07: 6D	111		RTS		
	112				
EC08: FF FF	113		HEX	FFFF	
EC0A: 85 F8	114	COU73	STA	COOL	;Salva caracter de saida
EC0C: A5 47	115		LOA	PGV	
EC0E: FD 05	116		BEQ	ISVD	;PGV=0 ?
EC10: A5 FB	117		LOA	COOL	
EC12: 4C 09 08	118		JMP	PR09	
	119				
EC15: A0 52 04	1120	ISVD	LOA	GRAFFLG	;Ocorreu um #F2 antes?
EC18: 0D 2D	1121		BNE	CHGRF	;Sim,entao e caracter grafico
EC1A: A5 49	1122		LOA	VCTB	;Ocorreu um CTRL-B antes?
EC1C: 4D 02	1123		BNE	ISCTB?	;Sim estao va p/ ISCTB?
EC1E: A5 FB	1124		LOA	COOL	;Fegar caractere
EC20: 3D 3E	1125		BMI	ISCTB?	;Se ASCII
EC22: 2D EA F7	1126		JSR	SELEFAX	;Seleciona graf.ou palavras BASIC
EC25: 8D 05	1127		BCS	ISGRF?	;Se graficos

EC27:	A5 F8	>128	LDN	COOL	:Pega caracter
EC29:	4C 49 F7	>129	JMP	ISCBAS	:Imprime comandos BASIC
		>130			
EC2E:	A5 49	>131	ISGRF?	LDN	VCTB
EC2E:	06 18	>132	BNE	ISRET?	
EC30:	A5 F8	>133	LDN	COOL	
EC32:	C9 34	>134	WP	MS34	:Se caracter !=34,entao ISRET?
EC34:	80 12	>135	BOS	ISRET?	
EC36:	CA	>136	BEX		:Se esta entre 1E e 34
EC37:	4C 3A FF	>137	JMP	BELL	gera um BELL
		>138			
EC3A:	A5 F8	>139	CHGRF	LDN	COOL
EC3C:	29 7F	>140	AND	MS7F	:Transforma em positivo ASCII
EC3E:	38	>141	SEC		
EC3F:	E9 40	>142	SBC	MS40	:Caracter graficos
EC41:	85 F8	>143	STA	COOL	
EC43:	A9 00	>144	LDN	MS00	
EC45:	80 52 04	>145	STA	GRAFFLG	
EC48:	A5 F8	>146	ISRET?	LDN	COOL
EC4A:	C9 80	>147	CMF	MS80	:E RETURN?
EC4C:	00 00	>148	BNE	HTRET	
EC4E:	40 82 04	>149	LDN	FIMLIN	
EC51:	F0 08	>150	BEQ	HTRET	
EC53:	20 A2 FC	>151	JSR	CR	
EC56:	A9 00	>152	LDN	MS00	
EC58:	80 82 04	>153	STA	FIMLIN	
EC5B:	A5 F8	>154	HTRET	LDN	COOL
EC5D:	4C A9 F1	>155	JMP	ICCTP	:Saida de caracter
EC60:	A5 F8	>156	ISCTB?	LDN	COOL
EC62:	C9 82	>157	CMF	MS82	:Compara com CTRL-B
EC64:	F0 04	>158	BEQ	ISCTB	
EC66:	C9 F2	>159	CMF	MSF2	
EC68:	08 C2	>160	BNE	ISGRF?	
EC6A:	A9 01	>161	LDN	MS01	
EC6C:	80 52 04	>162	STA	GRAFFLG	
EC6F:	60	>163	RTS		
		>164			
EC70:	A9 01	>165	ISCTB	LDN	MS01
EC72:	45 49	>166	EQN	VCTB	:Inverte VCTB
EC74:	85 49	>167	STA	VCTB	
EC76:	4C 2C EC	>168	JMP	ISGRF?	
		>169			
EC79:	A2 FF	>170	PRPREG	LDX	MSFF
EC7B:	0E 88 CD	>171	STX	PRF2	
EC7E:	EB	>172	DNZ		:Zera A46, PGVL, VCTB
EC7F:	84	>173	TXA		
EC80:	85 46	>174	STA	A46	
EC82:	85 47	>175	STA	PGVL	
EC84:	05 49	>176	STA	VCTB	
EC86:	20 0C F1	>177	JSR	ZPGA	:Zera pagina 4
EC89:	60	>178	RTS		
		>179			
EC8A:	A0 41 04	>180	TESFLG	LDN	TEST41
EC8D:	F0 03	>181	BEQ	SET442	
EC8F:	4C CA F1	>182	JMP	COMTDR	
EC92:	A0 01	>183	SET442	LDY	MS01
EC94:	8C 42 04	>184	STY	TEST42	

```

EC97: 4C C2 F1 1185      JMP  NTCTP
      >1186
EC9A: 20 9F ED 1187      SASET  JSR  TFM
EC9B: 28      1188      SEC
EC9E: A5 3F 1189      LDA  A2H
EC9D: E5 30 1190      SBC  A1H
ECA2: 80 46 04 1191      STA  BUF+4
ECA5: A5 3E 1192      LDA  A2L
ECA7: E5 3C 1193      SBC  A1L
ECA9: 90 03 1194      BCC  IF80W      ;Se borrow
ECAB: EE 46 04 1195      INC  BUF+6
ECAE: A2 00 1196      IF80W  LDY  #50
ECB0: 8E 50 04 1197      STX  TEST50
ECB3: CA      1198      DEX
ECB4: BE 67 04 1199      STX  BUF+7
ECB7: BE 51 04 1200      STX  TEST51
ECBA: AF 24 1201      LDA  #524
ECBC: AD 20 1202      MXZER10 LDY  #520
ECBE: 20 06 FC 1203      JSR  ZEROLY
ECC1: 80 F9 1204      BNE  MXZER00
ECC3: 6F FE 1205      ADC  #5FE
ECC5: 80 F5 1206      BCS  MXZER00
ECC7: 20 19 ED 1207      M80SW  JSR  PRP
ECCA: AF 40 1208      LDA  #540
ECCC: A0 50 1209      MXZER01 LDY  #550
ECCE: 20 06 FC 1210      JSR  ZEROLY
ECC1: 6F FE 1211      ADC  #5FE
ECDB: 80 F7 1212      BCS  MXZER01
ECC5: A0 21 1213      LDY  #521
ECD7: 20 06 FC 1214      JSR  WRBIT      ;Escribe un bit
ECDA: EE 67 04 1215      INC  BUF+7
ECDD: A0 67 04 1216      LDA  BUF+7
ECED: 80 53 04 1217      STA  TEST53
ECE3: AF FF 1218      LDA  #5FF      ;Inicializa con #5FF
ECC5: A0 22 1219      LDY  #522
ECE7: 20 24 ED 1220      JSR  M80W
ECEA: 48      1221      PHA      ;Salva CHRSUM
ECED: 20 30 ED 1222      JSR  CSST1
ECEE: 48      1223      PLA      ;Recupera CHRSUM
ECDF: 20 24 ED 1224      JSR  M80W      ;Escribe
ECF2: A0 21 1225      LDY  #521
ECF4: 20 ED FE 1226      JSR  WRBYTE      ;Escribe un CHRSUM
ECF7: A0 67 04 1227      LDA  BUF+7
ECFA: F0 08 1228      BEQ  M80SW
ECFC: A5 3C 1229      LDA  A1L
ECFE: 80 70 04 1230      STA  TEST70
ED01: A5 30 1231      LDA  A1H
ED03: 80 71 04 1232      STA  TEST71
ED06: A0 50 04 1233      LDA  TEST50
ED09: 00 02 1234      BNE  FINSWE
ED0B: F0 0A 1235      BEQ  M80SW
ED0D: A2 03 1236      FINSWE LDY  #503      ;Imprime a message "OK"
ED0F: A0 00 1237      LDY  #500
ED11: 20 F3 FA 1238      JSR  MENS
ED14: AF 07 1239      LDA  #587
ED16: 4C 09 FB 1240      JMP  BELLO
      1241

```



```

ED19: A9 60 1242 FRP LDA #60
ED1B: 85 3C 1243 STA A1L
ED1D: A9 04 1244 LDA #04
ED1F: 85 3D 1245 STA A1H
ED21: A9 67 1246 LDA #67
ED23: 85 3E 1247 STA A2L
ED25: A9 04 1248 LDA #04
ED27: 85 3F 1249 STA A2H
ED29: 60 1250 RTS
      1251
ED2B: A2 00 1252 WPRM LDX #00
ED2C: 41 3C 1253 EOR (A1L,X) ;OKSUM
ED2E: 40 1254 PHA
ED2F: A1 3C 1255 LDA (A1L,X)
ED31: 20 E0 FE 1256 JSR #FFFE ;Escreve um byte
ED34: 20 BA FC 1257 JSR #FBAF ;Proximo At ate AT=42
ED37: A0 10 1258 LDY #10
ED39: 68 1259 PLA
ED3A: 90 EE 1260 BCC WPRM
ED3C: 60 1261 RTS
      1262
ED3D: 10 1263 CSST1 CLC
ED3E: A0 53 04 1264 LDA TEST53
ED41: 00 0E 1265 BNE CSST2
ED43: A9 68 1266 LDA #68
ED45: 85 3E 1267 STA A2L
ED47: A0 1F 1268 LDY #1F
ED49: A0 51 04 1269 LDA TEST51
ED4C: 00 02 1270 BNE RETM
ED4E: A0 35 1271 LDY #35
ED50: 60 1272 RETM1 RTS
      1273
ED51: A0 1A 1274 CSST2 LDY #1A
ED53: A0 51 04 1275 LDA TEST51
ED56: 00 02 1276 BNE CSST3
ED58: A0 33 1277 LDY #33
ED5A: A0 70 04 1278 CSST3 LDA TEST70
ED5D: 85 3C 1279 STA A1L
ED5F: A9 FF 1280 ADC #FF
ED61: 85 3E 1281 STA A2L
ED63: A0 71 04 1282 LDA TEST71
ED66: 85 30 1283 STA A1H
ED68: A9 00 1284 ADC #00
ED6A: 85 3F 1285 STA A2H
ED6C: C0 73 04 1286 CMP TEST73
ED6F: F0 18 1287 BEQ CSST4
ED71: 90 20 1288 BCC RETM2
ED73: A0 0F 1289 CSST4 LDY #0F
ED75: A0 51 04 1290 LDA TEST51
ED78: 00 02 1291 BNE CSST5
ED7A: A0 31 1292 LDY #31
ED7C: A0 73 04 1293 CSST5 LDA TEST73
ED7F: 85 3F 1294 STA A2H
ED81: A0 72 04 1295 LDA TEST72
ED84: 85 3E 1296 STA A2L
ED86: A9 01 1297 LDA #01
ED88: 80 50 04 1298 STA TEST50

```

```

ED8B: A0      >299          RTS
ED8C: A0 10   >300 CSST6  LDY #510
ED8E: A0 51 04 >301          LDA TEST51
ED91: 00 02   >302          BNE CSST7
ED93: A0 32   >303          LDY #532
ED95: A0 72 04 >304 CSST7  LDA TEST72
ED98: C5 3E   >305          CMP A2L
ED9A: 90 07   >306          BCC CSST4
ED9C: F0 05   >307          BEQ CSST4
ED9E: A0      >308 RETM2   RTS
                >309
ED9F: A0 03   >310 TFM     LDY #503
EDA1: 89 9C 00 >311 MOX     LDA: AIL,Y
EDA4: 99 70 04 >312          STA TEST70,Y
EDA7: 99 68 04 >313          STA PGR1,Y
EDA9: 88      >314          DEY
EDAB: 10 F4   >315          BPL MOX
EDAD: A0      >316          RTS
                >317
EDA1: A0 05   >318 LOADT  LDY #505
EDB0: 89 60 04 >319 KILNAM  LDA BUF,Y
EDB3: 99 74 04 >320          STA TEST74,Y
EDB6: 80      >321          DEY
EDB7: 10 F7   >322          BPL KILNAM
EDB9: CB      >323          IMY
EDBA: 8C 50 04 >324          STY TEST50
EDBC: 8C 53 04 >325          STY TEST53
EDBD: 8C 56 04 >326          STY TEST56
EDC3: A2 FF   >327 INTSUM   LDX #5FF ;Inicializa CHKSUM com 5FF
EDC5: B6 2E   >328          STX CHKSUM
EDC7: 20 19 00 >329          JSR PRP
EDCA: A2 10   >330 PGMOM   LDX #510 ;Prepara para 1 byte
EDCC: A0 25   >331 CPLTBIT LDY #525
EDCE: 20 F0 FC >332          JSR ROBIT ;Le meio ciclo
EDD1: 90 F7   >333          BCC PGMOM ;É bit zero?
EDD3: CA      >334          BEX
EDD4: D0 F6   >335          BNE CPLTBIT
EDD6: 20 31 EE >336          JSR INPMOM
EDD9: 20 FA FC >337          JSR RO2BIT ;Le 2 meios ciclos
EDDB: A0 24   >338 LECL    LDY #524
EDDE: 20 F0 FC >339          JSR ROBIT ;Le meio ciclo
EDE1: 80 F9   >340          BCS LECL
EDE3: 20 F0 FC >341          JSR ROBIT ;Le meio ciclo
EDE6: A0 3A   >342          LDY #53A
EDD8: 20 20 EE >343          JSR LECHK ;Le byte
EDDB: 20 30 ED >344          JSR CSST5
EDEE: 20 20 EE >345          JSR LECHK ;Le byte
EDF1: A0 53 04 >346          LDA TEST53
EDF4: F0 C0   >347          BEQ INTSUM ;Le programa
EDF6: A0 34   >348          LDY #534
EDF8: 20 EC FC >349          JSR ROBYTE
EDFB: C5 2E   >350          CMP CHKSUM
EDFD: F0 D8   >351          BEQ OKCHKSUM
EDFF: A9 A0   >352          LDA #5A0 ;Imprime espaço
EE01: 20 ED F0 >353          JSR CONT
EE04: 4C 20 FF >354          JMP PRERR ;Imprime erro
EE07: A5 3C   >355 OKCHKSUM LDA AIL

```

```

EE09: 80 70 04 1354 STA TEST70
EE0C: A5 30 1357 LDA A5H
EE0E: 00 71 04 1358 STA TEST71
EE11: A0 50 04 1359 LDA TEST50
EE14: F0 A0 1360 BEB INTSRM
EE16: 20 31 EE 1361 JSR INPNAH
EE19: A9 14 1362 LDA #514
EE1B: 05 24 1363 STA CH
EE1D: 4C 00 ED 1364 JMP F0MSWE
                                     PUT PARTE2H,02
                                     344
                                     31
EE20: 20 EC FC 12 LECHK JSR R0RYTE ;Le programa e checa
EE23: 01 3C 13  STA (AHL,X)
EE25: 45 2E 14  EOR CHKSUM
EE27: 05 2E 15  STA CHKSUM
EE29: 20 BA FC 16  JSR NEXTAI
EE2C: A0 35 17  LDY #35
EE2E: 90 F0 18  BCC LECHK
EE30: 60 19 19  RTS
                                     310
EE31: A9 17 111 INPNAH LDA #517 ;Cursor na ultima linha
EE33: 05 25 112 STA CV
EE35: A0 01 113 LDY #01
EE37: 84 24 114 STY CH
EE39: 88 115 DEY
EE3A: 89 60 04 116 CPLTWOE LDA BUF,Y ;Imprime o nome do programa
EE3D: 20 ED FD 117 JSR COUT
EE40: C8 118 INY
EE41: C0 06 119 CPY #06
EE43: 00 F5 120 BNE CPLTWOE
EE45: A9 A0 121 INPSP LDA #5A0 ;Imprime espaco
EE47: 20 ED FD 122 JSR COUT
EE4A: 89 60 04 123 LDA BUF,Y ;Imprime comprimento do programa
EE4D: 20 04 FD 124 JSR PR0YTE
EE50: C8 125 INY
EE51: C0 08 126 CPY #08
EE53: 00 F0 127 BNE INPSP ;Imprime 2 bytes hex
EE55: A0 67 04 128 LDA BUF+7
EE58: 00 29 129 BNE RETK3
EE5A: A0 74 04 130 LDA TEST74
EE5D: C9 A0 131 CMP #5A0
EE5F: F0 0F 132 BEA INPWAIT
EE61: A0 00 133 LDY #00
EE63: 89 74 04 134 COMPNAME LDA TEST74,Y ;Compara nome do programa
EE66: 09 A0 04 135 CMP BUF,Y
EE69: 00 1C 136 BNE RETK3
EE6B: C8 137 INY
EE6C: C0 06 138 CPY #06
EE6E: 00 F3 139 BNE COMPNAME
EE70: A9 01 140 INPWAIT LDA #01
EE72: 80 53 04 141 STA TEST53
EE75: A2 05 142 LDX #05 ;Imprime " WAIT"
EE77: A0 03 143 LDY #03
EE79: 20 F3 FA 144 JSR MEMS
EE7C: A0 03 145 LDY #03
EE7E: 89 68 04 146 ATLCOP LDA PGR1,Y
EE81: 99 70 04 147 STA TEST70,Y

```

EE84:	88	148		DECY		
EE85:	10 F7	149		BPL	ATLCOF	;Atualiza comprimento do progr.
EE87:	60	150	RETRG	RTS		
		151				
EE88:	84 F8	152	TOSUB1	STY	COOL	;Armazena pos.de CRTBL em COOL
EE8A:	64 34	153		LDT	YSW	;T=posicao do buffer IN
EE8C:	E6 34	154		TMC	YSW	
EE8E:	C9 06	155		CMP	MSDA	;Verif. se e M (move)
EE90:	F0 0C	156		BEQ	ACOMPA1	;Esta acompanhado de A ?
EE92:	C9 E8	157		CMP	MSER	;Verifica se e R (read)
EE94:	F0 31	158		BEQ	ACOMPA2	;Esta acompanhado de A ?
EE96:	C9 F0	159		CMP	MSFD	;Verifica se e W (write)
EE98:	F0 1F	160		BEQ	ACOMPA3	;Esta acompanhado de A
EE9A:	C6 34	161		DEC	YSW	;Volta posicao do buffer IN
EE9C:	80 40	162		BNE	TSB	;TSB,se nao e posicao zero
EE9E:	89 00 02	163	ACOMPA1	LDA	IN,T	;Verif.se caracter de IN, e A ?
EEA1:	C9 C1	164		CMP	MSC1	
EEA3:	00 06	165		BNE	WRFP	;Nao, verifica se e P ?
EEA5:	20 00 EC	166		JSR	MA	;Se e MA, seta prim.pag.de video
EEA8:	4C E7 EE	167		JMP	ZM001	
EEAB:	C9 00	168	WRFP	CMP	MSD0	;Compara com P
EEAD:	F0 04	169		BEQ	STMP	;Se NP, seta seg.pag.de video
EEAF:	C6 34	170		DEC	YSW	;Volta posicao de IN
EEB1:	00 28	171		BNE	TSB	;Se nao posicao zero, TSB
EEB3:	20 F8 EB	172	STMP	JSR	NP	;Seta segunda pag. de video
EEB4:	4C E7 EE	173		JMP	ZM001	
EEB9:	89 00 02	174	ACOMPA2	LDA	IN,T	;Busca caracter de IN
EEBC:	C9 C1	175		CMP	MSC1	;Se for A,armazena no stack
EEBE:	F0 1E	176		BEQ	TSB	o endereco da rotina de WRITE
EEC0:	20 EF EE	177		JSR	KWORD	;Senao armazena o end. de SAVET
EEC3:	A2 01	178		LDX	MS01	coloca nome do progr.em BUF
EEC5:	80 0C	179		BNE	PHAAOR	
EEC7:	89 00 02	180	ACOMPA2	LDA	IN,T	;Pega caracter de IN
EECA:	C9 C1	181		CMP	MSC1	;Se for A,armazena no stack
EECC:	F0 18	182		BEQ	TSB	o end. da rotina de READ
EECE:	20 EF EE	183		JSR	KWORD	;Coloca nome do progr.em BUF
EEE1:	A2 03	184		LDX	MS03	armazena o end. de LSHOT
EEE3:	80 F8 FD	185	PHAAOR	LDA	SHVHEAD,X	
EEE6:	48	186		PHA		
EEE7:	CA	187		DEX		
EEE8:	80 F8 FD	188		LDA	SHVHEAD,X	
EEE9:	48	189		PHA		
EEEC:	00 09	190		BNE	ZM001	
EEED:	A4 F8	191	TSB	LDT	COOL	;T=posicao do CRTBL
EEED:	AF FE	192	TOSUB	LDA	MSFE	;Prepara end. alto das rotinas
EEE2:	48	193		PHA		
EEE3:	89 E1 FF	194		LDA	SUBTBL,Y	;End. baixo das rotinas
EEE6:	48	195		PHA		
EEE7:	A5 31	196	ZM001	LDA	MS0E	;Modo anterior
EEE9:	A0 00	197	ZM00	LDT	MS00	;Zera modo
EEE9:	84 31	198		STY	MS0E	
EEED:	60	199		RTS		
		1100				
EEEE:	C8	1101	KWORD1	SNY		;Proximo caracter do buffer IN
EEEF:	89 00 02	1102	KWORD	LDA	IN,T	
EEF2:	09 80	1103		ORA	MS00	;Transforma em NASCII
EEF4:	C9 A0	1104		CMP	MSA0	;Verifica se e espaco

EEF6:	F0 F6	1105	BEQ	KWORD1	;Sim,busca novo caracter	
EEF8:	C9 A2	1106	CMF	NSA2	;Verifica se e aspa	
EEFA:	00 01	1107	BNE	NSASP		
EEFC:	C0	1108	IMY		;Sim,incrementa buffer	
EEFD:	A2 00	1109	MOASP	LDX	NS00	;Carrega nome em BUF
EEFF:	E0 07	1110	NRCHNAM	CPX	NS07	;E fim do BUF ?
EF01:	F0 19	1111	BEQ	ZBUFNAME		
EF03:	B9 00 02	1112	LDA	IN,Y		
EF06:	09 00	1113	ORA	NS80	;Transforma em NASCII	
EF08:	C9 80	1114	CMF	NS80	;Se RETURN zera BUF	
EF0A:	F0 10	1115	BEQ	ZBUFNAME		
EF0C:	C9 00	1116	CMF	NS80		
EF0E:	F0 0C	1117	BEQ	ZBUFNAME		
EF10:	C9 A2	1118	CMF	NSA2	;Se for aspa,completa com espaco	
EF12:	F0 07	1119	BEQ	CMPLSP		
EF14:	90 60 04	1120	STA	BUF,X	;Coloca caracter em BUF	
EF17:	E8	1121	IMX			
EF18:	C8	1122	IMY			
EF19:	00 E4	1123	BNE	NRCHNAM		
EF1B:	C8	1124	CMPLSP	IMY	;Proximo caracter de IN	
EF1C:	E0 08	1125	ZBUFNAME	CPX	NS08	;Se fim do BUF
EF1E:	F0 08	1126	BEQ	FINNAMH	salva posicao de IN e retorna	
EF20:	AF A0	1127	LDA	NSA0	;preenche BUF com espaco	
EF22:	90 60 04	1128	STA	BUF,X		
EF25:	E8	1129	IMX			
EF26:	00 F4	1130	BNE	ZBUFNAME		
EF28:	84 34	1131	FINNAMH	STY	YSW	
EF2A:	60	1132	RTS			
		1133				
EF2B:	20 7F F8	1134	BUSCON	JSR	POCC	;prepara indexacao para cor
EF2E:	AA	1135	TAX			
EF2F:	80 70 F4	1136	LDA	TABCSCR,X	;busca byte de cor	
EF32:	60	1137	RTS			
		1138				
EF33:	A0 59 C0	1139	EXT101	LDA	PR1STBH	;Printer strobe high
EF36:	A0 51 C0	1140	LDA	IOPER	;Preto e branco	
EF39:	AF 20	1141	LDA	NS20	;Endereca primeira pag. de video	
EF3B:	85 48	1142	STA	PGWH		
EF3D:	AF 05	1143	LDA	NS05	;Envia dado	
EF3F:	80 88 C0	1144	STA	PRF2		
EF42:	A0 20	1145	LDP	NS20	;Gera um atraso	
EF44:	80	1146	LOOP20	DET		
EF45:	00 F0	1147	BNE	LOOP20		
EF47:	A2 10	1148	LDX	NS10	;Loop de 50F vezes enquanto	
EF49:	CA	1149	PRESF	DEX	;SC080 for diferente de 505	
EF4A:	F0 08	1150	BEQ	FINESP		
EF4C:	A0 80 C0	1151	LDA	PRF1		
EF4F:	C9 05	1152	CMF	NS05		
EF51:	00 F6	1153	BNE	PRESF		
EF53:	AF 01	1154	LDA	NS01		
EF55:	00 02	1155	BNE	TENCANT		
EF57:	AF 00	1156	FINESP	LDA	NS00	
EF59:	85 47	1157	TENCANT	STA	PGAL	;Prepara PGM. com 0 ou 1
EF5B:	AF FF	1158	LDA	NSFF		
EF5D:	80 88 C0	1159	STA	PRF2		
EF60:	AF 00	1160	LDA	NS00		
EF62:	80 F0 9F	1161	STA	TSCANT		

EF65:	AD FD 9F	1162	LDN	TSCANT		
EF68:	FD 00	1163	BEQ	ZTSUTIL		
EF6A:	AD F1 9F	1164	LDN	INDCMT		
EF6D:	85 D6	1165	STA	LOCK		
EF6F:	AD F2 9F	1166	LDN	INDCMT+1		
EF72:	85 D7	1167	STA	LOCK+1		
EF74:	AD FD 9F	1168	LDN	TSCANT		
EF77:	80 25 D4	1169	ZTSUTIL	STA	TSUTIL	
EF7A:	20 A8 C2	1170	JSR	TESTCAP		
EF7D:	60	1171	RTS			
		1172				
EF7E:	08	1173	CANBUF	PHP	:Salva status no stack	
EF7F:	48	1174	PHA		:Salva acumulador no stack	
EF80:	C9 90	1175	CMF	MSPO	:Compara com CTRL-F	
EF82:	FD 12	1176	BEQ	ISCTP	:Se e CTRL-F	
EF84:	A5 47	1177	TEPVAL	LDN	PGVL	
EF86:	D0 D9	1178	BNE	NOVID	:Testa PGM.	
EF88:	68	1179	PLA		:Recupera acumulador	
EF89:	28	1180	PLP		:Recupera status	
EF8A:	9D 00 D2	1181	STA	IN,X	:Armazena no buffer IN	
EF8D:	E8	1182	INX		:Proxima posicao do buffer	
EF9E:	4C 54 FD	1183	JMP	BSROUT		
EF91:	68	1184	NOVID	PLA	:Recupera acumulador	
EF92:	28	1185	PLP		:Recupera status	
EF93:	4C 06 D8	1186	JMP	PR56		
EF96:	AD 41 D4	1187	ISCTP	LDN	TEST41	
EF99:	C9 D2	1188	CMF	MSO2	:Se \$441=2, -> TEPVAL	
EF9B:	FD E7	1189	BEQ	TEPVAL		
EF9D:	68	1190	PLA		:Recupera acumulador	
EF9E:	28	1191	PLP		:Recupera status	
EF9F:	4C 54 FD	1192	JMP	BSROUT		
EFA2:	D8	1193	TSPQVL	PHP	:Salva status no stack	
EFA3:	48	1194	PHA		:Salva acumulador no stack	
EFA4:	A5 47	1195	LDN	PGVL	:Testa PGM.	
EFA6:	D0 D5	1196	BNE	NOVID1	:Se diferente de 0	
EFA8:	68	1197	PLA		:Recupera acumulador	
EFA9:	28	1198	PLP		:Recupera status	
EFAA:	4C 82 FD	1199	JMP	WFLIM	:Com status e acc intactos	
EFAE:	68	1200	NOVID1	PLA	:Recupera acumulador	
EFAE:	28	1201	PLP		:Recupera status	
EFAF:	4C 0F D8	1202	JMP	PR6F		
EFB2:	D8	1203	SVID	PHP	:Salva status no stack	
EFB3:	48	1204	PHA		:Salva acumulador no stack	
EFB4:	AD 66 D4	1205	LDN	TEST86	:Se \$48A=2 -> POSCH6	
EFB7:	C9 D2	1206	CMF	MSO2		
EFB9:	D0 D5	1207	BNE	RECAP	:Se diferente de 2	
EFBB:	68	1208	PLA		:Recupera acumulador	
EFBC:	28	1209	PLP		:Recupera status	
EFBD:	4C 0E FD	1210	JMP	POSCH6		
EFCD:	68	1211	RECAP	PLA	:Recupera acumulador	
EFCE:	28	1212	PLP		:Recupera status	
EFC2:	8E 80 D4	1213	STX	TEST8D		
EFC5:	4C C0 FD	1214	JMP	CHRTXT		
		1215				
EFCB:	A5 26	1216	PLOT1	LDN	GBASL	:Le coordenada de linha
EFCB:	85 26	1217	PLOT0	STA	GBASL	:Armazena coordenado de linha
EFC0:	84 27	1218	STY	GBASH		:Armazena coordenado de coluna

EFCE: 0A	1219		TXA		;Salva X e Y no stack
EFCE: 4B	1220		PHA		
EFDE: 9B	1221		TTA		
EFDE: 4B	1222		PHA		
EFDE: A5 30	1223		LDA COLOR		;Indexa em X um valor de 0-7
EFDE: 29 07	1224		AND #507		para buscar a cor dada por COLOR
EFDE: AA	1225		TAX		
EFDE: A5 27	1226		LDA BRASH		;Se a coluna a ser plot. e impar,
EFDE: 4A	1227		LSR		;seta carry
EFDE: 8D AA F8	1228		LDA TABCOR,X		;Carrega o byte a ser plotado
EFDE: 90 10	1229		BCC NIMWSEP		;Se a coluna for par, armazena
EFDE: 49 7F	1230		EOR #57F		o byte em RMDH,
EFDE: 85 4F	1231		STA RMDH		senao inverte os bytes de imagem
EFDE: 29 7F	1232		AND #57F		
EFDE: C9 00	1233		CMP #500		;Se for cor branca ou preta,
EFDE: F8 04	1234		BEQ DESIMV		carrega em RMDH sem inverter
EFDE: C9 7F	1235		CMP #57F		
EFDE: D0 04	1236		BNE LINCORD		
EFDE: 49 7F	1237	DESIMV	EOR #57F		
EFDE: 85 4F	1238	NIMWSEP	STA RMDH		
EFDE: A5 26	1239	LINCORD	LDA BRASH		;Coordenada da linha no acaual.
EFDE: 20 03 F8	1240		JSR PLOTH		;Prep.o end. base da microlinha 1
EFDE: 18	1241		CLC		
EFDE: A5 0B	1242		LDA R4L		
EFDE: 45 27	1243		ADC BRASH		;Posiciona o end.da coluna em R4L
EFDE: 85 08	1244		STA R4L		
EFDE: A0 00	1245		LDY #500		;Zera indexador Y
EFDE: A2 04	1246		LDX #504		;Prep.X para contar 4 loops
F0DE: A5 4F	1247	GRPLOT	LDA ENDH		;Coloca o dado na memoria
F0DE: 91 00	1248		STA (R4L),Y		da pagina de video
F0DE: 18	1249		CLC		
F0DE: A5 09	1250		LDA R4H		;Endereco a proxima microlinha
F0DE: 69 04	1251		ADC #504		
F0DE: 85 09	1252		STA R4H		
F0DE: CA	1253		DEX		;Repete 4 vezes
F0DE: D0 F2	1254		BNE GRPLOT		
F0DE: 68	1255		PLA		;retorna Y e X do stack
F0DE: A8	1256		TAY		
F0DE: 68	1257		PLA		
F0DE: AA	1258		TAX		
F0DE: 60	1259		RTS		
	345		PUT PARTE21,02		
	31				
F0DE: A9 00	12	IN3	LDA #500		;Torna \$480 negativo
F0DE: 8D 8D 04	13		STA TEST8D		
F0DE: 00 05	14		BNE SVYS		;Desvia para SVYS
F0DE: A9 00	15	IN4	LDA #500		;Torna \$480 positivo
F0DE: 8D 8D 04	16		STA TEST8D		
F0DE: 9B	17	SVYS	TYA		;Salva Y no stack
F0DE: 4B	18		PHA		
F0DE: A0 05	19	PROCTCL	LDY #505		
F0DE: A9 02	110	ATBDUC	LDA #502		;Bera atraso p/ elim. bouncing
F0DE: 20 9E 0A	111		JSR MATR		
F0DE: 20 43 F8	112		JSR SCAM1		;Varre teclado
F0DE: E6 4E	113		INC RMDL		;Incrementa o par (RMD)
F0DE: D0 02	114		BNE MATWC		
F0DE: E6 4F	115		INC RMDH		

F030:	20	04	116	MHINC	BIT	TESTBD	;Testa S480
F035:	30	04	117		BRI	TESTCL	;Se entrou por IN0,TESTCL
F037:	80	EY	118		BOS	PROCTCL	;Se tem tecla apertada, PROCTCL
F039:	90	02	119		BCC	KEY	;Se nao tem tecla apertada,KEY
F03B:	90	E5	120	TESTCL	BCC	PROCTCL	;Se nao tem tecla apertada
F03D:	88		121	KEY	DEY		;Decrementa ate a quinta vez
F03E:	00	E4	122		BNE	ATBOUC	;Volta para testar teclado
F040:	68		123		PLA		;Salva Y no stack
F041:	A8		124		TAY		
F042:	60		125		RTS		
			126				
F043:	80	SE	127	SCAN1	STA	BTCHTO	;Reseta bit CONTROL
F046:	84		128		TAA		;Salva X e Y no stack
F047:	48		129		PMA		
F048:	98		130		TYA		
F049:	48		131		PMA		
F04A:	A9	00	132		LDA	#500	;Zera GRASH
F04C:	85	27	133		STA	GRASH	;Caso contador de caracteres
F04E:	A9	01	134		LDA	#501	;Habilita linha 1 da matriz
F050:	85	26	135		STA	GRASL	
F052:	A2	08	136		LDX	#508	;Prepara X para contar 8 linhas
F054:	A5	26	137	HABLIN	LDA	GRASL	
F056:	80	00	138		STA	KBOUT	;Envia dado para o teclado
F059:	A0	10	139		LDA	KBTPIN	;Le dado do teclado (resposta)
F05C:	A0	06	140		LDY	#506	;Prepara Y para contar 6 colunas
F05E:	6A		141	KKCOL	ROR		;Testa bit atraves do carry
F05F:	90	03	142		BCC	KKKEY	;Mao e a tec.apert.,novo teste
F061:	20	81	143		JSR	IDFSCTL	;Se encontrou tecla apertada
F064:	E6	27	144	KKKEY	INC	GRASH	;Increment. contador do caracter
F066:	88		145		DEY		;Decrementa coluna
F067:	00	F5	146		BNE	KKCOL	;Se nao foi a ultima coluna
F069:	18		147		CLC		;Limpa carry
F06A:	26	26	148		ROL	GRASL	;Habilita proxima linha
F06C:	CA		149		DEX		;Decrementa contador de linha
F06D:	00	E5	150		BNE	HABLIN	;Mao foi a ultima linha,repita
F06F:	18		151		CLC		;Acabou teste, limpa carry
F070:	68		152	SAMYX	PLA		;Recupera Y e X do stack
F071:	A8		153		TAY		
F072:	68		154		PLA		
F073:	AA		155		TAX		
F074:	A9	00	156	CLRKEY	LDA	#500	;Limpa registrador KBOUT
F076:	80	00	157		STA	KBOUT	
F079:	90	03	158		BCC	RT1	
F07B:	20	05	159		JSR	SEPTAM	
F07E:	60		160	RT1	RTS		
F07F:	68		161	RT2	PLA		;Recupera a resposta do teclado
F080:	60		162		RTS		
			163				
F081:	48		164	IDFSCTL	PMA		;Salva resp.do teclado no stack
F082:	A5	27	165		LDA	GRASH	;Se e tecla SHIFT
F084:	F0	FY	166		BEB	RT2	;Retorna e continua teste
F086:	68		167		PLA		;Retira a resposta do stack
F087:	A9	00	168		LDA	#500	;Limpa KBOUT
F089:	80	00	169		STA	KBOUT	
F08C:	A9	01	170		LDA	#501	;Habilita a linha 1
F08E:	80	00	171		STA	KBOUT	
F091:	A0	10	172		LDA	KBTPIN	



F094: 6A	173	RON		:Testa col. 1, linha 1 (SHIFT)
F095: 90 06	174	BCC SETCTL		:Se nao e SHIFT
F097: A5 27	175	LDA GBASH		:Seta bit 7 do cont. de carac.
F099: 09 80	176	ORA #580		
F09B: 85 27	177	STA GBASH		
F09D: A9 08	178	SETCTL LDA #500		:Linha KBOUT
F09F: 80 00 C0 179		STA KBOUT		
F0A2: 80 5F C0 180		STA BITCNT1		:Seta CONTROL
F0A5: A0 10 C0 181		LDA KBTPIN		:Le KBIN
F0A8: 6A	182	RON		:Testa tecla control
F0AA: 90 06	183	BCC SCMEMD		:Nao e CONTROL, va para SCMEMD
F0AB: A5 27	184	LDA GBASH		:seta bit 6
F0AD: 09 40	185	ORA #540		:0o contador de caracteres
F0AF: 85 27	186	STA GBASH		
F0B1: 68	187	SCMEMD PLA		
F0B2: 68	188	PLA		
F0B3: 38	189	SEC		:Seta carry
F0B4: 80 8A	190	BCC SANYX		
F0B4: 84 24	191	PROVND STY CH		:Posiciona cursor horizontal
F0B8: 48	192	PROV01 PHA		:Salva acumulador no stack
F0B9: C9 80	193	CMF #580		:E RETURN?
F0BB: F0 40	194	BEQ FIM1		:Sim,retorna a rotina principal
F0BD: 4C 82 EF 195		JMP SVID		:Nao,rotina de saida de dados
F0CD: A2 F2	196	CHRTXT LDX #IGERCHR		:Endereco alt.da tab.de geracao
F0CE: 86 07	197	STX R3H		de caracteres de texto em R3H
F0C4: C9 80	198	CMF #580		:Se for ASCII -> TEXTO
F0CA: 80 06	199	BCC TEXTO		
F0CB: E9 80	100	SBC #500		
F0CA: A2 F4	101	LDX #IGERERF		:Endereco alt.da tab.de geracao
F0CE: 86 07	102	STX R3H		de caracteres graficos em R3H
F0CE: AE 80 04 103		TEXTO LDX TESTBD		
F0D1: 29 3F	104	AND #52F		:Seta valor entre 0 e 63
F0D3: 0A	105	ASL		:Multiplica por 8
F0D4: 0A	106	ASL		
F0D5: 0A	107	ASL		
F0D6: 85 06	108	STA R3L		
F0D8: A5 07	109	LDA R3H		
F0DA: 69 00	110	ADC #500		
F0DC: 85 07	111	STA R3H		
F0DE: 20 F3 FB 112		PROCHR JSR CV08		:Mult .CV por 8 ,resul .aa DELH
F0E1: 8A	113	TXA		:Salva X e Y no stack
F0E2: 48	114	PHA		
F0E3: 98	115	TYA		
F0E4: 48	116	PHA		
F0E5: A2 08	117	LDX #508		:i como contador
F0E7: A0 00	118	LDY #500		de 8 microlinhas de video
F0E9: 20 73 FC 119		IMPCH JSR EPVID		:prepara R4 como endereco base
F0EC: 18	120	CLC		: da pagina de video
F0ED: A5 08	121	LDA R4L		:Posiciona endereco de video
F0EF: A5 24	122	ADC CH		para coluna desejada
F0F1: 85 08	123	STA R4L		
F0F3: A5 09	124	LDA R4H		
F0F5: 69 00	125	ADC #500		
F0F7: 85 09	126	STA R4H		
F0F9: 81 06	127	LDA (R3L),Y		:Pega byte na tab.de enderecos
F0FB: 45 32	128	EOR IMWFLS		:Nodo normal ou inverso?
F0FD: 91 08	129	STA (R4L),Y		:Seta pagina de video

F0FF:	E6 E3	1130	INC	DELH	;Proxima microlinha
F101:	E6 06	1131	INC	R3L	;Proximo byte da tabela
F103:	CA	1132	DEX		;Microlinha 8 ?
F104:	00 E3	1133	BNE	JMPCN	;Mas, imprime proxima microlinha
F106:	68	1134	PLA		;Recup.Y e X do stack e retorna
F107:	48	1135	TAY		
F108:	68	1136	PLA		
F109:	AA	1137	TAX		
F10A:	68	1138	FINL	PLA	
F10B:	40	1139	RTS		
		1140			
F10C:	90 00 04	1141	ZPS4	STA	FG4,X
F10F:	E8	1142	INX		
F110:	00 FA	1143	RPZB4	BNE	ZPS4
F112:	89 20	1144	LDA	WS20	;PGR#820, seta pag 1 de video
F114:	85 48	1145	STA	PGM	;seta pag. de video
F116:	60	1146	RTS		
		1147			
F117:	40 82 04	1148	COOCC	LDA	FINLIN
F11A:	00 0E	1149	BNE	TS446	;se FINLIN >0 va para TS446
F11C:	40 41 04	1150	LDA	TEST41	
F11F:	F0 06	1151	BEQ	JNCR	;se S445=0 fazer va CR
F121:	48	1152	TAY		;Y#4
F122:	A6 25	1153	LDX	CV	;X=CV
F124:	20 00 C2	1154	JSR	JMPR	
F127:	4C 62 FC	1155	JNCR	JMP	CR
F12A:	A5 46	1156	TS446	LDA	A46
F12C:	C9 03	1157	CMR	WS03	;Se A46=3 JMP \$R036
F12E:	F0 03	1158	BEQ	JCART6	;Se A46<3 JMP \$B15
F130:	4C 15 08	1159	JMP	PRG15	
F133:	4C 06 90	1160	JCART6	JMP	CART6
		346	PUT	PARTE2J,02	
F136:	A5 24	11	SCROLL	LDA	CH
F138:	48	12	PMA		;Salva posicao de CH no stack
F139:	A5 22	13	LDA	WMDTOP	
F13B:	85 06	14	STA	R3L	;Coordenada do topo da tela
F13D:	E6 06	15	INC	R3L	em R3L
F13F:	20 F5 F8	16	JSR	A08	
F142:	20 73 FC	17	JSR	EPGVID	;Prepara DELH
F145:	A4 08	18	LDT	R4L	;Calcula o endereco base da
F147:	A5 09	19	LDA	R4H	pag de video em R4, a partir
F149:	18	110	CLC		de DELH
F14A:	69 20	111	ADC	WS20	;Prepara BASL e BASH com o end.
F14C:	AA	112	TAX		de mem.corresp.a microlinha 1
F14D:	38	113	SCRL1	SEC	;do topo da tela
F14E:	84 28	114	STY	BASL	;Calcula o endereco fisico de
F150:	8A	115	TXA		memoria correspondente a
F151:	E9 20	116	SBC	WS20	microlinha 1 da proxima
F153:	85 29	117	STA	BASH	linha de texto
F155:	98	118	TXA		
F156:	69 7F	119	ADC	WS7F	
F158:	A8	120	TAX		
F159:	8A	121	TXA		
F15A:	E9 1F	122	SBC	WS1F	
F15C:	AA	123	TAX		
F15D:	29 04	124	AND	WS04	
F15F:	F0 0A	125	BEQ	SCRL2	

F161:	98	126		TYA	
F162:	C9 50	127		CMF	#150
F164:	F0 32	128		BE9	CLRVL
F166:	69 28	129		ADC	#52B
F168:	A8	130		TAY	
F169:	A6 48	131		_LDX	P99H
F168:	B6 28	132	SCRL2	STX	BAS2H ;armazena o endereço em
F168:	B4 28	133		STY	BAS2L e BAS2H
F16F:	A2 08	134		LDX	#50B ;Prep.contador de 8 microlinhas
F171:	A4 20	135	SCRL3	LDY	WORDLFT ;No inicio da linha horizontal
F173:	B1 28	136	SCRL4	LDA	(BAS2L),Y ;transf.a microlinha de video
F175:	91 28	137		STA	(BASL),Y de (BAS2L) para (BASL)
F177:	C8	138		IMY	
F178:	C4 21	139		CPY	WORDHTRH ;fim da linha horizontal ?
F17A:	00 F7	140		BNE	SCRL4 ;Continua transferindo a linha
F17C:	18	141		CLC	;transfere a proxima microlinha
F17D:	A9 04	142		LDA	#504 ate a microlinha 8
F17F:	65 28	143		ADC	BAS2H
F181:	85 28	144		STA	BAS2H
F183:	A9 04	145		LDA	#504
F185:	65 29	146		ADC	BASH
F187:	85 29	147		STA	BASH
F189:	CA	148		DEX	
F18A:	00 E5	149		BNE	SCRL3 ;Se nao e a microlinha 8
F18C:	A4 28	150		LDY	BAS2L
F18E:	A6 28	151		LDX	BAS2H
F190:	E6 06	152		INC	R3L ;Desce uma linha de texto
F192:	A5 06	153		LDA	R3L
F194:	C5 23	154		CMF	WORDHTRH ;É ultima linha de texto?
F196:	09 85	155		BNE	SCRL1 ;Nao,proxima linha de texto
F198:	A0 00	156	CLRVL	LDY	#500 ;limpa a ultima linha de texto
F19A:	20 98 FC	157		JSR	CLEOLZ
F19D:	A6 23	158		LDX	WORDHTRH
F19F:	CA	159		DEX	
F1A0:	B6 25	160		STX	CV ;Posiciona o CV na ultima linha
F1A2:	68	161		PLA	;Retorna a posicao do CH
F1A3:	85 24	162		STA	CH
F1A5:	60	163		RTS	
		164			
F1A6:	4C 0A EC	165	CONT2	JMP	COUT3 ;Saida de dados pelo video
F1A9:	B4 35	166	IBCTP	STY	YSW1S ;Salva Y
F1AB:	B6 4E	167		STX	RNDL ;Salva X
F1AD:	4E	168		PNA	;Salva acumulador no stack
F1AE:	20 7D F8	169		JSR	VIDWALT ;Impriam caracter
F1B1:	68	170		PLA	;Recupera acumulador do stack
F1B2:	C9 00	171		CMF	#580 ;Verifica se e ua RETURN
F1B4:	F0 11	172		BE9	SETCR ;Se for RETURN -> SETCR
F1B6:	C9 90	173		CMF	#590 ;E CONTROL-P ?
F1B8:	00 08	174		BNE	HTCTP ;Nao, -> HTCTP
F1BA:	28 82 F8	175		JSR	POSCUR
F1B9:	A9 00	176		LDA	#500
F1BF:	4C E3 F1	177		JMP	ESPERA
F1C2:	A6 4E	178	HTCTP	LDX	RNDL
F1C4:	A4 35	179		LDY	YSW1S
F1CA:	60	180		RTS	
		181			
F1C7:	4C 8A EC	182	SETCR	JMP	TESPL6

```

F1C4: 08 183  COWCR  PHP           ;Salva status no stack
F1C8: A5 46 184  LDA  A46           ;Se A46=3 -> SF1F5
F1CC: C9 03 185  CMP  #503
F1CF: FD 24 186  BEQ  JCART9
F1D1: C9 04 187  CMP  #504           ;Se A46=4 -> SF1F9
F1D3: FD 24 188  BEQ  JPRG18
F1D5: 20 189  PLP
F1D6: A6 25 190  LDH  CV           ;Recupera status do stack
F1D8: AC 42 04 191  LDY  TEST42
F1DB: 98 192  TYA
F1DC: 48 193  PHA
F1DE: CA 194  DECY  DEY
F1E0: 88 195  DEY
F1E4: 00 FC 196  BNE  DECY
F1E1: 68 197  PLA           ;Retorna Y do stack
F1E2: A8 198  TAY
F1E3: 48 199  ESPERA  PHA
F1E4: A9 7F 100  LDA  #7F           ;Gera um atraso
F1E6: 20 A8 FC 101  JSR  WAIT
F1E9: 68 102  PLA
F1EA: 20 00 C2 103  JSR  INPR
F1ED: A0 01 104  LDY  #501           ;Carrega $442 com #01
F1F7: 8C 42 04 105  STY  TEST42
F1F2: 4C C2 F1 106  JMP  HTCT9           ;Recupera X e Y e retorna
F1F5: 28 107  JCART9  PLP
F1F4: 4C 09 90 108  JMP  CART9
F1F9: 28 109  JPRG18  PLP
F1FA: 4C 18 08 110  JMP  PRG18
F1FD: FF FF FF 111  HEX  FFFFFFFF

```

\*Tabela de geracao de caracteres de texto

```

112
113
114
F200: 00 1C 22 115  GERDHR  HEX  001C222A34A023C ; 0
F203: 2A 34 1A 02 3C
F206: 00 08 14 116  HEX  00081422223E2222 ; A
F208: 22 22 3E 22 22
F210: 00 1E 22 117  HEX  001E22221E22221E ; B
F213: 22 1E 22 22 1E
F218: 00 1C 22 118  HEX  001C22202020221C ; C
F21B: 02 02 02 22 1C
F220: 00 1E 22 119  HEX  001E22222222221E ; D
F223: 22 22 22 22 1E
F226: 00 3E 02 120  HEX  003E2021E02023E ; E
F228: 02 1E 02 02 3E
F230: 00 3E 02 121  HEX  003E2021E020202 ; F
F233: 02 1E 02 02 02
F238: 00 3C 02 122  HEX  003C0202022223C ; G
F23B: 02 02 32 22 3C
F240: 00 22 22 123  HEX  002222223E222222 ; H
F243: 22 3E 22 22 22
F246: 00 1C 08 124  HEX  001C080808081C ; I
F248: 08 08 08 08 1C
F250: 00 20 20 125  HEX  0020202020221C ; J
F253: 20 20 20 22 1C
F258: 00 22 12 126  HEX  0022120A0A1222 ; K
F25B: 0A 0A 12 22
F260: 00 02 02 127  HEX  00020202020202E ; L

```

```

F263: 02 02 02 02 3E
F268: 00 22 26 1129      HEX 0022262A2A222222 ; M
F26B: 2A 2A 22 22 22
F270: 00 22 22 1129      HEX 002222262A322222 ; N
F273: 26 2A 32 22 22
F278: 00 1C 22 1130      HEX 001C22222222221C ; O
F27B: 22 22 22 22 1C
F280: 00 1E 22 1131      HEX 001E22221E020202 ; P
F283: 22 1E 02 02 02
F288: 00 1C 22 1132      HEX 001C2222222A122C ; Q
F28B: 22 22 2A 12 2C
F290: 00 1E 22 1133      HEX 001E22221E0A1222 ; R
F293: 22 1E 0A 12 22
F298: 00 1C 22 1134      HEX 001C22021C20221C ; S
F29B: 02 1C 20 22 1C
F2A0: 00 3E 08 1135      HEX 003E080008080800 ; T
F2A3: 08 08 08 08 08
F2A8: 00 22 22 1136      HEX 002222222222221C ; U
F2AB: 22 22 22 22 1C
F2B0: 00 22 22 1137      HEX 0022222222221408 ; V
F2B3: 22 22 22 14 08
F2B8: 00 22 22 1138      HEX 002222222A2A3622 ; W
F2BB: 22 2A 2A 36 22
F2C0: 00 22 22 1139      HEX 0022221408142222 ; X
F2C3: 14 08 14 22 22
F2C8: 00 22 22 1140      HEX 0022221400080808 ; Y
F2CB: 14 08 08 08 08
F2D0: 00 3E 20 1141      HEX 003E20100804023E ; Z
F2D3: 10 08 04 02 3E
F2D8: 00 1F 06 1142      HEX 001F060606061F ;
F2DB: 06 06 06 06 1F
F2E0: 00 00 02 1143      HEX 0000020408102000 ; /
F2E3: 04 08 10 20 00
F2E8: 00 7C 30 1144      HEX 007C30303030307C ;
F2EB: 30 30 30 30 7C
F2F0: 00 00 00 1145      HEX 0000000814220000 ; *
F2F3: 08 14 22 00 00
F2F8: 00 00 00 1146      HEX 000000000000003E ;
F2FB: 00 00 00 00 3E
F300: 00 00 00 1147      HEX 0000000000000000 ; Espaco
F303: 00 00 00 00 00
F308: 00 08 08 1148      HEX 0008080808080008 ; !
F30B: 08 08 08 00 08
F310: 00 14 14 1149      HEX 0014141400000000 ; "
F313: 14 00 00 00 00
F318: 00 14 14 1150      HEX 0014143E143E1414 ; #
F31B: 3E 14 3E 14 14
F320: 00 10 78 1151      HEX 0010781438503C10 ; $
F323: 14 38 50 3C 10
F328: 00 04 26 1152      HEX 0004261008043230 ; %
F32B: 10 08 04 32 30
F330: 00 08 14 1153      HEX 0008141408542458 ; &
F333: 14 08 54 24 58
F338: 00 10 10 1154      HEX 0010101000000000 ; '
F33B: 10 00 00 00 00
F340: 00 08 04 1155      HEX 0008040202020408 ; (
F343: 02 02 02 04 08

```

F340: 00 00 10 1156	HEX 0000102020201008 ;
F340: 20 20 20 10 08	
F350: 00 00 2A 1157	HEX 00002A1C001C2A08 ; *
F353: 1C 00 1C 2A 08	
F358: 00 00 00 1158	HEX 000000003E000000 ; +
F358: 00 3E 00 00 00	
F360: 00 00 00 1159	HEX 0000000000000004 ; ,
F363: 00 00 00 00 04	
F368: 00 00 00 1160	HEX 000000003E000000 ; =
F368: 00 3E 00 00 00	
F370: 00 00 00 1161	HEX 0000000000000008 ; .
F373: 00 00 00 00 08	
F378: 00 00 20 1162	HEX 0000201000040200 ;
F378: 10 00 04 02 00	
	347
F380: 00 1C 22 11	PUT PARTE2L_02
F383: 32 2A 24 22 1C	HEX 001C22322A242221C ; 0
F386: 00 08 0C 12	HEX 00080C000000001C ; 1
F388: 08 08 08 08 1C	
F390: 00 1C 22 13	HEX 001C22201804023E ; 2
F393: 20 18 04 02 3E	
F398: 00 3E 20 14	HEX 003E20101820221C ; 3
F398: 10 18 20 22 1C	
F3A0: 00 10 18 15	HEX 00101814123E1010 ; 4
F3A3: 14 12 3E 10 10	
F3A8: 00 3E 02 16	HEX 003E021E2020221C ; 5
F3A8: 1E 20 20 22 1C	
F3B0: 00 38 04 17	HEX 003804021E22221C ; 6
F3B3: 02 1E 22 22 1C	
F3B8: 00 3E 20 18	HEX 003E201000040404 ; 7
F3B8: 10 08 04 04 04	
F3C0: 00 1C 22 19	HEX 001C22221C22221C ; 8
F3C3: 22 1C 22 22 1C	
F3C8: 00 1C 22 10	HEX 001C22223C20100E ; 9
F3C8: 22 3C 20 10 0E	
F3D0: 00 00 00 111	HEX 0000000000000000 ; !
F3D3: 08 00 00 00 00	
F3D8: 00 00 00 112	HEX 0000000000000004 ;
F3D8: 08 00 00 00 04	
F3E0: 00 10 08 113	HEX 0010080402040810 ; (
F3E3: 04 02 04 08 10	
F3E8: 00 00 00 114	HEX 0000003E003E0000 ; =
F3E8: 3E 00 3E 00 00	
F3F0: 00 04 08 115	HEX 0004081020100004 ; )
F3F3: 10 20 10 00 04	
F3F8: 00 1C 22 116	HEX 001C221808000000 ; ?
F3F8: 10 08 00 00 08	

118

119

\*Tabela de geracao de caracteres graficos

F400: 00 00 00 120	GERGRF	HEX 000000000000007F
F403: 00 00 00 00 7F		
F408: 00 00 00 121		HEX 0000000000007F7F
F40B: 00 00 00 7F 7F		
F410: 00 00 00 122		HEX 00000000007F7F7F
F413: 00 00 7F 7F 7F		
F418: 00 00 00 123		HEX 000000007F7F7F7F

F418:	00 7F 7F 7F 7F		
F420:	00 00 00 124	HEX	0000007F7F7F7F
F423:	7F 7F 7F 7F 7F		
F426:	00 00 7F 125	HEX	00007F7F7F7F7F
F428:	7F 7F 7F 7F 7F		
F430:	00 7F 7F 126	HEX	007F7F7F7F7F7F
F433:	7F 7F 7F 7F 7F		
F436:	01 01 01 127	HEX	01010101010101
F438:	01 01 01 01 01		
F440:	03 03 03 128	HEX	03030303030303
F443:	03 03 03 03 03		
F446:	07 07 07 129	HEX	07070707070707
F448:	07 07 07 07 07		
F450:	0F 0F 0F 130	HEX	0F0F0F0F0F0F0F
F453:	0F 0F 0F 0F 0F		
F456:	1F 1F 1F 131	HEX	1F1F1F1F1F1F1F
F458:	1F 1F 1F 1F 1F		
F460:	3F 3F 3F 132	HEX	3F3F3F3F3F3F3F
F463:	3F 3F 3F 3F 3F		
F466:	7F 7F 7F 133	HEX	7F7F7F7F7F7F7F
F468:	7F 7F 7F 7F 7F		
	134		
F470:	00 55 24 135	TABCSR	HEX 0055247F8005AAFF ;Tabela usada por SCR
F473:	7F 80 05 AA 7F		
	136		
	137	*Continuacao da tabela de geracao de caracteres graficos	
	138		
F478:	00 00 7F 139	HEX	00007F00007F0000
F47E:	00 00 7F 00 00		
F480:	08 08 78 140	HEX	0808780808780808
F483:	08 08 78 08 08		
F486:	08 08 7F 141	HEX	08087F08087F0808
F488:	08 08 7F 08 08		
F490:	08 08 0F 142	HEX	08080F08080F0808
F493:	08 08 0F 08 08		
F496:	00 40 60 143	HEX	0040607070707070
F498:	70 78 7C 7E 7F		
F4A0:	00 01 03 144	HEX	000103070F1F3F7F
F4A3:	07 0F 1F 3F 7F		
F4A6:	00 7F 7E 145	HEX	007F7E7C78766040
F4A8:	7C 78 70 60 40		
F4B0:	00 7F 2F 146	HEX	007F3F1F0F070201
F4B3:	1F 0F 07 03 01		
F4B6:	7F 00 00 147	HEX	7F00000000000000
F4B8:	00 00 00 00 00		
F4C0:	00 00 00 148	HEX	000000007F000000
F4C3:	00 7F 00 00 00		
F4C6:	08 08 08 149	HEX	0808080808080808
F4C8:	08 08 08 08 08		
F4D0:	10 10 10 150	HEX	1010101010101010
F4D3:	10 10 10 10 10		
F4D6:	00 40 20 151	HEX	004020106040201
F4D8:	10 08 04 02 01		
F4E0:	00 01 02 152	HEX	0001020408102040
F4E3:	04 08 10 20 40		
F4E6:	00 1C 3E 153	HEX	001C3E3E3E1C00
F4E8:	3E 3E 3E 1C 00		

```

F4F0: 00 1C 22 154      HEX 001C22222221C00
F4F3: 22 22 22 1C 00
F4F8: 08 08 08 155      HEX 080808087F00000
F4F9: 08 7F 00 00 00
F500: 00 00 00 156      HEX 000000007F08000
F503: 00 7F 08 08 08
F508: 08 08 08 157      HEX 08080808F080808
F509: 08 0F 08 08 08
F510: 08 08 08 158      HEX 080808087908080
F513: 08 78 08 08 08
F518: 00 08 1C 159      HEX 00081C3E7F7F1C3E
F51E: 3E 7F 7F 1C 3E
F520: 00 36 7F 160      HEX 00367F7F7F3E1C00
F523: 7F 7F 3E 1C 08
F528: 00 08 1C 161      HEX 00081C3E7F3E1C08
F529: 3E 7F 3E 1C 00
F530: 00 1C 1C 162      HEX 001C1C7F7F68083E
F533: 7F 7F 68 08 3E
F538: 00 00 00 163      HEX 000000007808808
F53E: 00 78 08 08 08
F540: 00 00 00 164      HEX 00000000F080808
F543: 00 0F 08 08 08
F548: 08 08 08 165      HEX 080808780000000
F54E: 78 00 00 00 08
F550: 08 08 08 166      HEX 0808080F0000000
F553: 0F 00 00 00 00
F558: 00 00 00 167      HEX 0000000060100808
F55E: 00 60 10 08 08
F560: 00 00 00 168      HEX 000000003040808
F563: 00 03 04 08 08
F568: 08 08 10 169      HEX 080810600000000
F56E: 60 00 00 00 08
F570: 08 08 04 170      HEX 080804030000000
F573: 03 00 00 00 00
F578: 41 22 14 171      HEX 4122140808142241
F57E: 08 08 14 22 41
F580: 00 08 08 172      HEX 000808007F000808
F583: 00 7F 00 08 08
F588: 08 08 08 173      HEX 080808087F080808
F58E: 08 7F 08 08 08

```

174

```

F590: 20 FF 00 175 DISKID HEX 20FF00FF03FF3C
F593: FF 03 FF 3C

```

176

```

F597: 80 80 80 177 TITULO AGC "0002K1"
F59A: 82 C8 04

```

178

179

180

\*Tabela de codigos do teclado

```

F59D: FF 181 TABASC HEX FF ;Espaco reservado sem tecla
F59E: FF 182 HEX FF
F59F: FF 183 HEX FF
F5A0: FF 184 HEX FF
F5A1: C2 185 HEX C2 ; 8
F5A2: 82 186 HEX 82 ; CTRL-8
F5A3: 22 187 HEX 22 ; SHIFT-8
F5A4: 12 188 HEX 12 ; SHIFT/CTR-8

```



FS45: 06	189	HEX 06	: V
FS46: 96	190	HEX 96	: CTRL-V
FS47: 28	191	HEX 28	: SHFT-V
FS48: 18	192	HEX 18	: SHF/CTR-V
FS49: C3	193	HEX C3	: C
FS4A: 83	194	HEX 83	: CTRL-C
FS4B: 2A	195	HEX 2A	: SHFT-C
FS4C: 1A	196	HEX 1A	: SHF/CTR-C
FS4D: 08	197	HEX 08	: X
FS4E: 98	198	HEX 98	: CTRL-X
FS4F: 2F	199	HEX 2F	: SHFT-X
FS50: 17	1100	HEX 17	: SHF/CTR-X
FS51: 0A	1101	HEX 0A	: Z
FS52: 9A	1102	HEX 9A	: CTRL-Z
FS53: 2E	1103	HEX 2E	: SHFT-Z
FS54: 16	1104	HEX 16	: SHF/CTR-Z
FS55: FF	1105	HEX FF	: Espaço reservado sem tecla
FS56: FF	1106	HEX FF	
FS57: FF	1107	HEX FF	
FS58: FF	1108	HEX FF	
FS59: C7	1109	HEX C7	: G
FS5A: 87	1110	HEX 87	: CTRL-G
FS5B: 20	1111	HEX 20	: SHFT-G
FS5C: 10	1112	HEX 10	: SHF/CTR-G
FS5D: C6	1113	HEX C6	: F
FS5E: 86	1114	HEX 86	: CTRL-F
FS5F: 29	1115	HEX 29	: SHFT-F
FS60: 19	1116	HEX 19	: SHF/CTR-F
FS61: C4	1117	HEX C4	: D
FS62: 84	1118	HEX 84	: CTRL-D
FS63: 28	1119	HEX 28	: SHFT-D
FS64: 18	1120	HEX 18	: SHF/CTR-D
FS65: 03	1121	HEX 03	: S
FS66: 93	1122	HEX 93	: CTRL-S
FS67: 20	1123	HEX 20	: SHFT-S
FS68: 15	1124	HEX 15	: SHF/CTR-S
FS69: C1	1125	HEX C1	: A
FS6A: 81	1126	HEX 81	: CTRL-A
FS6B: 2C	1127	HEX 2C	: SHFT-A
FS6C: 14	1128	HEX 14	: SHF/CTR-A
FS6D: A0	1129	HEX A0	: "Espaco"
FS6E: A0	1130	HEX A0	: CTRL-Espaco
FS6F: A0	1131	HEX A0	: SHFT-Espaco
FS70: 34	1132	HEX 34	: SHF/CTR-Espaco
FS71: 04	1133	HEX 04	: T
FS72: 94	1134	HEX 94	: CTRL-T
FS73: 1E	1135	HEX 1E	: SHFT-T
FS74: 0C	1136	HEX 0C	: SHF/CTR-T
FS75: 02	1137	HEX 02	: R
FS76: 92	1138	HEX 92	: CTRL-R
FS77: 27	1139	HEX 27	: SHFT-R
FS78: 08	1140	HEX 08	: SHF/CTR-R
FS79: C5	1141	HEX C5	: E
FS7A: 85	1142	HEX 85	: CTRL-E
FS7B: 26	1143	HEX 26	: SHFT-E
FS7C: 0A	1144	HEX 0A	: SHF/CTR-E
FS7D: 07	1145	HEX 07	: V

F5DE: 97	1146	HEX 97	: CTRL-6
F5DF: 25	1147	HEX 25	: SHFT-M
F5E0: 09	1148	HEX 09	: SHF/CTR-M
F5E1: 01	1149	HEX 01	: 0
F5E2: 91	1150	HEX 91	: CTRL-6
F5E3: 24	1151	HEX 24	: SHFT-6
F5E4: 08	1152	HEX 08	: SHF/CTR-6
F5E5: 88	1153	HEX 88	: (-
F5E6: 88	1154	HEX 88	: CTRL (-
F5E7: 88	1155	HEX 88	: SHFT (-
F5E8: 3C	1156	HEX 3C	: SHF/CTR (-
F5E9: 85	1157	HEX 85	: 5
F5EA: 85	1158	HEX 85	: CTRL-5
F5EB: A5	1159	HEX A5	: SHFT-5
F5EC: 05	1160	HEX 05	: SHF/CTR-5
F5ED: 84	1161	HEX 84	: 4
F5EE: 84	1162	HEX 84	: CTRL-4
F5EF: A4	1163	HEX A4	: SHFT-4
F5F0: 04	1164	HEX 04	: SHF/CTR-4
F5F1: 83	1165	HEX 83	: 3
F5F2: 83	1166	HEX 83	: CTRL-3
F5F3: A3	1167	HEX A3	: SHFT-3
F5F4: 03	1168	HEX 03	: SHF/CTR-3
F5F5: 82	1169	HEX 82	: 2
F5F6: 82	1170	HEX 82	: CTRL-2
F5F7: A2	1171	HEX A2	: SHFT-2
F5F8: 02	1172	HEX 02	: SHF/CTR-2
F5F9: 81	1173	HEX 81	: 1
F5FA: 81	1174	HEX 81	: CTRL-1
F5FB: A1	1175	HEX A1	: SHFT-1
F5FC: 01	1176	HEX 01	: SHF/CTR-1
F5FD: 95	1177	HEX 95	: -)
F5FE: 95	1178	HEX 95	: CTRL -)
F5FF: 95	1179	HEX 95	: SHFT -)
F600: 30	1180	HEX 30	: SHF/CTR -)
F601: 86	1181	HEX 86	: 6
F602: 86	1182	HEX 86	: CTRL-6
F603: A6	1183	HEX A6	: SHFT-6
F604: 06	1184	HEX 06	: SHF/CTR-6
F605: 87	1185	HEX 87	: 7
F606: A5	1186	HEX A5	: CTRL-7
F607: A7	1187	HEX A7	: SHFT-7
F608: 07	1188	HEX 07	: SHF/CTR-7
F609: 88	1189	HEX 88	: 8
F60A: 88	1190	HEX 88	: CTRL-8
F60B: A8	1191	HEX A8	: SHFT-8
F60C: 26	1192	HEX 26	: SHF/CTR-8
F60D: 89	1193	HEX 89	: 9
F60E: 89	1194	HEX 89	: CTRL-9
F60F: A9	1195	HEX A9	: SHFT-9
F610: 37	1196	HEX 37	: SHF/CTR-9
F611: 80	1197	HEX 80	: 0
F612: 80	1198	HEX 80	: CTRL-0
F613: AA	1199	HEX AA	: SHFT-0
F614: 35	1200	HEX 35	: SHF/CTR-0
F615: F1	1201	HEX F1	: ↓
F616: F1	1202	HEX F1	: CTRL ↓

F617: F1	1203	HEX F1	: SHIFT +
F618: 3E	1204	HEX 3E	: SHIFT/CTR +
	340	PUT PARTE2N,02	
F619: 09	31	HEX 09	: Y
F61A: 99	32	HEX 99	: CTRL-Y
F61B: 31	33	HEX 31	: SHIFT-Y
F61C: 00	34	HEX 00	: SHIFT/CTR-Y
F61D: 05	35	HEX 05	: U
F61E: 95	36	HEX 95	: CTRL-U
F61F: 30	37	HEX 30	: SHIFT-U
F620: 0E	38	HEX 0E	: SHIFT/CTR-U
F621: C9	39	HEX C9	: I
F622: 89	310	HEX 89	: CTRL-I
F623: A0	311	HEX A0	: SHIFT-I
F624: 0F	312	HEX 0F	: SHIFT/CTR-I
F625: CF	313	HEX CF	: O
F626: 8F	314	HEX 8F	: CTRL-O
F627: 8D	315	HEX 8D	: SHIFT-O
F628: 43	316	HEX 43	: SHIFT/CTR-O
F629: D0	317	HEX D0	: P
F62A: 90	318	HEX 90	: CTRL-P
F62B: A8	319	HEX A8	: SHIFT-P
F62C: 44	320	HEX 44	: SHIFT/CTR-P
F62D: F0	321	HEX F0	: +
F62E: F0	322	HEX F0	: CTRL +
F62F: F0	323	HEX F0	: SHIFT +
F630: 3F	324	HEX 3F	: SHIFT/CTR +
F631: C8	325	HEX C8	: H
F632: 88	326	HEX 88	: CTRL-H
F633: 21	327	HEX 21	: SHIFT-H
F634: 11	328	HEX 11	: SHIFT/CTR-H
F635: CA	329	HEX CA	: K
F636: 8A	330	HEX 8A	: CTRL-K
F637: 1F	331	HEX 1F	: SHIFT-K
F638: 1C	332	HEX 1C	: SHIFT/CTR-K
F639: C8	333	HEX C8	: J
F63A: 88	334	HEX 88	: CTRL-J
F63B: 0E	335	HEX 0E	: SHIFT-J
F63C: 41	336	HEX 41	: SHIFT/CTR-J
F63D: CC	337	HEX CC	: L
F63E: 8C	338	HEX 8C	: CTRL-L
F63F: C0	339	HEX C0	: SHIFT-L
F640: 42	340	HEX 42	: SHIFT/CTR-L
F641: 8A	341	HEX 8A	: I
F642: 8A	342	HEX 8A	: CTRL I
F643: 88	343	HEX 88	: SHIFT I
F644: 38	344	HEX 38	: SHIFT/CTR I
F645: 80	345	HEX 80	: RETURN
F646: 80	346	HEX 80	: CTRL-RETURN
F647: 80	347	HEX 80	: SHIFT-RETURN
F648: 40	348	HEX 40	: SHIFT/CTR-RETURN
F649: CE	349	HEX CE	: N
F64A: 8E	350	HEX 8E	: CTRL-N
F64B: 23	351	HEX 23	: SHIFT-N
F64C: 13	352	HEX 13	: SHIFT/CTR-N
F64D: C0	353	HEX C0	: M
F64E: 80	354	HEX 80	: CTRL-M

F64F:	32	355	HEX 32	:	SHIFT-M
F650:	10	356	HEX 10	:	SHF/CTR-M
F651:	AC	357	HEX AC	:	,
F652:	AC	358	HEX AC	:	CTRL ,
F653:	BC	359	HEX BC	:	SHIFT ,
F654:	39	360	HEX 39	:	SHF/CTR ,
		361			
F655:	AC	362	HEX AE	:	,
F656:	AE	363	HEX AE	:	CTRL ,
F657:	BE	364	HEX BE	:	SHIFT ,
F658:	3A	365	HEX 3A	:	SHF/CTR ,
F659:	BF	366	HEX BF	:	?
F65A:	BF	367	HEX BF	:	CTRL ?
F65B:	AF	368	HEX AF	:	SHIFT ?
F65C:	38	369	HEX 38	:	SHF/CTR ?
		370			
		371			Rotina de interpretacao do buffer IN
		372			
F65D:	A2 00	373	ZXY LDH #500	:	;Zera X e Y
F65F:	A0 00	374	LDY #500		
F661:	B9 00 02	375	PROXIN LDA IN,Y		
F664:	85 FD	376	STA RTEMP1	:	;Arazena caracter
F666:	30 17	377	BNI CARACT	:	;Se e caracter de texto
F66B:	A0 52 04	378	LDA GRAFFLG	:	;Se GRAFFLG (=) 0,houve CTRL-B
F66B:	D0 24	379	BNE BORAF		
F66D:	A5 FD	380	LDA RTEMP1		
F66F:	20 EA F7	381	JSR SELEFAX	:	;Seleciona o tipo de caracter
F672:	80 09	382	BCC CARACT	:	;Se caracter grafico
F674:	A5 FD	383	LDA RTEMP1		
F676:	0C 0A 06	384	STY ING+5A		
F679:	20 BE F6	385	JSR PLVBAG		
F67C:	4C 9C F6	386	JMP STDING		
F67F:	A5 FD	387	CARACT LDA RTEMP1		
F681:	C9 82	388	CHP #502	:	;E CTRL-B ?
F683:	D0 17	389	BNE STDING		
F685:	A9 01	390	LDA #501	:	;Sim, inverte GRAFFLG
F687:	A0 52 04	391	EQR GRAFFLG		
F68A:	80 52 04	392	STA GRAFFLG		
F68D:	C8	393	INY		
F68E:	4C 61 F6	394	JMP PROXIN		
F691:	A9 F2	395	BORAF LDA #graf	:	;Caracter de controle grafico
F693:	90 0E 06	396	STA ING+5E,X		
F696:	EB	397	DXR		
F697:	A5 FD	398	LDA RTEMP1		
F699:	18	399	CLC		
F69A:	A9 40	3100	ADC #540	:	;Converte p/ ASC graficos
F69C:	90 0E 06	3101	STDING STA ING+5E,X		
F69F:	C9 80	3102	CHP #580	:	;E RETURN ?
F6A1:	F0 04	3103	BEN ZGRAFFLG	:	;Sim
F6A3:	C8	3104	INY	:	;Nao
F6A4:	EB	3105	DXR		
F6A5:	D0 8A	3106	BNE PROXIN	:	;E fim do buffer ?
F6A7:	A2 00	3107	ZGRAFFLG LDH #500		
F6A9:	0E 52 04	3108	STX GRAFFLG		
F6AC:	20 80 F6	3109	JSR TRANSRAF		
F6AF:	60	3110	RTS		
		3111			

```

F68D: 8D 0E 06 1112 TRANSBUF LDA INQ+SE,X ;Reorganiza buffer IN
F68E: 9D 0D 02 1113 STA IN,X
F68F: C9 8D 1114 CMP #500 ;E RETURN ?
F690: FD 03 1115 BEQ F6MTRANS ;Sim
F691: E8 1116 IMX ;Nao
F692: 0D F3 1117 BNE TRANSBUF ;Continua transferindo o buffer
F693: 6D 1118 F6MTRANS RTS
1119
F69E: 2D 04 F6 1120 PLV8AS JSR EDT8AS ;Calcula em (COOL) o end. na tab.
F6A1: B1 F8 1121 OUTROCH LDA (COOL),Y da palavra BASIC desejada
F6A2: 3D 08 1122 BMI ULTCHR ;Se BASIC, entao e ultimo caracter
F6A3: 09 8D 1123 ORA #500 ;Transforma em BASICII
F6A4: 9D 0E 06 1124 STA INQ+SE,X
F6A5: 2D C1 FF 1125 JSR NRC00 ;Proximo caracter da palavra
F6A6: E8 1126 IMX ;Incrementa INQ
F6A7: 0D F1 1127 BNE OUTROCH ;Nao e o ultimo caracter ?
F6A8: AC 04 06 1128 ULTCHR LDY INQ+5A ;Ultimo caracter, entao retorna
F6A9: 6D 1129 RTS
1130
F6D4: C9 1E 1131 EDT8AS CMP #51E ;Comp. se o cod. ASCII e (= A ZIE
F6D5: 9D 02 1132 RCL INQ85 ;Entre #1 e #1D fica como esta
F6D6: E9 16 1133 SBC #516 ;Entre #45 e #34 subtrai #16
F6D7: AB 1134 INQ85 TAY ;Codigo em Y para indexar
F6D8: B8 1135 DEY ;tab.de deslocam. de palavras
F6D9: AD 07 F7 1136 LDA ENDT85 ;coloca o endereco de inicio
F6DA: 95 F8 1137 STA COOL da tab.de palavras BASIC
F6DB: AD 08 F7 1138 LDA ENDT85+1 ;#FB=#D
F6DC: 85 FC 1139 STA C08H ;#FC=#C
F6DD: B9 09 F7 1140 LDA INQ85AS,Y ;Faz desloc.de palavr.na tab.
F6DE: AD 0D 1141 LDY #500 ;Zera indexador Y
F6DF: 38 1142 SEC
F6E0: E9 8D 1143 SBC #500
F6E1: FD 16 1144 BEQ F6MBAS ;Se for = 10 retorna
F6E2: 8D 02 06 1145 STA INQ+52 ;#armazena o desloc. de palavras
F6E3: B1 F8 1146 CONTB LDA (COOL),Y ;Caract.da tab.de palav. BASIC
F6E4: 3D 05 1147 BMI F6MPA ;Se ultimo caracter da palavra
F6E5: 2D C1 FF 1148 JSR NRC00 ;Incrementa o par (COOL+H)
F6E6: 0D F7 1149 BNE CONTB
F6E7: 2D C1 FF 1150 F6MPA JSR NRC00 ;Incrementa o par (COOL+H)
F6E8: CE 02 06 1151 DEC INQ+52 ;Decrementa contad.de palavras
F6E9: FB 02 1152 BEQ F6MBAS ;Retorne se e a palavra
F6EA: 0D ED 1153 BNE CONTB ;Nao e a palavra,faz novo loop
F6EB: 6D 1154 F6MBAS RTS
F6EC: 8D C3 1155 ENDT85 DA TOKTABL ;End.da tab.de palavras BASIC
1156
1157 *Tabela de contagens de palavras BASIC
1158 #Fornece o deslocamento dentro da tabela
1159
F709: B3 1160 INQ85AS HEX B3
F70A: B8 1161 HEX B8
F70B: B7 1162 HEX B7
F70C: B6 1163 HEX B6
F70D: AC 1164 HEX AC
F70E: BC 1165 HEX BC
F70F: EC 1166 HEX EC
F710: 9E 1167 HEX 9E
F711: AE 1168 HEX AE

```

F712:	82	3169	HEX	82	
F713:	87	3170	HEX	87	
F714:	C1	3171	HEX	C1	
F715:	C7	3172	HEX	C7	
F716:	81	3173	HEX	81	
F717:	84	3174	HEX	84	
F718:	A8	3175	HEX	A8	
F719:	97	3176	HEX	97	
F71A:	80	3177	HEX	80	
F71B:	82	3178	HEX	82	
F71C:	C5	3179	HEX	C5	
F71D:	86	3180	HEX	86	
F71E:	88	3181	HEX	88	
F71F:	C2	3182	HEX	C2	
F720:	83	3183	HEX	83	
F721:	81	3184	HEX	81	
F722:	40	3185	HEX	40	
F723:	C4	3186	HEX	C4	
F724:	93	3187	HEX	93	
F725:	80	3188	HEX	80	
F726:	89	3189	HEX	89	
F727:	8F	3190	HEX	8F	
F728:	EB	3191	HEX	EB	
F729:	85	3192	HEX	85	
F72A:	E2	3193	HEX	E2	
F72B:	A0	3194	HEX	A0	
F72C:	88	3195	HEX	88	
F72D:	89	3196	HEX	89	
F72E:	96	3197	HEX	96	
F72F:	8E	3198	HEX	8E	
F730:	A2	3199	HEX	A2	
F731:	8F	3200	HEX	8F	
F732:	90	3201	HEX	90	
F733:	92	3202	HEX	92	
F734:	91	3203	HEX	91	
F735:	84	3204	HEX	84	
F736:	8A	3205	HEX	8A	
F737:	ED	3206	HEX	ED	
		349	PUT	PARTE2K_02	
		31			
F738:	48	32	BUSTOKEN	PHA	;Busca token
F739:	A5	33	LDI	A44	
F73A:	00	34	BNE	DECOOTOK	
F73B:	68	35	PLA		
F73C:	4C	36	JMP	CONTOK	
F741:	68	37	DECOOTOK	PLA	
F742:	84	38	STY	RTEHP1	;Salva Y
F744:	A0	39	LDI	MS2E	;Indexa no fim da tabela
F746:	09	40	PROXTOK	CMP	INFRAG,T
F749:	F0	41	BEQ	ETOKEN	;Sim
F74B:	88	42	DEY		
F74C:	10	43	BPL	PROXTOK	;Proximo
F74E:	A4	44	LDI	RTEHP1	;Se nao encontrou
F750:	4C	45	JMP	CONTOK	
F753:	C8	46	ETOKEN	INT	
F754:	98	47	TYA		
F755:	C9	48	CMP	MS3E	

F757:	90 03	119	BCC	COMVTOX	:Ajusta codigo
F759:	18	120	CLC		
F75A:	6F 16	121	ADC	#516	
F75C:	4A F0	122	COMVTOX	L0Y RTEMP1	
F75E:	20 8F	123	JSR	SEMO	:Imprime caracter
F761:	A9 00	124	LDA	#500	
F763:	80 06 04	125	STA	TEST06	
F764:	4C 1C CA	126	JMP	SMDCHR	
		127			
F769:	84 F0	128	ISCRAS	STY RTEMP1	:Salva Y
F76B:	8E 09 06	129	STX	IM0+59	:Salva X
F76E:	40	130	PHA		:Salva codigo ASCII
F76F:	A5 46	131	LDA	A46	:Verifica conteudo de A46
F771:	FD 0A	132	BEQ	IMPRAS	:Se = 0
F773:	C9 03	133	CMF	#503	
F775:	D0 03	134	BNE	JPRGC	
F777:	4C 00 9D	135	JMP	CART0	:Se = 3
F77A:	4C 0C 00	136	JPRGC	JMP PRGC	:Se 1,3 \$00C
F77D:	68	137	IMPRAS	PLA	:Retorna codigo ASCII
F77E:	20 04 F6	138	JSR	EDTBAS	:Prep.em (COOL),Y o end.sa tab.
F781:	81 F8	139	PRLETR	LDA (COOL),Y	da palavra BASIC desejada
F783:	30 0A	140	BNE	ULTVF	:Se e NASCII
F785:	09 80	141	ORA	#580	:Transforma em NASCII
F787:	20 A9 F1	142	JSR	IDCTP	:Identifica CONTROL-P
F78A:	20 C1 FF	143	JSR	WAC00	:Incrementa (COOL)
F78D:	80 F2	144	BNE	PRLETR	
F78F:	20 A9 F1	145	ULTVF	JSR IDCTP	
F792:	AE 09 06	146			
F795:	AA F0	147			:Recupera X
F797:	60	148			:Recupera Y
		149			
F79B:	A5 47	150	PRERET	LDA PGVL	:Testa PGVL
F79A:	F0 03	151	BEQ	VOLTBUF	:Se PGVL=0,prep.ret.or.do carac.
F79C:	4C 03 09	152	JMP	PRG3	
F79F:	80 FF 01	153	VOLTBUF	LDA IN-1,X	:Pega carac.anterior do buffer
F7A2:	30 38	154	BMI	CCTR	:Se e NASCII,verif.se e CTRL-B
F7A4:	AB	155	TAY		:Se e NASCII salva em Y
F7A5:	A5 49	156	LDA	VCTR	:Testa VCTR
F7A7:	D0 3E	157	BNE	JOECBUF	:Decr.buffer e busca novo dado
F7A9:	98	158	TYA		
F7AA:	20 EA F7	159	JSR	SELEFAX	:Se VCTR=0 recupera codigo de Y
F7AB:	80 38	160	BCC	JOECBUF	:Se graf.decr.buf e busca dado
F7AF:	A5 46	161	LDA	A46	:Testa A46
F7B1:	FD 0A	162	BEQ	VTBFE	:Volt.buf.e uma letra da palav.
F7B3:	C9 04	163	CMF	#504	
F7B5:	F0 03	164	BEQ	JPRG12	:Se = 04
F7B7:	4C 03 90	165	JMP	CART3	:Se () 04
F7BA:	4C 12 08	166	JPRG12	JMP PRG12	
F7BD:	80 FF 01	167	VTBFE	LDA IN-1,X	:Carrega com carac.ant.do buf.
F7C0:	20 04 F6	168	JSR	EDTBAS	:Tab. de palav. BASIC em (COOL)
F7C3:	20 82 F8	169	JSR	POSCLR	:Prepara cursor
F7C6:	A0 00	170	ldy	#500	
F7C8:	81 F8	171	RETORNA	LDA (COOL),Y	:Carrega codigo da palav.BASIC
F7CA:	30 06	172	BNE	FINDEC	:Se e ultima letra
F7CC:	20 18 FC	173	JSR	DESESR	:Mao,entao retorna o cursor
F7CF:	CB	174	INY		:Incrementa contador de letras
F7D0:	D0 F6	175	BNE	RETORNA	:Pega proxima letra

F702:	CA	176	FINDCC	DEX		;Decrementa buffer
F703:	20 18 FC	177		JSR DESEGR		;Retorna cursor <b>na</b> posicao
F704:	20 82 FB	178		JSR POSCUR		;Posiciona cursor
F709:	4C 7C FD	179		JMP JMS		;Pega novo dado no teclado
F70C:	C9 82	180	OCTB	CHP #502		;Compara com CONTROL-8
F70E:	00 07	181		BNE JDECRBUF		;Nao_decr.buf.e busca novo dado
F703:	A9 01	182		LDA #501		;Se e CONTROL-8 inverte VCTB
F7E2:	45 49	183		EDR VCTB		
F7E4:	85 49	184		STA VCTB		
F7E6:	CA	185		DEX		;Volta buffer
F7E7:	4C 18 FD	186	JDECRBUF	JMP DECRBUF		;Busca novo dado
F7E8:	C9 46	187	SELEFAX	CHP #5A6		;Se carac.=46, seta carry
F7EC:	80 09	188		BCS RTCS		(graficos)
F7EE:	C9 34	189		CHP #534		;Se (A6 e !=34, reseta carry
F7F0:	80 04	190		BCS RTCR		(palavras basic)
F7F2:	C9 1E	191		CHP #51E		;Se ( R34 e != 1E, seta carry
F7F4:	80 01	192		BCS RTCS		(BELL)
F7F6:	18	193	RTCR	CLC		;Se ( 1E, reseta carry (BASIC)
F7F7:	60	194	RTCS	RTS		
		195				
F7FB:	A5 27	196	GBHOLD	LDA GRASH		;GRASH = ABCIEFGH
F7FA:	09 10	197		ORA #510		
F7FC:	85 27	198		STA GRASH		
F7FE:	60	199		RTS		
		1000				
F7FF:	FF	1101	HEX	FF		
F800:	4C CA EF	1102	PLOT2	JMP PLOT0		
F803:	4A	1103	PLOTM	LSR		;Se linha impar, seta carry
F804:	00	1104		PHP		;Salva status no stack
F805:	20 F5 FB	1105		JSR ARB		;Acumulador #8 result. em delh
F808:	20 73 FC	1106		JSR EPSVID		;Prepara endor.da pag de video
F808:	A5 09	1107		LDA #4H		;Se for coord.par endereco AS 4
F80C:	28	1108		PLP		;primeiras microlin. de um bloco
F80E:	90 02	1109		BCC UPBLOC		de texto, senao as 4 segundas
F810:	69 0F	1110		ADC #50F		o endereco esta em B4
F812:	85 09	1111	UPBLOC	STA #4H		
F814:	60	1112		RTS		
		1113				
F815:	86 EA	1114		STX #EA		
F817:	EA	1115		WOP		
F818:	EA	1116		WOP		
F819:	20 00 FB	1117	NLINE	JSR PLOT2		;Plota um bloco LORES
F81C:	C4 2C	1118	NLINE1	CPY #2		;Se chegou na ultima coluna,
F81E:	80 11	1119		BCS RETM4		;retorna
F820:	C8	1120		INT		;Plota na proxima coluna
F821:	20 C8 EF	1121		JSR PLOT1		
F824:	90 F6	1122		BCC NLINE1		
		1123				
F826:	69 01	1124	NLINE2	ADC #501		
F828:	48	1125	NLINE	PHA		
F829:	20 00 FB	1126		JSR PLOT2		;Plota um bloco LORES
F82C:	68	1127		PLA		
F82D:	C5 20	1128		CHP #2		;E ultima linha ?
F82F:	90 F5	1129		BCC NLINE2		;Nao,plote novo bloco
F831:	60	1130	RET#4	RTS		;Sim,retorne
		1131				
F832:	AD 2F	1132	CLPSCR	LDY #50F		;Posicao da ultima linha horiz.



FB34:	00 02	1130	BME CLRSC2		;limpa toda tela
FB36:	A0 27	1134	CLRTOP LDT #527		;limpa topo da tela (modo MIXED)
FB3B:	04 20	1135	CLRSC2 STY V2		;ultima linha a ser apagada em V2
FB3A:	A0 27	1136	LDT #527		
FB3C:	A9 00	1137	CLRSC3 LDA #500		;Seta cor preta 1
FB3E:	05 30	1138	STA COLOR		
FB40:	20 28 FB	1139	JSR VLINE		;Plota linhas verticais da coluna
FB43:	08	1140	DEY		;39 ate a coluna 0
FB44:	10 F6	1141	BPL CLRSC3		
FB46:	60	1142	RTS		
		1143			
FB47:	48	1144	GBASCALC PHA		;Calcula o endereco base da pag.
FB48:	4A	1145	LSR		;de video referente a linha de
FB49:	29 03	1146	AND #503		baixa resolucao
FB4B:	05 48	1147	ORA PAVH		(da linha 0 a linha 24)
FB4D:	05 27	1148	STA GBASH		
FB4F:	68	1149	PLA		
FB50:	29 18	1150	AND #518		
FB52:	90 0C	1151	BCC GBALC		
FB54:	69 7F	1152	ADC #57F		
FB56:	05 26	1153	GBALC STA GBASL		
FB58:	0A	1154	ASL		
FB59:	0A	1155	ASL		
FB5A:	05 26	1156	ORA GBASL		
FB5C:	05 26	1157	STA GBASL		;0 endereco esta em GBASL-H
FB5E:	60	1158	RTS		
		1159			
FB5F:	A5 30	1160	MEXTCOL LDA COLOR		;Incrementa COLOR de 3
FB61:	18	1161	CLC		
FB62:	69 03	1162	ADC #503		
FB64:	29 0F	1163	SETCOL AND #50F		;Limita COLOR na faixa de 0-15
FB66:	05 30	1164	STA COLOR		
FB68:	60	1165	RTS		
		350	PUT PARTE20,02		
		11			
FB69:	4A	12	SORN LSR		;testa linha par ou impar
FB6A:	08	13	PHP		
FB6B:	20 47 FB	14	JSR GBASCALC		;Calc.emder.base da pag de video
FB6E:	28	15	PLP		
FB6F:	90 03	16	BCC PBCC		;Se e linha par
FB71:	20 FB F7	17	JSR GBHOU0		;lepar,endereca o bloco 2
FB74:	01 26	18	FBCC LDA (GBASL),Y		;A = byte cor
FB76:	48	19	PHA		;Salva A no stack
FB77:	98	110	TYA		;Testa se e coluna impar ou par
FB78:	4A	111	LSR		
FB79:	68	112	PLA		;Recupera A do stack
FB7A:	90 03	113	BCC PBCC		;Se coluna par,cor em POCC
FB7C:	20 28 EF	114	JSR BUSCOR		;Se impar, cor em BUSCOR
FB7F:	4A	115	POCC LSR		
FB80:	38	116	SEC		
FB81:	4C 65 FA	117	JMP SCRHC		;Numero da cor no des do acumul.
FB84:	EA	118	WOP		
FB85:	A6 3A	119	INSD51 LDX PCL		
FB87:	A4 38	120	LDT PCB		
FB89:	20 96 FD	121	JSR PRYX2		;leprime PC como ender.ca hex.
FB8C:	20 48 F9	122	JSR PRBLM		;leprime 3 espacos em branco
FB8F:	A1 3A	123	INSD52 LDA (PCL,X)		;Busca opcodes

FB91: A8	124	INSD63	TAY	:Guarda em Y
FB92: 4A	125		LSR	:Testa par ou impar
FB93: 90 09	126		BCC DEVEN	:Se par DEVEN
FB95: 6A	127		ROR	:Testa bit 1
FB96: 80 10	128		BCC ERR	:XXXXXXXX op invalido
FB98: C9 A2	129		CMF #5A2	
FB9A: FD 0C	130		BE9 ERR	:Opcode \$B9 invalido
FB9C: 2F 07	131		AND #5B7	:Mascara bits
FB9E: 4A	132	IEVEN	LSR	:LSR para carry para teste
FB9F: AA	133		TAX	
FBAD: 8D A2 F9	134		LDA #MT1,X	:Busca byte de index. de formato
FBAB: 20 A5 FA	135		JSR SCRM2	:Se carry:1 habilita OMS no des, :Se nao habilita so des
FBAA: 00 04	136	*		:Se nao erro GETFORMT
FBAA: 00 04	137		BNE GETFORMT	
FBAB: A0 8D	138	ERR	LDY #5B0	:Substitui \$B0 por op invalido
FBAA: AF 0D	139		LDA #5D0	:Indexa #MT2 com zero
FBAC: AA	140	GETFORMT	TAX	
FBAD: 8D A6 F9	141		LDA #MT2,X	:Indexacao para tabela de formato de impressao
FBBD: 85 2E	142		STA #FORMAT	
FB82: 2F 03	143		AND #5D3	:Mascara 2 bits de comecimento
FB84: 85 2F	144		STA LENGTH	:0=1Byte, 1=2Bytes, 2=3Bytes
FB86: 98	145		TIA	:Recupera opcode
FB87: 29 8F	146		AND #5BF	
FB89: AA	147		TAX	:Opcode masc. por XXXX010 ,em X
FB8A: 98	148		TYA	:Opcode
FB8B: A0 03	149		LDY #5D3	
FB8D: E0 8A	150		CPX #5BA	
FB8F: FD 0B	151		BE9 #NNDX3	
FB01: 4A	152	NNDX1	LSR	
FB02: 90 0B	153		BCC #NNDX3	:Forma indexacao para tabela de mnemonicos
FB04: 4A	154		LSR	
FB05: 4A	155	NNDX2	LSR	:1) XXXX1010 => 0010XXXX
FB06: 09 2D	156		ORA #52D	:2) XXXYYY01 => 0011XXXX
FB08: 88	157		DEY	:3) XXXYYY10 => 0011XXXX
FB09: 00 FA	158		BNE #NNDX2	:4) XXXYY100 => 0010XXXX
FB0B: C8	159		INY	:5) XXXXX000 => 0000XXXX
FB0C: 88	160	NNDX3	DEY	
FB0D: 00 F2	161		BNE #NNDX3	
FB0F: 60	162		RTS	
FB8D: 20 85 F8	163	INSTOSP	JSR INSD61	:Formato e copia,da instrucao
FB03: 48	164		FINA	:Salva indexada tabela mnemonicos
FB84: 81 3A	165	PRINTOP	LDA (PCL),Y	:Imprime Byte
FB06: 20 0A FD	166		JSR #PBYTE	
FB09: A2 01	167		LDX #5D1	:Imprime 1 espaco
FB0B: 20 4A F9	168	PRINTBL	JSR #PBL2	
FB0E: C4 2F	169		CPY LENGTH	:Compara quantos bytes devem ser impressos ( de 1 a 3 )
FB0F: 00 F1	170		INY	
FB03: A2 03	171		BCC #PRINTOP	:Nao imprime,todos^,entao o prox.
FB05: C0 04	172		LDX #5D3	:Tabela espaco para impr.mnemonic.
FB07: 00 04	173		CPY #5D4	
FB07: 90 F2	174		BCC #PRINTBL	
FB09: 68	175		PLA	:Recupera index. de mnemonic. em Y
FB0A: AB	176		TAX	
FB0B: 89 C3 F9	177		LDA #MNEHL,Y	:Busca 3 caract.mnemonic.em 2 Bytes
FB0E: 85 2C	178		STA #MNEH	
FB0D: 89 03 FA	179		LDA #MNEH,Y	
FB0F: 85 2D	180		STA #MNEH	

FBF5:	A9 00	181	PRM1	LDA	MS00	
FBF7:	AD 05	182		LDY	MS05	
FBF9:	06 20	183	PRM2	ASL	PMEN	
FBFB:	26 2C	184		RRL	LANE	
FBFD:	2A	185		ROL		:Desloca 5 bits do caract
FBFE:	89	186		DEI		:no acar. (limpa carry)
FBFF:	D0 F8	187		BNE	PRM2	
FC01:	A9 BF	188		ADC	M??	:Converte NASCII (soma OFFSET??)
FC03:	20 ED	189		JSR	COUT	:Imprime caractere (letra) ou (?)
FC06:	CA	190		DEX		
FC07:	D0 EC	191		BNE	PRM1	:Imprime ate a terceira letra
FC09:	20 48	192		JSR	PRBLM	:Imprime 3 espacos em branco
FC0C:	A4 2F	193		LDY	LENGTH	
FC0E:	A2 06	194		LDX	MS06	:Indexa CHAR1 e CHAR2
FC10:	ED 03	195	PRADR1	CPX	MS03	
FC12:	F0 1C	196		BEQ	PRADR5	:Se X=0 entao vai para PRADR5
FC14:	06 2E	197	PRADR2	ASL	FORMAT	:Prepara indicacao
FC16:	90 0E	198		BCC	PRADR3	:Imprime os form.(S,M,C,I,...,X,Y)
FC18:	80 83	199		LDA	CHAR1-1,X	
FC1B:	20 E9	100		JSR	COUT	
FC1E:	80 89	101		LDA	CHAR2-1,X	
FC21:	F0 03	102		BEQ	PRADR3	
FC23:	20 ED	103		JSR	COUT	
FC26:	CA	104	PRADR3	DEX		
FC27:	D0 E7	105		BNE	PRADR1	
FC29:	60	106		RTS		
		107				
FC2A:	88	108	PRADR4	DEY		
FC2B:	20 E7	109		BMI	PRADR2	
FC2D:	20 DA	110		JSR	PRBYTE	:Impr.contudo ou ender.de memr.
FC30:	45 2E	111	PRADR5	LDA	FORMAT	
FC32:	C9 08	112		CMP	MS08	
FC34:	81 3A	113		LDA	(PCL),Y	
FC36:	90 F2	114		BCC	PRADR4	
FC38:	20 56	115	RELADR	JSR	PCADJ3	
FC3F:	AA	116		TAX		:PCL,PCH+OFFSET+: PARA 8,Y
FC3C:	E8	117		INX		
FC3D:	D0 01	118		BNE	PRMTX	:+1 PARA Y,X
FC3F:	C8	119		INP		
FC4B:	98	120	PRMTYX	TYA		
FC41:	20 DA	121	PRMTAX	JSR	PRBITE	:Imprime o acumulador em HEX
FC44:	8A	122	PRMTX	TXA		
FC45:	4C DA	123		JMP	PRBITE	:Imprime o acumulador em HEX
FC4B:	A2 03	124	PRBLM	LDX	MS03	:Prepara X para contar 3 vezes
FC46:	A9 AD	125	PRBL2	LDA	M? ?	:Carrega o acumulador com "espaco"
FC4C:	20 E9	126	PRBL3	JSR	COUT	:Imprime espaco "X" vezes
FC4F:	CA	127		DEX		
FC50:	D0 F8	128		BNE	PRBL2	
FC52:	60	129		RTS		
		130				
FC53:	38	131	PCADJ	SEC		:LENGTH -1 0=1-byte , 1=2-bytes
FC54:	A5 2F	132	PCADJ2	LDA	LENGTH	2=3-bytes
FC56:	AA 38	133	PCADJ3	LDY	PCH	
FC5B:	AA	134		TAX		
FC59:	10 01	135		BPL	PCADJ4	:Testa sinal de deslocament
FC5B:	88	136		DEY		

```

F95C: 65 34 1137 PCADJ4 ADC PCL      :PCL + LENGTH (ou displacement) +1
F95E: 90 01 1138      BCC RETMS   para A carra em Y (POH)
F960: 08      1139      INY
F961: 60      1140 RETMS   RTS
      351      FUT  PARTE2P_D2
      31
      32 * BYTES FMT1 XXXXXX1YD INSTRS
      33 * Se Y=0, entao meio byte a esquerda
      34 * Se Y=1, entao meio byte a direita
      35 * (X=INDEX)
      36
F962: 04 20 54 17 FMT1   HEX 042054
F965: 30 00 18      HEX 3000
F967: 80 04 90 39      HEX 800490
F96A: 03 22 110      HEX 0322
F96C: 54 33 00 111      HEX 543300
F96F: 80 04 112      HEX 8004
F971: 90 04 20 113      HEX 900420
F974: 54 33 114      HEX 5433
F976: 00 80 04 115      HEX 008004
F979: 90 04 116      HEX 9004
F97B: 20 54 30 117      HEX 205430
F97E: 00 80 118      HEX 0080
F980: 04 90 00 119      HEX 049000
F983: 22 44 120      HEX 2244
F985: 33 00 08 121      HEX 330008
F988: 44 00 122      HEX 4400
F98A: 11 22 44 123      HEX 112244
F98D: 33 00 124      HEX 3300
F98F: 08 44 A9 125      HEX 0844A9
F992: 01 22 126      HEX 0122
F994: 44 33 00 127      HEX 443300
F997: 80 04 128      HEX 8004
F999: 90 01 22 129      HEX 900122
F99C: 44 33 130      HEX 4433
F99E: 00 80 04 131      HEX 008004
F9A1: 90 132      HEX 90
F9A2: 26 31 87 133      HEX 263187
F9A5: 9A 134      HEX 9A
F9A6: 00 135 FMT2   DFB 580      ERRO
F9A7: 21 136      DFB 521      IMEDIATO
F9A8: 81 137      DFB 501      2-PAGE
F9A9: 82 138      DFB 582      ABSOLUTO
F9AA: 00 139      DFB 503      AMPLIADO
F9AB: 00 140      DFB 500      ACUMULADOR
F9AC: 59 141      DFB 559      (2PAG),X1
F9AD: 40 142      DFB 540      (2PAG),Y
F9AE: 91 143      DFB 591      2PAGE,X
F9AF: 92 144      DFB 552      ABS,X
F9B0: 86 145      DFB 586      ABS,Y
F9B1: 4A 146      DFB 54A      (ABS)
F9B2: 85 147      DFB 585      2PAG,Y
F9B3: 90 148      DFB 590      RELATIVO
      149
F9B4: AC 150 CHR1   ASC ". ."
F9B5: A9 151      ASC ".)"
F9B6: AC 152      ASC ". ."

```

F9B7: A3	153	ASC	"W"
F9B8: A8	154	ASC	"V"
F9B9: A4	155	ASC	"Y"
	156		
F9BA: 09	157	CHAR2	ASC "Y"
F9BB: 03	158	HEX	00
F9BC: 08	159	ASC	"3"
F9BD: A4	160	ASC	"5"
F9BE: A4	161	ASC	"5"
F9BF: 00	162	HEX	00
	163		
	164	* NHEML e da Formas:	
	165	*	(A) XXXXX000
	166	*	(B) XXXYY100
	167	*	(C) XXXXSD10
	168	*	(D) XXXYY110
	169	*	(E) XXXYY101
	170	*	(X-INDEX)
	171		
F9C0: 1C BA 1C	172	NHEML	HEX 1CBA1C
F9C3: 23 5D 8B	173		HEX 235D8B
F9C6: 18 A1 9D	174		HEX 18A19D
F9C9: 8A 1D 23	175		HEX 8A1D23
F9CC: 9D 8B 1D	176		HEX 9D8B1D
F9CF: A1 00 29	177		HEX A10029
F9D2: 19 AE 6F	178		HEX 19AE6F
F9D5: A8 19 23	179		HEX A81923
F9D8: 24 53 18	180		HEX 245318
F9DB: 23 24 53	181		HEX 232453
F9DE: 19 A1	182		HEX 19A1
F9E0: 00 1A 5B	183		HEX 001A5B
F9E3: 38 A5 6F	184		HEX 38A56F
F9E6: 24 24	185		HEX 2424
F9E9: AE AE A8	186		HEX AEAEA8
F9EC: A0 29 00	187		HEX A02900
F9EF: 7C 00	188		HEX 7C00
F9F0: 15 9C 6D	189		HEX 159C6D
F9F3: 9C A5 6F	190		HEX 9CA56F
F9F6: 29 53	191		HEX 2953
F9F9: 84 13 34	192		HEX 841334
F9FB: 11 A5 6F	193		HEX 11A56F
F9FE: 23 40	194		HEX 2340
FA00: 00 62 5A	195	NHEML	HEX 00625A
FA03: A8 26 62	196		HEX A82662
FA06: 94 8B 54	197		HEX 948B54
FA09: 44 C8 54	198		HEX 44C854
FA0C: 68 44 EB	199		HEX 6844EB
FA0F: 94 00 84	1100		HEX 940084
FA12: 00 84 74	1101		HEX 008474
FA15: 84 28 6E	1102		HEX 84286E
FA18: 74 F4 CC	1103		HEX 74F4CC
FA1B: A4 72 F2	1104		HEX A472F2
FA1E: A4 BA	1105		HEX A4BA
FA20: 00 A4 A2	1106		HEX 00A4A2
FA23: A2 74 74	1107		HEX A27474
FA26: 74 72	1108		HEX 7472
FA29: 44 68 B2	1109		HEX 4468B2

```

FA29: 32 82 00 >110      HEX 328200
FA2E: 22 00 >111      HEX 2200
FA30: 1A 1A 26 >112      HEX 1A1A26
FA30: 26 72 72 >113      HEX 267272
FA36: 08 08 >114      HEX 0808
FA38: 04 0A 26 >115      HEX 040A26
FA38: 48 44 44 >116      HEX 484444
FA3E: A2 CB >117      HEX A2CB
FA40: 85 45 >118      IRR STA ACC      ;Salva acumulador em ACC
FA42: 68 >119      PLA
FA43: 48 >120      PHA
FA44: 04 >121      ASL
FA45: 0A >122      ASL
FA46: 0A >123      ASL
FA47: 30 03 >124      BHI BREAK      ;Teste para BREAK
FA49: 6C FE 03 >125      JMP (IRSLDCL)
      >126
FA4C: 28 >127      BREM      FLP
FA4D: 20 9F FB >128      JSR SAV2      ;Salva regs.no BREAK, incluindo PC
FA50: 48 >129      PLA
FA51: 85 3A >130      STA PCL
FA53: 48 >131      PLA
FA54: 85 38 >132      STA PCH
FA54: 6C FD 03 >133      JMP (BRKVL)      ;Normalmente JMP XBRK
FA59: 20 85 FB >134      JSR INSDS1      ;Imprime PC no BREAK
FA5C: 20 CF FA >135      JSR RGOSP1      ;Imprime conteúdo dos regs.
FA5F: 4C 5D FF >136      JMP NON      ;Retorna no início do monitor
      >137
      >138      *Ao ligar o aparelho o vetor RESET indica este endereço
      >139
FA62: 4C 6E FA >140      RESET      JMP RESET1
      >141
FA65: 90 04 >142      SCRN2      BCC RTNSKZ
FA67: 4A >143      LSR      ;transfere o DNS para o des
FA68: 46 >144      LSR
FA69: 4A >145      LSR
FA6A: 4A >146      LSR
FA6B: 29 0F >147      RTNSKZ      AND #50F      ;Zera o DNS
FA6D: 60 >148      RTS
      >149
FA6E: 08 >150      RESET1      CLD      ;limpa modo decimal
FA6F: 20 84 FE >151      JSR SETMON
FA72: 20 38 FB >152      JSR INT1
FA75: 20 93 FE >153      JSR SETVID      ;Inicia memória e formatos
FA7B: 20 89 FE >154      JSR SETVID      ;Video como disp.de saída de dados
FA7B: 20 89 FE >154      JSR SETKBD      ;Teclado como entrada de dados
FA7B: 20 3A FF >155      JSR BELL      ;Gera um BELL
FA7E: A0 F3 03 >156      LDA SOFTEW      ;Verif. se e warm ou cold start
FA81: 49 A5 >157      OR #A5
FA83: CD F4 03 >158      CNP PWREUP
FA84: 00 1D >159      BNE PFREG      ;Se diferente faz um cold start
FA8D: A0 F2 03 >160      LDA SOFTEW      ;faz um warm start
FA90: 00 12 >161      BNE EXT10      ;Se warm start vai para EXT10
FA90: A9 C2 >162      LDA #5C2
FA9F: CD F3 03 >163      CNP SOFTEW
FA9F: 00 08 >164      BNE EXT10
FA9F: A0 03 >165      FIXREV      LDY #5D3      ;SOFTEW= #5D3

```

```

FAR6: 8C F2 03 1166      STY  SOFTEVL
FAR9: 20 33 EF 1167      JSR  EXTIO1 ;Verifica presenca de perifericos
FARC: 4C 00 C2 1168      JMP  BASIC  ;Inicio da BASIC
FARF: 20 33 EF 1169      EXTIO JSR  EXTIO1 ;Verifica presenca de perifericos
FAR2: 4C F2 03 1170      MOFVX JMP  (SOFTEVL)
FAR5: 20 79 EC 1171      PPRGJ JSR  PPRGJ
FAR8: 20 5F FB 1172      PARUP JSR  DFTX2000 ;Limpa tela, imprime TK2000
FARB: A2 05 1173         LOX  MS05 ;Transfere a tabela VETSTART
FAR0: 80 97 FB 1174      SETPLP LDA  VISTRI-1,X ;para os vetores!
FAR3: 90 EF 03 1175      STA  BRXVL-1,X (BRXV), (SOFTEVL), PWDUPL
FAR3: CA 1176           DEX
FAR4: 00 F7 1177         BNE  SETPLP ;Carregue ate o ultimo vetor
FAR6: 86 00 1178         STX  LOCD ;(LOCD) = $C100
FAR8: A9 C1 1179         LDA  $C1
FAR8: 85 01 1180         STA  LOC1
FARC: A0 07 1181      SLOOP LBY  MS07 ;Y = 07
FAR1: 81 00 1182      NXYBYT LDA  (LOC0),Y
FAR0: 09 8F F5 1183      CMP  DISKID-1,Y ;Compara (LOC0), Y com DISKID, Y
FAR3: 00 CF 1184         BNE  FIXSEV ;Se for ( ) va para FIXSEV
FAR3: 88 1185           DEY
FAR3: 88 1186           DEY
FAR7: 10 F5 1187         BPL  NXYBYT ;Compara ate ultimo byte
FAR9: 4C 00 00 1188      JMP  (LOC0) ;Jump para diskdriver
1189
352
FAR2: 20 BE FD 11      RDGOSP JSR  CR0UT ;Faz um RETURN
FARF: A9 FD 12      RDGOSP1 LDA  $AREG ;A3- $07FD (AREG)
FAD1: 85 40 13         STA  A3L
FAD3: A9 07 14         LDA  $AREG
FAD5: 85 41 15         STA  A3H
FAD7: A2 FB 16         LOX  MSFB ;X como contador
FAD9: A9 A0 17      RDSP1 LDA  #"" ;imprime um espaco
FAD8: 20 ED FD 18      JSR  COUT
FAD1: 80 CA FD 19      LDA  RTBL-251,X ;imprime o nome do registrador
FAE1: 20 ED FD 110     JSR  COUT
FAE4: A9 80 111     LDA  #"" ;imprime o caracter = (igual)
FAE6: 20 ED FD 112     JSR  COUT
FAE9: 80 F5 04 113     LDA  AREG-251,X ;Impr.o conteudo do reg. em hex
FAEC: 20 DA FD 114     JSR  PRBYTE
FAEF: E8 115         INX
FAFD: 30 E7 116     BNE  RDSP1 ;Apresente todos registradores
FAF2: 60 117         RTS
118
FAF3: 8F F3 FD 119     NEMS LDA  NENTAB,Y ;imprime caracter de NENTAB,Y
FAF6: 20 ED FD 120     JSR  COUT
FAF9: C8 121         INY ;Proximo caracter
FAFA: CA 122         DEX
FAFB: 00 F6 123     BNE  NEMS ;Ate o ultimo
FAFD: 60 124         RTS
125
FAFE: FF FF FF 126     HEX  FFFFFFFF
FBD1: FF
FBD2: 20 14 FD 127     IM2 JSR  IM3
FBDS: A5 27 128     SEPTAB LDA  BRASH ;Separa a tabela de codigos ASCII
FBD7: 0A 129         ASL ;de 4 em 4
FBD8: 0A 130         ASL
FBDF: 24 27 131     BIT  BRASH

```

FB0B: 08	132		PHP		
FB0C: 49 00	133		ADC #500		;Se CTRL soma 1 ao acum. (carry)
FB0E: 28	134		PLP		
FB0F: 10 03	135		RPL WCTL		;Se SHIFT/CTRL, +3 ao acum. (2carry)
FB11: 18	136		CLC		;Se e SHIFT soma 2 ao acumul.
FB12: 49 02	137		ADC #502		
FB14: 0E 0E 04	138	NXCTL	STX XSW		;Salva X em XSW
FB17: 4A	139		TAX		;Prepara indexacao
FB1B: 80 9D F5	140		LDA TABASC,X		
FB1B: AE 0E 04	141		LDX XSW		;Recupera X em XSW
FB1E: 38	142		SEC		
FB1F: 60	143		RTS		
	144				
FB20: A4 24	145	NXTP05	LDY CH		;T= posicao horizontal do cursor
FB22: 20 88 FD	146		JSR PRCD1		;Imprime caracter no video
FB24: E6 24	147	NXTP051	INC CH		;Desloca cursor a direita
FB27: A5 24	148		LDA CH		;Ultrapassou a largura da tela ?
FB29: C5 21	149		CMP #WIDTH		;Sim, faz um COOCR
FB2B: 80 01	150		BCS JCOCR		
FB2B: 60	151		RTS		
	152				
FB2E: 4C 17 F1	153	JCOCR	JMP COOCR		
FB31: A4 20	154	POSE6	LDY #WDLFT		;Posiciona cursor a esquerda
FB33: 84 24	155		STY CH		
FB35: 60	156		RTS		
	157				
FB36: FF FF FF	158		HEX FFFFFFFF		
FB39: FF FF					
FB3B: 20 79 EC	159	INT1	JSR PRPR0		
FB3E: A9 00	160		LDA #500		;Zera status
FB40: 80 F3 07	161		STA STATUS		
FB43: A9 00	162		LDA #500		;Faz topo do texto na linha 0
FB45: F0 05	163		BEQ SETM0		
FB47: 20 36 F8	164		JSR CLRTP		;Limpa parte superior da tela
FB4A: A9 14	165		LDA #514		;Faz topo do texto na linha #514
FB4C: 85 22	166	SETM0	STA #WDTOP		
FB4E: A9 00	167		LDA #500		;0, como inicio a esquerda da tela
FB50: 85 20	168		STA #WDLFT		
FB52: A9 20	169		LDA #520		;Prepara largura da tela com #520
FB54: 85 21	170		STA #WIDTH		
FB56: A9 18	171		LDA #518		;Prepara a #518 como ultima linha
FB5B: 85 23	172		STA #WDSM		
FB5A: A9 17	173		LDA #517		;Cursor vertical na ultima linha
FB5C: 85 25	174	TABU	STA CV		
FB5E: 60	175		RTS		
	176				
FB5F: 20 58 FC	177	OPTX2000	JSR HOME		;Faz um HOME
FB62: A9 0E	178		LDA #50E		;Posiciona cursor horizontal
FB64: 85 24	179		STA CH		;Na coluna #50E
FB66: 40 06	180		LDY #506		
FB68: 89 96 F5	181	STITLE	LDA TITULO-1,Y		;Pega caracter de titulo
FB6B: 20 88 FD	182		JSR PRCD1		;Imprime o caracter
FB6E: E6 24	183		INC CH		;Proxima coluna
FB70: 88	184		DEY		
FB71: 00 F5	185		BNE STITLE		;Até o ultimo caracter
FB73: 60	186		RTS		
	187				



```

FB74: 00 F3 03 188      LDA  SORTEM
FB77: 49 45 189      EOR  #5A5
FB79: 80 F4 03 190      STA  PAREADP
FB7C: 60 191      RTS
      192
FB7D: C9 80 193      VDMAIT CMP  #580 ;Compara se e um RETURN
FB7F: 00 14 194      BNE  MONAIT
FB81: 20 43 F0 195      JSR  SCAM1 ;Se e RETURN varre teclado
FB84: 90 00 196      BCC  LDRET ;Se nao tem tecla apertada
FB84: C9 93 197      CMP  #593 ;E CONTROL-S ?
FB88: 00 09 198      BNE  LDRET
FB8A: 20 43 F0 199      RPSCH1 JSR  SCAM1 ;Tem tecla apertada ?
FB8D: 90 F8 100      BCC  RPSCH1 ;Loop ate ter tecla apertada
FB8F: C9 91 101      CMP  #591 ;E CONTROL-R
FB91: 00 F7 102      BNE  RPSCH1 ;Se nao, novo loop
FB93: A9 80 103      LDRET LDA  #580 ;Prepara para um RETURN
FB95: 4C F0 F8 104      MONAIT JMP  DFC00
      105
FB98: 59 106      VISTRT HEX  59
FB99: FA 107      HEX  FA
FB9A: 80 108      HEX  80
FB9B: C2 109      HEX  C2
FB9C: 67 110      HEX  67
      111
FB9D: 05 45 112      SAVE1 STA  ACC ;Salva o acumulador em ACC
FB9F: 8E F1 07 113      SW2  STX  XREG ;Salva X em XREG
FBA2: 45 45 114      LDA  ACC ;Salva o acumulador em AREG
FBA4: 80 F0 07 115      STA  AREG
FBA7: 4C 40 FF 116      JMP  SW3 ;Salva Y, status e stack pointer
      117
FBA8: 00 24 55 118      TABCOR HEX  002455F0A0B5FF
FBA0: 7F 80 AA 05 FF 119
FB82: 48 120      POSCUR PWA ;Salva acumulador e Y no stack
FB83: 98 121      TTA
FB84: 48 122      PWA
FB85: 20 F3 F8 123      JSR  CV08 ;DELH= CV#8
FB88: 20 73 FC 124      JSR  EPVID ;Prep.R4 com end. da pag de video
FB8B: A0 00 125      LDY  #500 ;Zera Y (indicador)
FB8D: 10 126      CLC ;Situa posic.horiz. na pag de vid.
FB8E: A5 08 127      LDA  R4L
FB8D: 45 24 128      ADC  CH
FB82: 85 08 129      STA  R4L
FB84: 81 08 130      INVCUR LDA  (R4L),Y ;Prepara cursor como caracter de
FB86: 49 7F 131      EOR  #57F 8 microlinhas no modo inverso
FB88: 91 08 132      STA  (R4L),Y
FB8A: A5 09 133      LDA  R4H ;Proxima microlinha
FB8C: 49 04 134      ADC  #504
FB8E: 85 09 135      STA  R4H
FB8D: 29 40 136      AND  #540
FB82: F0 F0 137      BEW  INVCUR ;Ata a ultima microlinha
FB84: 48 138      PLA ;Recupera Y e acumulador do stack
FB85: A8 139      TAY
FB86: 48 140      PLA
FB87: 40 141      RTS
      353      PUT  PARTE26_02

```

FBD0: EA	12		NOP		
FBD1: C9 07	13	BELL0	CHP	Bell	;Compara se o caracter BELL
FBD2: 00 12	14		BNE	RTS28	;Se nao for retorna
FBD3: A9 40	15	BELL1	LDA	#540	
FBD4: 20 A6 FC	16		JSR	WAIT	;Atraso de .002 segundos
FBE2: A0 C0	17		LDI	#5C0	
FBE4: A9 0C	18	BELL2	LDA	#50C	;Gera um som a 10KHz
FBE6: 20 A6 FC	19		JSR	WAIT	de duracao de .18953 segundos
FBE7: A0 30 C0	110		LDA	SPKR	
FBE8: 00	111		DET		
FBE9: 00 F5	112		BNE	BELL2	
FBEF: 40	113	RTS28	RTS		
	114				
FBFD: 4C 20 F8	115	NOPOS1	JMP	NOXPOS	;Imprime caracter e avanca cursor
FBF3: A5 25	116	CVXB	LDA	CV	;Multiplica CV * 8
FBF5: 0A	117	AXB	ASL		;Armazena resultado em DELH
FBF6: 0A	118		AGL		;Obs: CV continua intacto
FBF7: 0A	119		ASL		
FBF8: 05 E3	120		STA	DELH	
FBFA: 40	121		RTS		
	122				
FBFB: EA	123		NOP		
FBFC: EA	124		NOP		
FBFD: C9 F0	125	IOFC00	CHP	#5F0	
FBFF: F0 22	126		BEQ	UP	;Saba uma linha
FC01: C9 F1	127		CHP	#5F1	
FC03: F0 61	128		BEQ	LF	;Desce uma linha
FC05: C9 A0	129	VIDOUT	CHP	#5A0	;Se for != A0 ou != 7F
FC07: 80 E7	130		BCS	NOPOS1	ir para NOPOS1
FC09: A8	131		TAY		
FC0A: 10 E4	132		BPL	NOPOS1	
FC0C: C9 80	133		CHP	#580	;Se for RETURN faz um CR
FC0E: F0 52	134		BEQ	CR	
FC10: C9 8A	135		CHP	#58A	;Se for N8A (CTRL-E) faz um LF
FC12: F0 52	136		BEQ	LF	
FC14: C9 88	137		CHP	#588	;E #588 (CONTROL-8) ?
FC16: 90 C1	138		BNE	BELL0	;Se nao, BELL0
FC18: A5 24	139	DESE50	LDA	CH	;Se o cursor nao estiver
FC1A: C5 20	140		CHP	UNDLFT	totalmente a esquerda da tela,
FC1C: 00 02	141		BNE	DESE50	desloque uma posicao a esquerda
FC1E: E8	142		INX		
FC1F: 40	143		RTS		
FC20: C6 24	144	DESE50	DEC	CH	
FC22: 40	145		RTS		
	146				
FC23: A5 22	147	UP	LDA	UNDTOP	;Se o cursor nao esta no topo
FC25: C5 25	148		CHP	CV	sobre uma linha
FC27: 80 00	149		BCS	RTS4	
FC29: C6 25	150		DEC	CV	
FC2B: EA	151	RTS4	NOP		
FC2C: 40	152	RTS4	RTS		
	153				
FC2D: A9 0C	154	PRBES0	LDA	#50C	;Imprime o caracter "Y" (CTRL-Y)
FC2F: 20 E0 F0	155		JSR	OUT	
FC32: 40	156		RTS		
	157				
FC33: 20 18 F0	158	INX	JSR	INA	;Testa se tem tecla apertada

FC36: 20 0C FD	159	JSR	IMD	;Busca código de tela
FC39: 60	160	RTS		
	161			
FC3A: AE F1 07	162	RSTRI	LDX XREG	;Restaura X de XREG
FC3B: AC F2 07	163		LDY YREG	;Restaura Y de YREG
FC40: 28	164		PLP	;Restaura status do stack
FC41: 60	165		RTS	
	166			
FC42: 44 24	167	CLEERP	LDY CR	
FC44: A5 25	168		LDA CV	
FC46: 20 98 FC	169	CLEOP1	JSR CLEOLZ	;Limpa uma linha horizontal
FC49: E6 25	170		IMC CV	;Desce uma linha
FC4B: A5 25	171		LDA CV	
FC4D: C5 23	172		CMF WNBTH	;Compara se e última linha
FC4F: 90 F5	173		BCC CLEOP1	;Se não e última linha, CLEOP1
FC51: 84 24	174		STY CR	;Posiciona cursor horizontal
FC53: 44 22	175		LDY WNDTOP	;Pos. cursor vert. na linha 0
FC55: 84 25	176		STY CV	
FC57: 60	177		RTS	
	178			
FC5B: A5 22	179	HOME	LDA WNDTOP	;Pos. no topo da tela
FC5A: 85 25	180		STA CV	
FC5C: 20 31 FB	181		JSR P05E5B	;Pos. à esquerda da tela
FC5F: 4C 46 FC	182		JMP CLEOP1	;Limpa a tela
	183			
FC62: A5 20	184	CR	LDA WNDLFT	;Pos.cursor a esq. do vídeo
FC64: 85 24	185		STA CR	
	186			
FC66: E6 25	187	LF	IMC CV	;Desce cursor uma linha de texto
FC68: A5 25	188		LDA CV	
FC6A: C5 23	189		CMF WNBTH	;Se era última linha de vídeo,
FC6C: 90 80	190		BCC WTS4	sobe uma linha e faz um SCROLL
FC6E: C6 25	191		DEC CV	semão retorna
FC70: 4C 36 F1	192		JMP SCROLL	
	193			
FC73: A5 E3	194	EPNV10	LDA DELH	;Calcula a próxima alinalinha
FC75: 0A	195		ASL	;4a pag. de vídeo dentro do box
FC76: 0A	196		ASL	
FC77: 29 1C	197		AND #51C	;Limita ender. alto em #51C ou #5C
FC79: 85 09	198		STA R4H	
FC7B: A5 E3	199		LDA DELH	;Calcula o terceiro dígito
FC7D: 6A	100		ROR	;do endereço da pag.de vídeo
FC7E: 6A	101		ROR	
FC7F: 6A	102		ROR	
FC80: 6A	103		ROR	
FC81: 29 03	104		AND #503	
FC83: 05 09	105		ORA R4H	
FC85: 05 46	106		ORA P04H	;Soma com o quarto dígito para
FC87: 05 09	107		STA R4H	formar o endereço alto de vídeo
FC89: A5 E3	108		LDA DELH	;Calcula o segundo dígito do
FC8B: 6A	109		ROR	endereço que somado a CH VAI
FC8C: 29 ED	110		AND #5ED	formar o endereço baixo de vídeo
FC8E: 85 08	111		STA R4L	
FC90: 6A	112		ROR	
FC91: 6A	113		ROR	
FC92: 29 18	114		AND #518	
FC94: 05 08	115		ORA R4L	

```

FC96: 05 08 1116 STA #4L
FC98: 60 1117 RTS
1118
FC99: A4 24 1119 CLREQ LDY CH
FC9B: AF A0 1120 CLEQLZ LDA #''
FC9D: 20 B6 FD 1121 CLREPE JSR PROCMO ;MASCII(ESPCAO)
FC9E: C8 1122 IMY ;prime caracter
FC9F: C4 21 1123 CPY WNDWTH ;fite o fim da linha
FC99: 9D FB 1124 BCC CLREPE
FC9A: A4 20 1125 LDY WNDLFT ;Posiciona Y a esquerda da tela
FC97: 60 1126 RTS
1127
FC9B: 30 1128 WAIT SEC
FC99: 48 1129 WAIT2 PNA ;Atraso:
FC9A: E9 D1 1130 WAIT3 SBC #501 (26+27*4+5*4*2)
FC9C: D0 FC 1131 BNE WAIT3
FC9E: 68 1132 PLA
FC9F: E9 D1 1133 SBC #501
FC9B: D0 F6 1134 BNE WAIT2
FC9B: 60 1135 RTS
1136
FC94: E6 42 1137 NEXT44 INC #4L ;Incrementa A4
FC96: D0 D2 1138 BNE NEXT41
FC98: E6 43 1139 INC #4H
FC9A: A5 3C 1140 NEXT41 LDA #4L ;Incrementa A1
FC9C: C5 3E 1141 CMP #2L
FC9E: A5 3D 1142 LDA #1H
FC9D: E5 3F 1143 SBC #2H ;Se A1=A2 seta carry
FC9C: E6 3C 1144 INC #1L
FC9A: D0 D2 1145 BNE RTS40
FC96: E6 3D 1146 INC #1H
FC9B: 60 1147 RTS40 RTS
354 PUT PARTE2R_02
31
FC99: A0 48 12 HEADR LDY #540 ;Escreve longo sinal em TAPEOUT
FC9B: 20 D8 FC 13 JSR ZERDLY ;Meios ciclos de 450us cada
FC9E: D0 FF 14 BNE HEADR ;Repete X vezes
FC9D: 69 FE 15 AOC #5FE
FC9C: 8D F5 16 BCS HEADR
FC94: A0 21 17 LDY #521 ;Escreve um sinal de SYNC (400us)
FC96: 20 B6 FC 18 WRBIT JSR ZERDLY ;Escreve 2 meios ciclos de:
FC99: C8 19 IMY ;250us(1) 500us(1)
FC9A: C8 110 IMY
FC9B: 80 111 ZERDLY DEY ;Primeiro contador de tempo
FC9C: D0 FD 112 BNE ZERDLY
FC9E: 9D D5 113 BCC WRTAPE
FC9D: A0 32 114 LDY #532
FC9E: 88 115 ONEDLY DEY ;Segundo contador de tempo
FC9C: D0 FD 116 BNE ONEDLY
FC9A: AC 2D CD 117 WRTAPE LDY TAPEOUT ;Muda "TAPEOUT" de estado
FC9B: A0 2C 118 LDY #52C
FC9A: CA 119 DEX ;Numero de meios ciclos
FC9B: 60 120 RTS ;a serem escritos
121
FC9E: A2 D8 122 RDBYTE LDY #50B ;Le 8 bits para formar um Byte
FC9E: 40 123 RDBYTE2 PNA ;Salva acumulador no slack
FC9F: 20 FA FC 124 JSR RDBIT ;Le bit de TAPEIN

```

```

FCF2: 68 125 PLA ;Reconstitui acumulador do stack
FCF3: 2A 126 ROL ;Introduz pelo carry, o bit lido
FCF4: A0 3A 127 LDI #55A ;7 com #5A como contador de tempo
FCF6: CA 128 DER ;Le proximo bit ate o 8
FCF7: 00 F5 129 BNE RDBYTE2
FCF9: 60 130 RTS
131
FCFA: 20 FD FC 132 RDBIT JSR RDBIT ;2 meios ciclos de onda de TAPEIN
FCFB: 00 133 RDBIT DER ;Decrementa T ate transicao
FCFC: A0 10 C0 134 LDA #B7P1M da onda de TAPEIN
FCFD: 45 2F 135 EOR LASTIM
F003: 10 FB 136 BPL RDBIT
F005: 45 2F 137 EOR LASTIM ;#mac.em LASTIM o ultimo estado
F007: 05 2F 138 STA LASTIM da forma de onda de TAPEIN
F009: C0 00 139 CFI #500 ;Se o bit=1, seta carry
F00B: 60 140 RTS
141
F00C: 6C 38 00 142 IM0 JMP (KSM) ;Se teclado, indica #F00F
F00F: 4C 02 FB 143 IN1 JMP IN2
F012: 8A 144 CURE50 TXA
F013: FD 67 145 BEQ JIMS ;Se inicio do buffer, novos dados
F015: 4C 98 F7 146 JMP PRENET ;Verif.caract.anter.do buffer IN
F018: CA 147 DECBUF DEX
F019: 20 82 FB 148 JSR POSCUR ;Apaga o cursor
F01C: 20 18 FC 149 JSR DESE50 ;Desloca uma posicao a esquerda
F01F: 20 82 FB 150 JSR POSCUR ;Reposiciona o cursor
F022: 4C 7C FD 151 JMP JIMS
F025: 20 82 FB 152 SRZCUR JSR POSCUR ;Apaga cursor
F028: 20 25 FB 153 JSR NEXTPOS1 ;Desloca uma posicao a direita
F02B: 20 38 FD 154 JSR PSOR ;Repos.o cursor e increm.buffer
F02E: 00 4C 155 BNE JIMS ;Se nao completou buffer JIMS
F030: 4C 30 FD 156 JMP POSCUR1 ;Se completou buffer POSCUR1
157
F033: FF FF 158 HEX FFFF
159
F035: 4C 0C FD 160 INTC JMP IM0
161
F038: 20 82 FB 162 PSOR JSR POSCUR
F03B: EB 163 INX ;Incrementa posicao do Buffer
F03C: 60 164 RTS
165
F03D: C9 00 166 POSCUR1 CMP #50B ;Compara com (- ou CTRL-@)
F03F: FD 01 167 BEQ CURE50 ;Se igual desloca a esquerda
F041: C9 95 168 CMP #595 ;Compara com (-) ou CTRL-U
F043: FD ED 169 BEQ SRZCUR ;Igual,desl.direita,incr. Buffer
F045: C9 F0 170 CMP #5FD ;Compara com ^
F047: FD 0F 171 BEQ SRCUR ;Sobe cursor se igual
F049: C9 F1 172 CMP #5F1 ;Compara com
F04B: FD 00 173 BEQ SRCUR ;Desce cursor se igual
F04D: C9 9B 174 CMP #59B ;Compara com CTRL-X
F04F: FD 13 175 BEQ JPRBES0 ;Reinicia Buffer se igual
F051: 4C 7E EF 176 JMP CARBUF ;Carrega no Buffer IN
F054: 00 05 177 SRROUT BNE SROUT ;Nao complet.buffer, impr.caracter
F056: FD 0C 178 BEQ JPRBES0 ;Completo Buffer, reinicia Buffer
F05B: 20 82 FB 179 SRZCUR JSR POSCUR ;Prepara cursor
F05D: 20 ED FD 180 SROUT JSR CONIT ;Imprime caracter
F05E: 20 82 FB 181 JSR POSCUR ;Reposiciona cursor

```

FD61:	4C 7C FD	182	JMP	JIMS	:Testa estrada de dados p/teclado	
		183				
FD64:	20 20 FC	184	JMPRESA	JSR	PRRESA	:Imprime "\n"
FD67:	20 9C FD	185	GETLAZ	JSR	CRONT	:faz um CR
FD6A:	85 33	186	GETLN	LDA	PRORPT	:Impr. carac.de inicio"Q,I,J"
FD6C:	20 ED FD	187		JSR	COU1	
FD6F:	8F AD	188	GETLN1	LDA	" "	:Limpa buffer IN
FD71:	A2 00	189		LDX	WS00	:X como cont. de compr. do buffer
FD73:	90 00 02	190	CLRBN	STA	IN,X	
FD76:	EB	191		INX		
FD77:	90 FA	192	BNE	CLRBN		:Limpa buffer ate o fia
FD79:	26 82 FB	193	JSR	POSCUR		:Coloca o cursor no video
FD7C:	20 33 FC	194	JMS	JSR	IMS	:Verif.entrada de dados p/teclado
FD7F:	4C A2 EF	195	JMP	JSP	TSPGAL	:Se PDA=0, continua em WFLIN
FD82:	C9 80	196	WFLIN	CMP	WS00	:Compara com RETURN
FD84:	90 87	197	BNE	POSCUR1		:Se nao, faz nova comparacao
FD85:	90 00 02	198	ADDIMP	STA	IN,X	:Se e RETURN coloca no buffer IN
FD89:	EA	199		HOP		
FD8A:	EA	100		HOP		
FD8B:	20 86 FF	101	CROUT1	JSR	CLREOL1	:Limpa linha a partir do cursor
FD8E:	8F 90	102	CROUT	LDA	WS00	:Carrega codigo ASCII de RETURN
FD90:	00 58	103	BNE	COU1		:Saida de dados
FD92:	AA 30	104	PRAI	LDD	ASH	:#ASH
FD94:	A6 3C	105		LDD	AIL	:#AIL
FD9A:	20 9C FD	106	PRYX2	JSR	CROUT	:faz um RETURN
FD99:	20 40 F9	107	JSR	PRNTYX		:Impr.Y e X hexadecimal
		108	*			:(Endereco contido em An)
FD9C:	A0 00	109	LDD	WS00		
FD9E:	8F AD	110	LDA	"-"		:Imprime um traco "-"
FD9D:	4C ED FD	111	JMP	COU1		
FD93:	A5 3C	112	XAMB	LDA	AIL	:Ajusta A2 para imprimir
FD95:	09 07	113		ORA	WS07	endereco de final 0 ou 8
FD97:	85 3E	114		STA	A2L	
FD99:	A5 30	115		LDA	A1H	
FD9B:	85 3F	116		STA	A2H	
FD9D:	A5 3C	117	NOOBCH	LDA	AIL	:Verif.se e ender.de final 0 ou 8
FD9F:	29 07	118		AND	WS07	
FD91:	00 03	119	BNE	DATAOUT		:Nao, imprime conteudo do endereco
FD93:	20 92 FD	120	XAM	JSR	PRAI	:Sim, impr.o ender. contido em A1
FD9A:	8F AD	121	DATAOUT	LDA	" "	:Imprime um espaco
FD9B:	20 ED FD	122		JSR	COU1	
FD9B:	81 3C	123		LDA	(A1),Y	:Impr.conteudo do end.A1 em hexa
FD9D:	20 DA FD	124		JSR	PRYTE	
FD9D:	20 BA FC	125		JSR	WEXTAL	:Incrum. A1, se A1=A2 seta carry
FD9D:	90 EB	126		BCC	NOOBCH	:Verif. se e vez de impr. ender.
FD9D:	60	127	RTS4C	RTS		
		128				
FD9A:	4A	129	XAMPN	LSR		:E' acdo XAM, ADD ou SUB
FD97:	90 EA	130		BCC	XAM	
FD99:	4A	131		LSR		:Verifica se e soma ou subtracao
FD9A:	4A	132	TSTOP	LSR		
FD9B:	A5 3E	133		LDA	A2L	:Carrega A com A2L
FD9D:	90 02	134		BCC	ADD	:Se for soma faz ADD
FD9F:	49 FF	135		EDR	WSFF	:Se sub, faz complemento 2
FD91:	65 3C	136	ADD	ADC	AIL	:Soma AIL com carry ao acumulador
FD93:	4B	137		PHA		:Salva resultado no stack
FD94:	8F 60	138		LDA	"*"	:Imprime "*" seguido o resultado

FD04:	20	ED	FD	1139	JSR	COUT	
FD09:	68			1140	PLA		;Retorna resultado do stack
FD0A:	48			1141	PRBYTE	PHA	;Salvo byte hexa a ser impresso
FD0B:	4A			1142		LSR	;Transfere o DMS para o ds
FD0C:	4A			1143		LSR	
FD0D:	4A			1144		LSR	
FD0E:	4A			1145		LSR	
FD0F:	20	ES	FD	1146	JSR	PRHEXZ	;Subrotina de impressao em Hexa
FD2:	68			1147	FLA		;Retorna byte
FD2:	29	DF		1148	PRHEX	AND #50F	;Imprime o ds em hexadecimal
FD2:	09	80		1149	PRHEXZ	ORA #580	;Transforma em ASCII
FD7:	C9	8A		1150	CMR	#58A	;Se for entre 0 e 9 impr.dirto,
FD9:	90	02		1151	BCC	COUT	senao soma 6 e imprime letra
FD9:	69	06		1152	ADC	#506	de A a F
FD2:	4C	36	00	1153	COUT	JMP (CSM,)	;End. da rotina de saida de dados
FD0:	4C	A6	F1	1154	COUT1	JMP COUT2	*Normalmente = #50F0
				1155	PUT	PARTE25,02	
				11			
FD3:	40	CF	CB	12	NEXTAB	ASC	" OK WAIT"
FD4:	40	D7	Cl	C9	DA		
				13			
FD8:	99	EC		14	SAVEADR	DA	SAVE1-1
FD9:	40	ED		15	LOADADR	DA	LOAD1-1
				16			
FDFF:	FF			17		HEX	FF
				18			
FE0:	C6	34		19	BLI	DEC	YSRV
FE2:	F0	9F		110	BEQ	XAMB	;Volta uma posicao no Buffer IN,Y
FE0:	CA			111	BLANK	DEX	;Se pos.0, impr.conteudo em hexa
FE5:	00	16		112	BNE	SETM0Z	;Seta MODE e termina BLI
FE7:	C9	8A		113	CMR	#:"	;Compara com ":"
FE9:	00	88		114	BNE	XAMPH	;Se nao, imprime ender.e conteudo
FE8:	85	31		115	STOR	STA MODE	;Sim, seta MODE e transf.conteudo
FE0:	A5	3E		116	LDA	A2L	de A2L para o endereco indicado
FE0:	91	40		117	STA	(ASL),Y	por A0
FE1:	E6	40		118	INC	A3L	;Proximo endereco em A3
FE1:	00	02		119	BNE	RTSS	
FE1:	E6	41		120	INC	A3H	
FE1:	60			121	RTSS	RTS	
				122			
FE1:	A4	34		123	SETM00	LDF	YSRV
FE1:	89	FF	01	124	LDA	IM-1,Y	;Salva cod.de (+), (+), (-), (-)
FE1:	85	31		125	SETM0Z	STA MODE	
FE1:	40			126		RTS	
				127			
FE2:	A2	01		128	LT	LDA #501	;X para transferir o DMS primeiro
FE2:	85	3E		129	LT2	LDA A2L,X	;Copia 2 bytes de A2 em A4 e A5
FE2:	95	42		130		STA A4L,X	
FE2:	95	44		131		STA A5L,X	
FE2:	CA			132		DEX	
FE2:	10	F7		133		BPL	LT2
FE2:	60			134		RTS	
				135			
FE2:	81	3C		136	MOVE	LDA (ASL),Y	;Carrega 1 byte de A1 para A4
FE2:	91	42		137		STA (AHL),Y	
FE3:	20	84	FC	138	JSR	NXTA4	;Incrementa A1 e A4 ate A1=A2
FE3:	90	F7		139	BCC	MOVE	;Se A1<A2 carrega proximo byte

FE35: 60	>40		RTS	
	>41			
FE36: 81 3C	>42	WY	LDA (A4L),Y	;Compara A1 com A4
FE38: 01 42	>43		CMP (A4L),Y	
FE3A: FD 1C	>44		BEQ WYOK	;Se igual compara proximo Byte
FE3C: 28 92 FD	>45		JSR PRAS	;Se (), impr.ender. contido em A1
FE3F: 81 3C	>46		LDA (A4L),Y	;Impr.byte contido em A1 em hex.
FE41: 20 0A FD	>47		JSR PRBYTE	
FE44: A9 A0	>48		LDA #' "	;Imprime um espaco
FE46: 20 ED FD	>49		JSR COUT	
FE49: A9 A0	>50		LDA #' ('	;Abre parenteses
FE4B: 20 ED FD	>51		JSR COUT	
FE4E: 81 42	>52		LDA (A4L),Y	;Imprime o Byte contido no ender.
FE50: 20 0A FD	>53		JSR PRBYTE	A4 em hexadecimal
FE53: A9 A9	>54		LDA #' ('	;Fecha parenteses
FE55: 20 ED FD	>55		JSR COUT	
FE58: 20 04 FC	>56	WYOK	JSR NRTA4	;Incrementa A1 e A4 ate A1=A2
FE5B: 90 09	>57		BCC WY	;Se A1<A2 entao verif.prox.byte
FE5D: 60	>58		RTS	
	>59			
FE5E: 20 75 FE	>60	MLIST	JSR AIPC	;Move A1 p/PC, se end.especificado
FE61: A9 14	>61		LDA #514	;Disassembla 20 instracoes
FE63: 40	>62	L1ST2	PHA	
FE64: 20 00 FB	>63		JSR INSTOSP	;Imprime a instracao
FE67: 20 53 F9	>64		JSR PCADJ	;Ajusta PC para proxima instracao
FE6A: 85 3A	>65		STA PCL	
FE6C: 84 38	>66		STY PCH	
FE6E: 68	>67		PLA	
FE6F: 30	>68		SEC	
FE70: E9 01	>69		SBC #501	;Disassembla ate a vigesima inst.
FE72: 00 EF	>70		BNE L1ST2	
FE74: 60	>71		RTS	
	>72			
FE75: 8A	>73	AIPC	TXA	;Se X=1, transfere A1 para PC,
FE76: FD 07	>74		BEQ A1PCRTS	;Se X=0, nao transfere nada
FE78: 85 3C	>75	A1PCLP	LDA A1L,X	
FE7A: 95 3A	>76		STA PCL,X	
FE7C: CA	>77		DEX	
FE7D: 10 F9	>78		BPL A1PCLP	
FE7F: 60	>79	A1PCRTS	RTS	
	>80			
FE80: A0 7F	>81	SETIMV	LDY #57	;Carrega IMVFLG com #57,
FE82: 00 02	>82		BNE SETIFLG	produzindo video no modo inverso
FE84: A0 00	>83	SETHORN	LDY #500	;Carrega IMVFLG com #500,
FE8A: 84 32	>84	SETIFLG	STY IMVFLG	produzindo video no modo normal
FE8D: 60	>85		RTS	
	>86			
FE89: A9 00	>87	SETKBD	LDA #500	;Simula porto 0 em A2L para
FE8B: 85 3E	>88	IMPORT	STA A2L	especificar teclado como entrada
FE8D: A2 30	>89	IMPRT	LX MCSM,	
FE8F: A0 0F	>90		LDY KIH1	
FE91: 00 00	>91		BNE IOPRT	
FE93: A9 00	>92	SETVID	LDA #500	;Simula porto 0 em A2L para
FE95: 85 3E	>93	OUTPORT	STA A2L	especificar video como saida
FE97: A2 34	>94	OUTPRT	LX MCSM,	
FE99: A0 F0	>95		LDY #COUT1	
FE9B: A5 3E	>96	IOPRT	LDA A2L	;Seta RAM VETORES IN/OUT



```

FE9D: 29 0F 197      AND  W50F      ;Testa des
FE9E: F0 06 198      BEQ  IOPRT1
FE9F: 09 98 199      ORA  W590      ;M99=(A<5F
FEA0: A0 00 100      LDY  W500      ;Y=000
FEA1: F0 02 101      BEQ  IOPRT2
FEA2: A9 FD 102 IOPRT1 LDA  W5F0      ;End.º sign. das rotinas de I/O
FEA3: 94 00 103 IOPRT2 STY  LOCL,X    ;# Normalmente:
FEA4: 95 01 104      STA  LOCL,X    ;(CSML)=SFDF0 (IM1)
FEA5: 60 105      RTS          ;(CSML)=SFDF0 (OUT1)
           106
FEAE: EA 107      NOP
FEAF: EA 108      NOP
           109
FER0: 4C 00 C2 110 XBASIC JMP  BASIC      ;Va para o BASIC inicializado
FER1: 4C 03 C2 111 BASCONT JMP  BASIC2     ;Continua no BASIC
           112
FER4: 20 75 FE 113 50   JSR  ALPC      ;Carrega PC com conteúdo de A1
FER5: 20 3F FF 114      JSR  IORESTORE ;Restaura todos os registradores
FER6: 6C 3A 00 115      JMP  (PCL)     ;Executa no ender. indicado por PC
           116
FERF: 4C CC FA 117 REGZ  JMP  REGOSP     ;Iap.º cont. dos regs. A,X,Y,P,S
FEC2: 60 118      TRET  RTS
FEC3: EA 119      NOP
FEC4: 60 120 TIRET  RTS
           121
FEC5: C1 122 RTBL  ASC  "A"
FEC6: D8 123      ASC  "X"
FEC7: D9 124      ASC  "Y"
FEC8: D0 125      ASC  "P"
FEC9: D3 126      ASC  "S"
           127
FECA: 4C FB 03 128 USER  JMP  USRADR
           129
FEDD: A9 40 130 WRITE  LDA  W540      ;Emite um sinal piloto de 10 seg.
FEDF: 20 C9 FC 131      JSR  HEADR
FEE2: A0 27 132      LDY  W527
FEE4: A2 00 133 WR1    LDX  W500
FEE6: 41 3C 134      EOR  (ALL,X) ;Check Byte
FEE8: 48 135      PHA
FEE9: A1 3C 136      LDA  (ALL,X)
FEEB: 20 E0 FE 137      JSR  WRBYTE   ;Escreve um Byte de A1
FEEC: 20 8A FC 138      JSR  NEXTA1   ;Incrementa A1
FEE1: A0 10 139      LDY  W510
FEE3: 68 140      PLA
FEE4: 90 EE 141      BCC  WR1      ;Se A1/A2 pega novo Byte de A1
FEE6: A0 22 142      LDY  W522
FEE8: 20 E0 FE 143      JSR  WRBYTE   ;Escreve um check sum
FEEB: F0 40 144      BEQ  BELL     ;Termina com BELL
FEED: A2 10 145 WRBYTE LDX  W510      ;Escreve um Byte
FEEF: 0A 146 WRBYTE2  ASL
FEFD: 20 04 FC 147      JSR  WRBIT    ;Escreve 1 meio ciclo de 1 bit
FEF3: 00 FA 148      BNE  WRBYTE2
FEF5: 60 149      RTS
           156
           11
FEF4: 20 00 FE 12 CRNH  JSR  BL1      ;Interpreta o conteúdo do Buffer
FEF9: 68 150      PLA

```

FEFA:	68	34	PLA		
FEFB:	00	64	BMC	MONZ ;Retorna no inicio do modo monit.	
FEFD:	20	FA	FC	16 READM JSR R02BIT ;Busca sinal piloto de TAPEIN	
FEFE:	A9	16	17 LDA	RS16 ;Atrazo de 3,5 segundos	
FEFG:	20	C9	FC	18 READ2 JSR HEADR	
FEFH:	05	2E	19 STA	CHSUM ;Inicializa CHSUM=FFFF	
FEFI:	20	FA	FC	100 JSR R02BIT ;Busca sinal piloto de TAPEIN	
FEFJ:	A0	24	111 LDY	RS24 ;Busca sinal de SYNC	
FEFK:	20	F0	FC	112 JSR R0BIT	
FEFL:	00	F9	113 BCS	NR02 ;Loop ate encontrar	
FEFM:	20	F0	FC	114 JSR R0BIT ;Segundo SYNC (meio ciclo)	
FF00:	A0	38	115 LDY	RS38 ;Indica Y para teste de 0/1	
FF01:	20	EC	FC	116 R03 JSR R0BYTE ;Le 1 Byte	
FF02:	01	3C	117 STA	(A1,X) ;Coloca em A1	
FF03:	45	2E	118 ENR	CHSUM ;Atualiza CHSUM	
FF04:	05	2E	119 STA	CHSUM	
FF05:	20	BA	FC	120 JSR HEXTAL ;Incrementa A1	
FF06:	A0	35	121 LDY	RS35 ;Compara Y para teste de 0/1	
FF07:	90	F0	122 BCC	R03 ;Le outro Byte ate A1=42	
FF08:	20	EC	FC	123 JSR R0BYTE ;Le CHSUM	
FF09:	C5	2E	124 CMP	CHSUM ;Compara com CHSUM	
FF0A:	F0	00	125 BEQ	BELL ;Se igual gera um BELL e retorna	
FF0B:	A9	C5	126 PRERR LDA	"E" ;Imprime a mensagem "ERR"	
FF0C:	20	ED	FD	127 JSR	COUT
FF0D:	A9	02	128 LDA	"R"	
FF0E:	20	ED	FD	129 JSR	COUT
FF0F:	20	ED	FD	130 JSR	COUT
FF10:	A9	07	131 BELL LDA	MBell ;Carrega codigo NASCII de BELL	
FF11:	4C	ED	FD	132 JMP	COUT ;Produce o beep
FF12:	A0	F3	07	133 IORESTORE LDA	STATUS ;Armazena STATUS no stack
FF13:	48		134 PHA		
FF14:	A5	45	135 LDA	ACC ;Prestara ACC no acumulador	
FF15:	4C	3A	FC	136 JMP	RSTR1
FF16:	EA		137 NOP		
FF17:	EA		138 NOP		
FF18:	4C	90	FB	139 IOSAVE JMP	SAVE1 ;Salva acumulador e regist. X
FF19:	BC	F2	07	140 SAV3 STY	YREG ;Salva Y em RES
FF20:	08		141 PHP		
FF21:	48		142 PLA		
FF22:	80	F3	07	143 STA	STATUS ;Salva regist.de status em STATUS
FF23:	0A		144 TSX		
FF24:	0E	F4	07	145 STX	SPNT ;Salva stack pointer em SPNT
FF25:	08		146 CLD		
FF26:	60		147 RTS		
			148		
FF27:	FF	FF	149	HEX	FFFF
			150		
FF28:	08		151 MON CLD		
FF29:	20	3A	FF	152 JSR	BELL ;Limpa modo decimal
FF2A:	A9	CD	153 MONZ LDA	"P" ;Gera um BELL	
FF2B:	05	33	154 STA	PROMPT ;Pronto de inicio do modo MONITOR	
FF2C:	20	47	FD	155 JSR	GETLNZ ;Obtem linha de entrada de dados
FF2D:	4C	A7	FF	156 JMP	ZN00
			157		
FF2E:	4C		158	HEX	4C
FF2F:	20	E9	EE	159 ZN00 JSR	ZN00 ;Zera modo
FF30:	20	AA	FF	160 NEXTITH JSR	GETNUM ;Busca numero hex. no Buffer IN

FF72:	04 34	161		STY	YSW	;Salva posicao de IN em YSW
FF74:	40 17	162		LDY	MS17	;Verif. se e uma das funcoes
FF76:	88	163	CHRSRCH	DEY		;do monitor em CRTBL
FF77:	30 E4	164		BNL	NOM	
FF79:	09 CA	165		CMF	CNTRL, Y	
FF7C:	00 F8	166		BNE	CHRSRCH	
FF7E:	20 88	167		JSR	TOSUM1	;Encontra funcao, entao armazena
FF81:	44 34	168		LDY	YSW	o ender. da func. no stack
FF83:	4C 4F	169		JMP	NXTITH	;Reposiciona IN, busca novo dado
FF86:	20 99	170	CLREOL1	JSR	CLREOL	;Completa uma linha com "espaco"
FF89:	20 50	171		JSR	ZXY	;Interpreta buffer IN
FF8C:	60	172		RTS		
		173				
		174				*Coloca um endereco de ate 4 digitos em A2
		175				
FF8D:	A2 03	176	DIG	LDX	MS03	;des do acumulador no des de AZL
FF8F:	0A	177		ASL		
FF90:	0A	178		ASL		
FF91:	0A	179		ASL		
FF92:	0A	180		ASL		
FF93:	0A	181	NXTBIT	ASL		;Passa o bit atraves do carry
FF94:	26 3E	182		ROL	A2L	;des de A2L para DMS de A2L
FF96:	26 3F	183		ROL	A2H	;DMS de A2L para des de A2H
FF9B:	CA	184		DEX		;des de A2H para DMS de A2H
FF99:	10 F8	185		SPL	NXTBIT	;Desloca ate completar 4 bits
FF9B:	A5 31	186	NXTBAS	LDX	MS0E	;Se nao modo zero, va para NRTBAS2
FF9D:	00 06	187		BNE	NXTBAS2	
FF9F:	85 3F	188		LDX	A2H, X	;Se modo 0, carrega A1 e A3 com A2
FFA1:	95 30	189		STA	A1H, X	
FFA3:	95 41	190		STA	A3H, X	
FFA5:	EB	191	NXTBAS2	JMX		
FFA6:	F0 F3	192		BEQ	NXTBAS	
FFA8:	00 06	193		BNE	NXTCHR	;Proximo caracter do Buffer IN
FFAA:	A2 00	194	GETNUM	LDX	MS00	;Zera A2
FFAC:	86 3E	195		STX	A2L	
FFAE:	86 3F	196		STX	A2H	
FFB0:	89 00 02	197	NXTCHR	LDX	IN, Y	;Busca caracter do Buffer IN
FFB3:	C8	198		JNY		;Proxima posicao do Buffer IN
FFB4:	4F 80	199		EOB	MS80	;Transforma ASCII de 80-89 em 0-9
FFB6:	C9 0A	1100		CMF	MS0A	;Entre 0-9 processa o numero
FFB8:	95 03	1101		BCC	008	
FFBA:	6F 88	1102		ADC	MS88	;ASCII de C1-C6 transf. em FA-FF
FFBC:	C9 FA	1103		CMF	MSFA	;Entre FA e FF processa o numero
FFBD:	80 CD	1104		BCS	010	
FFCD:	60	1105		RTS		;Ret.se nao e um numero hex.
FFC1:	EE FB 00	1106	NXC00	INC	C00L	;Incrementa o par (C00L-H)
FFC4:	00 03	1107		BNE	C00RTS	
FFC6:	EE FC 00	1108		INC	C00H	
FFC9:	60	1109	C00RTS	RTS		
		1110				
FFCA:	8C	1111	CNTRL	HEX	8C	;F(CTRL-C)
FFCB:	82	1112		HEX	82	;F(CTRL-Y)
FFCC:	8E	1113		HEX	8E	;F(CTRL-E)
FFCD:	82	1114		HEX	82	;F(CTRL-Y)
FFCE:	EF	1115		HEX	EF	;F(0)
FFCF:	C4	1116		HEX	C4	;F(CTRL-K)
FFD0:	82	1117		HEX	82	;F(CTRL-Y)

FFD1: A9	1118	HEX A9	:F(CTRL-#)	
FFD2: B8	1119	HEX B8	:F(CTRL-0)	
FFD3: A6	1120	HEX A6	:F("=")	
FFD4: A4	1121	HEX A4	:F("=")	
FFD5: 06	1122	HEX 06	:F("M")	
FFD6: 95	1123	HEX 95	:F("=")	
FFD7: 07	1124	HEX 07	:F("M")	
FFD8: 02	1125	HEX 02	:F("T")	
FFD9: 05	1126	HEX 05	:F("L")	
FFDA: F0	1127	HEX F0	:F("M")	
FFDB: 00	1128	HEX 00	:F("0")	
FFDC: EB	1129	HEX EB	:F("R")	
FFDD: 93	1130	HEX 93	:F("=")	
FFDE: A7	1131	HEX A7	:F("=")	
FFDF: C6	1132	HEX C6	:F("CR")	
FFE0: 99	1133	HEX 99	:F("BLANK")	
FFE1: B2	1134	SUBTBL	HEX B2	:WASCOM-1
FFE2: C9	1135	HEX C9	:MUSR-1	
FFE3: BE	1136	HEX BE	:MREGZ-1	
FFE4: C1	1137	HEX C1	:MRET-1	
FFE5: 35	1138	HEX 35	:MVF-1	
FFE6: BC	1139	HEX BC	:MWHPR-1	
FFE7: C3	1140	HEX C3	:MTR-1	
FFE8: 96	1141	HEX 96	:MOUTPRT-1	
FFE9: AF	1142	HEX AF	:MOSASIC-1	
FFEA: 17	1143	HEX 17	:MSETH00-1	
FFEB: 17	1144	HEX 17	:MSETH00-1	
FFEC: 2B	1145	HEX 2B	:MNOVE-1	
FFED: 1F	1146	HEX 1F	:MLT-1	
FFEE: 83	1147	HEX 83	:MSETH00A-1	
FFEF: 7F	1148	HEX 7F	:MSETLW-1	
FFF0: 50	1149	HEX 50	:MHLIST-1	
FFF1: CC	1150	HEX CC	:MWRITE-1	
FFF2: B5	1151	HEX B5	:MDO-1	
FFF3: FC	1152	HEX FC	:MREADM-1	
FFF4: 17	1153	HEX 17	:MSETH00E-1	
FFF5: 17	1154	HEX 17	:MSETH00E-1	
FFF6: F5	1155	HEX F5	:MCRMON-1	
FFF7: 03	1156	HEX 03	:MBRANK-1	
FFF8: 5D FF	1157	DA MM	:VECTOR (MONITOR ENTER)	
FFF9: FB 03	1158	DA MM	:VECTOR (INTERRUPTAO NAO MASCARADA)	
FFFC: 62 FA	1159	DA RESET	:VECTOR (RESET)	
FFFE: 4D FA	1160	DA DR9	:VECTOR (INTERRUPT REQUEST)	

--End assembly, 16128 bytes, Errors: 0

Symbol table - alphabetical order:

A1H	=53D	A1L	=53C	A1PC	=5FE75	A1PLP	=5FE78
ALPCRTS	=5FE7F	A2H	=53F	A2L	=53E	A3H	=541
A3L	=540	A46	=546	A4H	=543	A4L	=542
A5L	=544	ABS	=5DECF	ACC	=545	ACOMPA1	=5EE9E
ACOMPA2	=5EEC7	ACOMPA3	=5EEB9	ADD	=5DAD3	ADL	=5D4ED
AD2	=5D4EB	AD3	=5D6D4	AD4	=5D6D8	AD5	=5D614

ADD	=SF001	ADDACC	=SF0EF	ADDIMP	=SF086	ADDON	=SC080
ADXC2	=S002A	ADXC3	=S0035	ADXC4	=S0040	ADEXP	=S0028
ADJEXP	=SF0B8	ADKDB	=SEBF5	ADMM	=S06AF	ADX	=S07BA
ADFER	=S03F5	ADQ	=S0270	ADUB?	=SEB87	ADRD	=S07FD
ADR	=S45	ADSGN	=SAA	ADTTH	=S00E0	ADRYT	=S0438
ADYFOUN0	=S0480	ADYLOOF	=S0487	ADYMAN?	=S0493	ADYPAT	=S94
ADYSTR	=S0626	ADYTAB	=S68	ADYWA2	=S07E4	ADYWA3	=S07E8
ADYWR	=S070C	ADYV00	=S07F3	ASC	=S09FF	ASCHEX	=S0207
AGS	=SC192	ASSEND	=SC189	ASSEMB1	=SC180	ASSEMB2	=SC1C8
AT7	=SE56C	ATROUC	=SF024	ATLCOF	=SEE7E	ATH	=SE288
ATW1	=SE30D	ATN2	=SE3CE	ATN3	=SE3E1	ATMSER	=SE3E8
AXB	=SF8F5	ATYMT	=S0426	AtofAC	=S06AA	AtofACS	=S08AC
BADMM	=S0314	BAS2H	=S28	BAS2L	=S2A	BASCONT	=SFEB3
BASH	=S29	BASIC	=SC200	BASIC2	=SC203	BASL	=S28
BASLOC	=S47	BASLOC1	=S80	BELL	=SFF3A	BELLO	=SFB09
? BELL1	=SFB00	BELL2	=SFBE4	BORAF	=SF6P1	BILLMO	=S02DE
? BILAEIM	=SE029	? BIGND	=SE775	BIGND1	=SE770	BL1	=SFED0
? BLANK	=SFE04	BLENGTH	=S91	BLTU	=SC682	BLTU2	=SC689
BPL1	=S0824	BPL2	=S0916	BPL3	=S0949	BPL4	=S0999
BREAK	=SFAAC	BREAKIN	=SC4AC	BRVAL	=S03FD	BSB	=S0263
BSROUT	=SF054	BTCONT	=SC05E	BTCHT1	=SC05F	BUF	=S0460
BUSBYT	=SC110	BUSCOR	=SEF28	BUSESP	=SC23A	BUSTOKEN	=SFF38
BYPASS	=S032C	CALL	=SE536	CARACT	=SF67F	CARBUF	=SEF7E
CART	=SC209	CART0	=S9080	CART3	=S90C0	CART6	=S90D6
CART9	=S90D9	CAT	=S0881	CCT8	=SF70C	CEMHard	=SE024
CH	=S24	CHAB1	=SF984	CHAB2	=SF98A	CHABAC	=S00
CHGRF	=SECC3A	CHKCLS	=S01E0	CHKCOM	=S01E6	CHKDIN	=S04CA
CHGSP	=S0F80	CHKIFLT	=SE89E	CHKFST	=SE800	CHKHEX	=SC6C5
CHKMM	=S0092	CHKOPN	=S01E3	CHKSTR	=S0094	CHKSUM	=S2E
CHKT	=SEBEE	CHKTYP	=S030C	CHKVAL	=S0095	CHRGET	=S81
CHKGOT	=S87	CHRSRCH	=SFF76	CHRSTR	=S0960	CHRTEL	=SFFCA
CHRTXT	=SFDC0	CLEAR	=SC981	CLEARC	=SC983	CLEGL2	=SFF98
CLEP31	=SFC46	CLEREP	=SFC90	CLBRIN	=SFD73	CLREGL	=SFC99
CLREGL1	=SFF86	CLREOP	=SFC42	? CLKEY	=SFD74	CLASC2	=SFB38
CLKSC3	=SFB3C	? CLRSOR	=SFB32	CLRTOP	=SFB36	CLVAL	=SFD98
CMTABL	=SC206	CMPBL1	=SE07C	CMPDME	=S02E9	CMPHM	=SE087
CNPL06F	=S0282	CNPLSP	=SEF18	COCCR	=SF117	COOH	=SFC
COOL	=SFB	COORIS	=SFFC9	COLCNUMT	=S8A	COLDET	=SE442
COLONT?	=SC840	COLOR	=S30	COLOR*	=SESAC	COMBYTE	=S0666
COMILEG	=SC5E5	COMPARE	=S010C	COMPNAME	=SEE63	COMDEF	=SC5A4
CON	=SC8C6	CONENT	=S0A15	CONT	=SC888	CONTRASS	=SC195
CONTR	=SFAF3	CONTECH	=SC108	CONTRC	=SFLCA	CONTTOK	=SCA50
CONPK	=S0CFC	CONWTK	=SFF75C	COPSTR	=SC0BF	COPY	=SC2DC
COS	=SE304	COSTBL	=SE938	COUNTED	=SE101	COUNTH	=S10
COUT	=SFD0D	COUT1	=SFDFO	COUT2	=SFLA6	COUT3	=S0206
CPLTBIT	=SE0CC	CPLTMOE	=SEE3A	CPRMAG	=S16	CPROP	=S00C0
CPTYP	=S89	CR	=SFC62	CRD0	=SCE20	? CRON	=SFEF6
CRONT	=SFD8E	? CRDUT1	=SFD88	CSHF12	=SE7FE	? CSHFT	=SE7FC
CSST1	=S0E30	CSST2	=SE051	CSST3	=SE05A	CSST4	=SE073
CSST5	=SE07C	CSST6	=SE08C	CSST7	=SE095	CSAL	=S16
CTNCT?	=SCB91	CURE56	=SFD12	CURLIN	=S75	CURLSV	=S6
CV	=S25	CV2	=S009C	CVX8	=SFBF3	CY	=S0779
DATA	=SC0BA	DATAFLG	=S13	DATAM	=SCCC8	DATANUT	=SFD84
DATIM	=S0F91	DATLIM	=S78	DATPTR	=S70	DECRUF	=SFD18
DECR0T0M	=S741	DECTBL	=SE183	DECCY	=SFL00	DEF	=S0620
DEL	=SE692	DELH	=SE3	DESC?	=SC081	DESE56	=SFC18
DESE561	=SFC20	DESIMU	=SEFED	DEST	=S60	DFT01M	=S0511

DI0	+5FFB0	DIA	+502F9	DIFLGL	+510	DINLUP	+50560
DIMK	+50509	DIMK2	+5058A	DIR?	+50EEC	DIRECT	+50CAF
DISK10	+5F590	DISPAG1	+5C054	DISPAG2	+5C055	DIV	+50078
DIVID	+500A7	DIW2	+500F8	DIVV2R0	+5C505	DWTH	+50169
DOCP	+50200	DOGRMS0	+5C700	DMATH	+5D160	DMATH	+50160
DOHE	+5C927	DORREHT	+5C855	DOGPC	+5C254	DOWN	+5E885
DOWN1	+5E8A0	DOWN2	+5E8A0	DOWK	+5E8A4	DP1	+5EAC1
DP2	+5EAC7	DP3	+5EAE6	DPDIO	+50FA1	DPFLG	+598
DPL	+5E089	DPLEFT	+5DFC3	DPLCC	+5E080	DPRIGHT	+5DFCC
DPYK2000	+5F25F	DRAM	+5EAE9	DRAM1	+5E986	DRW2	+5E9A7
DRAM3	+5E981	DRAM4	+5E98E	DRAWS	+5E9C9	DRAMO	+5E982
DRWPNT	+5EAA0	DSOLEM	+58F	DSCPTR	+58C	DSCTHP	+590
DSK	+5C100	DURACAO	+5C252	DVL	+50852	DV2	+5085C
DVAR	+50830	DVARS	+50833	DVAR75	+508AC	DVH	+5D1
DRL	+5D0	DY	+502	EDTBAS	+5F604	EH	+5D5
EL	+5D4	END	+5C895	END2	+5C896	END3	+5C8A0
END4	+5C8AF	ENDCHR	+50E	ENCFOR	+507D	ENOMASK	+535
ENDTMP	+5007A	ENDRWG	+5C908	ENDTBS	+5F707	EPGV10	+5FC73
EQIC	+5E93A	EQ3	+5E840	EQ4	+5E888	EQL	+5D1F0
EQG0P	+5D1C0	ERFLG7	+5C888	ERL1H	+5CEA4	ERLUP	+5C70E
ERR	+5F8A0	ERRD1H	+50A20	ERRFLG	+500	ERR1H	+5C63F
ERR1LH	+5D4	ERRMUN	+50E	ERROR	+5C701	ERRPOS	+50C
ERRSTA	+50F	ESPERA	+5F1E3	ESPRIAK	+5A7FF	ESTOR	+5C58F
ETOKEN	+5F753	EVAL	+5DF78	EVALVAR	+5E5A4	EVO	+5DF08
EVDOME	+50F03	EVAR	+5EA71	EXCSTRG	+5C5FF	EXECUTE	+50837
EXIG	+50007	EXIT	+5D185	EXP	+5E223	EXPON	+59A
EXPSER	+5E1FA	EXPS0M	+59C	EXT10	+5F4FF	EXT101	+5E733
EXTRAFAC	+5AC	EXTRASV	+592	EXMEM	+5C596	FAC	+590
FACSGM	+5A2	FACSTRNG	+5E050	FAD0	+50A00	FADDH	+5D40A
FA007	+50A00	FALSE	+5D085	FC1	+5DF03	FC2	+5DF09
FCOMP	+5DECC	FCOMP2	+5DECE	FD1	+5D09A	FD2	+5D080
FD3	+5D080	FD4	+5D0C0	FDS	+5D0CE	FD6	+5D0E8
FD7	+5D0EF	FD8	+5D0A2	FDIV	+5D080	FDIVT	+5D083
FEL	+5D72F	FE2	+5D739	FEMBAS	+5F7D4	FEND	+5D711
FEVLOOP	+5D0A6	FEM1	+5F10A	FEMNAM	+5E728	FEMCAR	+5C2C1
FINDC	+5F702	FINESP	+5EFS7	FINFEM0	+5C222	FML1H	+5D482
FINDOTA	+5C263	FINPA	+5F6FC	FINSANE	+5E000	FINTENP	+5C284
FINTRANS	+5F680	FIN	+5DF64	FIN2	+5DF68	FIN3	+5DF77
FINDOTA	+5CFC8	FINDOOL	+5D000	FIRST	+5F0	FIXSEV	+5F4F4
FL1	+5C935	FL2	+5C94C	FLGRM	+5D440	FL01	+5D8E5
FL02	+5D8EA	FL0AT	+5D8A0	FL0P	+5D84C	FH1	+5DCCA
FR2	+5DCCF	FR3	+5DCC2	FH4	+5DCEE	FHT1	+5F962
FR12	+5F9A6	FH0	+5DCA1	FHULT	+5DCC9	FHULTT	+5DCC9C
FR?	+5D1CC	FR?	+5D458	FINDATA	+5D4C9	FINDAM	+59A
FRARY	+5D483	FNDLEN	+5D6A5	FNDFOR	+5C459	FNDL1H	+5C931
FRANR	+5D7A2	FOR	+5CA80	FOR2	+5CA91	FORNAT	+52E
FOR1MST	+5C120	FORNAMCX	+5CA0C	FORPNT	+585	FOUT	+5E0AE
FRGEN	+5A4	FRMT	+5E1B1	FR1	+5D88E	FR2	+5D89A
FR3	+5D8A7	FRE	+5D5F8	FRE2	+5D5FF	FREFAC	+5D91A
FRE4	+5C25A	FRESPC	+571	FRESTR	+5D917	FRETNP	+5D91E
FRETHS	+5D94F	FRETOP	+5A7	FREVL	+5D0A3	FREVL2	+5D08D
FRMUN	+5D08F	FROUND	+5D8A9	FS08	+5D4C1	FSORT	+5D4CA
FULL	+5D79E	FUNCT	+5DAAE	FUNDEF	+5CA2C	GARBAG	+5D79E
GBFL6	+513	GBASCALC	+5F847	GBASH	+527	GBASL	+526
GBCALC	+5F854	GBH010	+5F778	GC	+5CA48	GO	+5D413
GBRIFS	+5C84E	GBXCHR	+5F200	GERDRF	+5F400	GERR	+5D025
GET	+5C0C5	GETAM	+5D46C	BETARY	+5D407	BETARY2	+5D409

DETARYPT	=SE859	DETRYT	=S0412	GETCHR	=SC443	GETEXP	=SE002
GETFMT	=SF8AC	GETIM	=SC8AE	GETLINK	=SC955	GETLN	=SF06A
GETLMS	=SF06F	GETLNM	=SFD67	GETMLM	=SFFAA	GETLO	=S04C3
GETSPA	=S074C	GETSPC	=S07AE	GETSTR	=SD9F6	GETVAL	=S018D
GGERR	=SE6A7	GT9	=S0C42	GT9HPT	=S0A0C	GG	=SC885
GNE	=S058A	GO	=SFE8A	GO1	=SCC7A	GO2	=SCC7E
GOCHO	=S0B42	GOCHO2	=S0B44	GOCHO3	=S0C52	GOCHO	=S0B4B
GOERR	=S055A	GOEX	=S016D	GOEXE	=S0FA4	GOGO	=S0B2E
GO19	=S0A0C	GOELNE	=S0C5A	GOEWST	=S0D7A	GOOUT	=S0267
GOSTART	=S0B8B	GOSTROUT	=S03	GOEUB	=S0C46	GOTFOR	=S0037
GOTO	=S0C63	GOTO7	=S0D19	GOTSPA	=S0783	GOVARI	=S00
GR	=SE6F9	GR1	=SE6F1	GRAFFLG	=S0452	GRFPAS	=S087C
GRPLOT	=SF001	GSE	=S0583	GSTNER	=S0C6A	GTRTTC	=S0A0F
GTFORPNT	=S0A54	GILT	=S032B	GTMLR	=S0A6D	H2	=S0C
HABLIN	=SF054	HALF	=S017E	HANDLER	=S064A	HCLR	=SE771
HCOLR	=SE6A6	HCOLR1	=S1C	HCOLR2	=S0E	HCOLT	=SE90C
HEADR	=SFC29	HFM0	=SE94C	HFM01	=SE971	HFM02	=SE97F
HFMS	=SE43A	HFMS1	=SE44E	HGLIN	=SE88B	HGR	=SE738
HGR1	=SE74F	HGR2	=SE74B	HGRFLG1	=S985A	HGRFLG2	=S985E
HISFFF	=SE4CF	HIGHOS	=S94	HIGRTR	=S96	HJEN	=SE5E7
HLIM	=SE586	HLIN2	=SE801	HLIN3	=SE8E9	HLINA	=SE91F
HLINE	=S9819	HLINE1	=S981C	HLINML	=SE8B1	HNASK	=S3D
HNDX	=SE5	HONE	=SFC5B	HP2	=SE48B	HP3	=SE48F
HPAG	=SE6	HPL0T	=SE47E	HPL0T0	=SE7D7	HPOSH	=S079D
HPGSH1	=SE7C1	HPGSH2	=SE7C2	HTAB	=SE867	HTAB1	=SE86C
HTAB2	=SE87A	HUM	=S0C2D	IDCTP	=S01A9	IDFC00	=S0FFD
IDFECTL	=SFD81	IEVEN	=S989E	IF	=SC0EE	IFBDM	=SE0AE
IFTRUE	=S0C06	IMEDIA7	=S01FA	IMPBAS	=S077D	IMPCH	=S90E9
IMPNAM	=SE031	IMPSSL	=S0A22	INPR	=S020D	IMPSP	=SE045
IMPWAT	=SE07D	IM	=S020D	IN0	=SFD0C	IN1	=SFD0F
IN2	=SFB02	IN3	=SFD14	IN4	=SFD18	IN5	=SFC33
INCHR	=S0B6B	INCOMP	=S0CF3	INOS	=SFD0A	INORR	=S018D
INORR1	=S0186	INDEX	=S0E	INDILEG	=S058B	INOPBAS	=S07D9
INDX	=S99	IN6	=S0A0D	INI000	=SE4E7	INIBAS	=SE4E9
INIBAS3	=SE51A	INICART	=S9FF1	INICPOL	=S0C7D	INISUM	=SE0C3
INIT1	=SFB3B	INLIN	=S0B1B	INLIN2	=S0B1D	INLIND	=S0B2A
INLINA	=S0B2A	INLINS	=S0B4B	INMSD	=S0A47	INPDONE	=S0CFE
INPERR	=S0E4B	INPFIH	=S0CF1	IMPORT	=SFE8B	IMPRT	=S03D3
INPTR	=S7F	INPUT	=S0ED7	INPUTERR	=S0E96	INPUTFLG	=S15
INS1	=S097A	INS2	=S099D	INS3	=S0981	INSA	=S0982
INS5	=S098C	INS01	=SFB05	INS02	=SFB0F	INS03	=SFB91
INSRTD16	=S0FD8	INSRTLIN	=S07D9	INSTANT	=S0F53	INSTOSP	=SFB8D
INSTRN	=S09D3	INT	=S0F3D	INTC	=SFD05	INTFLG	=S12
INTPART	=SE0A3	INTWAR?	=S034D	INVALID	=S0181	INLCUR	=SFB04
INVERSE	=SE50D	INPLG	=S32	IOCLOR	=S0C5D	IOPER	=S0C51
IOPR1	=SFE9B	IOPR11	=SFEA7	IOPR12	=SFEA9	IORESTONE	=SFF3F
IOSAME	=SFF4A	IOERR	=S0483	IRN	=SFA4D	IRNLCL	=S03FE
ISCBAS	=S0769	ISCNIC	=S0B7B	ISCTB	=S0C7D	ISCTB?	=S0E6D
ISCTP	=S0F96	ISGRF?	=S0C2C	ISLETC	=S0397	ISRET?	=S0E4B
ISTOK?	=S0B67	ISVD	=S0C15	JARY	=S036D	JCAR16	=S0133
JCAR19	=S01F5	JCOOCR	=S982E	J1D	=S0099	JDECRUF	=S07E7
JER	=S0485	JERR	=S074C	JERROR	=S004D	JF	=S02F2
JGCRD	=S0C0E	JIMS	=S07D7	JLNC	=S0C12	JM1D	=S0E92
JJCR	=S0127	JNH	=S0S77	JPHANS	=S9D	JOV	=S0C5D
JPHESB	=S0D64	JPHG12	=S078A	JPR1D	=S01F9	JPRC	=S077A
JPROCESS	=S0C21	JST	=S0C66	JSTN	=S068F	JSTNER	=S0E32
KROUT	=S0C0D	KRTPIN	=S0D1D	KSM	=S3B	KWORD	=SE0E2

KWORD1	=SEEEE	LASTIN	=S2F	LASTOP	=S87	LASTPT	=S53
LOBET	=SFR93	LECHK	=SEE20	LECL	=SEDOC	LEFT	=SE7E7
LEFT1	=SE7F9	LEFT2	=SE7F6	LEFTSTR	=S0974	LEN	=S09F0
LENGTH	=S2F	LET	=SCD48	LET2	=SCD68	LETREAL	=SCD9C
LETSTR	=SCD9F	LF	=SFC66	LFRIT	=SE7E5	LG2	=S0C65
LIM	=S0B8F	LINC004	=SE550	LINC000	=SEFF1	LIMBET	=S0C31
LIMSET	=SC7E1	LIMMUM	=S50	LIMPT	=SE03E	LIST	=S0Y8C
LIST2	=SFE43	LISTED	=SCA38	LSTLOOP	=SCA15	LMEM	=S0C
LOAD	=SCBEE	LOADADR	=SFD0D	LOADT	=SE04E	LOCO	=S00
LOC1	=S01	LOC2	=S02	LOC3	=S03	LOC4	=S04
LOCK	=S06	LOG	=S0C58	LOGGER	=S0C32	LOG4015	=S0C56
LOGe	=SE1F5	LOMEM	=SE607	LOOP20	=SEF44	LOWR	=S9B
LR1	=SE7EE	LRUD1	=SEB33	LRUD2	=SEB34	LRUD3	=SEB44
LRUD4	=SEB48	LRUDX1	=SEB1C	LRUDX2	=SEB1D	LSTLIM	=SCA0E
LST0*	=SCA0C	LT	=SFE20	LTD	=SFE22	MA	=SC000
MAF	=SDEB0	MAF2	=SDEB2	MAINWP	=S2F10	MAILST	=S0F23
MAIARY	=S0408	MAIDIGIT	=SE10B	MAINT	=S041C	MAISTR	=SE037
MARXEND	=SE174	MATHBL	=SC292	MB	=SE071	MEMORR	=SC6FF
MEMSIZ	=S73	MENS	=SFAF3	MENSERR	=SC54F	MENTAB	=SFD03
MFA	=SDEAF	MFA2	=SDE73	MII	=S0433	MIZ	=S0435
MIOSTR	=S094B	MIN	=S01F6	MIMPRT	=SFE80	MIMUS	=SC3A7
MISHTCH	=S009C	MINT	=S0422	MLIST	=SFE5E	MLOWD	=SEB92
MIXL	=SFFC0	MICHR	=SFA00	MINDX1	=SFBCL	MINDX2	=SFB0C5
MINDX3	=SFBCC	MIDCHK	=SFDAD	MIDC	=S31	MIR	=SE742
MIR	=SFF50	MIRZ	=SFF61	MIRMAN	=S0321	MISTOFF	=SC032
MOTOR	=SEB70	MOTRA	=SEB80	MOVIF	=SDE38	MOVZF	=SDE38
MOVAF	=SDE7D	MOVDM	=SEAC6	MOVE	=SFE2C	MOVESTR	=S0900
MOVEX	=SEBFD	MOVEX2	=SEY02	MOVFA	=SDE6D	MOVFN	=S0E13
MOVINS	=S0BEE	MOVIT	=SE4C8	MOVNF	=SDE45	MOVNL	=SDE30
MOVSTR	=S0BFC	MP	=SEBFB	MRO2	=SFT0A	MSAVE	=SEBC4
MSKTB	=SE92D	MSLUP	=SE008	MUI	=S05D0	MU2	=S050A
MU3	=S05F3	MUL10	=S0053	MULT	=S0C07	MVBACK	=SE133
MVBT	=SC6B2	MVDD	=S0011	MVDWN	=SC796	MVPRG	=S07C0
MVSD	=S09D4	MVSD	=S0900	MVZP	=SE46C	MWario	=SE051
MAKOK	=S0317	NB	=S094A	MCR	=S0A29	MD1	=S0269
MD2	=S0116	NDP	=S0FE2	MDX	=SE6E9	MEDEX?	=SE140
MEG2	=S0BBE	NEGATE	=S0C25	MEGFAC	=S0BB9	MEGRUN	=S0418
MEGDP	=SE1EA	NEW	=SC76D	MEMDESC	=SCDC6	MEMLH?	=SC744
MEWMOX	=SE7F8	NEWETT	=SCAEC	MEMWR	=S00B6	NEXT	=S0021
NEXTAL	=SFCBA	NEXTCHR	=SCE54	NEXTCOL	=SFB5F	NHWC	=SFD032
NTO	=SE42B	NI1	=SE488	NI2	=SE402	NI3	=SE408
NI4	=SEY0F	NIS	=SE435	NIMEDIAT	=SC20D	NIH	=S0350
NINT	=S04FE	NIXTA	=S0SAC	NDVSEP	=SEFEF	NI1	=SC78E
NI1	=S03FB	NOASP	=SEEF0	NOFTX	=SFA42	NI1	=SCB61
NOLAN	=SC2AF	NORMAL	=SE50A	NOISUCH	=SC99C	NOIT?	=S018B
NOIFMO	=S03A1	NOTFOUMO	=S0402	NOTNATH	=S015D	NOTOK	=S028F
NOTRACE	=SE5D6	NOTZ	=S01C7	NOTZE	=SE069	NOTinstr	=S0262
NOVID	=SEFF1	NOVID1	=SEFAD	NOVMT	=SFB95	NRH	=SE50F
NRTS4	=SFC28	NTCTL	=SFB14	NTCTP	=SFLC2	HTD	=S0CF9
NUL	=S0B01	NTRET	=SECS8	NTXT	=SE76A	NM10	=S0D6A
NUMBER	=S0191	NURCHP	=S0288	MUNDIN	=S0F	NM1N	=S0788
NV1	=S0375	NV2	=S0307	NVW	=S03CC	NX	=S038E
NVARY	=S0442	NXBASLOC	=SE4FA	NXBSW	=SECC7	NXBYT	=S0C84
NXCHAN	=SCE6F	NXCHANM	=SEEFF	NXDIR	=S0BB2	NXDLN	=S0C33
NXDOO	=SFFC1	NXCOL	=SFO5E	NXD18	=SC0A5	NXD18IT	=S0F7E
NXDIN	=S02F6	NXDIN	=S0298	NXDR	=SC67A	NXDIN	=S0701
NXIMP	=S2F16	NXKEY	=SFA64	NXLIN	=S0748	NXLINK	=S07E9



KOLNMH	=SEDB0	KRLST	=SCWY1	KRM	=SD041	KRRUM	=SC025
KOPFR	=SD0FE	KRPA0	=SC680	KRPF	=SD048	KRPF01	=SF8F0
KOPTR	=SD080	KRS	=SCFE1	KRSP	=SD0F6	KRSPC	=SC051
KRTAA	=SFCB4	KRTBAS	=SFF99	KRTBAS2	=SFFA5	KRTB1T	=SFF93
KRTB1T	=SFAME	KRTCHR	=SFFB0	KRTD1M	=SD446	KXTERM	=SE280
KRT11M	=SFF6F	KRTPOS	=SFB20	KRTPOS1	=SFB25	KXVAR	=K0373
KXY	=SF030	KXZER00	=SELC8C	KXZER01	=SECC0C	KXseF0N	=K034F
KY	=SCBB9	KZ	=SD723	KXDATA	=SC573	KXTEND	=K1L30
K00SER	=SE276	KXKSUM	=SEED07	KXDL1M	=S77	KXTEXT	=S79
K0CNT	=SCD10	KXEDLY	=SFC02	KXERR	=SE62C	KXSTEP	=SC480
K0G0T0	=SCD11	K0R	=SE23E	K0R	=S0277	K0RMASK	=SF3
K0UD0	=SECB1	K0UFRMG	=SD045	K0UTPORT	=SFE95	K0UTPR1	=SE97
K0UQUES	=SCE7F	K0UTROCH	=SFA01	K0UTSP	=SCE7C	K0VERFLOW	=K080F
K0CHK	=SD10A	K0UTTY?	=SEDF0F	K0PSE	=SCB81	K0CADJ	=SFF93
K0CADJ2	=SFF94	K0CADJ3	=SFF95A	K0CADJ4	=SFF95C	K0CR	=S08
K0CL	=S0A	K0CL	=S0271	K0CL0	=S0D04	K0CNR	=S0CF74
K0EXE	=S047E	K0ENCHR	=S0F73	K0GA	=S0400	K0CCC	=S0F74
K0VMH	=SEDCA	K0R1	=S0468	K0R2	=S046A	K0VR	=S087A
K0AL	=S47	K0NADR0	=SEED0	K0iiv2	=SE380	K0ive2e2	=SE385
K0LAT	=SE579	K0L0D	=SEFCA	K0L0T1	=SEFC8	K0L0T2	=SFB00
K0L0T0N0	=SE540	K0L0M	=SFB03	K0L0?	=SC918	K0LUS	=SC340
K0LUD0FS	=S0ED	K0LUBAS	=SFA8E	K0MTR	=S0F	K0MSTK	=SE179
K0CC	=SFB7F	K0LNT	=S0A57	K0KE	=S0A95	K0P	=SCC90
K0S	=S0619	K0SCHR	=SFA0E	K0SCDR	=SFB82	K0SCUR1	=SFF00
K0SE58	=SFB31	K0S0P	=S0C00	K0PRE0	=SFAA5	K0R1	=S0F92
K0R1	=SFF10	K0R1R2	=SFF14	K0R1R3	=SFF26	K0R1R4	=SFF2A
K0R5	=SFF30	K0R5S8	=SFC20	K0R1R2	=SFF94A	K0R1R3	=SFF94C
K0RMLK	=SFF40	K0R1TTE	=SFD0A	K0R1R4	=SFF08A	K0R1R01	=SFF08B
K0REFWC	=S011E	K0R1TEST1	=S00F5	K0R1R1R	=SFF79B	K0R1R1R	=SFF20
K0RESP	=SEF49	K0R1F	=S0C00	K0R1F2	=S0C08B	K0R1	=S0000
K0R1G12	=S0R12	K0R1S	=S0R15	K0R1G18	=S0R18	K0R13	=S0003
K0R6	=S0R06	K0R1P	=S0R09	K0R1C	=S0R0C	K0R1EM0	=S4F
K0R1F	=S0R0F	K0R1EX	=SFA03	K0R1EX2	=SFA05	K0R1NT	=S0CFA
K0R1M72	=S0R0FC	K0R1ETR	=SFA7B1	K0R1R1	=SFA8F5	K0R1M2	=SFA8F9
K0R1TAX	=SFA91	K0R1TBL	=SFA89	K0R1TFC	=S0C48	K0R1M?	=S0C20
K0R1TOP	=SFA84	K0R1TX	=SFA44	K0R1TXX	=SFA40	K0R1C0SS	=S015C
K0R1C0SS1	=S017C	K0R1TCL	=SFA02	K0R1G10	=S0C26	K0R1MPT	=S33
K0R1C0M	=S0144	K0R1XIN	=SFA61	K0R1MPT?	=S0C268	K0R1TOK	=SFA746
K0R1P	=S0D19	K0R1REG	=S0C79	K0R1STR	=S0D40	K0R1STR1NG	=S0CFA
K0R1OK	=S0A6A	K0R1T18H	=S0C59	K0R1X2	=SFA96	K0R1C	=SFA08
K0R1F	=S0138	K0R1FACK	=S0130	K0R1M40	=S0125	K0R1GET	=S0303
K0R1R2T2	=S0C08	K0R1R2T3	=S0C08A	K0R1L3	=S0C2EA	K0R1SHFAC	=S0148
K0R1CHR	=SFA67	K0R1DP	=SEDC4	K0R1TEMP	=S074F	K0R1EX	=SE150
K0R1M	=S0CB4	K0R1LINK	=S0C7FE	K0R1MEX	=S0744	K0R1STR	=S0D40
K0R1TOK	=S0CB2	K0R1	=SE18A	K0R12	=SE1D4	K0R1EDUP	=S0CFA
K0R1UP	=SFAA8	K0R1NT	=S0C	K0R1	=S0F20	K0R1	=S0F2C
K0R1T	=S0F0C	K0R1T	=S0CE9	K0R1TER	=SE38A	K0R1?	=S071E
K0R1G	=S077F	K0R1	=S07	K0R1	=S06	K0R1	=S077F
K0R1	=S07	K0R1	=S08	K0R1	=S04FB	K0R1	=S0582
K0R1	=SE2EA	K0R1	=S0C01	K0R1B1T	=SFCFA	K0R1	=SFF16
K0R1T	=SFCFD	K0R1TTE	=SFC0E	K0R1TTE2	=SFC0E	K0R1P1	=SFA09
K0R10	=SFCDF	K0R1T2	=SFFD2	K0R1R1R	=S0EAD	K0R1	=SFEFD
K0R10M	=S0A02	K0R1CALL	=SE72C	K0R1C	=SEFC0	K0R1S	=S0017
K0R10SP	=SFA0C	K0R12	=SFA0F	K0R1LAW	=SFF98	K0R1AT	=S01C0
K0R1AT1	=S010C	K0R1AT2	=S0116	K0R1R1R	=S0119	K0R1	=S0C01
K0R1?	=S0B99	K0R1	=S0CC8	K0R1TX	=SFB	K0R1SFAC	=S0E00
K0R1T	=SFA62	K0R1T1	=SFA0E	K0R1P1R1	=S0EAC	K0R1T1R1T	=S0C78

RESTORE	-SCB49	RESULT	-S42	RESUME	-SE679	RET1	-SC681
RET10	-SC227	RET11	-S0066	RET12	-S0096	RET2	-SC4FE
RET3	-SC95F	RET4	-SC9AD	RETS	-SCB77	RET6	-SCB04
RET7	-SCC8F	RETB	-SCCC7	RETY	-SCD30	RETCBA	-SF7C3
RETW1	-SCD50	RETW2	-SE09E	RETH0	-SEEB7	RETW4	-SF831
RETMS	-SFV61	? RETST	-SC988	RETURN	-SCCA9	RetinMR	-SC5C5
ROOSP1	-SFACF	RIGHT	-SEB0A	RIGHTSTR	-SD9AD	RHL	-SCC03
RK2	-SCC08	RINEN	-S20	RND	-SE2C8	RWADJ1	-SE2C0
RMDAL2	-SE2CA	RND8	-SDE9C	RNDH	-S4F	RMDL	-S4E
RMDSEED	-SC9	ROT	-SEAA1	ROTZ	-SF9	ROUN	-SEDA0
ROUND	-SDE94	RPSCK1	-SF8BA	? RPZ64	-SF110	RS1	-SC4DC
RS2	-SCAED	RS3	-SC6EB	RSTR1	-SFCSA	RTL	-SF07E
RT2	-SF07F	RTBL	-SFEC5	RTCR	-SF7F6	RTCS	-SF7F7
RTEMP1	-SFD	RTMSKZ	-SFA68	RTL1	-SD3A0	RTW10	-SDEA9
RTW11	-SDF28	RTW12	-SDF63	RTW13	-SE1F4	RTW14	-SE2BF
RTW15	-SECE7	RTW16	-SE43C	RTW2	-SDSC6	RTW3	-SD95F
RTW4	-SD489	RTW5	-SD887	RTW6	-SD8EE	RTW7	-SD0FC
RTW8	-SDC69	RTW9	-SDE88	RTWw88	-SC563	RTS1	-SEB09
RTS2	-SE9B1	RTS28	-SF8EF	RTS3	-SEA7D	RTS4	-SFC2C
RTS48	-SFCC8	? RTS4C	-SFD05	RTS5	-SFE17	RUN	-SFC37
RUNLINE	-SCC40	SAFEFOR?	-SC66E	SAV2	-SF89F	SAV3	-SFF4D
SAVD1M	-SD501	SAVE	-SCB35	SAVEL	-SF89D	SAVEADR	-SFFD8
SAVET	-SEED9A	SAVBLD	-SD692	SAVOP	-SDOFF	SAV1	-SE123
SAVYX	-SFD70	SCALE	-SEAA7	SCALEZ	-SE7	SCAM1	-SFD43
SCDCH	-SD357	SCNEND	-SF0B1	SCREEN	-SD221	SCRL1	-SF140
SCRL2	-SF168	SCRL3	-SFL71	SCRL4	-SF173	SCRM	-SF869
SCRN2	-SFA65	SCROLL	-SFL36	SCRTR	-SCP62	SE	-SC88D
SELEFAX	-SF7EA	SEMO	-SCEB9	SENDCH	-SCA19	SEMED	-SC278
SEPTAB	-SF805	SERIES	-SE28C	SERLEN	-S43	SERL00P	-SE2K3
SERMATN	-SE290	SERFMT	-S4D	SET4	-SC2C2	SET442	-SE292
SET48R	-SE4C1	SETFLG	-SC28E	SETCOL	-SF864	SETCR	-SF1C7
SETCTL	-SF090	SETON	-SCB73	SETOP	-SDF82	SETMO	-SCA66
SETFOR	-SDE41	SETHI	-SE5FA	? SETHPS	-SE769	SETIFLG	-SEFB6
? SETINW	-SFE80	SETXBD	-SFE89	SETNDZ	-SFE10	? SETXBD	-SFE18
SETXWRN	-SFE84	SETPLP	-SFAA0	SETPRMS	-SC97C	SETSGM	-SDF9F
SETVID	-SFE93	SETVPNT	-SD3FB	SETWMD	-SFBAC	SFS	-SCD0D
SF56	-SE058	SGM	-SDEAA	SGM?	-SD100	SGMCHK	-SDF6A
SGMCPR	-S4B	SGMFLT	-SD618	SGMIF	-SD843	SHS	-SDC17
SH2	-SDC10	SH2	-SDC28	SHAPEH	-S18	SHAPEL	-S1A
SHAPEPNT	-SE8	SHFTIN	-SC900	SHFTR	-SDC21	SHFTRES	-SDBF4
SHIFT	-SECCD	SHIN	-SC989	SHLWD	-SEAF5	? SI	-SESE1
SI1	-SE330	SI2	-SE340	SI3	-SE340	SI6M	-SE0VC
SI6M1	-SE0A0	SI6M2	-SE0A2	SI6NFLG	-S16	SI6MIF	-SDB48
SI6M	-SE308	SINGER	-SE38F	SINTEPR0	-SC558	SK2	-SC914
SKIP	-SD18C	SKIPFOK	-SC910	SKP	-SCF77	SKPTR	-SCA60
SKPV	-SD02A	SL1	-SEB16	SL2	-SEB20	SL2	-SEB23
SL1	-SE71F	? SLOOP	-SF48C	SMOCH	-SCA1C	SMO8?	-SDF47
SMYXERR	-SD135	SOFTTEW	-SD3F3	SOFTTEVL	-SD3F2	SOUND	-SCC40
SPC?	-SCE46	SPEED	-SESC9	SPEEDZ	-SF1	SPKR	-SCD30
SPMT	-SD074	SN	-ME167	SNdais	-SDC4C	SNRratio	-SDCA7
SPCUR	-SFD58	SROUT	-SFD58	SDZJR	-SFD25	SS	-SE29F
SSF	-SCB87	ST1	-SD182	STACK	-SD100	STATUS	-SD073
STC0R	-SE289	STE	-SE07A	STEP	-SCAC9	STEX	-SD08F
STHINEN	-SE401	STITLE	-SFB68	STXDW1	-SC99A	STMP	-SEEB0
STOING	-SFA9C	STOP	-SCB93	? STOR	-SFE08	STORE	-SE70F
STR	-SDA0F	STRCMP	-SD265	STREND	-S60	STR1M1	-SDBEF
STRIP	-SCB56	STRLIT	-SD701	STRLT2	-SD707	STRW61	-S4B

STRMG2	+S40	STRMG7	+S0336	STROUT	+SC5F	STRPRT	+SCA2
STRSPA	+S0AF7	STRTRNG	+SC9C8	STRTXT	+S01A9	STAP	+SC4F
STRXPT	+SC9AE	SUBERR	+S04B0	SUBFL6	+S14	SUBNAM	+S0820
SUBTBL	+SFFE1	SUAR	+S070F	SWRSG0	+S0707	SWRS	+S07C3
SVID	+SEF82	SWP	+S0402	SVY	+S0E05	SVYS	+SFB20
SWPNT	+SCF66	SYNCHR	+S01E8	SYNDR	+S01F1	? TIRET	+SFC0A
T4BK	+SE4A9	TAB	+SCE2B	TABASC	+SFSY0	TABCOR	+SFB8A
TABCSOR	+SF470	TABIT	+SCE50	TABV	+SFB5C	TABHERE	+SCE38
TARHST	+SC118	TAN	+SE354	TAK2	+SE37C	TAPEOUT	+SC020
TAPEWIT	+SEB3C	TENCART	+SEFS9	TEMP1	+S93	TEMP2	+S98
TEMP3	+S8A	TEMPPT	+S52	TEMPST	+S55	? TEMP	+SE1A7
TEPDAL	+SEF84	TESFLG	+S0C8A	TEST41	+S0441	TEST42	+S0442
TEST50	+S0450	TEST51	+S0451	TEST53	+S0453	TEST170	+S0470
TEST71	+S0471	TEST72	+S0472	TEST73	+S0473	TEST174	+S0474
TEST86	+S0486	TEST80	+S0480	TESTCAR	+SC0AB	TESTCL	+SFC08
TEXT	+SE703	TEXT0	+MFOCE	TFN	+S0E9F	? TIPEMO	+SC10E
TIPEND1	+SC1F8	TIPEND2	+SC1F9	TITL0	+SFS97	TK2000	+S03F8
TNPEP	+S99	TK07	+SC09F	TKDOME	+SCA79	TKENY?	+SCA4B
TKLFP	+SCA63	TKLUP	+SCA6F	TKTABL	+SC380	TKNAL	+SC24F
? TOSUB	+SEEE0	TOSUB1	+SEEB8	TRACE	+SE504	TRACE?	+SC31F
TRAMSBU	+SF680	TRCFLG	+SF2	? TRET	+SFE02	TRUE	+S0288
TRUE?	+SC0F0	TSAA6	+SF12A	TSB	+SE0DE	TSCART	+S9FF0
TSPAL	+SEFA2	? TSTOP	+SFOCA	TSUTL	+S0435	TVAR	+S078A
TW68K	+S0384	TXPSV	+S87	TXTPSV	+SF4	TXTPTB	+S88
TXTTAB	+S67	ULCHR	+SF680	ULTVF	+SF78F	UNARY	+S0234
UNDER1	+SCCA1	UMDFCN	+S0628	UMFC	+SC356	UNOT	+SC03A
UP	+SFC23	? UP0	+SE855	UP1	+SC8A8	UP2	+SE878
UP3	+SE87D	UP4	+SE87F	UPS	+SE876	UPBLOC	+SFB12
UPDOWN	+SE853	UPDOWN1	+SE881	? USER	+SFECA	USR	+S0A
USRAD0	+S00F8	V2	+S20	VAL	+S0A21	VALILEG	+SC583
VALTYP	+S11	VAR?	+S0194	VAR1	+S01F0	VARNAM	+S81
VARNT	+S0027	VARPNT	+S83	VARTAB	+S69	VART0	+SCC15
VCTB	+S49	VDOME	+S0877	VFCOM	+SC138	VFLIN	+SFB82
VY	+SFE36	VFYK	+SFE50	VIDOUT	+SFC05	VIDWRT	+SFB70
VL2	+S0A29	VL3	+S0A41	VLIN	+SE595	VLIME	+SFB28
VLIME2	+SFB26	VOLTBUF	+S779F	VPNT	+S40	VR1	+S0200
VR2	+S021E	VRFP	+SEEA8	VSEARCH	+S0369	VTAB	+SE580
VRFE	+S780	VTSTRT	+SFB98	WAIT	+SFCAB	WAIT2	+SFC49
WAIT3	+SFC6A	WAITB	+S0A9E	WHATX	+SE15C	WNB0TH	+S23
WDLFT	+S20	WNDOTP	+S22	WNDUP	+SE171	WNBWTH	+S21
WOK	+SCF9A	WPRM	+SE02A	WR1	+SFE04	WR0TT	+SFC06
WR0Y12	+SFEFF	WR1YTE	+SFEED	WRITE	+SFECD	WRITAPE	+SFC05
WT2	+S0440	WT3	+S0A81	X8H	+SE1	XDL	+SE0
X1	+SCE233	X2	+SE241	X3	+SE251	XAM	+SFB83
XAN8	+SFB63	XAMP8	+SFDCA	? X8ASIC	+SFE80	? X8XK	+SFA59
XDRAN	+SEAEF	? XDRAND	+SE90E	XDRAN1	+SE9E2	XDRAN2	+SEAC03
XDRAN3	+SEAB0	XDRAN4	+SEAD1A	XDRAN5	+SEAE25	XRE0	+S07F1
XSAY	+S048E	Y0	+SE2	YCNTC	+SEF	YRE0	+S07F2
YSAY	+S34	YSAY1	+S35	ZNRT	+S053A	ZBUFNAME	+S0F1C
ZEM0	+S0722	ZERDLY	+SFC08	ZER0	+S0248	ZEROFAC	+S0848
ZF	+S0F0E	ZFAC	+S0F5A	ZORAFFLO	+SFB67	ZLUP	+S0543
ZNO0	+SEEE9	ZN000	+SFF6C	ZM001	+SEEE7	ZP0A	+SFL0C
ZPGM	+SC208	ZFSTUFF	+SE425	ZTSUTL	+SEF77	ZK?	+SFA50
at	+SC5	bell	+S87	data	+S83	equal	+S00
fn	+SC2	for	+S81	gsub	+S80	goto	+S48
graf	+SF2	leftstr	+S8B	minus	+S29	net	+SC6
plus	+SC8	pop	+S81	print	+S8A	rea	+S82

scrn	=507	sgn	=502	spc	=503	step	=507
tab	=500	then	=504	to	=501		

Symbol table - numerical order:

DOWNH	=500	LOC0	=500	LOC1	=501	LOC2	=502
LOC3	=503	GOSTROUT	=503	LOC4	=504	R3L	=506
R3H	=507	R4L	=508	R4H	=509	USR	=50A
CHARAC	=500	EMOCHR	=50E	MJMDIH	=50F	PATR	=50F
DIRFLG	=510	VALTYP	=511	INTFLG	=512	DATAFLG	=513
GARFLG	=513	SUBFLG	=514	INPUTFLG	=515	CPRMASK	=516
SIGFLG	=516	SHAPEL	=51A	SHAPEH	=51B	HCOLOR1	=51C
COUNTH	=510	MDLFT	=520	WDMOTH	=521	WMDTOP	=522
WMDPTH	=523	CH	=524	CV	=525	GBASL	=526
GBASH	=527	BASL	=52B	BASH	=529	BASZL	=52A
BASZH	=52B	LXNEN	=52C	H2	=52C	V2	=52D
RMBEN	=520	FORMAT	=52E	CHKSUM	=52E	LASTIH	=52F
LENGTH	=52F	HMBK	=530	COLOR	=530	MODE	=531
IMFLG	=532	PROMPT	=533	YSAV	=534	ENDMASK	=535
YSAV1	=535	CSWL	=536	KSWL	=53B	PCL	=53A
PCH	=53B	AIL	=53C	A1H	=53D	A2L	=53E
A2H	=53F	A3L	=540	A3H	=541	A4L	=542
A4H	=543	ASL	=544	ACC	=545	A4S	=546
PGA	=547	PGH	=54B	VCTB	=549	RNDL	=54E
RNDH	=54F	LJMHM	=550	TEMP1	=552	LASTPT	=553
TEMPST	=555	TMCK	=55E	DEST	=560	RESULT	=562
TXITAB	=567	BASLOC	=567	VARTAB	=569	ARYTAB	=56B
STREND	=560	FRETOP	=56F	FRESPC	=571	MEMSIZ	=573
CURLH	=575	OLDLIM	=577	OLDTEXT	=579	DATLH	=57B
DATPTR	=57D	INPTR	=57F	VARHAR	=581	for	=581
VARPNT	=583	data	=583	FORPNT	=585	TXPSV	=587
LASTOP	=587	br11	=587	CPRTYP	=589	FMCHM	=58A
TEMP3	=58A	DSCPTR	=58C	DSOLEM	=58F	JMPADR5	=590
BLENGTH	=591	EXTRASV	=592	TEMP1	=593	WDRMS	=594
ARYPNT	=594	HIGHTR	=596	TEMP2	=59B	INDX	=599
TRPEXP	=599	EXPON	=59A	DPFLG	=59B	LOWTR	=59B
EXPSEM	=59C	FAC	=59D	DSCTRP	=59D	VPNT	=5A0
pop	=5A1	FACSEM	=5A2	SERLEN	=5A3	FFGEN	=5A4
ARG	=5A5	ARRSEM	=5A6	SGMCFR	=5A8	STRM1	=5A8
goto	=5A8	EXTRAFAC	=5AC	SERPNT	=5AD	STRM2	=5AD
FFSEM0	=5AF	BASLOC1	=5B0	gosh	=5B0	CRIBET	=5B1
rea	=5B2	CHRGOT	=5B7	TXTPTR	=5B8	print	=5BA
tab	=5C0	to	=5C1	fn	=5C2	spc	=5C3
then	=5C4	at	=5C5	not	=5C6	step	=5C7
plus	=5C8	ENSEED	=5C9	minus	=5C9	DOL	=5D0
equal	=5D0	DOH	=5D1	DY	=5D2	sgn	=5D2
DOHNT	=5D3	EL	=5D4	EH	=5D5	LOCK	=5D6
scrn	=5D7	ERRFLG	=5D8	ERRLEN	=5D8	ERRPOS	=5D8
ERRNUM	=5DE	ERRSTX	=5DF	XDL	=5E0	RDH	=5E1
YD	=5E2	DELH	=5E3	HOLORZ	=5E4	HMOX	=5E5
HPAG	=5E6	SCALEZ	=5E7	SHAPLPT	=5E8	left str	=5E8
COLORCNT	=5EA	YDNTC	=5EF	FRST	=5F0	SPEEDZ	=5F1
TROFLG	=5F2	graf	=5F2	ORMASK	=5F3	TXTPSV	=5F4
CURLSV	=5F6	RDNSTX	=5FB	RTZ	=5F9	COLL	=5FB
COOH	=5FC	RTEMP1	=5FD	STACK	=5FD0	IN	=5FD0D
BREV	=5G0F0	SOFTEN	=5G0F2	SOFTENH	=5G0F3	PAREDM	=5G0F4

AMPER	=902F5	TK2000	=903F8	USRAOR	=903FB	MMI	=903FB
2BLOCCL	=903FE	PGA	=90400	TSUTIL	=90435	FLGRAM	=90440
TEST41	=90441	TEST42	=90442	TEST50	=90450	TEST51	=90451
GRAFFLG	=90452	TEST53	=90453	BUF	=90460	PER1	=90468
PGR2	=9046A	TEST70	=90470	TEST71	=90471	TEST72	=90472
TEST73	=90473	TEST74	=90474	FIMLIN	=90482	TEST86	=90486
TEST80	=90480	XSHV	=9048E	IMG	=90600	AREG	=907F0
XREG	=907F1	TREG	=907F2	STATUS	=907F3	SPMT	=907F4
R16K	=907FF	PRG1	=90800	PRG3	=90803	PRG6	=90806
PRG9	=90809	PRG2	=9080C	PRG7	=9080F	PRG12	=90812
PRG15	=90815	PRG18	=90818	ESPR16K	=947FF	R48K	=907FF
CART0	=99000	CART3	=99003	CART6	=99006	CARTY	=99009
HEFLE1	=99854	HEFLE2	=99855	TSCART	=99FF0	IMICART	=99FF1
K8OUT	=9C000	K8PIN	=9C010	TAPEOUT	=9C020	SPKR	=9C030
IOCOLM	=9C050	IOPEB	=9C051	NOTA0FF	=9C052	DISPAG1	=9C054
DISPAG2	=9C055	PRST0N	=9C059	BTOUT0	=9C05E	BTOUT1	=9C05F
POLD	=9C064	DNICPOL	=9C070	PRF1	=9C080	PRF2	=9C088
DSK	=9C100	RELAT1	=9C100	RELAT1	=9C10C	RELAT2	=9C116
RELEAR	=9C119	TAM0MT	=9C11B	BUS0YT	=9C11D	FORM0MT	=9C120
VFC0M	=9C138	0ETEND	=9C13D	PR0XCOM	=9C144	PROCESS	=9C15C
PROCESS1	=9C17C	INDERR1	=9C186	IM0ERN	=9C188	ASS	=9C192
CONTASS	=9C195	IMVALID	=9C1B1	ASSEMB	=9C1B9	ASSEMB1	=9C1BD
ASSEMB2	=9C1CB	CONTECH	=9C1C8	? TIPEND	=9C1BC	TIPEND1	=9C1FB
TIPEND2	=9C1F9	IMEDIAT?	=9C1FA	IMPR	=9C200	ZPGN	=9C208
NMEDIAT	=9C200	FIMFNO	=9C222	.JPROCESS	=9C231	BUSESP	=9C234
SOUND	=9C240	TONAL	=9C24F	0UNACAO	=9C252	FREQ	=9C25A
FEMOTA	=9C263	PR0X0T	=9C268	POL	=9C271	SEWES	=9C27B
FINTEMP	=9C28A	TESTCAR	=9C298	NOLAN	=9C2AF	SET0FLG	=9C2BE
FINDAN	=9C2C1	SET4	=9C2C2	CART	=9C2C9	BASIC	=9C2D0
BASIC2	=9C2B9	CH0TABL	=9C2D6	UNFNC	=9C356	MATH0BL	=9C392
NTRMS	=9C3A7	UN0?	=9C3AA	PLUS	=9C3A0	TOKTABL	=9C3B0
MEMORR0	=9C54F	MOseFOR	=9C54F	SINTERR0	=9C55B	RTMse55B	=9C563
MseDATA	=9C573	VALILEG	=9C583	ESTOR	=9C58F	ExcMEM	=9C596
COMDEF	=9C5A4	IMDLEB	=9C588	Reli0ERR	=9C5C5	01Vpa2R0	=9C595
CONDELE	=9C5E5	IMC0MP	=9C5F3	EXCSTRG	=9C5FF	FORM0C0	=9C60C
IMP0SSL	=9C622	FIMDEF	=9C62C	ERRIM	=9C63F	IMNG0	=9C647
BRAKIM	=9C64C	0TF0MPHT	=9C654	FIMFOR	=9C659	SAHEFOR?	=9C66E
NIFOR	=9C678	RET1	=9C681	BLTU	=9C682	BLTU2	=9C689
SETEND	=9C6A6	M8YTT	=9C6A2	N8YTT	=9C6A6	N8PAG	=9C6B0
CHMEM	=9C6C5	REASON	=9C6D2	RS1	=9C6DC	RS2	=9C6E0
RS3	=9C6EB	RET2	=9C6FE	MEMERR	=9C6FF	ERROR	=9C701
DOERRMSB	=9C708	ERLUP	=9C70E	PRINTM?	=9C720	RESTART	=9C72B
NOLIN	=9C748	ML	=9C78E	W8NM	=9C796	NEMLM?	=9C7A4
W8P8G	=9C7C0	INSRTLIN	=9C7D9	LTKSET	=9C7E1	NOLINK	=9C7ED
PULLINK	=9C7FE	FIMDEOL	=9C800	IMLIN	=9C818	IMLINK	=9C810
IMLINK	=9C824	IMLINA	=9C82A	IMLIMS	=9C848	GOB8FS	=9C84E
STRIP	=9C856	NO1	=9C861	IMCH	=9C868	GETIM	=9C86E
PARSE	=9C8B1	KYCHR	=9C882	SE	=9C880	TOK?	=9C8FF
ISTOK?	=9C8A7	NY	=9C889	KX	=9C88E	LIN	=9C8BF
PUTTOK	=9C8E2	PUTIM	=9C8E4	SSF	=9C8F7	EM?	=9C8F9
SHFITM	=9C900	SHIM	=9C909	SKIPTOK	=9C910	SK2	=9C914
PLU?	=9C918	DOHE	=9C927	FIMLIN	=9C931	FL1	=9C935
FL2	=9C94C	GETLINK	=9C955	M8KUCH	=9C95E	RET3	=9C95F
NEW	=9C968	SCRICH	=9C962	SETPTRS	=9C97C	CLEAR	=9C981
CLEARC	=9C980	STRIME	=9C996	RET4	=9C9A0	STXTP?	=9C9A6
? RETST	=9C988	LIST	=9C99C	STRONG	=9C9C8	EMING	=9C9D8
MAINLST	=9C9E3	NOLST	=9C9F1	LST0?	=9C9C0	LSTILIN	=9C9DE

LISTLOP	=SCA15	SENDCH	=SCA19	SHOCHR	=SCA1C	NDR	=SCA29
LISTED	=SCA20	GETCHR	=SCA43	DC	=SCA4B	TOKENP	=SCA4B
CONTOK	=SCA50	SKPTX	=SCA60	TOKLP	=SCA63	PRTK	=SCA6A
TOKLUP	=SCA6F	TOKSWE	=SCA79	FOR	=SCA80	FOR2	=SCA91
STEP	=SCA92	ONESTEP	=SCA80	NEWST1	=SCAEC	DIRECT	=SCAFF
TRACE?	=SCB1F	EXECUTE	=SCB37	GOEND	=SCB40	GOCHO	=SCB42
GOAND2	=SCB44	GOCHO3	=SCB52	NOTOK	=SCB5F	COLON?	=SCB62
J5Y	=SCB66	RESTORE	=SCB69	SETDM	=SCB73	RETS	=SCB77
ISCMTC	=SCB78	OK	=SCB85	DRFLG?	=SCB88	CTRC?	=SCB91
STOP	=SCB93	END	=SCB95	END2	=SCB96	END3	=SCB9B
END4	=SCB9F	DOSTART	=SCBB8	CONT	=SCBB8	CON	=SCB96
RET6	=SCB04	SAVE	=SCB05	LOAD	=SCBEE	JUNK	=SCC12
WRTD0	=SCC15	PROG10	=SCC26	RUN	=SCC27	RUNLINE	=SCC40
GO5UB	=SCC46	GLINE	=SCC5A	GOTO	=SCC63	G01	=SCC7A
G02	=SCC7E	RET7	=SCC8F	POP	=SCC90	UNDEFR	=SCCAL
OSTMER	=SCCA6	RETURN	=SCCA8	DATA	=SCCBA	ADON	=SCCB0
RETB	=SCC27	DATAN	=SCC2B	REH	=SCC2B	RH1	=SCC30
RH2	=SCC28	PULL3	=SCC2A	DF	=SCC2E	TRUE?	=SCC3D
REN	=SCC01	DFTRUE	=SCC06	JGCHO	=SCC0E	ONGOTO	=SCD11
GOTO?	=SCD19	ONCAT	=SCD10	KUNUM	=SCD25	RETR	=SCD30
LINGET	=SCD01	ASCHEX	=SCD37	WDIG	=SCD65	LET	=SCD68
LET2	=SCD88	LETREAL	=SCD9C	LETSTR	=SCD9F	PUTSTR	=SCD90
DESC?	=SCD81	COPSTR	=SCD8F	NEWDESC	=SCD85	COPY	=SCD9C
PRSTRNG	=SCD04	PRINT	=SCDFA	PRINT2	=SCDFC	CR00	=SCD20
HEGATE	=SCD25	RET10	=SCD27	TAB	=SCD28	KRCLN	=SCD33
TABHERE	=SCD30	SPC?	=SCD46	TAB1	=SCD50	KASPC	=SCD51
NEXTCHR	=SCD54	DUPIC	=SCD5A	STRBUT	=SCD5F	STRPRT	=SCD62
KCHWR	=SCD69	OUTSP	=SCD7C	OUTQUES	=SCD7F	OUTD0	=SCD81
SEND	=SCD89	INPUTERR	=SCD96	READERR	=SCD90	ERLIN	=SCD94
INFERR	=SCD9B	RESPERR	=SCD9C	DREENT	=SCD95	GET	=SCD25
INPUT	=SCD07	ROUT	=SCD09	DIR?	=SCD2C	KXIN	=SCD01
READ	=SCD07	ZF	=SCD0E	HAZIMP	=SCD10	KXIMP	=SCD16
SNDG?	=SCD47	STRP	=SCD4F	INSTART	=SCD53	PUTCH	=SCD67
PENCHR	=SCD73	PCCHR	=SCD74	SKP	=SCD77	KUNIN	=SCD98
DATIN	=SCD91	WRX	=SCD9A	SWPMT	=SCD96	IMFFIN	=SCD01
FINDATA	=SCD0B	KOS	=SCD0E1	INPCOME	=SCD0E	NTD	=SCDFF9
RET11	=SCD006	EXIG	=SCD007	REDIG	=SCD017	NEXT	=SCD021
WWRKT	=SCD027	SKPV	=SCD02A	GERR	=SCD035	GOFFOR	=SCD037
GOHEMT	=SCD07A	ENDFOR	=SCD07D	FRMLN	=SCD08F	CHKMLN	=SCD092
CHRSTR	=SCD094	CHRVAL	=SCD095	RET12	=SCD09B	CV2	=SCD09C
KISATCH	=SCD09E	JERR0R	=SCD0A0	FRNEVL	=SCD0A3	HD1	=SCD0A9
FEVLOOP	=SCD0AE	FRNEVL2	=SCD0B0	CPROP	=SCD0C0	CHKTF?	=SCD0C0
KLTH	=SCD0E3	PREFTEST	=SCD0F5	KXOP	=SCD0FE	SAVOP	=SCD0FF
COMPARE	=SCD10C	HD2	=SCD116	PREFMC	=SCD11E	PSHND	=SCD125
SMTXERR	=SCD125	PSHF	=SCD130	PSHFACX	=SCD130	PUSHFAC	=SCD148
NOTNATH	=SCD150	DOEX	=SCD160	DORHT	=SCD162	DRHT	=SCD169
DOHATH	=SCD160	EXIT	=SCD185	GETVAL	=SCD180	SKIP	=SCD19C
NUMBER	=SCD191	WAR?	=SCD194	STRTXT	=SCD1A9	ST1	=SCD1B2
NOT?	=SCD180	EWOP	=SCD1C0	WRTZ	=SCD1C7	FW?	=SCD1CC
SGH?	=SCD103	PARCHK	=SCD1D4	CHVCL5	=SCD1ED	CHKOPN	=SCD1E3
CHKCON	=SCD1E6	SYNCHR	=SCD1EB	SYNERR	=SCD1F1	NIN	=SCD1F6
EMAL	=SCD1F8	VMAL	=SCD1FD	VR1	=SCD200	VR2	=SCD21E
SCREEN	=SCD221	UNARY	=SCD22A	NOTinstr	=SCD262	DOROUT	=SCD267
OK	=SCD277	ANC	=SCD27D	FALSE	=SCD285	TRUE	=SCD288
POSP	=SCD280	STRCMP	=SCD2A5	SFS	=SCD2CD	CHL00P	=SCD2D2
MUNCHP	=SCD298	DOCHP	=SCD29D	CHPDONE	=SCD2E9	JF	=SCD2F2
KODIN	=SCD2F6	DIN	=SCD2F9	PTARGET	=SCD360	PTARGET2	=SCD368

FTRGET3	=S030A	BADNAM	=S0314	MANOK	=S0317	MORNAM	=S0321
GLTL	=S0328	BYPASS	=S032C	STRMG7	=S0336	INTVAR7	=S0340
MJH	=S0350	SCDCH	=S0357	JARY	=S0360	BSB	=S0363
VSEARCH	=S036F	KXVAR	=S0373	MU1	=S0375	MW2	=S037F
NXPTR	=S0380	ISLETC	=S0397	RTN1	=S03A0	NOTFND	=S03A1
THDBRK	=S0384	NEWAR	=S0384	MW	=S03CC	SETUPHT	=S03FB
SHP	=S0402	GETARY1	=S0407	GETARY2	=S0409	GO	=S0413
NEGMUM	=S0418	MAXINT	=S041C	MAXINT	=S0422	AYINT	=S0426
MJ1	=S0433	MJ2	=S0435	ARRAY	=S0430	KXTD1H	=S0446
FNDARY	=S0463	ARYLOOP	=S0467	ARYNAM7	=S0493	MGARY	=S04A2
SUBERR	=S0480	TRERR	=S0483	JER	=S0485	ARYFOUND	=S048F
CHQD1H	=S04CA	NOTFOUND	=S0402	MARARY	=S0408	NINT	=S04EF
RAR	=S04FB	SAVD1H	=S0501	DFTD1H	=S0511	JARY	=S0534
ZLUP	=S0543	KXP5	=S0548	FNDEL1H	=S0545	D1HL1P	=S0560
OSE	=S0583	GME	=S0586	D1MOK	=S0589	D1MOK2	=S058A
MOH	=S0598	HINTA	=S05AC	RARY	=S05B2	RTK2	=S05CA
MJL.T	=S05C7	MJ1	=S05D0	MJ2	=S05D4	MJ3	=S05F3
FRE	=S05FB	FRE2	=S05FF	GDVWF	=S060C	FOS	=S0619
SDFL1	=S0618	ERRD1R	=S0620	UMDFNC	=S0628	DEF	=S062D
FNC?	=S0658	FUNCT	=S066E	SAVLD	=S0692	GETOLD	=S06C2
FNDATA	=S06C9	STR	=S06DF	STRD1H	=S06EF	STRSPA	=S06F7
STRLET	=S0701	STRL12	=S0707	FEND	=S0711	QUO?	=S071E
ZEND	=S0722	NZ	=S0723	FE1	=S072F	FE2	=S0739
FUTHEW	=S0744	JERR	=S074C	PUTENP	=S074F	GETSPA	=S074C
GETSPC	=S076E	CT	=S0779	GETSPA	=S0783	FULL	=S078E
GABGAG	=S079E	FNDVAR	=S07A2	TUAR	=S07BA	SIMAS	=S07C3
SUAR	=S07CF	SHRGO	=S07D7	ARYVAR	=S07DC	ARYVA2	=S07E4
ARYVA3	=S07EB	ARYY00	=S07F3	BPL1	=S082A	ARYSTR	=S0826
LG0	=S082E	DVAR5	=S0833	DVAR	=S0830	0V1	=S0852
DV2	=S085C	DVAR5	=S086C	VDONE	=S0877	GRBPAS	=S087C
CAT	=S0881	MTL	=S08D1	NOVIMS	=S08EE	NOVSTR	=S08FC
NOVSTR	=S0900	WIS2	=S0904	MV3	=S0900	BPL2	=S0916
FRESTR	=S0917	FREFAC	=S091A	FRETM	=S091E	BPL3	=S0949
HB	=S094A	FRETHS	=S094F	RTK3	=S095F	CHRSTR	=S09A0
LEFTSTR	=S0974	MSL	=S097A	IMS2	=S0980	IMS3	=S0981
IMS4	=S0982	BPL4	=S0999	RIGHTSTR	=S09A0	KIDSTR	=S09A8
IMS5	=S098C	DNSTRMS	=S0903	LEM	=S09FD	GETSTR	=S09F6
ASC	=S09FF	DT16	=S0A0C	STRYTC	=S0A0F	GETBYT	=S0A12
CONJMT	=S0A15	VAL	=S0A21	VL2	=S0A2F	VL3	=S0A41
POINT	=S0A57	GTNUM	=S0A60	CMRYTE	=S0A66	GETADR	=S0A6C
PEEK	=S0A7E	POKE	=S0A95	W1T8	=S0A9E	W12	=S0A0D
MT3	=S0AB1	RTN4	=S0AB9	FADCH	=S0AB4	FSUR	=S0AC1
FSUBT	=S0AC4	AD0	=S0AD3	FAD0	=S0A08	FADOT	=S0A08
AD1	=S0AED	AD2	=S0AEB	AD3	=S0BD4	AD4	=S0BD8
ADS	=S0B14	SUBNAM	=S0B20	SGMTF	=S0B43	SIGHIF	=S0B48
FLOOP	=S0B4C	ZEROFAC	=S0B6B	AtaFAC	=S0B6A	AtaFACS	=S0B6C
ADMAN	=S0B6F	FRI	=S0B8E	FR2	=S0B9A	FR3	=S0BA7
FROOM	=S0B89	RTMS	=S0B87	MEGFAC	=S0B88	MEG2	=S0BBE
PLUSZPS	=S0BEO	RTN6	=S0BEE	OVERFLOW	=S0BEF	SHFTRES	=S0BF4
NOFT	=S0BF6	SHIFT	=S0C0A	SH1	=S0C17	SH2	=S0C1D
SHFR	=S0C21	SH3	=S0C2B	HIM	=S0C2D	LOSSER	=S0C32
SIMaelo	=S0C47	SIMaelo	=S0C4C	Memelo	=S0C51	LOSS1S	=S0C5A
LOG	=S0C58	GTN	=S0CA2	LG2	=S0C45	FWULT	=S0C99
FNA.LT	=S0C9C	FNU	=S0CA1	FH1	=S0CCA	FH2	=S0CCF
FNG	=S0CB2	FMA	=S0CEE	RTN7	=S0C7C	CONUPK	=S0CFD
ADEXP	=S0CB8	ADXC2	=S0CB2A	ADXC3	=S0C35	ADEX4	=S0C40
OUTFORMS	=S0C45	ZERO	=S0C46	JAV	=S0C50	MJL10	=S0C53

RTNB	+S0649	MUMD	+S006A	DIV10	+S006F	DIV	+S007B
EMWIMP	+S007A	FDIV	+S0080	FDIVT	+S0083	FD1	+S009A
FDH	+S00A2	FD2	+S00B0	FD3	+S00B3	FD4	+S00C0
FDS	+S00CE	FD6	+S00E8	FD7	+S00EF	DIVZ	+S00F8
RESXFAC	+S0E0D	NOVFN	+S0E13	NOVCF	+S0E38	NOV1F	+S0E3B
NOVNL	+S0E3D	SETF08	+S0E41	NOVNF	+S0E45	NOVFA	+S0E4D
NFA	+S0E4F	NFAZ	+S0E73	NOVNF	+S0E7D	NAF	+S0E8D
NAF2	+S0E82	RTWP	+S0E9B	RND8	+S0E9C	ROUND	+S0E9A
SDM	+S0E9C	SIDM1	+S0EAD	SIGH2	+S0EAD	RTK10	+S0EAD
SGN	+S0EAA	FLOAT	+S0EAD	FLO1	+S0EB5	FLO2	+S0EBA
NBS	+S0EC9	FCOMP	+S0ECC	FCOMP2	+S0ECE	FC1	+S0FD3
FC2	+S0FD9	QINT	+S0F0C	Q1	+S0F20	RTK11	+S0F2B
Q22	+S0F2C	DNT	+S0F3D	ZFAC	+S0F5A	RTK12	+S0F63
FTM	+S0F44	FIN2	+S0F68	FIND	+S0F77	EWL	+S0F7B
NOIDBIT	+S0F7C	CHNDP	+S0F8D	SETSDN	+S0F9F	DP01G	+S0FAL
GODEX	+S0FAA	SGNCH	+S0FA6	SETOP	+S0FB2	ADJEP	+S0FB8
ACX	+S0FBA	DPLEFT	+S0FC3	DPRIGHT	+S0FCC	EVDSM	+S0FD3
EVD	+S0FD8	DNRAT0G	+S0FD8	NDF	+S0FE2	ADACC	+S0FEF
GETEXP	+S0E02	INDG	+S0E01	STEX	+S0E1F	CEMNAEDC	+S0E74
ETLweLM	+S0E09	SILHAG	+S0E2E	INPRT	+S0E33	LDPRT	+S0E3E
PRINTFAC	+S0E48	PRSTR	+S0E48	FOUT	+S0E4E	FACSTRNG	+S0E5D
SFSG	+S0E5B	NR1ZE	+S0E6F	NR	+S0E71	STE	+S0E7A
CPBN1	+S0E7C	CPNM	+S0E87	JM10	+S0E92	JO10	+S0E99
ROU	+S0E9D	INTPART	+S0EAD	DPLDC	+S0EB8	DPL	+S0EB9
PUDOP	+S0EC4	SVY	+S0E05	MAKSTR	+S0E07	NSLUP	+S0E08
PAILITY?	+S0E0F	COUNTED	+S0E01	NAW0151	+S0E08	SP1Y	+S0E123
NRBACK	+S0E13D	WDEX?	+S0E14D	PUTEA	+S0E15D	WMTX	+S0E15C
WDEP	+S0E171	HARKEM?	+S0E174	PWTSTR	+S0E179	HALF	+S0E17E
DECTBL	+S0E183	TEVD	+S0E1A7	SBR	+S0E1A7	PPMR?	+S0E1B1
PM1	+S0E18A	PM2	+S0E1D4	REGOP	+S0E1EA	RTK13	+S0E1F4
LOGe	+S0E1F5	EXPSER	+S0E1FA	EXP	+S0E22D	X1	+S0E23D
NR	+S0E23E	X2	+S0E241	X3	+S0E251	000SER	+S0E274
SERDES	+S0E28C	SERMALN	+S0E29D	SS	+S0E29F	SERL08P	+S0E2A3
NOTERH	+S0E2B0	RTN14	+S0E2BF	RNDADJ1	+S0E2D0	RNDADJ2	+S0E2D4
RND	+S0E2CB	RD1	+S0E2E6	RD2	+S0E2E1	DOS	+S0E304
SIM	+S0E30B	SIL	+S0E33D	S12	+S0E34D	S13	+S0E34D
TAN	+S0E354	TAN2	+S0E37C	P1div2	+S0E38D	P1weck2	+S0E3E5
QUARTER	+S0E38A	SIMSER	+S0E38F	ATN	+S0E38B	ATM1	+S0E3D0
ATN2	+S0E3CE	ATN3	+S0E3E1	RTM15	+S0E3E7	ATMSER	+S0E3E8
ZFSTUFF	+S0E425	N10	+S0E42B	RTN16	+S0E43C	COLDST	+S0E442
WVZP	+S0E44C	TARH	+S0E44F	SET48K	+S0E4C1	HISFFF	+S0E4CF
STIMEH	+S0E4B1	TH1000	+S0E4E7	TABAG	+S0E4E9	NRASLDC	+S0E4FA
INBRAS3	+S0E51A	CALL	+S0E53A	PLOTFA5	+S0E74D	GOERR	+S0E75A
LINC30R	+S0E55D	AT7	+S0E5AC	PLOT	+S0E79F	MLM	+S0E5B6
VLIM	+S0E595	COLOR?	+S0E5A3	EWLVAR	+S0E5AA	STON	+S0E5B9
VTAB	+S0E5B0	SPEED	+S0E5C9	TRACE	+S0E5D4	NOTRACE	+S0E5D6
NORMAL	+S0E5D8	INVERSE	+S0E5D0	NRH	+S0E5DF	SI	+S0E5E1
HIREH	+S0E5E7	JHM	+S0E5F7	SETHI	+S0E5FA	LOWEN	+S0E607
QMERK	+S0E62C	HMAOLERR	+S0E64A	RESUME	+S0E679	JSTN	+S0E68F
DEL	+S0E692	N11	+S0E6BB	NOVDN	+S0E6C6	NOVIT	+S0E6CB
N12	+S0E6D2	N13	+S0E6D8	NOX	+S0E6E9	GR1	+S0E6F1
OR	+S0E6F9	TEXT	+S0E703	STORE	+S0E70F	SL1	+S0E71F
RECALL	+S0E73C	NOX	+S0E74C	HGR2	+S0E74B	HGR1	+S0E74F
HGR	+S0E75B	NTXT	+S0E764	SETHP0	+S0E769	HCLR	+S0E771
BRNGD	+S0E775	BRNGD1	+S0E77D	HP95M	+S0E79D	HP95M1	+S0E7C1
HP06H2	+S0E7C2	HP10T0	+S0E7D7	LFRIT	+S0E7E5	LEFT	+S0E7E7
LRL	+S0E7EE	LEFT1	+S0E7F1	LEFT2	+S0E7F6	NRWGX	+S0E7F8



?	CSHIFT	+SE7FC	CSHFT2	+SE7FE	RTS1	+SE8D9	RIGHT	+SE8DA
	LRUD01	+SE81C	LRUD02	+SE81D	LRUD1	+SE83D	LRUD2	+SE83A
	LRUD3	+SE844	LRUD4	+SE84B	EQ3	+SE84D	UPDOWN	+SE853
?	UP0	+SE85S	UP1	+SE868	UP5	+SE876	UP2	+SE87B
	UP3	+SE87D	UP4	+SE87F	UPDOWN1	+SE8B1	DOWN	+SE8B5
	EQ4	+SE88B	DOWN3	+SE8A4	DOWN1	+SE8AB	DOWN2	+SE8AD
?	HLINRL	+SE8B1	HGL1H	+SE8BB	HLINC	+SE8D1	HLINC	+SE8E9
	MOVEX	+SE8FD	MOVEX2	+SE902	HCOLUNT	+SE90C	HL1H4	+SE91F
	NSKTBL	+SE923	EQ1C	+SE93A	COSTBL	+SE93B	? HFIND	+SE94C
	HFIND1	+SE971	HFIND2	+SE97F	RTS2	+SE981	? DRAW0	+SE982
	DRAW1	+SE986	DRAW2	+SE9A7	DRAW3	+SE9B1	DRAW4	+SE9BE
	DRAMS	+SE9C9	RT4	+SE9D9	? XDRAM0	+SE9DC	XDRAM1	+SE9E2
	XDRAM2	+SEAD3	XDRAM3	+SEAD0	XDRAM4	+SEAL8	XDRAM5	+SEAC5
	HIS	+SEAC5	HFSH	+SEAD8	HFSH1	+SEAE4	GBERR	+SEAE7
	HCOLOR	+SEAE6	EWAR	+SEAF1	RTS3	+SEAF0	HPLOT	+SEAF3
	HP2	+SEAB8	HP3	+SEABF	ROT	+SEAM1	SCALE	+SEAF7
	DRAPWT	+SEAF0	DP1	+SEAC1	DP2	+SEAC7	DP3	+SEAE6
	DRAW	+SEAE9	XORAW	+SEAEF	SHLOAD	+SEAF5	SL1	+SEB16
	SL2	+SEB20	SL3	+SEB23	TAPEPNT	+SEB3C	GETARYPT	+SEB59
	HTAB	+SEB67	HTAB1	+SEB6C	HTAB2	+SEB7A	MOTOR	+SEB7D
	AOUB?	+SEB67	MOTRA	+SEB8D	NLOAD	+SEB92	CHKDFLT	+SEB9E
	NSAVE	+SEB84	CHKDFST	+SEBDD	CHRT	+SEBEE	JSYNERH	+SEBFF
	AKTINH	+SEB75	NP	+SEBFB	NA	+SECD0	COUT3	+SECCD
	ISVD	+SECL5	ISGRF?	+SECC2C	CHGRF	+SECC3A	ISREY?	+SECC4B
	MTRK?	+SECC5B	ISCTH?	+SECC6D	ISCTB	+SECC70	PRPRES	+SECC7Y
	TESFLG	+SECC8A	SET442	+SECC92	SAVET	+SECC9A	TFORW	+SECCAE
	MXZER00	+SECC8C	NOBSAV	+SECC7	MXZER01	+SECCC	FMSAVE	+SECCD0
	PRP	+SECD19	WGRN	+SECD2A	CSST1	+SECCD0	RETM1	+SECCD8
	CSST2	+SECD51	CSST3	+SECD5A	CSST4	+SECD73	CSST5	+SECD7C
	CSST6	+SECD8C	CSST7	+SECD95	RETK2	+SECD9E	TFH	+SECD9F
	NRN	+SEDA1	LOADT	+SEDAE	NKLNAM	+SEDD0	INKSUM	+SEDD3
	PSNMH	+SEDDCA	CPLTRIT	+SEDDC	LEOL	+SEDDC	OKSUM	+SEDD7
	LEDCH	+SEED0	INPNAM	+SEED1	CPLTHWE	+SEED3A	DNFSP	+SEED45
	COMPNAME	+SEED3	DNFWAIT	+SEED70	ATLCP	+SEED7E	RETK3	+SEED87
	TOSUM1	+SEED8B	ACOMPAS	+SEED9E	WMP	+SEEDAB	STMP	+SEEDB3
	ACOMPAG	+SEED9	ACOMPAG2	+SEEDC7	PHADR	+SEEDD3	TSD	+SEEDDE
?	TOSUB	+SEEDD	ZMOD	+SEED7	ZMOD	+SEED7	KWRD1	+SEEDDE
	KWRD	+SEEEF	NOASP	+SEED70	KWDHAM	+SEEEF	CHPLSP	+SEF18
	ZBUFNAME	+SEF1C	FDMNAM	+SEF2B	BUSCON	+SEF2B	EXT101	+SEF33
	LOOP20	+SEF44	FRESF	+SEF49	FTHESP	+SEF57	TENCART	+SEF59
	ZTSUTL	+SEF77	CARBUF	+SEF7E	TEPGUL	+SEF84	MOV10	+SEF91
	ISCTP	+SEF96	TSPQWL	+SEFA2	MOV101	+SEFA0	SVTD	+SEF82
	RECAP	+SEFC0	PLAT1	+SEFC8	PLOT0	+SEFCA	DESINV	+SEFFD
	NIMASEP	+SEFEF	LINCORD	+SEFF1	GRPLOT	+SF001	IMG	+SF014
	JMA	+SF018	SVYS	+SF020	PROCTOL	+SF022	ATBOU	+SF024
	NRHC	+SF032	TESTCL	+SF03B	KXY	+SF03D	SCAN1	+SF043
	HMBL1H	+SF054	MOXUL	+SF05E	KXKEY	+SF064	SAVIX	+SF070
?	CLREY	+SF074	RT1	+SF07E	RT2	+SF07F	INFCTCL	+SF081
	SETCTL	+SF09D	SCMEND	+SF0B1	PRCMD0	+SF0B6	PRCMD1	+SF0B8
	CHRTXT	+SF0DD	TEXT0	+SF0DE	FOSDR	+SF0DE	IMPCH	+SF0E9
	FJH	+SF10A	ZFG4	+SF10C	? RP204	+SF11D	COOCR	+SF117
	JMR	+SF127	TSAA6	+SF12A	JCAR16	+SF139	SCROLL	+SF136
	SCR11	+SF140	SCR12	+SF148	SCR13	+SF171	SCR14	+SF173
	CLRVL	+SF198	COU2	+SF1A6	IOCTP	+SF1A9	HTCTP	+SF1C2
	SETCR	+SF1C7	CONTR	+SF1CA	DECK1	+SF1D0	ESPERA	+SF1C3
	JCAR1Y	+SF1F5	JPRB18	+SF1F9	GERCHR	+SF200	GERDRF	+SF400
	TABCSR	+SF470	DISK10	+SF590	TITULO	+SF597	TAB3C	+SF590

ZET	=SF25D	PROXIH	=SF661	CARACT	=SF67F	BOGRA	=SF691
STODMG	=SF69C	ZGRAFFLG	=SF6A7	TRANSBUF	=SF6B0	FIMTRAYS	=SF6B0
PLUBAS	=SF6BE	OUTROCH	=SF6C1	ULTCHR	=SF6D0	EDTBAS	=SF6D4
TDDBS	=SF6DA	CJNTR	=SF6F3	FDMFA	=SF6FC	FEMBAS	=SF706
ENDTBS	=SF707	INPBAS	=SF709	BUSTOKEN	=SF736	DEGUT14	=SF741
PRCTOK	=SF746	ETOKEN	=SF753	COMVTK	=SF75C	IJCBAS	=SF769
JRBC	=SF77A	IMPBAS	=SF77D	PRLETR	=SF7B1	ULTVF	=SF7BF
FIERET	=SF798	VOLBLUF	=SF79F	JRBE12	=SF78A	VBSE	=SF78D
NETCHNA	=SF7C3	FINDCC	=SF7D2	CTB	=SF7D0	JDECBUF	=SF7E7
SELEFAX	=SF7EA	RTR	=SF7F6	RTCS	=SF7F7	GRNBLD	=SF7FB
PLDT2	=SF800	PLDTH	=SF803	UPBLAC	=SF812	HLINE	=SF819
HLINE1	=SF81C	VLINER	=SF826	VLIME	=SF82B	RETR4	=SF831
? CLRSCR	=SF832	CLRTOP	=SF836	CLRSCT	=SF83B	CLRSCT	=SF83C
? DBASCALC	=SF847	CSALC	=SF856	? NEXTCOL	=SF85F	SETCOL	=SF864
SCRH	=SF869	PRMCC	=SF874	FOCC	=SF87F	INSD11	=SF885
? INSD52	=SF88F	INSD53	=SF891	TEVEN	=SF89E	ERR	=SF8AB
GETFHT	=SF8AC	MNNDL2	=SF8C1	MNNDL2	=SF8C5	MNNDL3	=SF8CC
INSD30P	=SF8D0	PRMTP	=SF8D4	PRMTRL	=SF8D8	FRML	=SF8F5
PRM2	=SF8F7	PRADR1	=SF910	PRADR2	=SF914	PRADR3	=SF926
FRADR4	=SF92A	PRADR5	=SF930	? RELADR	=SF93B	FRHT14	=SF940
? PRMTAX	=SF941	? PRMTX	=SF944	PRBLNK	=SF94B	PRBL2	=SF94B
? PRBL3	=SF94C	PCADJ	=SF953	? PCADJ2	=SF954	PCADJ3	=SF956
PCADJ4	=SF95C	REHNS	=SF961	FMT1	=SF962	FMT2	=SF96A
CHAR1	=SF98A	CHAR2	=SF98A	MNHL	=SF9C0	MENR	=SFAD0
ZRQ	=SFA40	BREAK	=SFA4C	? XBRK	=SFA59	RESET	=SFA62
SCRN2	=SFA65	RTRSKZ	=SFA68	RESET1	=SFA6E	FLXSEV	=SFA94
EXT10	=SFA9F	? NOFIX	=SFAA2	PPREG	=SFAA5	? PWRUP	=SFAAB
SETPLP	=SFAAD	? SLOOP	=SFABC	MXTRY1	=SFA8E	REGDSP	=SFACC
RDOSP1	=SFACF	RDSP1	=SFA09	NEWS	=SFAF3	INC	=SFB02
SEP1AB	=SFB05	NTCL	=SFB14	MXTP05	=SFB20	MXTP051	=SFB25
JCOOGR	=SFB2E	POSES4	=SFB31	INT1	=SFB3B	SETM40	=SFB4C
TABV	=SFB5C	OPTK2000	=SFB5F	STITLE	=SFB6B	VDWAIT	=SFB7D
WPCN1	=SFB6A	LDRET	=SFB93	NDWATT	=SFB95	VTSRT1	=SFB9B
SMEL	=SFB9D	SMZ	=SFB9F	TABCO8	=SFBAA	POSCUR	=SFB82
INVCUR	=SFBCA	BELL0	=SFB0F	? BELL1	=SFB0D	BELL2	=SFB04
RTS28	=SFB0F	WOP01	=SFBFD	CVB8	=SFBF3	ARB	=SFBF5
DFC00	=SFBFD	? VIDOUT	=SFC05	DESE90	=SFC18	DESE91	=SFC20
LP	=SFC23	RTS4	=SFC26	RTS4	=SFC2C	PRRES0	=SFC20
IMS	=SFC23	RSTR1	=SFC3A	CLREOP	=SFC42	CLEOP1	=SFC46
HOME	=SFC58	CR	=SFC62	LF	=SFC66	EPVID	=SFC73
CLRECL	=SFC99	CLREL2	=SFC98	CLREP	=SFC9D	WLT	=SFC8A
WAIT2	=SFC9F	WAIT3	=SFC9A	HET44	=SFCB4	NEXTAL	=SFCBA
RTS48	=SFC0B	HEADR	=SFC09	WRBIT	=SFC06	ZEROLY	=SFC08
ONEDL1	=SFC02	WRTAPE	=SFC05	ROBYTE	=SFC0C	ROBYTE2	=SFC0E
RD2BIT	=SFCFA	ROBIT	=SFCFD	IMO	=SFD0C	INI	=SFD0F
CURE50	=SFD12	DECBUF	=SFD18	SRZCUR	=SFD25	? INTC	=SFD25
PSCR	=SFD3B	POSCUR1	=SFD30	BSROUT	=SFD54	SRCUR	=SFD50
SROUT	=SFD58	JRBE50	=SFD64	GETLMZ	=SFD67	GETLM	=SFD6A
? GETLH1	=SFD6F	CLREJH	=SFD73	JIMS	=SFD7C	WFLH	=SFD82
? AODMP	=SFD86	? CROUT1	=SFD8B	CROUT	=SFD8E	PAR1	=SFD92
PRY22	=SFD96	XAMB	=SFDAB	NOBOCN	=SFDAD	XAM	=SFD93
DATAOUT	=SFD86	? RTS4C	=SFD05	XAMPN	=SFD0A	? TSTOP	=SFD0A
NO	=SFD01	PRBYTE	=SFD0A	? PRHEX	=SFD03	PRHEX2	=SFD05
COUT	=SFD0D	COUT1	=SFD0F	MENTAB	=SFD0F	SAVEADR	=SFD0F
? LOADADR	=SFD0D	BLI	=SFD0D	? BLANK	=SFD04	? STOR	=SFD0B
RTS5	=SFE17	? SETM40	=SFE18	SETM42	=SFE1D	? LT	=SFE20
LT2	=SFE22	MOVE	=SFE2C	VFY	=SFE36	VFYK	=SFE28

? MLIST	=SFE5E	LIST2	=SFE63	AIPC	=SFE75	ASPCLP	=SFE78
AIPCATS	=SFE7F	? SETIMV	=SFE80	SETNRN	=SFE84	SETDPLG	=SFE86
SETK80	=SFE89	? INPRT	=SFE88	? NIDPRT	=SFE8D	SETVID	=SFE93
? OUTPRT	=SFE95	? OUTPRT	=SFE97	IOFRT	=SFE98	IOFRT1	=SFEA7
IOFRT2	=SFEA9	? XBASIC	=SFE80	? BASCONT	=SFE83	? GO	=SFE86
? REGZ	=SFE8F	? TNET	=SFE82	? TNET	=SFE84	RTBL	=SFE85
? USER	=SFE84	WRITE	=SFE8D	WRI	=SFE84	WRBYTE	=SFE8D
WRBYT2	=SFE8F	? CRASH	=SFE86	READN	=SFE8D	READ2	=SFE8E
MR02	=SFE8A	RD3	=SFE86	PRRR	=SFE8D	BELL	=SFE8A
IORESTORE	=SFE8F	? IOSAVE	=SFE8A	SWD3	=SFE8D	HOW	=SFE8D
HOWZ	=SFE81	DN00	=SFE8C	NXTIEM	=SFE8F	CHRDRCH	=SFE76
CLREOL1	=SFE86	DIG	=SFE8D	NXTBIT	=SFE93	NXTBAS	=SFE98
NXTBAS0	=SFE85	GETNUM	=SFE8A	NXTCHR	=SFE8D	NXC00	=SFE81
COBRTS	=SFE89	CHGTBL	=SFE8A	SUBTBL	=SFE81		

## BIBLIOGRAFIA

Brown, P. J. - **Writing Interactive Compilers and Interpreters**. 3a ed., New York, John Wiley & Sons, [1981].

Leventhal, Lance A. - **6502 Assembly Language Programming**. 1a ed., Berkeley, Osborne/McGraw-Hill, [1979].

McIntire, Thomas C. - **Software Interpreters for Microcomputers**. 1a ed., New York, John Wiley & Sons, [1978].

Microdigital Eletrônica Ltda. - **Manual Técnico do TK 2000 Color**.



