

HEWLETT-PACKARD

Advanced Scientific Calculator

Manual do Proprietário



Atenção

Se você tiver dúvidas sobre o uso da calculadora e não encontrar a resposta neste manual, entre em contato com o

SERVIÇO DE ATENDIMENTO AO CLIENTE

Tesis Informática S/A

Al. Rio Negro, 750 — Alphaville

06400 — Barueri — S. Paulo

(011) 421-4619, das 8h as 12h

Ajude-nos a ajudá-lo!

Tomando um momento para preencher este cartão, você nos ajudará a melhor entender suas necessidades. Por favor, primeiro leia as questões e, então, preencha o cartão. Obrigado.

AJUDE-NOS A AJUDÁ-LO!

Modelo _____ Data de aquisição _____

Nome _____

Endereço _____

Cidade, Estado, CEP. _____

Idade _____ Telefone _____ Comercial _____ ou Residencial _____

1 Qual é a sua POSIÇÃO OU OCUPAÇÃO? (Por favor, marque apenas uma opção.)

- | | | |
|----------------------------|--|------------------|
| 101 estudante | 105 presidente, ger. geral | 109 autônomo |
| 102 professor, pesquisador | 106 proprietário, diretor, vice-presidente | 110 aposentado |
| 103 assessor | 107 representante externo | 111 outros _____ |
| 104 gerente de nível médio | 108 técnico | |

2 Qual é a sua AREA DE ATIVIDADE ou CAMPO DE TRABALHO/ESTUDO? (Por favor, marque apenas uma opção.)

- | | |
|----------------------------|---|
| 201 engenharia mecânica | 209 compras, programação de produção, controle de armazém |
| 202 engenharia civil | 210 contabilidade, auditoria |
| 203 engenharia elétrica | 211 finanças, análise de investimentos |
| 204 engenharia química | 212 administração |
| 205 engenharia _____ | 213 marketing |
| 206 topografia | 214 vendas |
| 207 processamento de dados | 215 assistência ao cliente, manutenção |
| 208 controle de qualidade | 216 outra _____ |

3 Em que INDÚSTRIA você trabalha? (Não responda esta se você for estudante ou aposentado. Por favor, marque apenas uma opção)

- | | |
|---|--|
| 301 educação | 310 química, refinação de petróleo |
| 302 banco comercial/invest, financeira | 311 agricultura, reflorest, agropastoral |
| 303 seguros | 312 processamento/distribuição de alimentos |
| 304 corretagem de imóveis | 313 fabricação de equipamentos industriais |
| 305 serviços comerciais/de consultoria | 314 fabricação de bens de consumo |
| 306 consultoria técnica | 315 transportes |
| 307 soft. serviços de proc. de dados | 316 empresas de telecomunicação gás, energia elétrica. |
| 308 construção, arquitetura | 317 administração pública, militar |
| 309 mineração, perfuração e expl. de petróleo | 318 outras _____ |

4 Onde você adquiriu sua calculadora HP? (Por favor, marque apenas uma opção.)

- | | |
|--|---|
| 401 revendedor de micro-computadores | 407 mala direta |
| 402 revendedor de equipamentos para escritório | 408 revendedor especializado (áudio, vídeo, fotografia) |
| 403 livraria/papelaria | 409 comprada pela cia./inst. de pesquisa ou ensino. |
| 404 loja de departamentos | 410 diretamente da TESIS |
| 406 catálogo | 411 outra _____ |

5 Como você ficou sabendo sobre este modelo?

- | | |
|--|---|
| 501 já era cliente HP | 505 mala direta |
| 502 informação de amigos, colegas, professor | 506 vendedor |
| 503 anúncio de jornal ou revista | 507 folheto ou literatura obtida na loja. |
| 504 artigos na imprensa | 508 outras _____ |

ISR 40-0574/87
UP AG. Barueri
DR/São Paulo

CARTÃO RESPOSTA COMERCIAL

Não é necessário selar

O selo será pago por

Tesis

Tesis Informática SA

06499 Barueri SP

Comentários sobre o Manual do Proprietário da HP-28S

Agradecemos antecipadamente o envio de sua avaliação. Seus comentários e sugestões serão de grande valia no aprimoramento de nossas publicações futuras.

Título e data de impressão do manual (constantes à página do título):

Solicitamos a gentileza de fazer um círculo em torno do conceito que melhor expressa sua opinião. Use o espaço denominado **Comentários** para acrescentar opiniões adicionais, se desejar.

1= Concordo plenamente 4= Não concordo
2= Concordo 5= Desaprovo fortemente
3= Neutro

- | | | | | | |
|--|---|---|---|---|---|
| ■ O manual está bem organizado | 1 | 2 | 3 | 4 | 5 |
| ■ Consigo encontrar a informação que desejo | 1 | 2 | 3 | 4 | 5 |
| ■ A informação do manual é precisa | 1 | 2 | 3 | 4 | 5 |
| ■ Consigo compreender as instruções com facilidade | 1 | 2 | 3 | 4 | 5 |
| ■ O layout e o formato do manual são úteis e atraentes | 1 | 2 | 3 | 4 | 5 |
| ■ As ilustrações são claras e elucidativas | 1 | 2 | 3 | 4 | 5 |

O conteúdo do manual é: () longo () adequado () insuficiente

Quais capítulos do Manual do Proprietário você utiliza com maior frequência?

Quais tópicos no Manual de Referência você utiliza mais? _____

Comentários: _____

Nome: _____

Endereço: _____

CEP/Cidade/Estado: _____

Profissão: _____

Mande as respostas à:

Tesis Informática S/A
Al. Rio Negro, 750 - Alphaville
06400 - Barueri - S. Paulo

ISR 40-0574/87
UP AG. Barueri
DR/São Paulo

CARTÃO RESPOSTA COMERCIAL

Não é necessário selar

O selo será pago por

Tesis

Tesis Informática SA

06499 Barueri SP

HP-28S

Calculadora Científica Avançada

Manual do Proprietário



Edição 1 agosto 1988

Número do Manual para Pedidos Avulsos: 00028-90095

AVISO

Para informações sobre garantia e normas de segurança para esta calculadora, veja páginas 289 e seguintes.

Este manual e os programas aqui contidos são fornecidos na forma em que se encontram e estão sujeitos a modificações sem aviso prévio. A Tesis e a Hewlett-Packard Company não oferecem nenhum tipo de garantia com respeito a este manual ou aos programas aqui contidos. As empresas acima não poderão ser responsabilizadas pelos erros nem por danos decorrentes da utilização do material descrito.

©Hewlett-Packard Co. 1987. Este manual possui informações de propriedade das empresas acima, protegidas por direitos autorais. Todos os direitos estão reservados. Este documento e os programas nele contidos não podem ser copiados, adaptados ou traduzidos para outra língua, no seu todo ou em parte, sem autorização prévia, por escrito, das empresas acima.

Os direitos sobre os programas que controlam sua calculadora também são reservados pela Tesis e Hewlett-Packard Company. Não é permitida a reprodução, a adaptação ou tradução desses programas sem autorização prévia, por escrito, das empresas mencionadas.

Histórico da Tiragem

Edição 1

agosto 1988

Nº de fabr: 00028-90096

Bem-vindo à HP-28S

Parabéns! Com a HP-28S, você pode facilmente resolver problemas complicados, incluindo problemas que você não poderia resolver em uma calculadora anteriormente. A HP-28S combina poderosa capacidade de computação numérica com uma nova dimensão - *computação simbólica*. Você pode formular um problema simbolicamente, encontrar uma solução simbólica que mostra o comportamento global do problema e obter resultados numéricos da solução simbólica.

A HP-28S oferece os seguintes recursos:

- manipulação algébrica. Você pode desenvolver, agrupar ou rearranjar termos de uma expressão e você pode, simbolicamente, resolver uma equação para uma variável.
- cálculo diferencial e integral. Você pode calcular derivadas, integrais indefinidas e integrais definidas.
- soluções numéricas. Utilizando o HP Solve, na HP-28S, você pode resolver uma expressão ou equação para qualquer variável. Você também pode resolver um sistema de equações lineares. Com múltiplos tipos de dados, você pode utilizar números complexos, vetores e matrizes tão facilmente como números reais.
- traçado de gráficos. Você pode traçar gráficos de expressões, equações e dados estatísticos.
- conversão de unidades. Você pode converter entre quaisquer combinações de 120 unidades equivalentes incorporadas à máquina. Também pode definir suas próprias unidades.
- estatística. Você pode calcular estatísticas com amostras simples ou com dados emparelhados e probabilidades.
- bases de números binários. Você pode calcular com números binários, octais e hexadecimais e executar manipulação de bits.
- entrada direta de fórmulas algébricas, mais lógica RPN para cálculos interativos.

O *Manual do Proprietário da HP-28S* (este manual) contém três partes. A parte 1, "Princípios Básicos", demonstra como trabalhar com problemas simples. A parte 2, "Sumário dos Recursos da Calculadora", baseia-se na parte 1 para auxiliá-lo a aplicar os exemplos em seus próprios problemas. A parte 3, "Programação", descreve recursos de programação e demonstra-os em uma série de exemplos de programação.

O *Manual de Referência da HP-28S* fornece informações detalhadas sobre os comandos. Este é um dicionário de menus, descrevendo os conceitos e comandos para cada menu.

Recomendamos que primeiro você execute os exemplos da parte 1 do Manual do Proprietário, para familiarizar-se com a calculadora e, então, passe à parte 2 para obter um entendimento mais amplo de suas operações. Quando desejar saber mais sobre algum comando específico, procure no Manual de Referência. Quando desejar aprender sobre programação, leia a parte 3 do Manual do Proprietário.

Estes manuais mostram como utilizar a HP-28S para efetuar cálculos matemáticos, mas eles não ensinam matemática. Pressupomos que você já esteja familiarizado com os princípios matemáticos relevantes. Por exemplo, para utilizar os recursos de cálculos diferencial e integral da HP-28S efetivamente, você precisa conhecer cálculo diferencial e integral elementar.

Por outro lado, você não necessita entender todos os tópicos matemáticos na HP-28S para utilizar as partes que lhe interessam. Por exemplo, você não necessita entender cálculo diferencial e integral para utilizar a capacidade estatística.

ÍNDICE

| | |
|-----------|----------------------------------|
| 15 | Como Utilizar Este Manual |
| 15 | O Que Este Manual Contém |
| 16 | Para Maiores Informações |

Parte 1: Princípios Básicos

| | | |
|----------|-----------|---|
| 1 | 18 | Para Iniciar |
| | 18 | Informações Preliminares |
| | 18 | Abrindo e Fechando o Estojo |
| | 19 | Localizando a Tampa da Bateria e a Saída de Impressão |
| | 20 | Ligando e Desligando a HP-28S |
| | 20 | Apagando Toda a Memória (Inicializando a Memória) |
| | 21 | Ajustando o Contraste do Visor |
| | 21 | Cálculos no Teclado |
| | 25 | Uma Visão Geral da Calculadora |
| | 25 | Principais Recursos e Conceitos |
| | 31 | O Catálogo de Comandos |
| 2 | 34 | Efetuando Cálculos Aritméticos |
| | 36 | Entrando e Apresentando Números |
| | 36 | Mudando o Ponto Decimal |
| | 37 | Selecione a Vírgula Como a Marca de Raiz |
| | 39 | Digitando Números |
| | 40 | Funções de Um Número |
| | 41 | Funções de Dois Números |
| | 41 | Adição e Subtração |
| | 41 | Multiplicação e Divisão |
| | 42 | Potências e Raízes |
| | 43 | Porcentagens |

- 43 Trocando os Níveis 1 e 2
- 44 Apagando Objetos da Pilha
- 45 Cálculos Encadeados
- 47 Se Você Executar a Função Errada

3

- 48 **Utilizando Variáveis**
- 48 Introdução a Variáveis
- 49 Criando uma Variável Numérica
- 50 Recuperando uma Variável Numérica
- 50 Processando uma Variável Numérica
- 51 Mudando o Valor de uma Variável
- 52 Eliminando uma Variável
- 52 Mudando o Nome de uma Variável
- 54 Criando o Conteúdo da Variável
- 56 Recuperando uma Variável Programa
- 56 Processando uma Variável Programa
- 57 Nomes com Aspas e sem Aspas

4

- 58 **Repetindo Cálculos**
- 58 Criando uma Expressão
- 60 Criando um Diretório
- 63 Utilizando o Solver Para Repetir um Cálculo
- 66 Utilizando um Conjunto Diferente de Valores
- 68 Utilizando uma Expressão Diferente
- 71 Voltando ao Diretório HOME
- 72 Sumário

5

- 73 **Funções de Números Reais**
- 73 Funções Trigonométricas
- 73 Selecionando Medidas Angulares
- 74 Utilizando π
- 76 Convertendo Medidas Angulares
- 77 Funções Logarítmicas, Exponenciais e Hiperbólicas
- 78 Outras Funções Reais
- 79 Definindo Novas Funções

| | | |
|-----------|------------|--|
| 6 | 82 | Funções Com Números Complexos |
| | 82 | Utilizando Números Complexos |
| | 84 | Utilizando Coordenadas Polares |
| | 86 | Uma Função do Usuário para Adição Polar |
| 7 | 89 | Traçando Gráficos |
| | 91 | Imprimindo um Gráfico |
| | 91 | Mudando a Escala do Gráfico |
| | 93 | Transladando o Traçado de um Gráfico |
| | 94 | Redefinindo os Extremos do Gráfico |
| | 97 | Traçando Gráficos de Equações |
| 8 | 98 | O Solver |
| | 98 | Encontrando um Zero de uma Expressão |
| | 100 | Encontrando um Mínimo ou Máximo |
| | 103 | Valor do Dinheiro no Tempo |
| 9 | 107 | Soluções Simbólicas |
| | 107 | Encontrando os Zeros de uma Expressão Quadrática |
| | 109 | Isolando uma Variável |
| | 110 | Desenvolvendo e Agrupando |
| | 112 | Utilizando FORM |
| 10 | 117 | Cálculo |
| | 117 | Diferenciando uma Expressão |
| | 118 | Diferenciação Passo-a-Passo |
| | 120 | Diferenciação Completa |
| | 120 | Integrando uma Expressão |
| | 121 | Integração Simbólica de Pólinômio |
| | 122 | Integração Numérica de Expressões |

| | | |
|-----------|------------|--|
| 11 | 124 | Vetores e Matrizes |
| | 124 | Vetores |
| | 124 | Entrando um Vetor |
| | 125 | Multiplicando e Dividindo um Vetor por um Número |
| | 125 | Adicionando e Subtraindo Vetores |
| | 126 | Calculando o Produto Vetorial |
| | 126 | Calculando o Produto Escalar |
| | 126 | Matrizes |
| | 127 | Entrando uma Matriz |
| | 127 | Vendo uma Matriz Grande |
| | 128 | Invertendo uma Matriz |
| | 128 | Calculando o Determinante |
| | 128 | Multiplicando Dois Arranjos |
| | 128 | Multiplicando Duas Matrizes |
| | 129 | Multiplicando uma Matriz e um Vetor |
| | 130 | Resolvendo um Sistema de Equações Lineares |
| 12 | 131 | Estatísticas |
| | 132 | Entrando Dados |
| | 133 | Editando Dados |
| | 134 | Estatísticas com uma Única Amostra |
| | 134 | Calculando a Média |
| | 135 | Calculando o Desvio Padrão |
| | 135 | Calculando a Variância |
| | 135 | Estatísticas com Dados Emparelhados |
| | 136 | Especificando um Par de Colunas |
| | 136 | Calculando a Correlação |
| | 136 | Calculando a Covariância |
| | 137 | Calculando a Regressão Linear |
| | 137 | Efetuando Projeções |
| 13 | 138 | Aritmética Binária |
| | 138 | Selecionando o Tamanho da Palavra |
| | 139 | Selecionando a Base |
| | 139 | Entrando Inteiros Binários |
| | 140 | Calculando com Inteiros Binários |

| | | |
|-----------|------------|--|
| 14 | 141 | Conversões de Unidades |
| | 141 | O Catálogo UNITS |
| | 143 | Convertendo Unidades |
| | 144 | Convertendo Cadeias de Unidades |
| | 146 | Verificando se as Unidades estão Corretas |
| | 147 | Função do Usuário para Conversão de Unidades |
| 15 | 149 | Imprimindo |
| | 149 | Imprimindo o Visor |
| | 150 | Imprimindo um Registro Contínuo |
| | 151 | Imprimindo o Nível 1 |
| | 152 | Imprimindo a Pilha |
| | 152 | Imprimindo uma Variável |

Parte 2: Sumário dos Recursos da Calculadora

| | | |
|-----------|------------|--------------------------------------|
| 16 | 154 | Objetos |
| | 155 | Números Reais |
| | 155 | Números Complexos |
| | 156 | Inteiros Binários |
| | 156 | Cadeias |
| | 157 | Arranjos |
| | 158 | Listas |
| | 159 | Nomes |
| | 160 | Programas |
| | 161 | Algébricos |
| | 161 | Expressões |
| | 162 | Equações |
| | 163 | Constantes Simbólicas |
| 17 | 164 | Operações, Comandos e Funções |

| | | |
|-----------|------------|--|
| 18 | 166 | A Linha de Comando |
| | 166 | O Menu do Cursor |
| | 168 | Algumas Teclas de Entrada |
| | 169 | Delimitadores de Objetos e Separadores |
| | 169 | Modos de Entrada |
| | 171 | Exceções |
| | 171 | Seleção Manual |
| | 172 | Como o Cursor Indica os Modos |
| | 173 | Executando a Linha de Comando |
| | 173 | Editando Objetos Existentes |
| | 174 | Recuperando Linhas de Comando |
| | 175 | A Linha de Comando como uma Cadeia |
| | | |
| 19 | 176 | A Pilha |
| | 176 | Revisão de Conceitos da Pilha |
| | 177 | Vendo a Pilha |
| | 177 | Manipulando a Pilha |
| | 179 | Variáveis Locais |
| | 179 | Recuperando os Últimos Argumentos |
| | 180 | Restaurando a Pilha |
| | 181 | A Pilha como uma Lista |
| | | |
| 20 | 182 | Memória |
| | 182 | Memória do Usuário |
| | 182 | Variáveis Globais |
| | 183 | Diretórios |
| | 187 | Recursos para Recuperação |
| | 188 | Pouca Memória |
| | 190 | Maximizando o Desempenho |
| | | |
| 21 | 192 | Menus |
| | 193 | Menus de Comandos |
| | 194 | Menus de Operações |
| | 194 | Menus de Variáveis |
| | 195 | Menus Personalizados |

| | | |
|-----------|------------|--|
| 22 | 196 | Catálogo de Comandos |
| | 197 | Encontrando um Comando |
| | 197 | Verificando a Utilização de um Comando |
| 23 | 198 | Processamento |
| | 199 | Objetos Classe-Dados |
| | 199 | Objetos Classe-Nome |
| | 200 | Processamento de Nomes Locais |
| | 200 | Processamento de Nomes Globais |
| | 201 | Objetos Classe-Procedimento |
| | 201 | Processamento de Programas |
| | 202 | Processamento de Algébricos |
| | 203 | Processamento de Funções |
| 24 | 205 | Modos |
| | 205 | Modos Gerais |
| | 207 | Modos de Entrada e Apresentação no Visor |
| | 210 | Modos de Recuperação |
| | 211 | Exceções Matemáticas |
| | 212 | Modos de Impressão |
| 25 | 215 | Operações de Sistema |
| | 216 | Imprimindo o Visor |
| | 216 | Controle de Comando |
| | 216 | Operações de Apagar |
| | 216 | Atenção |
| | 217 | Parada de Sistema |
| | 217 | Inicializando a Memória |

| | |
|------------|--------------------|
| 218 | Operações de Teste |
| 218 | Teste Contínuo |
| 219 | Teste de Teclado |

Parte 3: Programação

| | | |
|-----------|------------|---|
| 26 | 222 | Estruturas de Programas |
| | 222 | Estruturas de Variável Local |
| | 223 | Estruturas Condicionais |
| | 226 | IF...THEN...ELSE ...END |
| | 226 | IFTE (Função If-Then-Else-End) |
| | 227 | IF...THEN...END |
| | 227 | IFT (Comandos If-Then-End) |
| | 227 | Armadilhas de Erro |
| | 228 | Estruturas de "Loop" Definido |
| | 228 | START...NEXT |
| | 229 | FOR <i>counter</i> ...NEXT |
| | 230 | ... <i>incremento</i> STEP |
| | 231 | Estruturas de "Loop" Indefinido |
| | 231 | DO...UNTIL...END |
| | 232 | WHILE ...REPEAT...END |
| | 233 | Estruturas de Programa Superpostas |
| 27 | 234 | Programas Interativos |
| | 234 | Solicitando uma Entrada |
| | 235 | Solicitando uma Escolha |
| | 235 | Um Exemplo Mais Complicado |
| 28 | 240 | Exemplos de Programas |
| | 241 | Funções Caixa |
| | 241 | BOXS (Superfície de uma Caixa) |
| | 244 | BOXS Sem Variáveis Locais |
| | 245 | BOXR (Relação entre Superfície e o Volume de uma Caixa) |
| | 246 | Números de Fibonacci |
| | 247 | FIB1 (Números de Fibonacci, Versão Recursiva) |
| | 248 | FIB2 (Números de Fibonacci, Versão "Loop") |
| | 249 | Comparação de FIB1 e FIB2 |
| | 250 | Execução Passo-a-Passo |

- 253** Desenvolvendo e Agrupando Completamente
- 253** MULTI (Execução Múltipla)
- 255** EXCO (Desenvolvendo e Agrupando Completamente)
- 257** Apresentando no Visor um Inteiro Binário
- 257** PAD (Preenche com Espaços no Início)
- 258** PRESERVE (Salva e Restabelece no Programa BDISP)
- 259** BDISP (Apresentação de Binário no Visor)
- 262** Estatísticas com Somatórios
- 263** SUMS (Matriz de Somatórios Estatísticos)
- 265** Σ GET (Obtém um Elemento de Σ COV)
- 266** Σ X2 (Soma dos Quadrados de x)
- 266** Σ Y2 (Soma dos Quadrados de y)
- 267** Σ XY (Soma dos Produtos de x e y)
- 270** Mediana de Dados Estatísticos
- 270** SORT (Classifica uma Lista)
- 272** LMED (Mediana de uma Lista)
- 273** MEDIAN (Mediana de Dados Estatísticos)
- 275** Mudando de Diretórios
- 276** UP (Muda para um Diretório-Pai)
- 277** DOWN (Muda para um Subdiretório)

Apêndices e Índices

| | | |
|----------|------------|---|
| A | 282 | Atendimento ao Usuário, Baterias e Assistência Técnica |
| | 282 | Respostas a Perguntas Frequentes |
| | 286 | Baterias |
| | 289 | Manutenção da Calculadora |
| | 289 | Limites Ambientais |
| | 289 | Determinando se a Calculadora Necessita de Reparos |
| | 291 | Garantia Limitada de Um Ano |
| | 293 | Se a Calculadora Necessitar de Reparos |
| | 295 | Informações Sobre a Assistência Técnica Internacional |
| B | 296 | Notas Para Usuários de Calculadoras HP RPN |
| C | 302 | Notas Para Usuários de Calculadoras Algébricas |
| D | 306 | Diagramas dos Menus |
| | 327 | Índice das Teclas |
| | 332 | Índice por Assunto |

Como Utilizar Este Manual

Se você dispõe de tempo e tem inclinação para isso, você pode ler este manual da primeira à última página, executando todos os exemplos. Caso contrário, recomendamos o seguinte método para iniciar:

1. leia os primeiros cinco capítulos na parte 1, "Princípios Básicos", para familiarizar-se com a calculadora.
2. existem dois apêndices que comparam a HP-28S com outros tipos de calculadoras.
 - Se você está familiarizado com outras calculadoras Hewlett-Packard que utilizam RPN, leia o apêndice B, "Notas para Usuários de Calculadoras HP RPN", à página 296.
 - Se você está familiarizado com calculadoras que utilizam alguma forma de entrada algébrica, leia o apêndice C, "Notas para Usuários de Calculadoras Algébricas", à página 302.
3. se você estiver interessado em um tópico descrito mais adiante na parte 1, você pode ir à frente e tentar os exemplos naquele capítulo.

O Que Este Manual Contém

A parte 1, "Princípios Básicos", demonstra como executar alguns problemas simples. Ao resolvê-los, você estará aprendendo os princípios básicos das operações da HP-28S, tipos de objetos e menus.

A parte 2, "Sumário dos Recursos da Calculadora", acrescenta informações às que você aprendeu na parte 1. Esta parte apresenta mais detalhes sobre a utilização da calculadora, incluindo opções e recursos não discutidos na parte 1. Utilizando a parte 2, você pode extrapolar os exemplos da parte 1 para resolver seus próprios problemas.

A parte 3, "Programação", descreve os recursos de programação da HP-28S. O último capítulo, "Exemplos de Programação", contém uma série de programas curtos que demonstram técnicas de programação.

Para Maiores Informações

À medida que você executa os exemplos deste manual, poderá ter dúvidas quanto aos recursos demonstrados ou mencionados nos exemplos. Tanto este manual como o Manual de Referência contêm informações adicionais.

- Se você encontrar dificuldades, veja "Respostas a Perguntas Frequentes", à página 282.
- Para uma breve descrição do que cada tecla faz, veja "Índice das Teclas", à página 327.
- Para uma breve descrição dos comandos em cada menu, veja o apêndice D, "Diagramas dos Menus", à página 306.
- Para informações detalhadas sobre um menu, consulte o Manual de Referência. Todos os menus (mais alguns tópicos adicionais) aparecem em ordem alfabética. O conteúdo do dicionário está descrito na última capa do Manual de Referência.
- Para informações detalhadas a respeito de um comando específico, procure no "Índice de Operações", à página 381 do Manual de Referência, onde você encontrará referência a um elemento do dicionário (usualmente um menu) e da página para o comando particular.

Parte 1

Princípios Básicos

| | | |
|---------------|------------|--|
| Página | 18 | 1: Para Iniciar |
| | 34 | 2: Efetuando Cálculos Aritméticos |
| | 48 | 3: Utilizando Variáveis |
| | 58 | 4: Repetindo Cálculos |
| | 73 | 5: Funções Reais |
| | 82 | 6: Funções Complexas |
| | 89 | 7: Traçando Gráficos |
| | 98 | 8: O Solver |
| | 107 | 9: Soluções Simbólicas |
| | 117 | 10: Cálculo |
| | 124 | 11: Vetores e Matrizes |
| | 131 | 12: Estatística |
| | 138 | 13: Aritmética Binária |
| | 141 | 14: Conversões de Unidades |
| | 149 | 15: Imprimindo |

Para Iniciar

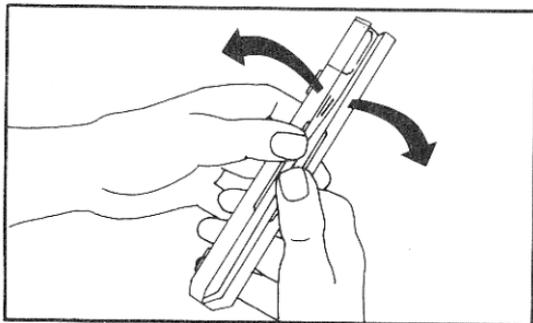
Este capítulo descreve os recursos básicos da calculadora e demonstra um cálculo simples. A seguir, uma ilustração do teclado com indicações salienta as principais características do teclado e visor. Por fim, você aprenderá sobre o catálogo de comandos, que é um guia prático dos comandos, e como utilizá-los.

Informações Preliminares

Esta seção descreve as características básicas da calculadora.

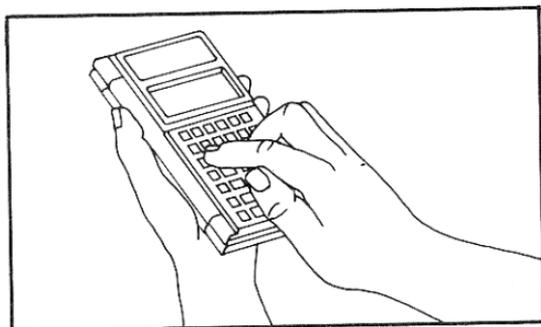
Abrindo e Fechando o Estojo

A calculadora forma seu próprio estojo, abrindo e fechando como um livro. Para abrir a calculadora, segure-a com o fecho voltado para você e abra-a com seus polegares.



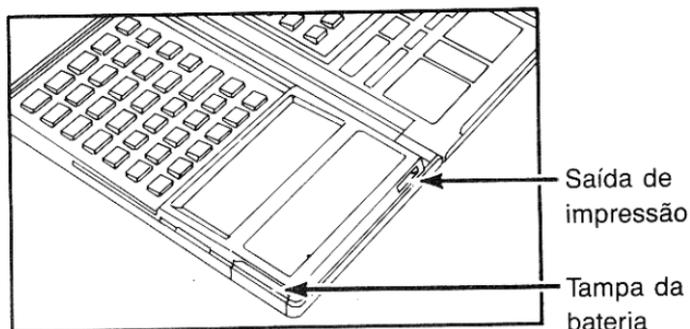
Para fechar a calculadora, dobre os dois lados como se estivesse fechando um livro e pressione até ouvir um "clac".

Você pode dobrar o lado esquerdo da calculadora para trás completamente até encostá-lo no lado direito. Esta é uma maneira cômoda para trabalho de campo quando você quer segurar a calculadora com uma mão e operá-la com a outra ou, simplesmente, para economizar espaço em sua mesa.



Localizando a Tampa da Bateria e a Saída de Impressão

Com a calculadora aberta, note a localização da tampa da bateria e da saída de impressão.



A HP-28S é alimentada por três baterias alcalinas tipo N. A calculadora é vendida com baterias. Se estas não estiverem instaladas, siga as instruções que se iniciam à página 286.

A HP-28S, quando usada com a impressora, envia informações através de um sinal infra-vermelho. Este sinal é emitido da saída de impressão da calculadora e recebido pela impressora. As operações da impressora são descritas no capítulo 15.

Ligando e Desligando a HP-28S

Pressione para ligar a calculadora. A HP-28S possui *memória contínua* de tal forma que todos os dados na calculadora, incluindo-se o conteúdo do visor, permanecem inalterados desde a última vez que você a utilizou.

Enquanto a calculadora está ligada, atua como a tecla ATTN (*atenção*), conforme impresso em branco abaixo da tecla. Pressionar apaga qualquer texto que você digitou e interrompe programas.

Pressione para desligar a calculadora. (“Pressione ” significa “pressione a tecla de prefixo , então pressione a tecla com OFF impressa acima dela.”)

Se a calculadora ficar inativa em torno de 10 minutos, automaticamente é desligada para conservar a bateria. Pressione para ligá-la novamente.

Apagando Toda a Memória (Inicializando a Memória)

Você pode retornar a calculadora ao estado original, inicializando a memória. Nesse caso, toda informação se perde. Quaisquer modos que você tenha alterado (formato de apresentação dos números no visor, modo angular, etc) retornam as suas definições *padrão*.

Para Inicializar a Memória:

1. pressione e mantenha pressionada .
2. pressione e mantenha pressionada (no canto superior esquerdo do teclado da direita).
3. pressione e solte (no canto superior direito do teclado da direita).
4. solte .
5. solte .

A calculadora emite um sinal audível e apresenta no visor a mensagem MEMORY LOST. A mensagem automaticamente desaparece quando você pressiona qualquer tecla.

Caso você tenha iniciado o processo para inicializar a memória e mude de idéia, continue pressionando **ON**, enquanto pressiona **DEL** (na fileira superior, ao lado de **INS**) e, então, solte **ON**. Pressionando **DEL**, a seqüência de inicialização é cancelada.

Ajustando o Contraste do Visor

Você pode ajustar o contraste do visor para compensar os diversos ângulos de visão e intensidade luminosa.

Para ajustar o contraste:

1. pressione e mantenha pressionada **ON**.
2. pressione **+** uma ou mais vezes para escurecer o visor ou pressione **-** uma ou mais vezes para clarear o visor.
3. solte **ON**.

Cálculos no Teclado

Tente o seguinte cálculo.

$$(15 + 23) \times \text{sen } 30^\circ$$

Os passos básicos são os mesmos que você executaria se estivesse usando lápis e papel. Primeiro, você calculará $15 + 23$, que produzirá um resultado intermediário. A seguir, calculará $\text{sen } 30^\circ$, o qual produzirá o outro resultado intermediário. Finalmente, você combinará os resultados intermediários para obter a resposta.

Se você cometer um erro enquanto estiver digitando um número, você pode:

- pressionar **◀** para apagar o último dígito entrando.
- pressionar **ON** para apagar todos os dígitos entrados.

Inicie com uma folha de papel em branco.

```
4:
3:
2:
1:
```

O visor mostra a sua área de trabalho, que é denominada *pilha*. Neste momento, a pilha está vazia.

Pressione para escrever 15 na *linha de comando*.

15

```
3:
2:
1:
15
```

Note que a pilha se move para cima, abrindo espaço para a linha de comando, de modo que apenas três níveis da pilha são apresentados no visor.

Coloque 15 na pilha.

```
4:
3:
2:
1: 15
```

O número vai para o *nível 1 da pilha*, como indicado pelo 1: à esquerda. Note que a linha de comando desaparece, assim quatro níveis da pilha são apresentados no visor novamente.

Pressione para escrever 23 na linha de comando.

23

```
3:
2:
1: 15
23
```

Coloque 23 no nível 1.

```
4:
3:
2: 15
1: 23
```

O número 15, que estava no nível 1, é elevado para o nível 2.

Adicione 15 e 23.

```
4:
3:
2:
1: 38
```

Os números 15 e 23 são removidos da pilha e sua soma, 38, é devolvida ao nível 1. Você deixará este resultado intermediário na pilha, enquanto calcula o segundo resultado intermediário.

Para calcular $\sin 30^\circ$, você utilizará o menu TRIG (trigonometria).

```
3:
2:
1: 38
SIN ASIN COS ACOS TAN ATAN
```

A linha inferior do visor mostra seis comandos no menu TRIG. Os seis rótulos de menu (SIN até ATAN) definem as seis teclas do menu (as teclas imediatamente abaixo do visor).

Pressione para escrever 30 na linha de comando.

30

```
2:
1: 38
30
SIN ASIN COS ACOS TAN ATAN
```

Coloque 30 no nível 1.

```
3:
2: 38
1: 30
SIN ASIN COS ACOS TAN ATAN
```

O resultado anterior, $15 + 23 = 38$, é elevado ao nível 2.

Calcule $\sin 30^\circ$.

```
3:
2: 38
1: .5
SIN ASIN COS ACOS TAN ATAN
```

O número 30 é removido do nível 1 e o seu seno, .5, é devolvido ao nível 1 em seu lugar. O resultado anterior, 38, permanece no nível 2.

Calcule $38 \times .5$.



Os números 38 e .5 são removidos dos níveis 1 e 2 e o seu produto, 19, é devolvido ao nível 1.

Este passo completa o cálculo:

$$(15 + 23) \times \text{sen } 30^\circ = 19.$$

Para sumarizar, damos a seguir um procedimento geral para o cálculo que você acabou de completar.

1. Digite um número na *linha de comando*.
2. Pressione para colocar o número na *pilha*.
3. Pressione a tecla que executa o comando. (Se o comando não aparece no teclado, selecione o *menu* que contém o comando e pressione a *tecla do menu* abaixo do *rótulo do menu* apropriado.)

O exemplo anterior demonstrou que todos os cálculos ocorrem na pilha. Para ressaltar esta idéia, você pressionou para colocar cada número na pilha. Na prática, *you precisa pressionar somente para separar dois números digitados seqüencialmente* — como no exemplo, para separar 15 e 23. Tente repetir o exemplo, omitindo o segundo e terceiro s.

O estilo de cálculo ilustrado acima, no qual você entra números na pilha, antes de executar as funções matemáticas, é denominado RPN (*Reverse Polish Notation = Notação Polonesa Reversa*), notação pós-fixada, ou lógica de pilha ("stack logic"). Quase todos os comandos da HP-28S, não apenas os cálculos, utilizam lógica RPN. Este sistema usa duas regras simples:

- as entradas requeridas por uma função, denominadas os *argumentos* da função, precisam estar na pilha antes que a função seja executada.
- os resultados de uma função são devolvidos à pilha, onde eles estão disponíveis como argumentos para a próxima função.

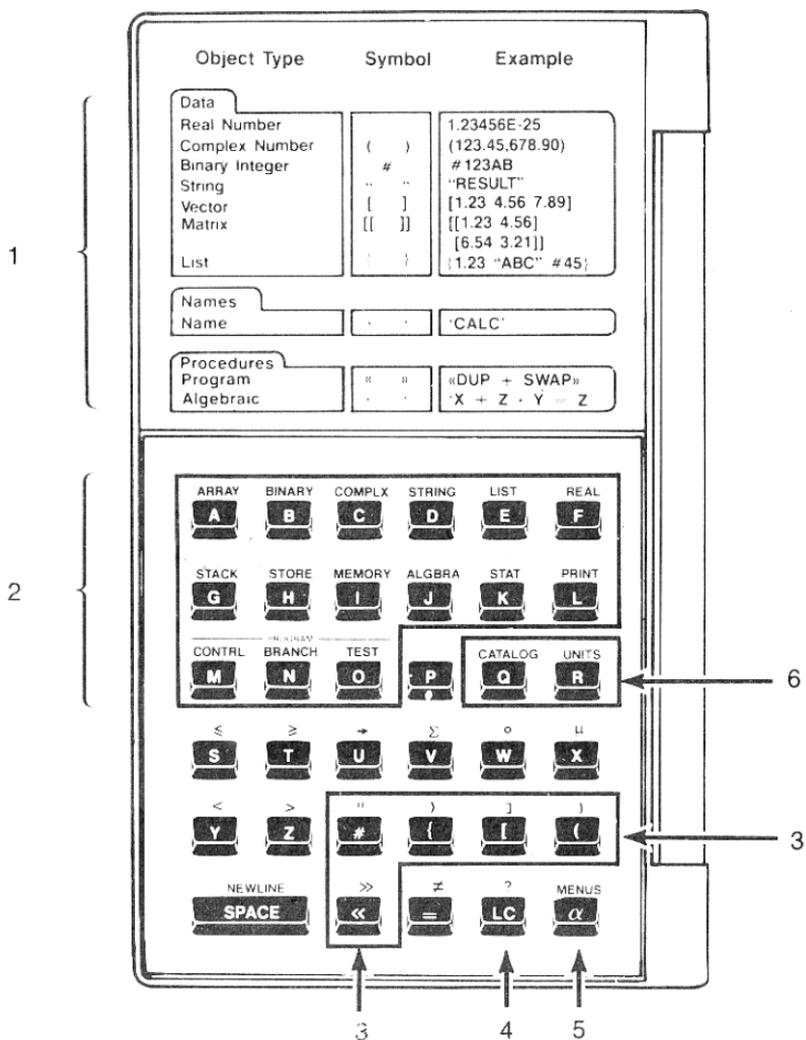
Você também pode calcular entrando uma expressão em forma algébrica, como ela poderia aparecer escrita num livro. No próximo capítulo, você executará o mesmo cálculo acima, utilizando uma expressão algébrica.

Uma Visão Geral da Calculadora

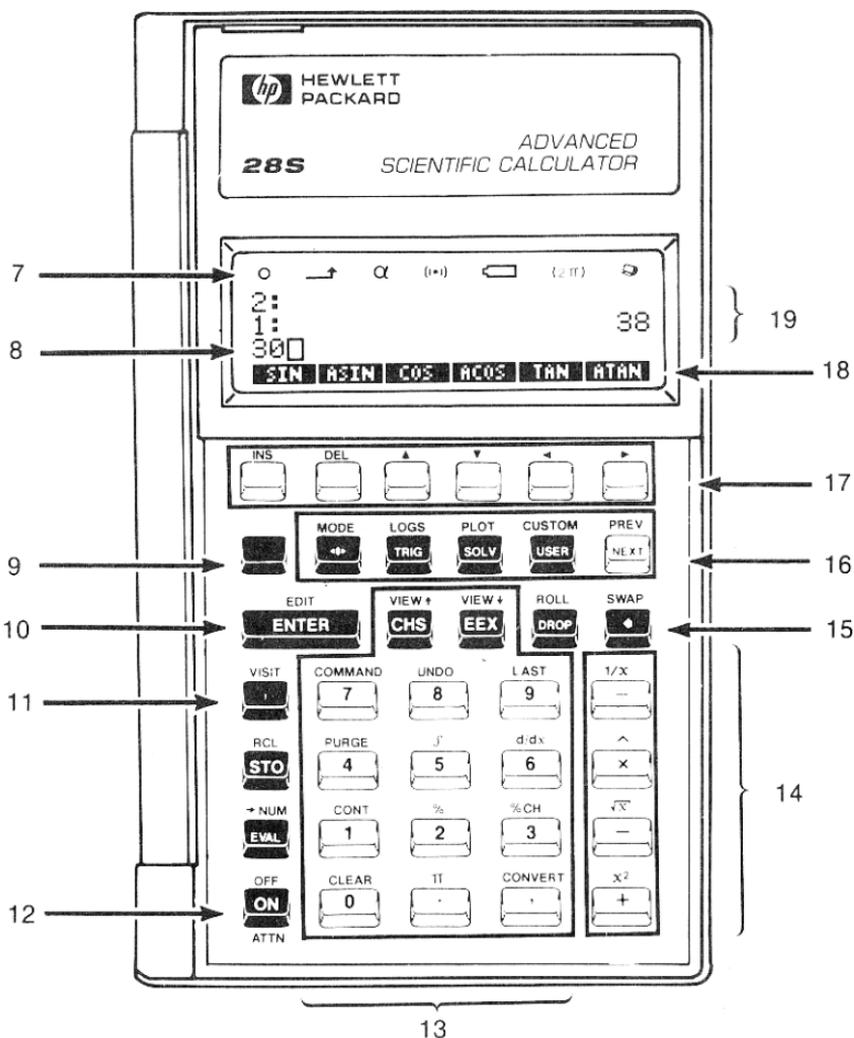
Esta seção ressalta alguns dos principais recursos da calculadora, incluindo um catálogo de comandos que relaciona e descreve cada comando.

Principais Recursos e Conceitos

As ilustrações nas páginas 26 e 27 mostram o teclado da calculadora e o visor, identificando recursos importantes. Os números nas descrições a seguir correspondem aos números nas ilustrações.



1. Tipos e formatos de objetos
2. Seleção de menu (com tecla de prefixo)
3. Delimitadores de objetos
4. Letras minúsculas
5. Modo de entrada
6. Listas de comandos e unidades (com tecla de prefixo)



- | | |
|---|--|
| 7. Anúncios | 13. Entrada de números |
| 8. Linha de comando | 14. Aritmética |
| 9. Tecla de prefixo | 15. Retrocesso (volta uma posição) |
| 10. Entra linha de comando | 16. Seleção de menu, próxima linha de menu |
| 11. Delimitador para objetos simbólicos | 17. Teclas do menu |
| 12. Liga e desliga; apaga linha de comando; interrompe programa | 18. Rótulos de menu |
| | 19. Níveis da pilha |

1. Tipos e formatos de objetos. Esta tabela mostra os delimitadores corretos e os exemplos para os 10 tipos básicos de objetos. Um “objeto” é qualquer um dos itens individuais que você opera na calculadora. Os 10 tipos básicos de objetos são:

- números reais, tais como 5 ou -4.3×10^{15} .
- números complexos, que são um par de números reais representando um número complexo $x + iy$ ou um ponto em um plano.
- inteiros binários, que são inteiros sem sinal utilizados em ciência de computação.
- cadeias (strings), que contêm seqüências arbitrárias de caracteres.
- vetores, que são arranjos (arrays) uni-dimensionais, utilizados em álgebra linear.
- matrizes, que são arranjos (arrays) bi-dimensionais, utilizados em álgebra linear.
- listas, que contêm seqüências arbitrárias de objetos.
- nomes, que possibilitam a você atribuir nomes e armazenar outros objetos e executar cálculos simbólicos.
- programas, que permitem que você crie seus próprios comandos.
- algébricos, que representam expressões matemáticas e equações.

2. Seleção de menu (com tecla de prefixo). Utiliza as teclas de seleção de menu para associar comandos às teclas do menu. Por exemplo, pressione  ARRAY para selecionar o menu ARRAY. Para selecionar um menu diferente, pressione outra tecla de seleção de menu.

Existem teclas adicionais de seleção de menu no teclado da direita (ver item 16).

3. Delimitadores de objetos. Estes símbolos identificam os diferentes tipos de objetos (ver item 1). Por exemplo,  identifica inteiros binários, enquanto  e  identificam programas.

Números reais não requerem delimitadores. Objetos simbólicos (nomes e algébricos) requerem o delimitador , localizado no teclado da direita (ver item 11).

4. Letras minúsculas. Pressione  para digitar letras minúsculas. O modo letras minúsculas continua até que você pressione  uma segunda vez, ou pressione  para processar a linha de comando ou pressione  para apagar a linha de comando.

5. Modo de entrada. A linha de comando tem três modos de entrada, cada um adequado para entrar certos tipos de objetos. Modos de entrada mudam automaticamente à medida que você digita os objetos mas, algumas vezes, você pode desejar controle manual; então a tecla α permite que você selecione o modo de entrada desejado.

6. Listas de comandos e unidades (com tecla de prefixo). Pressione \blacksquare $\boxed{\text{CATALOG}}$ para uma listagem de todos os comandos da HP-28S e seus argumentos necessários (página 31). Pressione \blacksquare $\boxed{\text{UNITS}}$ para listagem das unidades reconhecidas na conversão de unidades (página 141).

7. Anúncios. Os anúncios indicam o estado da calculadora. Quando um anúncio é visível, ele indica o seguinte:

| Anúncio | Significado |
|--|--|
|  | Programa suspenso. |
|  | A tecla de prefixo \blacksquare foi pressionada. |
| α | Modo de entrada Algébrico. |
| $((\bullet))$ | Ocupada, não está pronta para entrada. |
|  | Bateria fraca. |
| (2π) | Modo radiano. |
|  | Enviando dados para impressora. |

8. Linha de comando. O texto que você digita vai para a linha de comando.

9. Tecla de prefixo. Pressione a tecla de prefixo colorida \blacksquare para executar um comando impresso em cor acima da tecla.

10. Entrar linha de comando. Pressione $\boxed{\text{ENTER}}$ para processar o texto na linha de comando.

11. Delimitador para objetos simbólicos. Delimitadores são pontuações que identificam tipos de objetos; objetos simbólicos são nomes e algébricos. Para digitar um objeto simbólico, pressione $\boxed{\text{'}}$ no início e, quando necessário, no fim do objeto.

Números reais não requerem delimitadores. Os delimitadores para outros tipos de objetos estão no teclado da esquerda (ver itens 1 e 3).

12. Ligar e desligar, apagar linha de comando, interromper programa. Para ligar a calculadora, pressione **[ON]**; para desligar, pressione **[OFF]**. (OFF está impresso no teclado acima de **[ON]**. “Pressione **[OFF]**” significa pressionar a tecla de prefixo **[]** e, então, pressionar **[ON]**.)

Enquanto a calculadora está ligada, **[ON]** também atua como a tecla ATTN (*atenção*) para apagar texto na linha de comando ou para interromper um programa em execução. (ATTN está impresso no teclado abaixo de **[ON]**.)

13. Entradas numéricas. Para digitar números, utilize as teclas de dígitos de **[0]** até **[9]**, **[CHS]** (*troca sinal*), e **[EEX]** (*entra expoente*). Admitindo que você utilize o ponto como ponto decimal (página 36), use **[.]** para separar a parte inteira da parte fracionária. Entradas numéricas estão descritas à página 39.

14. Aritmética. As funções aritméticas estão descritas em “Funções de Um Número”, à página 40 e em “Funções de Dois Números”, à página 41.

15. Retrocesso. Pressione **[←]** para apagar o último caractere que você digitou.

16. Seleção de menu, próxima linha de menu. Utilize as teclas de seleção de menu para atribuir comandos às teclas de menu. Por exemplo, pressione **[TRIG]** para selecionar o menu TRIG. Para selecionar um menu diferente, pressione outra tecla de seleção de menu.

Quando não existem rótulos de menu visíveis, o *menu do cursor* está ativo. As operações no menu do cursor (**[INS]** até **[▶]**) estão impressas em branco acima das teclas do menu. Quando os rótulos de menu são visíveis, pressione **[↔]** para selecionar o menu do cursor. Para voltar ao menu anterior, pressione **[↔]** uma segunda vez.

Um menu pode conter mais de uma linha, com até seis comandos em cada linha. Pressione **[NEXT]** para apresentar a próxima linha do menu corrente. Pressione **[PREV]** para apresentar a linha anterior.

Existem teclas adicionais de seleção de menu no teclado da esquerda (ver item 2). Para uma relação em ordem alfabética de todos os menus, incluindo uma breve descrição dos comandos em cada menu, veja o apêndice D, "Diagramas dos Menus."

17. Teclas do menu. As teclas do menu são definidas pelos rótulos de menu. Se não há rótulos visíveis, estas teclas executam as operações do menu do cursor, gravadas em branco acima das teclas.

18. Rótulos de menu. Os rótulos de menu mostram as definições correntes das teclas do menu.

19. Níveis da pilha. A pilha mostra os objetos com os quais você está trabalhando no momento. Cada nível numerado da pilha (nível 1, nível 2, etc.) pode conter um objeto.

O Catálogo de Comandos

A HP-28S contém um catálogo de todos os comandos relacionados alfabeticamente. Para cada comando, o catálogo mostra sua utilização, isto é, os argumentos requeridos por este comando. Para uma descrição completa de qualquer comando relacionado no catálogo, ver "Índice de Operações", à página 381 do Manual de Referência.

Inicie o catálogo.

■ CATALOG

```
ABORT
NEXT PREV USE FETCH QUIT
```

O primeiro comando é ABORT.

A operação normal da calculadora é suspensa enquanto o catálogo está ativo. As operações **NEXT** e **PREV** movem o catálogo para outros comandos. A operação **USE** apresenta os argumentos requeridos pelo comando corrente. As operações **FETCH** e **QUIT** terminam o catálogo, restabelecendo a operação normal da calculadora.

Tente pressionar **NEXT** e **PREV** para mover-se através do catálogo. Você pode manter as teclas pressionadas para movimentos repetidos.

Você pode mover-se para o primeiro item de uma letra específica do catálogo, pressionando a tecla alfabética correspondente. Tente "T".

T

```
TAN
NEXT PREV USE FETCH QUIT
```

O primeiro comando na letra "T" é a função TAN. Se você pressionar uma tecla de símbolo (não alfabética) no teclado da esquerda, o catálogo se move para o primeiro elemento com aquele símbolo. Tente "Σ".

Σ

```
Σ+
NEXT PREV USE FETCH QUIT
```

O primeiro comando "Σ" é o comando Σ+. Se você pressionar uma tecla de símbolo no teclado da esquerda e não existirem comandos iniciando com aquele símbolo, o catálogo move-se para +, o primeiro comando não alfabético.

#

```
+
NEXT PREV USE FETCH QUIT
```

Verifique a utilização de +.

USE

```
USAGE: +
2: Real Number
1: Real Number
NEXT PREV USE FETCH QUIT
```

Este exemplo mostra que você pode adicionar dois números reais. Verifique a próxima combinação.

NEXT

```
USAGE: +
2: Real Number
1: Complex Number
NEXT PREV USE FETCH QUIT
```

Este mostra que você pode adicionar números reais e números complexos. Verifique a próxima combinação.

NEXT

```
USAGE: +
2: Complex Number
1: Real Number
NEXT PREV USE FETCH QUIT
```

Este mostra que os números reais e complexos podem estar em qualquer ordem.

Verifique as 14 combinações restantes. A última combinação tem a seguinte forma:

```
USAGE: +
2: Any Object
1: List
NEXT PREV      USE  QUIT
```

Após verificar as combinações, volte ao catálogo principal.

QUIT

```
+
NEXT PREV      USE  FETCH QUIT
```

Você pode agora mover-se para outro elemento do catálogo e verificar suas combinações de argumentos. Depois de examinar o catálogo, retome a operação normal da calculadora.

QUIT

```
3:
2:
1: 19
SIN ASIN COS ACOS TAN ATAN
```

Alternativamente, você pode sair do catálogo pressionando `FETCH`, que também escreve o nome do comando corrente na linha de comando.

2

Efetuando Cálculos Aritméticos

Existem duas maneiras de efetuar-se cálculos aritméticos na HP-28S. Utilizando a pilha, como no capítulo anterior ou entrando uma *expressão* representando o cálculo. No capítulo anterior você calculou:

$$(15 + 23) \times \text{sen } 30^\circ$$

A seguir, vejamos como fazer o mesmo cálculo utilizando uma expressão.

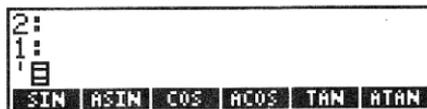
Apague a pilha e selecione o menu TRIG.

 CLEAR  TRIG



Inicie a expressão.

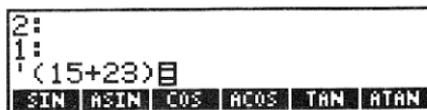




O cursor muda, indicando *modo de Entrada Algébrico*. Você verá os efeitos deste modo à medida que digitar a expressão.

Digite a primeira parte da expressão.

 15  23 



Por estar no modo Algébrico, pressionar $\boxed{+}$ escreve o caractere + na linha de comando, ao invés de executar o comando.

Continue a expressão.

\boxed{X} \boxed{SIN}

```
2:
1:
1: '(15+23)*SIN(
```

| | | | | | |
|-----|------|-----|------|-----|------|
| SIN | ASIN | COS | ACOS | TAN | ATAN |
|-----|------|-----|------|-----|------|

Por estar no modo Algébrico, pressionar \boxed{X} escreve * na linha de comando e pressionar \boxed{SIN} escreve SIN(na linha de comando, ao invés de executar os comandos.

Complete a expressão e coloque-a na pilha.

30 \boxed{ENTER}

```
3:
2:
1: '(15+23)*SIN(30)'
```

| | | | | | |
|-----|------|-----|------|-----|------|
| SIN | ASIN | COS | ACOS | TAN | ATAN |
|-----|------|-----|------|-----|------|

O fechamento de parênteses) e o fechamento do delimitador ' são adicionados automaticamente.

Processe a expressão.

\boxed{EVAL}

```
3:
2:
1: 19
```

| | | | | | |
|-----|------|-----|------|-----|------|
| SIN | ASIN | COS | ACOS | TAN | ATAN |
|-----|------|-----|------|-----|------|

A expressão é removida da pilha e o resultado, 19, é devolvido ao nível 1.

O cálculo está completo:

$$(15 + 23) \times \text{sen } 30^\circ = 19.$$

Para executar um cálculo que já está escrito como uma expressão, tal como em um livro texto, é mais fácil digitar a expressão e processá-la. Alternativamente, para ver os resultados intermediários de seus cálculos, ou para ir executando cálculos contínuos, é mais fácil calcular na pilha. Os resultados são os mesmos.

A relação entre cálculos na pilha e expressões é demonstrada no capítulo 4, "Repetindo Cálculos", onde você calcula na pilha, utilizando nomes ao invés de números, para produzir uma expressão.

Entrando e Apresentando Números

Existem dois modos que afetam a maneira como os números são apresentados no visor. Para demonstrar as escolhas, coloque o número $\frac{2}{3}$ na pilha.

Coloque 2 na pilha.

2

| | | | | | |
|----------------------------|--|--|--|--|----|
| 3: | | | | | |
| 2: | | | | | 19 |
| 1: | | | | | 2 |
| SIN ASIN COS ACOS TAN ATAN | | | | | |

Divida por 3.

3

| | | | | | |
|----------------------------|--|--|--|--|----------------|
| 3: | | | | | |
| 2: | | | | | 19 |
| 1: | | | | | .6666666666667 |
| SIN ASIN COS ACOS TAN ATAN | | | | | |

O resultado, $\frac{2}{3}$, é devolvido ao nível 1. Este resultado é a aproximação decimal de $\frac{2}{3}$, como apresentado no visor pela seleção padrão para ponto decimal e modo de apresentação de número. A próxima seção descreve outras possibilidades.

Mudando o Ponto Decimal

Nos Estados Unidos um ponto é utilizado para separar a parte inteira de um número da parte fracionária. No Brasil, utiliza-se a vírgula decimal para esta finalidade. Neste papel, o ponto é chamado um ponto decimal e o termo genérico para esta pontuação numérica é uma *marca de raiz*.

Muitos outros países utilizam a vírgula como a marca de raiz. Você pode selecionar a vírgula como segue.

Selecione o menu MODE.

| | | | | | |
|---------------------------|--|--|--|--|----------------|
| 3: | | | | | |
| 2: | | | | | 19 |
| 1: | | | | | .6666666666667 |
| STD= FIX SCI ENG DEG= RAD | | | | | |

A primeira linha do menu MODE aparece. Apresente a segunda linha do menu MODE.

NEXT

```
3:
2:                               19
1:                               .6666666666667
CMD= UNDO= LAST= ML= RDX= FRMO
```

Selecione a vírgula como a marca de raiz.

RDX,

```
3:
2:                               19
1:                               ,6666666666667
CMD= UNDO= LAST= ML= RDX= FRMO
```

Os pontos decimais são substituídos pela vírgula decimal, e um quadrado aparece no rótulo do menu RDX, para indicar que o modo RDX está ativado.

Desative o modo RDX, para restabelecer o ponto decimal.

RDX,

```
3:
2:                               19
1:                               .6666666666667
CMD= UNDO= LAST= ML= RDX= FRMO
```

Selecionando o Modo de Apresentação de Números

Você pode escolher quantas posições decimais são apresentadas no visor.

Volte para a primeira linha do menu MODE.

NEXT

```
3:
2:                               19
1:                               .6666666666667
STD= FIX= SCI= ENG= DEG= RAD
```

Você pode mover-se da última linha para a primeira, pressionando . Uma vez que o menu MODE tem apenas duas linhas, ao pressionar , volta-se para a primeira linha.

O menu mostra quatro escolhas básicas para o modo de apresentação de números: STD (padrão), FIX (fixa), SCI (científica) e ENG (engenharia). O rótulo para STD, neste momento, inclui um quadrado, indicando que STD é a escolha corrente. No modo STD, o número de posições decimais depende do valor.

Para um inteiro, não são mostradas casas decimais; para o exemplo apresentado acima, o máximo de 12 casas decimais foi mostrado.

Os outros formatos do visor mostram um dado número de casas decimais — de 0 a 11 — independentemente do número apresentado. Demonstraremos cada um dos outros formatos de apresentação de números com duas posições decimais. Somente as apresentações dos números no visor são arredondadas — internamente, os números não são modificados.

Apresente $2/3$ arredondados para duas casas decimais.

2 **FIX**

| | |
|-------------------------|-------|
| 3: | |
| 2: | 19.00 |
| 1: | 0.67 |
| STD FIX SCI ENG DEG RAD | |

Apresente $2/3$ como uma *mantissa* e um *expoente*, com a mantissa arredondada para duas casas decimais.

2 **SCI**

| | |
|-------------------------|---------|
| 3: | |
| 2: | 1.90E1 |
| 1: | 6.67E-1 |
| STD FIX SCI ENG DEG RAD | |

O valor do número é o produto da mantissa por 10 elevado à potência do expoente. A mantissa é sempre um número entre 1 e 9.9999999999.

Apresente $2/3$ como uma mantissa e um expoente, com a mantissa arredondada para duas posições decimais e o expoente um múltiplo de três.

2 **ENG**

| | |
|-------------------------|---------|
| 3: | |
| 2: | 19.0E0 |
| 1: | 667.E-3 |
| STD FIX SCI ENG DEG RAD | |

Volte para o modo padrão de apresentação de números.

STD

| | |
|-------------------------|---------------|
| 3: | |
| 2: | 19 |
| 1: | .666666666667 |
| STD FIX SCI ENG DEG RAD | |

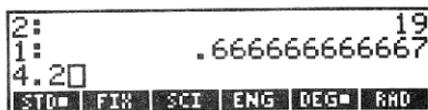
Digitando Números

Você pode entrar números como uma mantissa e um expoente, onde o valor do número é o produto da mantissa por 10 elevado à potência do expoente. A mantissa ou o expoente ou ambos podem ser negativos.

Por exemplo, digite o número -4.2×10^{-12} .

Primeiro entre os dígitos para a mantissa.

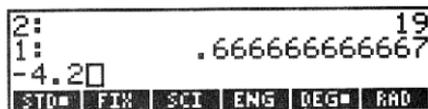
4.2



Se você cometer um erro, pressione \leftarrow para apagá-lo e, então, digite os números corretos.

Em seguida, troque o sinal da mantissa.

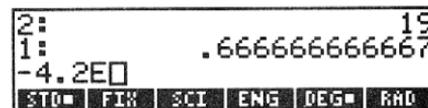
CHS



“CHS” significa “trocar sinal” (change sign) — pressionar CHS uma segunda vez, tornaria a mantissa positiva novamente.

Agora inicie o expoente.

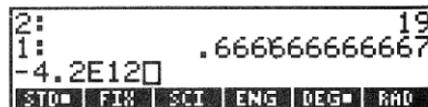
EEX



“EEX” significa “entrar expoente.” O E na linha de comando marca o expoente do número. Se você pressionar EEX por engano para um número sem expoente, você pode apagar o E pressionando \leftarrow , da mesma forma que você apagaria um dígito incorreto.

Digite os algarismos para o expoente.

12



Torne o expoente negativo.

CHS

```
2: 19
1: .66666666666667
-4.2E-12
STO= FIX SCI ENG DEG= RAD
```

Coloque o número na pilha.

ENTER

```
3: 19
2: .66666666666667
1: -4.2E-12
STO= FIX SCI ENG DEG= RAD
```

Não se esqueça de utilizar **CHS** para digitar números negativos. Por exemplo, se este manual mostrar as teclas -4 **X**, você necessitará pressionar **4** **CHS** **X**.

Funções de Um Número

Funções que atuam sobre um único número — por exemplo, transformando um número em negativo ou extraindo uma raiz quadrada — são chamadas funções de um número. Todas atuam no número que está no nível 1. Existem quatro funções de um número no teclado:

- pressione **CHS** para transformar um número em negativo.
- pressione **$\frac{1}{x}$** para calcular o inverso do número.
- pressione **\sqrt{x}** para extrair a raiz quadrada do número.
- pressione **x^2** para elevar o número ao quadrado.

Se você estiver digitando um número, não será necessário pressionar **ENTER** antes de executar a função de um número — pressionar a tecla de função automaticamente executa o ENTER. Por exemplo, você pode calcular $1/8$ como segue:

8 **$\frac{1}{x}$**

```
3: .66666666666667
2: -4.2E-12
1: .125
STO= FIX SCI ENG DEG= RAD
```

Funções de Dois Números

Funções que atuam sobre dois números — tal como a soma — são chamadas funções de dois números. Todas atuam sobre os números nos níveis 1 e 2.

Quando você está digitando ambos os argumentos da função, como quando você dividiu 2 por 3, à página 36, você precisa pressionar **ENTER** para separar os dois argumentos. Quando um ou os dois argumentos já estiverem na pilha de cálculos anteriores, não será necessário pressionar **ENTER**.

Adição e Subtração

Calcule $36 + 17$.

36 **ENTER**
17 **+**

```
3: -4.2E-12
2: .125
1: 53
STD= FIX SCI ENG DEG= RAD
```

O resultado é 53.

A ordem dos números não importa para a adição. Entretanto, é importante para a subtração. A seguir, calcule $91 - 27$.

91 **ENTER**
27 **-**

```
3: .125
2: 53
1: 64
STD= FIX SCI ENG DEG= RAD
```

O resultado é 64.

Multiplicação e Divisão

Calcule 13×6 .

13 **ENTER**
6 **X**

```
3: 53
2: 64
1: 78
STD= FIX SCI ENG DEG= RAD
```

O resultado é 78.

A ordem dos números não importa para a multiplicação. Entretanto, é importante para a divisão. A seguir, calcule $182/14$.

182
14

| | |
|---------------------------|----|
| 3: | 64 |
| 2: | 78 |
| 1: | 13 |
| STD= FIX SCI ENG DEG= RAD | |

O resultado é 13.

Potências e Raízes

A ordem dos números é importante tanto para potências como para raízes. Calcule 5^3 .

5
3

| | |
|---------------------------|-----|
| 3: | 78 |
| 2: | 13 |
| 1: | 125 |
| STD= FIX SCI ENG DEG= RAD | |

O resultado é 125.

Para calcular $\sqrt[4]{2401}$, primeiro coloque 2401 na pilha.

2401

| | |
|---------------------------|------|
| 3: | 13 |
| 2: | 125 |
| 1: | 2401 |
| STD= FIX SCI ENG DEG= RAD | |

Agora eleve 2401 à potência $1/4$.

4

| | |
|---------------------------|-----|
| 3: | 13 |
| 2: | 125 |
| 1: | 7 |
| STD= FIX SCI ENG DEG= RAD | |

O resultado é 7.

Percentagens

Calcule 40% de 85.

85
40

| | |
|---------------------------|-----|
| 3: | 125 |
| 2: | 7 |
| 1: | 34 |
| STO= FIX SCI ENG DEG= RAD | |

O resultado é 34.

A ordem dos números não importa para “percentagens”. Entretanto, é importante para “variação percentual”. Calcule a variação percentual de 60 para 75.

60
75

| | |
|---------------------------|----|
| 3: | 7 |
| 2: | 34 |
| 1: | 25 |
| STO= FIX SCI ENG DEG= RAD | |

O resultado é +25, significando que 75 é 25% mais do que 60.

Trocando os Níveis 1 e 2

A ordem dos números é importante para todas as funções — subtração, divisão, potenciação, radiciação e variação percentual. Você pode trocar a ordem dos números pressionando . Por exemplo, você tem 25 na pilha; suponha que você deseja calcular $30 - 25$.

Digite 30.

30

| | |
|---------------------------|----|
| 2: | 34 |
| 1: | 25 |
| 30 | |
| STO= FIX SCI ENG DEG= RAD | |

Troque a ordem de 25 e 30.

| | |
|---------------------------|----|
| 3: | 34 |
| 2: | 30 |
| 1: | 25 |
| STO= FIX SCI ENG DEG= RAD | |

Note que pressionar executou ENTER por você.

Sutraia 25 de 30.

-

| | |
|---------------------------|----|
| 3: | 7 |
| 2: | 34 |
| 1: | 5 |
| STD= FIX SCI ENG DEG= RAD | |

O resultado é 5.

Apagando Objetos da Pilha

À medida que você trabalhou estes exemplos, você acumulou uns tantos números na pilha. A pilha cresce sem limite à medida que você coloca mais objetos na mesma e esses objetos permanecem, até que você os utilize em operações ou até que você os apague.

Você pode apagar objetos um de cada vez ou todos de uma só vez.

Apague o número no nível 1.

DROP

| | |
|---------------------------|-----|
| 3: | 125 |
| 2: | 7 |
| 1: | 34 |
| STD= FIX SCI ENG DEG= RAD | |

Objetos em níveis mais altos da pilha movem-se um nível para baixo cada.

Apague todos os objetos da pilha.

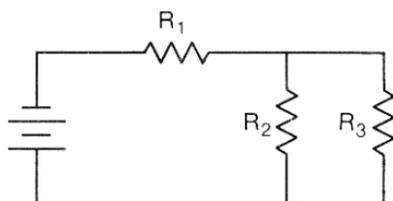
CLEAR

| | |
|---------------------------|--|
| 3: | |
| 2: | |
| 1: | |
| STD= FIX SCI ENG DEG= RAD | |

É uma boa idéia apagar a pilha sempre que você inicia um problema. À medida que você trabalhar no problema, saberá que todos os objetos na pilha são relevantes para o problema corrente e não referentes ao problema anterior.

Cálculos Encadeados

Quando você executa cálculos complicados, a pilha serve como um armazenamento temporário, para manter resultados intermediários. Este armazenamento temporário atua automaticamente. Por exemplo, suponha que você queira calcular a resistência total do seguinte circuito:



A fórmula para a resistência total neste circuito é:

$$R_{total} = R_1 + \frac{1}{\frac{1}{R_2} + \frac{1}{R_3}}$$

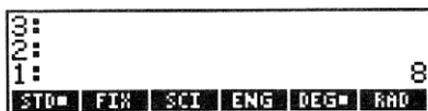
Se R_1 , R_2 , e R_3 têm resistências de 8, 6, e 3 ohms, respectivamente, calcule o seguinte:

$$R_{total} = 8 + \frac{1}{\frac{1}{6} + \frac{1}{3}}$$

Calcule como demonstrado a seguir.

Coloque 8 na pilha.

8



Você deixará 8 na pilha até que seja a hora de adicioná-lo ao resto dos cálculos.

Coloque 1/6 na pilha.

6 $\frac{1}{x}$

```
3:
2:
1: .1666666666667
STD= FIX SCI ENG DEG= RAD
```

Coloque 1/3 na pilha.

3 $\frac{1}{x}$

```
3:
2: .1666666666667
1: .3333333333333
STD= FIX SCI ENG DEG= RAD
```

Some os inversos de 6 e 3.

+

```
3:
2:
1: .5000000000000
STD= FIX SCI ENG DEG= RAD
```

Calcule o inverso da soma.

$\frac{1}{x}$

```
3:
2:
1: 2.0000000000000
STD= FIX SCI ENG DEG= RAD
```

Complete o cálculo de R_{total} .

+

```
3:
2:
1: 10
STD= FIX SCI ENG DEG= RAD
```

O resultado é 10 ohms.

Se Você Executar a Função Errada

A HP-28S inclui recursos de recuperação para auxiliá-lo a “voltar sobre suas próprias pegadas” quando você executa uma função por engano. Os passos seguintes revertem os efeitos de uma função, quer seja de um número ou de dois números.

1. Pressione  UNDO para recuperar o conteúdo anterior da pilha.
2. Se um número estava na linha de comando quando você cometeu um erro, pressione  COMMAND para recuperar o conteúdo anterior da linha de comando.
3. Continue o cálculo.

Utilizando Variáveis

Variáveis permitem que você se refira a objetos pelo nome. Você cria uma variável associando um objeto nome com qualquer outro objeto. O objeto nome define o nome da variável; o outro objeto define o conteúdo da variável. Você pode então utilizar o nome da variável para referir-se ao conteúdo da variável.

Variáveis são armazenadas na *memória do usuário*, uma parte da memória da calculadora separada da pilha. A pilha foi projetada para armazenamento temporário dos objetos com os quais você está trabalhando no momento, tais como os números que você está utilizando em um cálculo. A memória do usuário foi projetada para armazenamento de variáveis por um longo prazo, tais como números que você utiliza freqüentemente.

Neste capítulo, você criará uma variável numérica, que pode ser um conceito familiar para você; também criará uma variável programa que é, provavelmente, um conceito com o qual você não está familiarizado. Na HP-28S, um programa não tem nome intrínscico— é simplesmente um outro tipo de objeto. Você atribui um nome ao programa armazenando-o em uma variável, da mesma forma que você faria com um número e, então, pode executar o programa pelo nome.

Os passos para criar, recuperar, processar, modificar, mudar o nome, ou eliminar uma variável são idênticos para todas as variáveis, independentemente do seu conteúdo. Esta uniformidade torna a HP-28S fácil de ser utilizada e poderosa porque existem menos regras especiais e mais flexibilidade.

Introdução a Variáveis

As variáveis mais simples são as variáveis numéricas. Esta seção mostra como criar, recuperar e processar uma variável numérica.

Criando uma Variável Numérica

Suponha que você freqüentemente utilize uma medida de volume de 133 em seus cálculos. Crie uma variável denominada VOL ("volume") como segue: apague a pilha e selecione o menu USER.

```
3:
2:
1:
```

O menu USER mostra suas variáveis. Ele está em branco porque você ainda não criou nenhuma variável.

Coloque o número no nível 1.

133

```
3:
2:
1: 133
```

Coloque o nome 'VOL' no nível 1.

' VOL

```
3:
2:
1: 'VOL' 133
```

Note que o fechamento da ' e adicionado automaticamente. O número 133 é elevado para o nível 2. (Na prática, você não necessita pressionar , mas ele está incluído aqui para maior clareza.)

Crie a variável VOL.

```
3:
2:
1:
VOL
```

O número e o nome são removidos da pilha, criando uma variável denominada VOL com o valor de 133. Note que VOL agora aparece no menu USER.

Recuperando uma Variável Numérica

Agora que você criou a variável VOL, coloque seu valor na pilha.

Coloque o nome VOL na pilha, tirando partido do menu USER.

' VOL ENTER

| | | | | |
|-----|--|--|--|-------|
| 3: | | | | |
| 2: | | | | |
| 1: | | | | 'VOL' |
| VOL | | | | |

Recupere o conteúdo de VOL.

RCL

| | | | | |
|-----|--|--|--|-----|
| 3: | | | | |
| 2: | | | | |
| 1: | | | | 133 |
| VOL | | | | |

Este é o número que você armazenou em VOL.

Se você está acostumado com uma calculadora que possui registradores de armazenamento, recuperar números é um processo familiar. Na HP-28S, as variáveis são recuperadas com pouca frequência; mais usualmente elas são *processadas*.

Processando uma Variável Numérica

Para variáveis numéricas, “processando” tem o mesmo significado que “recuperando” — processar uma variável numérica devolve o número à pilha. Você verá que o cálculo é fácil. (Quando você criar uma variável programa, mais adiante neste capítulo, verá que processando e recuperando podem ter efeitos muito diferentes.)

Devolva o valor de VOL à pilha por processamento.

VOL

| | | | | |
|-----|--|--|--|-----|
| 3: | | | | |
| 2: | | | | 133 |
| 1: | | | | 133 |
| VOL | | | | |

Você também pode processar VOL digitando seu nome *sem aspas*.

VOL ENTER

| | | | | |
|-----|--|--|--|-----|
| 3: | | | | 133 |
| 2: | | | | 133 |
| 1: | | | | 133 |
| VOL | | | | |

Mudando o Valor de uma Variável

Você pode mudar o valor de uma variável, utilizando o mesmo procedimento que utilizou para criá-la. O novo valor substitui o valor anterior.

Agora mude o valor da variável VOL para 151.

Escreva o novo valor na linha de comando.

151

```
2: 133
1: 133
151
VOL
```

Note que o cursor aparece como um quadrado vazio. O cursor mudará no próximo passo.

Escreva o nome da variável na linha de comando.

VOL

```
2: 133
1: 133
151 VOL
VOL
```

O cursor mudou quando você pressionou para indicar o novo *modo de entrada* — como a calculadora responde quando você pressiona teclas.

Inicialmente, a linha de comando estava em *modo de execução imediato*, adequado para cálculos de teclado. Quando você pressionou , que indica um nome ou uma expressão, a linha de comando mudou para *modo de entrada algébrico* adequado para entrar-se nomes e expressões:

- pressionar uma tecla de função tal como escreve o caractere + , ao invés de executar o comando.
- pressionar uma tecla do menu USER escreve o nome da variável ao invés de processá-la.

Agora armazene o novo valor na variável.

STO

```
3: 133
2: 133
1: 133
VOL
```

Verifique o novo valor.

VOL

| | |
|-----|-----|
| 3: | 133 |
| 2: | 133 |
| 1: | 151 |
| VOL | |

Eliminando uma Variável

Quando você terminar a utilização da variável VOL, elimine-a da memória do usuário.

Escreva o nome da variável na linha de comando.

VOL

| | |
|------|-----|
| 2: | 133 |
| 1: | 151 |
| 'VOL | |
| VOL | |

(A aspa é necessária para evitar que se processe a variável.)

Elimine a variável VOL da memória do usuário.

PURGE

| | |
|----|-----|
| 3: | 133 |
| 2: | 133 |
| 1: | 151 |
| | |

Note que o rótulo VOL desaparece do menu USER.

Mudando o Nome de uma Variável

Você pode efetivamente mudar o nome de uma variável, criando uma nova variável com o mesmo valor e eliminando a variável original.

Nesta seção, primeiramente, você executará os passos necessários para mudar o nome de uma variável, a seguir escreverá um programa contendo os mesmos passos e, finalmente, armazenará o programa em uma variável para depois executá-lo pelo nome.

Em preparação, crie uma variável de forma que você tenha algo para mudar o nome — uma variável A com valor 10.

Coloque o valor 10 na pilha.

10

| | |
|----|-----|
| 3: | 133 |
| 2: | 151 |
| 1: | 10 |
| A | |

Crie a variável A.

' A

| | |
|----|-----|
| 3: | 133 |
| 2: | 133 |
| 1: | 151 |
| A | |

Note que A aparece no menu USER.

Suponha que você deseje mudar o nome de A para B. Coloque o nome anterior na pilha.

' A

| | |
|----|-----|
| 3: | 133 |
| 2: | 151 |
| 1: | 'A' |
| A | |

Coloque o novo nome na pilha.

' B

| | |
|----|-----|
| 3: | 151 |
| 2: | 'A' |
| 1: | 'B' |
| A | |

Isto completa a preparação: a variável existe, o nome anterior e o novo nome estão na pilha. Os nomes antigo e novo são os *argumentos* para o programa que admitirá que os mesmos estão na pilha nesta ordem. Os passos que se seguem são aqueles que estarão no programa.

Os passos incluem três comandos simples e manipulação da pilha: OVER, ROT e SWAP. Você verá como eles operam à medida que os passos são executados.

Copie o nome anterior no nível 1. (Use o comando OVER no menu STACK.)

OVER

| | |
|-------------------------------|-----|
| 3: | 'A' |
| 2: | 'B' |
| 1: | 'A' |
| DUP OVER DUP2 DROP2 ROT LIST+ | |

Recupere o conteúdo da variável.

RCL

```
3: 'A'  
2: 'B'  
1: 10  
DUP OVER DUF2 DROF2 ROT LIST
```

Mova o nome anterior para o nível 1. (Utilize o comando ROT, para "girar".)

ROT

```
3: 'B'  
2: 10  
1: 'A'  
DUP OVER DUF2 DROF2 ROT LIST
```

Elimine a variável anterior. (Eliminando a variável anterior antes de criar-se a nova, você evita que uma cópia extra do valor seja feita.)

PURGE

```
3: 151  
2: 'B'  
1: 10  
DUP OVER DUF2 DROF2 ROT LIST
```

Coloque o conteúdo e o novo nome na ordem correta.

SWAP

```
3: 151  
2: 10  
1: 'B'  
DUP OVER DUF2 DROF2 ROT LIST
```

Crie a nova variável.

STO

```
3: 133  
2: 133  
1: 151  
DUP OVER DUF2 DROF2 ROT LIST
```

Agora você pode criar um programa que automatiza estes passos.

Criando uma Variável Programa

Primeiro você digitará o programa e, então, armazená-lo-á em uma variável.

Comece o programa com o delimitador de programa.

«

```

2: 133
1: 151
«
┌ DUF OVER DUF2 DROF2 ROT LIST ┐

```

Note que a forma do cursor mudou e que o anúncio α apareceu, ambos indicando *modo de entrada alfanumérico*. Pressionar uma tecla de qualquer operação programável, escreve o nome da operação na linha de comando. Somente operações não programáveis, tal como pressionar \blacktriangleleft para apagar um caractere, são executadas.

Agora digite os passos que você executou anteriormente.

OVER RCL
ROT PURGE
SWAP STO
ENTER

```

2: 151
1: « OVER RCL ROT PURGE
   SWAP STO »
┌ DUF OVER DUF2 DROF2 ROT LIST ┐

```

Note que o fechamento do delimitador » foi adicionado automaticamente.

Armazene o programa em uma variável RENAME.

' RENAME STO

```

3: 133
2: 133
1: 151
┌ DUF OVER DUF2 DROF2 ROT LIST ┐

```

Verifique o menu USER.

USER

```

3: 133
2: 133
1: 151
┌ RENA E ─── ─── ─── ─── ─── ┐

```

Note que RENAME (em forma abreviada) aparece no menu USER.

Agora você está pronto para executar RENAME. Primeiramente, fará no método indireto, utilizando RCL e, então, pelo método normal, utilizando o menu USER. A diferença nos métodos destaca a diferença entre recuperar e processar uma variável programa.

Recuperando uma Variável Programa

Para este exemplo, mude o nome da variável B para C.

Coloque o nome anterior e o novo nome na pilha.

' B ENTER
' C ENTER

| | | |
|--------|-----|-----|
| 3: | | 151 |
| 2: | 'B' | |
| 1: | 'C' | |
| RENA B | | |

Recupere o programa RENAME

' RENA RCL

| | | |
|--------|------------------------------------|-----|
| 2: | | 'C' |
| 1: | « OVER RCL ROT PURGE SWAP STO » | |
| RENA B | | |

Para qualquer variável, RCL simplesmente devolve o conteúdo da variável à pilha.

Processando uma Variável Programa

Para executar um programa na pilha, você precisa *explicitamente* processá-lo.

EVAL

| | | |
|--------|--|-----|
| 3: | | 133 |
| 2: | | 133 |
| 1: | | 151 |
| C RENA | | |

O menu USER mostra que o nome B foi mudado para C.

Não foi necessário recuperar o programa na pilha para execução, mas este passo demonstrou como RCL opera para programas e como EVAL faz com que os programas sejam executados. A seguir, você verá uma maneira fácil de executar seu programa.

Desta vez, mudará o nome C para D. Coloque o nome anterior e o novo nome na pilha.

' C ENTER
' D ENTER

| | | |
|--------|-----|-----|
| 3: | | 151 |
| 2: | 'C' | |
| 1: | 'D' | |
| C RENA | | |

Mude o nome C para D.

RENA

| | | |
|----|---|------|
| 3: | | 133 |
| 2: | | 133 |
| 1: | | 151 |
| | D | RENA |

O menu USER mostra que o nome C foi mudado para D. Você executou o programa simplesmente pressionando uma tecla no menu USER.

Nomes com Aspas e sem Aspas

Nos exemplos acima, você utilizou nomes de variáveis de duas maneiras, com aspas e sem aspas. As aspas `'` são importantes porque distinguem o *nome* de uma variável do *conteúdo* da variável. Aqui está um resumo das finalidades de nomes com e sem aspas.

- Utilize um nome com aspas para representar o nome em si. As aspas impedem o processamento do nome (EVAL), de modo que ele vai para a pilha e pode ser um argumento para um comando. Neste capítulo, você utilizou nomes com aspas como argumentos para STO, RCL, PURGE e o programa RENAME.
- Utilize um nome sem aspas para processar (EVAL) a variável com aquele nome. O nome sem aspas não vai para a pilha — ao invés disso, o objeto armazenado na variável é manipulado de acordo com seu tipo: variáveis numéricas são devolvidas à pilha e programas são executados. Você verá o que ocorre com outros tipos de variáveis, mais adiante, neste manual.

Se você digitou um nome sem aspas que não está associado a uma variável, a forma do nome entre aspas vai para a pilha.

Repetindo Cálculos

Neste capítulo, você criará uma expressão contendo variáveis numéricas e utilizará um recurso da calculadora denominado Solver, para processar a expressão para diversos valores das variáveis numéricas.

No capítulo 2, você fez um cálculo digitando uma expressão que continha *números* e, então, processou a expressão. Neste capítulo, você criará uma expressão calculando na pilha e usando *nomes* como argumentos simbólicos. Você utilizará o Solver para atribuir valores às variáveis e processar a expressão. Cada vez que você processar a expressão, o cálculo será efetuado com os valores correntes das variáveis. Se você modificar o valor de uma ou mais variáveis, poderá simplesmente processar a expressão novamente para recalculá-la com os novos valores.

No capítulo 3, você criou variáveis numéricas e uma variável programa. Neste capítulo, você criará variáveis expressão e variáveis nome. (Lembre-se, qualquer objeto pode ser armazenado em uma variável.) Você também aprenderá sobre *diretórios*, que são conjuntos de variáveis.

Criando uma Expressão

Repetiremos o cálculo de resistência de “Cálculos Encadeados”, no capítulo 2, somente que desta vez utilizaremos nomes, ao invés de números, como argumentos. Lembre-se de que a fórmula para o circuito é:

$$R_{total} = R_1 + \frac{1}{\frac{1}{R_2} + \frac{1}{R_3}}$$

Apague a pilha e selecione o cursor do menu.

CLEAR

```
4:
3:
2:
1:
```

Se um menu for apresentado, pressione para selecionar o cursor do menu.

Coloque o nome 'R1' na pilha.

R1

```
4:
3:
2:
1: 'R1'
```

Note que a ' terminando o nome é colocada automaticamente. Você deixará R1 na pilha até que seja o momento de adicioná-la ao resto do cálculo.

Coloque o inverso de R2 na pilha.

R2

```
4:
3:
2: 'R1'
1: 'INV(R2)'
```

Coloque o inverso de R3 na pilha.

R3

```
4:
3: 'R1'
2: 'INV(R2)'
```

Adicione os inversos de R2 e R3.

```
4:
3: 'R1'
2: 'INV(R2)'
```

Calcule o inverso da soma.

```
3:
2: 'R1'
1: 'INV(INV(R2)+INV(R3))'
```

Adicione R1 e o inverso.

+

```
20:
21:
1: 'R1+INV(INV(R2)+INV(R3))'
```

A expressão resultante representa R_{total} .

Você poderia digitar esta expressão diretamente, tomando cuidado para adicionar parênteses quando necessário. Cada expressão é equivalente a um cálculo da pilha, de modo que você pode escolher o método que for mais fácil.

Mais adiante neste capítulo, você armazenará esta expressão em uma variável, mas primeiro crie um diretório para agrupar os exemplos deste capítulo.

Criando um Diretório

Um diretório é um conjunto de variáveis. Agora mesmo você está trabalhando no diretório HOME— um diretório permanente da calculadora, que existe mesmo após a inicialização da memória (MEMORY RESET). Neste capítulo, você criará um subdiretório dentro de HOME e, então, subdiretórios dentro daquele subdiretório.

Aqui estão alguns conceitos que você utilizará neste capítulo.

- Somente um diretório pode ser o *diretório corrente*; somente suas variáveis aparecem no menu USER.
- Se um diretório A contém um diretório B, então A é chamado o *diretório pai* de B, e B é chamado um *subdiretório* de A.
- Se você iniciar no diretório corrente e encontrar seu diretório pai e, então, o próximo diretório pai, e assim por diante, você sempre retornará para HOME. Esta seqüência de diretórios (na ordem inversa) é chamada a *trilha corrente*.

Você pode verificar a trilha corrente executando o comando PATH.

Selecione o menu MEMORY.

MEMORY

```
2:
1: 'R1+INV(INV(R2)+INV(R3))'
MEM MENU ORDER PATH HOME CRDIS
```

Verifique a trilha corrente.

PATH

```
3:
2: 'R1+INV(INV(R2)+INV...
1: ( HOME )
MEM MENU ORDER PATH HOME CROIR
```

A lista que PATH devolve sempre começa com HOME e termina com o diretório corrente. HOME é o ponto de partida para todas as trilhas e, desde que você não tenha criado outros diretórios ainda, HOME é também o diretório corrente.

Para agrupar todos os seus problemas de engenharia elétrica, crie um subdiretório denominado EE.

EE CROIR

```
3:
2: 'R1+INV(INV(R2)+INV...
1: ( HOME )
MEM MENU ORDER PATH HOME CROIR
```

Mude para o diretório EE.

EE

```
3:
2: 'R1+INV(INV(R2)+INV...
1: ( HOME )
MEM MENU ORDER PATH HOME CROIR
```

Verifique a trilha corrente novamente.

PATH

```
3: 'R1+INV(INV(R2)+INV...
2: ( HOME )
1: ( HOME EE )
MEM MENU ORDER PATH HOME CROIR
```

Agora o diretório corrente é EE. Para ver um efeito da mudança para o diretório EE, apresente o menu USER.

```
3: 'R1+INV(INV(R2)+INV...
2: ( HOME )
1: ( HOME EE )
MEM MENU ORDER PATH HOME CROIR
```

Note que o programa RENAME (criado no último capítulo) não aparece. Somente variáveis no *diretório corrente* (EE) aparecem no menu USER; RENAME está no diretório HOME.

Entretanto, você ainda poderia executar RENAME, digitando seu nome, já que qualquer variável, cujo diretório está na *trilha corrente* (HOME EE), pode ser encontrada pelo nome.

Este é um dos benefícios dos diretórios: se você colocar programas de uso geral, tal como *RENAME*, no diretório *HOME*, você sempre poderá executá-los, mas eles não irão atrapalhar, preenchendo o menu *USER*.

Agora você pode trabalhar no novo diretório *EE*.

Elimine (*DROP*) as duas listas de trilha da pilha.

```
3:
1: 'R1+INV(INV(R2)+INV(R3))'
```

Armazene a expressão em uma variável denominada *EQ1* (*equação 1*). Você verá a razão para este nome mais adiante.

EQ1

```
3:
2:
1:
EQ1
```

A variável *EQ1* aparece em um menu *USER*.

Vamos admitir que você utilizará esta expressão para uma variedade de problemas, cada um dos quais você quer tratar independentemente. Para fazer isso, você pode colocar os valores para cada problema em um subdiretório específico para cada problema.

Crie um subdiretório *SP1* (*série-paralelo 1*) para o primeiro problema.

SP1
CRDIR

```
3:
2:
1:
SP1 EQ1
```

O nome do novo subdiretório aparece no menu *USER*. Pressione a tecla do menu para mudar para *SP1*.

SP1

```
3:
2:
1:
```

O menu *USER* está vazio novamente porque o diretório corrente (*SP1*) está vazio.

N.T. O comando *DROP* efetivamente deixa cair toda a pilha de 1 nível e o objeto no nível 1 é eliminado (cai fora da pilha). Por esta razão, muitas vezes neste manual, referimo-nos a *DROP* como *eliminar*.

Verifique a trilha corrente.

PATH

```
3:
2:
1:      ( HOME EE SP1 )
```

Você pode encontrar qualquer variável nos diretórios HOME ou EE pelo nome, porque estes diretórios estão na trilha corrente (HOME EE SP1), mas o menu USER somente mostra as variáveis no diretório corrente (SP1).

Agora você está pronto para utilizar o Solver com a expressão EQ1.

Utilizando o Solver para Repetir um Cálculo

Existem três passos básicos para utilizar-se o Solver com uma expressão.

1. Armazene a expressão (ou o nome da expressão) em uma variável denominada EQ (*equação*). O Solver requer uma variável com este nome.
2. Utilize o menu Solver para atribuir valores às variáveis.
3. Utilize o Solver para processar a expressão.
4. Repita os passos 2 e 3 para outros valores.

Aqui estão os passos para o exemplo presente.

Passo 1: Armazene o nome EQ1 na variável EQ.

Este passo pode surpreendê-lo — por que armazenar um nome em uma variável? Por que não armazenar a expressão em si em EQ? A razão mais simples é que o nome EQ1 é mais curto e mais fácil de ser lembrado do que a equação toda. Você verá mais tarde, também, que este passo torna mais fácil ir e voltar entre diferentes equações.

Coloque o nome EQ1 na pilha.

```
3:
2:      ( HOME EE SP1 )
1:      'EQ1'
```

Se você se esqueceu da aspa , você obteve a expressão em si na pilha. Neste caso, pressione para eliminá-la e tente novamente.

Selecione o menu SOLVE.

SOLV

```
3:
2:      ( HOME EE SP1 )
1:      'EQ1'
STEQ RCEQ SOLVR ISOL QUAD SHOW
```

Utilize STEQ (*Store Equation = armazenar equação*) para armazenar o nome EQ1 na variável EQ.

STEQ

```
3:
2:
1:      ( HOME EE SP1 )
STEQ RCEQ SOLVR ISOL QUAD SHOW
```

Passo 2: Atribua valores às variáveis.

Apresente o menu Solver.

SOLVR

```
3:
2:
1:      ( HOME EE SP1 )
R1 R2 R3 EXPR=
```

As variáveis na equação corrente aparecem no menu Solver. (Se a equação contém mais de seis variáveis, pressionar **NEXT** apresenta linhas adicionais de variáveis.)

Este menu tem aparência diferente do menu USER porque trabalha diferentemente: o menu Solver *armazena valores* nas variáveis, ao invés de processar variáveis.

Agora você pode atribuir valores às variáveis R1, R2 e R3. Primeiro, armazene o número 8 na variável R1.

8 R1

```
R1: 8
2:
1:      ( HOME EE SP1 )
R1 R2 R3 EXPR=
```

Pressionar **R1** é equivalente a colocar 'R1' na pilha e pressionar **STO**. Note que a linha de cima do visor mostra o nome da variável e o valor.

Armazene o número 6 na variável R2.

6 R2

```
R2: 6
2:
1:      ( HOME EE SP1 )
R1 R2 R3 EXPR=
```

Armazene o número 3 na variável R3.

3

```
R3: 3
2:
1: ( HOME EE SP1 )
R1 R2 R3 EXPR=
```

Passo 3: Processe a expressão.

O rótulo de menu significa “expressão igual a” — pressioná-la processa a expressão.

```
EXPR=10
2:
1: ( HOME EE SP1 ) 10
R1 R2 R3 EXPR=
```

O valor (10) é devolvido ao nível 1 e aparece em caracteres brancos com fundo preto, na linha superior do visor.

Passo 4: Repita passos 2 e 3 para outros valores. Por exemplo, qual o resultado se R3 é 12?

Armazene o número 12 na variável R3.

12

```
R3: 12
2:
1: ( HOME EE SP1 ) 10
R1 R2 R3 EXPR=
```

Processe a expressão para os valores correntes de suas variáveis.

```
EXPR=12
2:
1: ( HOME EE SP1 ) 10 12
R1 R2 R3 EXPR=
```

O novo valor (12) é devolvido ao nível 1 e aparece em caracteres brancos com fundo preto, na linha superior do visor.

Utilizando um Conjunto Diferente de Valores

Suponha que você deseje trabalhar num problema diferente, com valores diferentes de R1, R2 e R3 e, mais tarde, voltar aos valores agora atribuídos a essas variáveis. Você poderia reentrar todos os valores a cada vez que mudasse de problema, mas esta seção irá mostrar-lhe uma maneira mais fácil. Existem três passos:

1. crie um novo diretório para os novos valores.
2. defina a mesma expressão como sendo EQ.
3. use o Solver como antes para atribuir valores e processar a expressão.

Este processo mostra outro benefício dos diretórios: *dentro de um diretório, somente uma variável pode existir com um nome particular; mas qualquer número de diretórios pode conter uma variável com um nome particular.*

Passo 1: Crie um novo diretório.

Desde que o novo diretório é uma alternativa a SP1, chame-o SP2 e crie-o dentro do mesmo diretório-pai (nominalmente EE). Isto será o primeiro "ramo" dentro da sua estrutura de diretório — dois subdiretórios (SP1 e SP2) dentro do mesmo diretório-pai (EE).

Para criar um subdiretório dentro de EE, você precisa tornar EE o diretório corrente. (Qualquer subdiretório que você criasse agora seria dentro de SP1).

Mude para o diretório EE.

EE

| | |
|---------------------------------|----|
| No Current Equation | |
| 2: | 10 |
| 1: | 12 |
| STEQ RCEQ SOLVE ISOL GUARD SHOW | |

A calculadora emite um sinal, apresenta NO CURRENT EQUATION (Não Há Equação Corrente) e ativa o menu SOLVE. Isto ocorre porque não existe equação corrente 'EQ' no diretório EE.

Crie um diretório SP2.

MEMORY

| | |
|--------------------------------|----|
| [HOME EE SP1] | |
| 2: | 10 |
| 1: | 12 |
| MEM MENU ORDER PATH HOME CRDIR | |

Mude para o diretório SP2.

SP2

```
3:      ( HOME EE SP1 )
2:      10
1:      12
MEM MENU ORDER PATH HOME CDRR
```

Verifique a trilha corrente.

PATH

```
3:      10
2:      12
1:      ( HOME EE SP2 )
MEM MENU ORDER PATH HOME CDRR
```

Note que HOME e EE estão na trilha corrente, como eles estavam quando SP1 era o diretório corrente, mas SP1 não aparece agora. Como resultado, você ainda pode encontrar as variáveis em HOME (tal como RENAME) e em EE (tal como EQ1), mas não as variáveis SP1 (EQ, R1, R2 e R3); agora você pode criar novas variáveis R1, R2 e R3.

Passo 2: Defina a mesma expressão para ser EQ.

Como antes, use STEQ para armazenar o nome EQ1 na variável EQ.

```
3:      10
2:      12
1:      ( HOME EE SP2 )
STEQ EQEQ SOLVR ISOL QURD SHOW
```

Passo 3: Utilize o Solver como antes para atribuir valores e processar a expressão. Suponha que os valores para o novo problema sejam:

$$R1 = 11, \quad R2 = 21, \quad R3 = 7$$

Selecione o menu Solver.

SOLVR

```
3:      10
2:      12
1:      ( HOME EE SP2 )
R1 R2 R3 EXPR=
```

Atribua os valores.

11
21
7

```
R3: 7
2:      12
1:      ( HOME EE SP2 )
R1 R2 R3 EXPR=
```

Processe a expressão.

EXPR=

```
EXPR=16.25
2: ( HOME EE SP2 )
1: 16.25
R1 R2 R3 EXPR=
```

Para retornar ao problema anterior, você executaria EE (para mudar para o diretório EE), executaria SP1 (para mudar para o diretório SP1) e pressionaria **SOLV** **SOLVR** (para ativar o menu Solver); todos os valores de variáveis seriam os mesmos que existiam quando você deixou SP1.

Utilizando uma Expressão Diferente

Agora que você tem dois conjuntos de valores para utilizar com a expressão EQ1, tente criar uma segunda expressão EQ2 que você possa utilizar com quaisquer dos conjuntos de valores. Há dois passos básicos:

1. mude para o diretório EE, crie a nova expressão e armazene-a em uma variável EQ2.
2. mude para o diretório SP1 ou SP2; mude o valor de EQ de 'EQ1' para 'EQ2' e utilize o Solver para calcular esta expressão.

Passo 1: Mude para o diretório EE; crie a nova expressão e armazene-a em uma variável EQ2.

Mude para o diretório EE.

EE **ENTER**

```
No Current Equation
2: ( HOME EE SP2 )
1: 16.25
STEQ RCEQ SOLVR ISOL QUAD SHOW
```

Crie a nova expressão. Neste exemplo, EQ2 será uma cópia editada da expressão EQ1.

Devolva a expressão armazenada em EQ1 à pilha.

USER EQ1

```
2: 16.25
1: 'R1+INV(INV(R2)+INV(
R3))'
```

SP2 SP1 EQ1

Devolva a expressão à linha de comando.

EDIT

```
1: 'R1+INV(INV(R2)+INV(
R3))'
```

```
R1+INV(INV(R2)+INV(
R3))'
```

A expressão aparece em caracteres invertidos no nível 1 (claro em fundo escuro), para alertá-lo que ela será substituída pelo conteúdo da linha de comando. O anúncio alfa α aparece, indicando que o modo alfanumérico está ativo.

Agora edite a expressão para representar a fórmula:

$$R_{total} = R_1 + \frac{1}{\frac{1}{R_2} + \frac{1}{R_3} + \frac{1}{R_3}}$$

Mova o cursor para a linha inferior da linha de comando. (As operações para mover o cursor estão no *menu do cursor* — os rótulos impressos em branco bem acima das teclas do menu.)

▼

```
1: 'R1+INV(INV(R2)+INV(
R3))'
```

```
R1+INV(INV(R2)+INV(
R3))'
```

O menu do cursor está ativo toda vez que a linha de comando existe e nenhum menu é apresentado. Você pode ligar e desligar o menu do cursor pressionando **↔**. Pressionar **EDIT** automaticamente liga o menu do cursor.

Mova o cursor para imediatamente após o termo contendo R3.

```
1: 'R1+INV(INV(R2)+INV(
R3))'
```

```
R1+INV(INV(R2)+INV(
R3)█'
```


Mude a definição de EQ de EQ1 para EQ2.

EQ2 SOLV STEQ

```
3: 12
2: ( HOME EE SP2 )
1: 16.25
STEQ RCEQ SOLVR ISOL QUAD SHOW
```

Processe a expressão EQ2 com os valores de SP1.

SOLVR EXPR=

```
EXPR=11
2: 16.25
1: 11
R1 R2 R3 EXPR=
```

Para processar EQ2 com os valores de SP2, você poderia executar EE (para mudar de diretório de volta para EE) e, então, repetir o passo 2 acima, substituindo SP1 por SP2.

Voltando ao Diretório HOME

Admitindo que você tenha terminado por agora com o seu programa de engenharia elétrica, você pode voltar ao diretório HOME. Uma vez que HOME é um diretório permanente, seu nome está incluído no menu MEMORY.

Mude para o diretório HOME.

MEMORY HOME

```
3: ( HOME EE SP2 )
2: 16.25
1: 11
MEM MENU ORDER PATH HOME CROIR
```

Verifique o menu USER.

USER

```
3: ( HOME EE SP2 )
2: 16.25
1: 11
EE D. REMA
```

O rótulo de menu **EE** é o único sinal de tudo que você criou neste capítulo — EQ1, EQ2, os subdiretórios SP1 e SP2 e todas as variáveis neles contidas. Esta é uma das grandes vantagens dos diretórios: *visto de seu diretório-pai, um diretório completo — suas variáveis e seus próprios subdiretórios — aparece simplesmente como o nome do diretório.*

Sumário

Eis aqui a estratégia que você seguiu neste capítulo.

- Criar um diretório para cada conjunto de problemas relacionados.
- Armazenar cada expressão necessária para os problemas em uma variável.
- Criar um subdiretório para os valores específicos em cada problema.
- Utilizar o Solver com qualquer combinação de expressões e valores.

Funções de Números Reais

Este capítulo apresenta os menus TRIG, LOGS e REAL. O menu TRIG contém funções trigonométricas e comandos pertinentes a medidas angulares. O menu LOGS contém funções logarítmicas, exponenciais e hiperbólicas. O menu REAL contém comandos adicionais para números reais.

Todos os comandos destes menus são descritos brevemente no apêndice D, "Diagramas dos Menus". Para descrições completas, ver "TRIG", "LOGS" ou "REAL", no Manual de Referência.

Funções Trigonométricas

Esta seção mostra como selecionar o modo angular corrente, calcular com π e converter medidas angulares.

Selecionando Medidas Angulares

A calculadora pode interpretar argumentos angulares e resultados como graus ($1/360$ de um círculo) ou como radianos ($1/2\pi$ de um círculo). A escolha padrão é modo angular graus (Degrees). Para os exemplos desta seção, mude para o modo angular radianos.

Apague a pilha e selecione o menu MODE.

CLEAR MODE

| |
|---------------------------|
| 3: |
| 2: |
| 1: |
| STO= FIX SCI ENG DEG= RAD |

Os dois rótulos de menu mais à direita, **DEG** (*degrees = graus*) e **RAD** (*radians = radianos*) representam suas escolhas de um modo angular. Note que o rótulo **DEG** mostra um pequeno quadrado, indicando que o modo angular corrente é graus.

Selecione modo angular radianos.

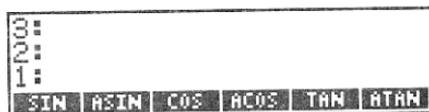
RAD



O anúncio de radianos (2π) aparece e os rótulos de menu mudam. (A maioria das ilustrações neste manual não mostra os anúncios. Para localizar o anúncio (2π), veja a ilustração à página 27.)

Apresente a primeira linha do menu TRIG.

TRIG



Estas são funções de um número, agindo sobre um número no nível 1. Para números reais, o modo angular afeta como **SIN** (*seno*), **COS** (*coseno*), e **TAN** (*tangente*) interpretam seus argumentos; e como **ASIN** (*arco seno*), **ACOS** (*arco coseno*) e **ATAN** (*arco tangente*) expressam seus resultados.

Você utilizará a função **SIN** na discussão sobre π que vem a seguir.

Utilizando π

O número transcendental π não pode ser representado exatamente em uma forma decimal finita. Em geral, a aproximação de 12 dígitos da calculadora (3.14159265359) resulta em uma precisão de 12 dígitos, a qual é suficiente para a maioria das aplicações.

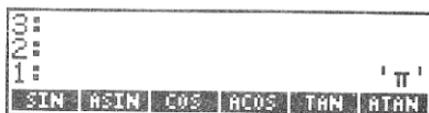
A HP-28C também oferece uma constante simbólica π que representa π exatamente. Em modo angular radianos, as funções **SIN**, **COS** e **TAN** reconhecem a constante simbólica π e produzem um resultado exato. As funções **SIN** e **COS** também reconhecem $\pi/2$.

Para outras funções, a constante simbólica π produz uma expressão contendo π . Se você forçar um número real como resultado, a calculadora utiliza a aproximação de 12 dígitos.

Para demonstrar a diferença entre 3.14159265359 e π , calcule o seno de cada um deles.

Coloque ' π ' no nível 1.

 π 

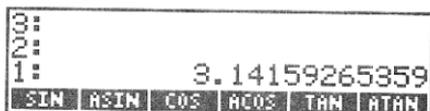


3:
2:
1: π
SIN ASIN COS ACOS TAN ATAN

Embora este objeto pareça um nome, é na realidade uma expressão com um único termo: a constante simbólica π .

Force um número real como resultado, utilizando \rightarrow NUM (para número)

 \rightarrow NUM

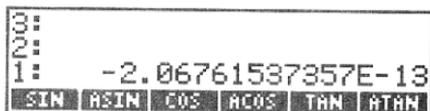


3:
2:
1: 3.14159265359
SIN ASIN COS ACOS TAN ATAN

A aproximação de 12 dígitos para π (3.14159265359) é devolvida ao nível 1.

Calcule o seno da aproximação de π .

SIN

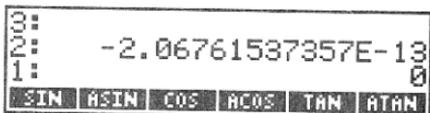


3:
2:
1: -2.06761537357E-13
SIN ASIN COS ACOS TAN ATAN

O resultado ($-2.06761537357 \times 10^{-13}$) não é exatamente 0, porque o argumento (3.14159265359) não é exatamente π .

Agora calcule o seno de π .

 π 



3:
2:
1: -2.06761537357E-13
SIN ASIN COS ACOS TAN ATAN

A função SIN reconhece a constante simbólica π e devolve o resultado exato (0).

Convertendo Medidas Angulares

O menu TRIG contém comandos que convertem um ângulo de um sistema de medidas para outro. Estes comandos estão na terceira linha do menu TRIG. Dê uma rápida examinada na segunda linha, antes de continuar para a terceira.

Apresente a segunda linha do menu TRIG.

NEXT

```
3:
2:  -2.06761537357E-13
1:  0
P→R  R→P  R→C  C→R  ARG
```

Estes comandos operam com números complexos e são duplicados no menu COMPLEX. Números complexos são descritos no próximo capítulo.

Apresente a terceira linha do menu TRIG.

NEXT

```
3:
2:  -2.06761537357E-13
1:  0
→HMS  HMS→  HMS+  HMS-  D→R  R→D
```

Você utilizará os comandos HMS→ e D→R para converter um ângulo expresso em graus, minutos e segundos para um ângulo expresso em radianos.

Os quatro comandos HMS (*horas-minutos-segundos*) permitem que você calcule com números cujas partes fracionárias estão expressas como minutos e segundos. Tais números precisam ter o seguinte formato especial, denominado o formato HMS:

$$h.MMSSs$$

onde h representa horas (ou graus), MM representa minutos, SS representa segundos e s representa fração decimal de segundos. MM e SS representam cada um dois dígitos; h e s representam qualquer número de dígitos.

Os comandos →HMS (*decimal-para-HMS*) e HMS→ (*HMS-para-decimal*) convertem um número real do formato decimal normal para o formato especial HMS e vice-versa. Os comandos HMS+ (*HMS mais*) e HMS- (*HMS menos*) adicionam e subtraem números no formato HMS, com resultado também no formato HMS.

Por exemplo, converta $141^{\circ} 26' 15''$ para graus decimais.

Entre o número no formato HMS.

141.2615

```
3: -2.06761537357E-13
1: 0
141.2615
→HMS HMS→ HMS+ HMS- D→R R→D
```

Converta o número do formato HMS para graus decimais.

HMS→

```
3: -2.06761537357E-13
2: 0
1: 141.4375
→HMS HMS→ HMS+ HMS- D→R R→D
```

As outras duas funções nesta linha de menu, $D \rightarrow R$ (*graus-para-radianos*) e $R \rightarrow D$ (*radianos-para-graus*) convertem um número real de medida angular graus para medida angular radianos e vice-versa.

Converta o número no nível 1 de graus para radianos.

D→R

```
3: -2.06761537357E-13
2: 0
1: 2.46855006079
→HMS HMS→ HMS+ HMS- D→R R→D
```

Resumindo, você calculou:

$$141^{\circ} 26' 15'' = 141.4375^{\circ} = 2.46855006079 \text{ radianos.}$$

Funções Logarítmicas, Exponenciais e Hiperbólicas

O menu LOGS contém funções logarítmicas e exponenciais, comum e natural e funções hiperbólicas. Para uma descrição detalhada destas funções, ver "LOGS" no Manual de Referência.

Apresente a primeira linha do menu LOGS.

LOGS

```
3: -2.06761537357E-13
2: 0
1: 2.46855006079
LOG ALOG LN EXP LNFI EXPM
```

As funções LOG (*logaritmo comum*) e ALOG (*antilogaritmo comum*) calculam logaritmos e exponenciais na base 10. As funções LN (*logaritmo natural*) e EXP (*exponencial natural*) calculam logaritmos e exponenciais na base e . (e é um número transcendental aproximadamente igual a 2.71828182846.)

Para um argumento x , a função LNP1 (*ln mais 1*) calcula $\ln(x + 1)$, e a função EXPM (*exp menos 1*) calcula $(\exp x) - 1$. Para argumentos próximos de 0, cada uma destas funções oferece maior precisão do que a seqüência correspondente de funções. (Um exemplo utilizando LNP1 aparece em "Valor do Dinheiro no Tempo", à página 103.)

Apresente no visor a segunda linha do menu LOGS.

NEXT

```
3: -2.06761537357E-13
2: 0
1: 2.46855006079
SINH ASINH COSH ACOSH TANH ATANH
```

Estas são as funções hiperbólicas e suas inversas: SINH (*seno hiperbólico*) e ASINH (*arco-seno hiperbólico*), COSH (*cosseno hiperbólico*) e ACOSH (*arco-cosseno hiperbólico*) e TANH (*tangente hiperbólica*) e ATANH (*arco-tangente hiperbólica*). Estas funções derivam de e^x , a função exponencial natural. Todas são funções de um número que atuam sobre o objeto no nível 1.

Outras Funções Reais

O menu REAL contém funções que se aplicam primordialmente a números reais.

Selecione o menu REAL.

REAL

```
3: -2.06761537357E-13
2: 0
1: -2.46855006079
NEG FACT RAND RDZ MAXB MINB
```

A função NEG (*tornar negativo*) devolve $-x$ para um argumento x . A função FACT (*fatorial*) devolve $n!$ para um inteiro positivo n ou a função gama $\Gamma(x + 1)$ para um argumento não inteiro x . O comando RAND (*random number = número aleatório*) devolve um número aleatório calculado de uma semente especificada por RDZ (*randomize*).

As funções MAXR (*máximo real*) e MINR (*mínimo real*) devolvem constantes simbólicas para o maior e o menor números reais positivos representáveis na HP-28S. (Para forçar um resultado numérico para uma constante simbólica, ver “Utilizando π ” à página 74.)

Esta seção mostra como utilizar a função NEG. Por conveniência, você pode executar NEG pressionando $\boxed{\text{CHS}}$ (*trocar sinal*) se não houver linha de comando presente. Para entrar o comando NEG na linha de comando — por exemplo, quando você está digitando um programa — pressione $\boxed{\text{NEG}}$ ou $\boxed{\text{N}} \boxed{\text{E}} \boxed{\text{G}}$.

Agora troque o sinal do número no nível 1 duas vezes, uma vez pressionando $\boxed{\text{CHS}}$ e outra pressionando $\boxed{\text{NEG}}$.

Troque o sinal do número no nível 1.

$\boxed{\text{CHS}}$

```
3: -2.06761537357E-13
2: 0
1: 2.46855006079
NEG FACT RAND RD2 MAXR MINR
```

Troque o sinal do número uma segunda vez.

$\boxed{\text{NEG}}$

```
3: -2.06761537357E-13
2: 0
1: 2.46855006079
NEG FACT RAND RD2 MAXR MINR
```

Definindo Novas Funções

Você pode criar variáveis programa que operam como as funções pré-programadas da HP-28S — você pode até mesmo utilizá-las em expressões. Tais variáveis programa, denominadas *funções do usuário*, precisam preencher dois requisitos:

- indicar seus argumentos explicitamente.
- devolver exatamente um resultado.

Por exemplo, você pode definir uma função COT para a função cotangente, onde $\cot x = 1/\tan x$.

Inicie o programa.

◀

```
2:
1: 2.46855006079
◀
NEG FACT RAND RDE MARK MINR
```

Indique o argumento.

▶ SPACE LC × LC

```
2:
1: 2.46855006079
◀ → x
NEG FACT RAND RDE MARK MINR
```

A seta para a direita indica que o nome seguinte representa uma *variável local*, que existe somente dentro deste programa.

É útil seguir alguma convenção para distinguir suas variáveis locais de suas variáveis ordinárias ou “globais”. Este manual utiliza letras minúsculas para distinguir variáveis locais. (Pressionar **LC** uma vez, muda para letras minúsculas; pressionar **LC** uma segunda vez, muda de volta para letras maiúsculas.)

Defina a função.

' ◀ 1/x (TAN (LC ×

```
2:
1: 2.46855006079
◀ → x' INV (TAN(x)
NEG FACT RAND RDE MARK MINR
```

Entre o programa.

ENTER

```
2:
1: ◀ → x 'INV(TAN(x))'
»
NEG FACT RAND RDE MARK MINR
```

O fechamento de parênteses e delimitadores é efetuado automaticamente.

Este programa significa: tome um argumento da pilha (em sintaxe RPN) ou da expressão (em sintaxe algébrica) e chame-o de x ; então processe a expressão $1/\tan x$, utilizando a definição local de x .

Armazene o programa na variável COT.

`' COT STO`

| | |
|-----------------------------|--------------------|
| 3: | -2.06761537957E-13 |
| 2: | 0 |
| 1: | 2.46855006079 |
| NEG FACT RAND RDE MARK MINB | |

Agora, você pode utilizar COT tanto em sintaxe RPN como algébrica, da mesma forma que as funções trigonométricas pré-programadas.

Calcular $\cot 45^\circ$ utilizando RPN.

`MODE DEG`
45 `USER COT`

| | |
|-----------------|---------------|
| 3: | 0 |
| 2: | 2.46855006079 |
| 1: | 1 |
| COT EE D RENA . | |

Calcule $\cot -45^\circ$ utilizando sintaxe algébrica.

`' COT (-45 ENTER`

| | |
|---------------|---------------|
| 3: | 2.46855006079 |
| 2: | 1 |
| 1: | 'COT(-45)' |
| COT EE D RENA | |

Processe a expressão.

`EVAL`

| | |
|---------------|---------------|
| 3: | 2.46855006079 |
| 2: | 1 |
| 1: | -1 |
| COT EE D RENA | |

6

Funções com Números Complexos

A HP-28S inclui um tipo de objeto que representa números complexos. Por exemplo, o número complexo $z = 3 + 4i$ é representado pelo objeto $(3, 4)$. Como cada número complexo é um único objeto, você pode calcular com números complexos tão facilmente quanto com números reais.

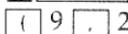
O par de números reais num número complexo pode representar as coordenadas de um ponto em um plano. Por exemplo, a HP-28S utiliza números complexos para representar coordenadas para traçar um gráfico. A segunda seção, neste capítulo, descreve dois sistemas de coordenadas, retangulares e polares, e mostra como converter um ponto de um sistema para o outro.

Utilizando Números Complexos

A maioria das funções que são válidas para números reais, são válidas da mesma maneira para números complexos. Por exemplo, você efetua cálculos aritméticos com números complexos da mesma forma que você o faz com números reais — coloca os números na pilha e executa a função. Tente calcular o seguinte:

$$((9 + 2i) + (-4 + 3i)) \times (6 + i)$$

Apague a pilha e entre $9 + 2i$.

| | |
|---------------|-------|
| 3: | |
| 2: | |
| 1: | (9,2) |
| COT EE 0 RENR | |

Adicione $-4 + 3i$. (Lembre-se de pressionar $\boxed{4}$ $\boxed{\text{CHS}}$ para entrar -4 .)

$\boxed{(-}$ $\boxed{4}$ $\boxed{,}$ $\boxed{3}$ $\boxed{+}$

```
3:
2:
1: (5,5)
COT EE D REIM
```

Multiplique por $6 + i$.

$\boxed{(}$ $\boxed{6}$ $\boxed{,}$ $\boxed{1}$ $\boxed{\times}$

```
3:
2:
1: (25,35)
COT EE D REIM
```

Algumas vezes um argumento real pode produzir um número complexo como resultado.

Calcule $\sqrt{-4}$.

-4 $\boxed{\sqrt{x}}$

```
3:
2: (25,35)
1: (0,2)
COT EE D REIM
```

Calcule arco-seno 2.

2 $\boxed{\text{TRIG}}$ $\boxed{\text{ASIN}}$

```
2: (0,2)
1: (1.57079632679,
-1.31695789692)
SIN ASIN COS ACOS TAN ATAN
```

Funções especificamente para números complexos estão no menu COMPLEX.

Selecione o menu COMPLEX.

$\boxed{\text{COMPLX}}$

```
2: (0,2)
1: (1.57079632679,
-1.31695789692)
R+C C+R RE IM CONJ SIGN
```

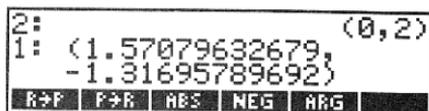
Todos os comandos no menu COMPLEX são descritos brevemente no apêndice D, "Diagramas dos Menus". Para descrições completas, ver "COMPLEX" no Manual de Referência.

- $R \rightarrow C$ (*real-para-complexo*) converte dois números reais x e y em um número complexo (x,y) .
- $C \rightarrow R$ (*complexo-para-real*) converte um número complexo (x, y) em dois números reais x e y .
- RE (*parte real*) devolve x para um argumento complexo (x, y) .

- IM (*parte imaginária*) devolve y para um argumento complexo (x, y) .
- CONJ (*conjugado*) devolve $(x, -y)$ para um argumento complexo (x, y) .
- SIGN devolve $(x/\sqrt{x^2 + y^2}, y/\sqrt{x^2 + y^2})$ para um argumento complexo (x, y) .

Apresente no visor a próxima linha do menu COMPLEX.

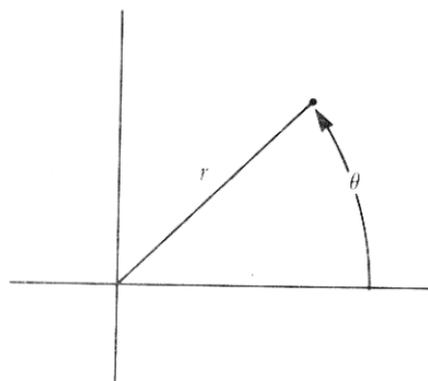
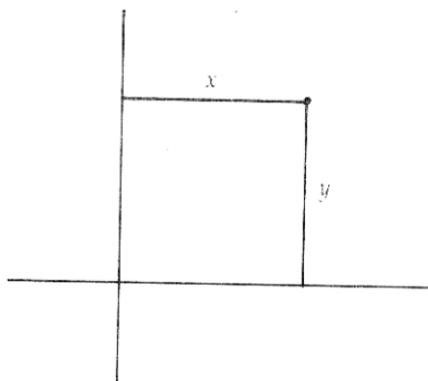
NEXT



Estas funções (com exceção de NEG) relacionam-se com números complexos em coordenadas polares.

Utilizando Coordenadas Polares

Um ponto em um plano pode ser descrito por dois sistemas diferentes de coordenadas. A ilustração abaixo mostra um ponto descrito de duas maneiras, em notação retangular (x, y) e em notação polar (r, θ) .



- $R \rightarrow P$ (*retangular-para-polar*) converte um número complexo em notação retangular (x, y) para notação polar (r, θ) .
- $P \rightarrow R$ (*polar-para-retangular*) converte um número complexo em notação polar (r, θ) para notação retangular (x, y) .
- ABS (*valor absoluto*) devolve r para um argumento complexo (x, y)
- NEG devolve $(-x, -y)$ para um argumento complexo (x, y) .
- ARG devolve θ para um argumento complexo (x, y) .

Note que somente $P \rightarrow R$ interpreta um número complexo como coordenadas polares; todas as outras funções — aritmética, trigonométrica, logarítmica, hiperbólica etc — interpretam um número complexo como coordenadas retangulares. Lembre-se desta regra importante: *qualquer número complexo em coordenadas polares precisa ser convertido para coordenadas retangulares antes que você possa utilizá-lo em um cálculo*.

Como um exemplo de aritmética com coordenadas polares, suponha que você viaje duas milhas com a bússola indicando 36° e, então, três milhas com a indicação de 65° . Qual é a distância resultante e a direção, calculadas com duas casas decimais?

Selecione modo angular graus (DEG) e FIX 2 para apresentação dos números.

MODE DEG 2 FIX

| | |
|---------------------------|---------------|
| 3: | (25.00,35.00) |
| 2: | (0.00,2.00) |
| 1: | (1.57,-1.32) |
| STO FIX= SCI ENG DEG= RAD | |

Entre a primeira distância e a direção angular.

(2 , 36

| | |
|---------------------------|--------------|
| 2: | (0.00,2.00) |
| 1: | (1.57,-1.32) |
| (2,36 | |
| STO FIX= SCI ENG DEG= RAD | |

Converta para coordenadas retangulares.

COMPLX **NEXT** **P→R**

| | |
|---------------------|--------------|
| 3: | (0.00,2.00) |
| 2: | (1.57,-1.32) |
| 1: | (1.62,1.18) |
| R→P P→R ABS NEG ARG | |

Entre a segunda distância e a direção angular.

(3 , 65

| | |
|---------------------|--------------|
| 2: | (1.57,-1.32) |
| 1: | (1.62,1.18) |
| (3,65 | |
| R→P P→R ABS NEG ARG | |

Converta para coordenadas retangulares.

P→R

```
3: (1.57,-1.32)
2: (1.62,1.18)
1: (1.27,2.72)
R→P P→R ABS NEG ARG
```

Adicione as coordenadas retangulares.

+

```
3: (0.00,2.00)
2: (1.57,-1.32)
1: (2.89,3.89)
R→P P→R ABS NEG ARG
```

Converta para coordenadas polares.

R→P

```
3: (0.00,2.00)
2: (1.57,-1.32)
1: (4.85,53.46)
R→P P→R ABS NEG ARG
```

A distância resultante é 4.85 milhas, e a direção angular resultante é 53.46°.

Uma Função do Usuário para Adição Polar

Aqui está um programa simples, PSUM (*soma polar*), para automatizar o processo que você executou manualmente, na seção anterior.

Inicie o programa.

«

```
2: (1.57,-1.32)
1: (4.85,53.46)
«
R→P P→R ABS NEG ARG
```

Indique os argumentos. (Use um espaço para separar os dois argumentos.)

→ LC × SPACE y

```
2: (1.57,-1.32)
1: (4.85,53.46)
«→ x y
R→P P→R ABS NEG ARG
```

A seta à direita indica que os nomes seguintes são *variáveis locais*, que só existirão dentro desse programa.

Defina a função.

$\boxed{R \rightarrow P}$ $\boxed{(}$ $\boxed{P \rightarrow R}$ $\boxed{(}$ $\boxed{\times}$ $\boxed{)}$ $\boxed{+}$
 $\boxed{P \rightarrow R}$ $\boxed{(}$ \boxed{y} \boxed{ENTER}

```

2: (4.85,53.46)
1: * → × y 'R→P(P→R(x)+
   P→R(y))' *
R→P P→R ABS NEG ARG
  
```

O fechamento do parênteses e delimitadores é colocado automaticamente.

Este programa significa: tome dois argumentos da pilha (em sintaxe RPN) ou da expressão (em sintaxe algébrica) e chame-os x e y ; então calcule as coordenadas polares da soma das coordenadas retangulares de x e y .

Armazene o programa na variável PSUM.

\boxed{PSUM} \boxed{STO}

```

3: (0.00,2.00)
2: (1.57,-1.32)
1: (4.85,53.46)
R→P P→R ABS NEG ARG
  
```

Agora, utilize PSUM para repetir o cálculo anterior, uma vez em sintaxe RPN e outra em sintaxe algébrica.

Entre a primeira distância e a direção angular.

$\boxed{(}$ $\boxed{2}$ $\boxed{,}$ $\boxed{36}$ \boxed{ENTER}

```

3: (1.57,-1.32)
2: (4.85,53.46)
1: (2.00,36.00)
R→P P→R ABS NEG ARG
  
```

Entre a segunda distância e a direção angular.

$\boxed{(}$ $\boxed{3}$ $\boxed{,}$ $\boxed{65}$

```

2: (4.85,53.46)
1: (2.00,36.00)
(3,65
R→P P→R ABS NEG ARG
  
```

Execute PSUM.

\boxed{USER} \boxed{PSUM}

```

3: (1.57,-1.32)
2: (4.85,53.46)
1: (4.85,53.46)
PSUM COT EE D RENN
  
```

O resultado coincide com a resposta anterior.

Agora tente a sintaxe algébrica.

`PSUM ((2,36) (3,65))`
`65` `ENTER`

| | | | | |
|------|---------------------|----|---|------|
| 2: | (4.85,53.46) | | | |
| 1: | PSUM((2,36),(3,65)) | | | |
| PSUM | COT | EE | D | RENR |

Os parênteses externos e a vírgula central definem os argumentos para PSUM; os outros parênteses e vírgulas são partes da sintaxe dos números complexos. Não esqueça que você necessita de *dois* conjuntos de parênteses, quando utiliza um número complexo como um argumento em sintaxe algébrica.

Processe a expressão.

`EVAL`

| | | | | |
|------|--------------|----|---|------|
| 3: | (4.85,53.46) | | | |
| 2: | (4.85,53.46) | | | |
| 1: | (4.85,53.46) | | | |
| PSUM | COT | EE | D | RENR |

Traçando Gráficos

Este capítulo introduz traçado de gráficos na HP-28S. Traçar gráficos é útil por si só, dando um entendimento visual de como uma expressão ou equação se comporta. Em adição, o traçado do gráfico de uma expressão torna fácil a estimativa de raízes, máximos ou mínimos. O próximo capítulo, "O Solver", mostra como utilizar o Solver para transformar estimativas em números precisos.

Neste capítulo, você aprenderá como utilizar alguns dos comandos no menu PLOT. Todos os comandos no menu PLOT são descritos brevemente no apêndice D, "Diagramas dos Menus". Para descrições completas, ver "PLOT" no Manual de Referência.

Como primeiro exemplo, você traçará o gráfico de $\sin x$ em modo angular radianos, mas existem passos preliminares para assegurar que o seu visor coincidirá com as ilustrações.

O traçado de gráficos utiliza uma variável denominada PPAR, para armazenar uma lista de parâmetros de traçado de gráficos. Elimine ("purge") qualquer PPAR existente, para assegurar que o próximo gráfico utilizará os parâmetros padrão de traçado de gráficos.

Apague a pilha e selecione o menu PLOT.



Apresente a segunda linha do menu PLOT.



Elimine (“purge”) qualquer PPAR existente.

PPAR PURGE

```
3:
2:
1:
PPAR RES AWES CENTR HW SH
```

Selecione modo angular radianos e modo de apresentação de números STD.

MOD RAD STD

```
3:
2:
1:
STD= FIX SCI ENG DEG RAD=
```

Agora entre a expressão.

TRIG SIN × ENTER

```
3:
2:
1:
'SIN(X) '
SIN ASIN COS ACOS TAN ATAN
```

Armazene a expressão como *equação corrente* — uma variável normal com o nome especial EQ. (Esta é a mesma convenção que você seguiu com o Solver, no capítulo 4.)

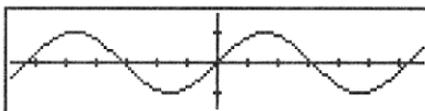
PLOT STEQ

```
3:
2:
1:
STEQ RCEQ PMIN PMAW INDEF DRAW
```

Pressionar STEQ é equivalente a pressionar EQ STO.

Trace o gráfico da expressão.

DRAW



Espera até o anúncio ((•)) desaparecer, indicando que o gráfico está completo.

A linha horizontal é o eixo para a variável independente (x neste exemplo), e a linha vertical é o eixo para a variável dependente (o valor da expressão $\sin x$). As marcas, em ambos os eixos, assinalam intervalos de comprimento 1.

Imprimindo um Gráfico

Se você possui uma impressora HP 82240A, você pode imprimir uma imagem do gráfico que acabou de traçar da seguinte maneira:

1. posicione a impressora de acordo com as instruções no manual da impressora.
2. pressione e mantenha pressionado .
3. pressione (a tecla com "PRINT" escrito acima dela).
4. solte .

Esta seqüência de teclas é o equivalente, no teclado, ao comando PRLCD (*imprima LCD*, encontrada na primeira linha do menu PRINT). Você pode utilizar estas teclas para imprimir o visor, praticamente, a qualquer tempo, sem perturbar a operação da calculadora.

Se você escrever um programa para traçar o gráfico de uma expressão e imprimir o resultado, utilize a seguinte seqüência de comandos:

...CLLCD DRAW PRLCD...

Retornando ao exemplo presente, restabeleça a apresentação normal da pilha no visor.



```
3:  
2:  
1:  
-----  
STEQ RCEQ PMIN PMAX INDEF DRAW
```

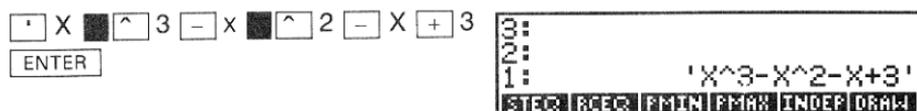
Mudando a Escala do Gráfico

Em geral, traçar o gráfico de uma expressão não produz resultados elegantes da primeira vez. Quando você estiver traçando o gráfico de uma expressão, que não lhe é familiar, poderá ser necessário ajustar a região do traçado do gráfico — definida pelos parâmetros de traçado do gráfico — para mostrar as características relevantes da expressão.

Se você sabe, antecipadamente, a região que deseja mostrar no gráfico, pode mudar diretamente os parâmetros de traçado de gráfico em PPAR. (PPAR é descrito em detalhe em "PLOT", no Manual de Referência). Mais freqüentemente, necessita experimentar, para encontrar a região que você deseja traçar no gráfico. Este manual mostra como utilizar os comandos no menu PLOT para convergir suas tentativas para o traçado desejado.

Para o segundo exemplo, você traçará o gráfico da expressão $x^3 - x^2 - x + 3$.

Coloque a expressão no nível 1.



Armazene a expressão como a equação corrente.



Trace o gráfico da expressão.

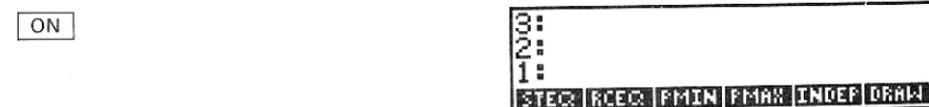


A linha horizontal é o eixo para x e a linha vertical é o eixo para o valor da expressão $x^3 - x^2 - x + 3$.

Este gráfico mostra um *zero* da expressão — um valor de X para o qual o valor da expressão é zero. O zero está localizado onde o gráfico da expressão cruza o eixo X . No próximo capítulo, utilizaremos o Solver para encontrar o número preciso para este zero.

Para mostrar mais do gráfico, expanda a escala vertical e trace o gráfico novamente.

Restabeleça a apresentação normal da pilha no visor.



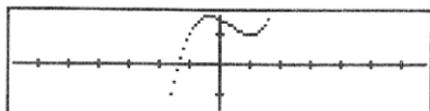
Expanda a altura por um fator de escala de 2, utilizando a operação *H (vezes altura) na próxima linha do menu.

NEXT 2 *H



Trace o gráfico novamente com os novos parâmetros de traçado.

PREV DRAW



As marcas no eixo horizontal ainda assinalam intervalos de comprimento 1, mas agora as marcas no eixo vertical marcam intervalos de comprimento 2.

A seguir você transladará o gráfico, movendo a parte interessante para o centro do visor.

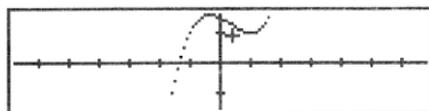
Transladando o Traçado de um Gráfico

Após traçar cada gráfico, a calculadora deixa uma retícula no centro do visor. (Você não pode ver a retícula quando os eixos estão no centro do visor.) Você pode utilizar a retícula para *digitalizar* qualquer ponto no visor, devolvendo as coordenadas do ponto para a pilha. Digitalizaremos o ponto que queremos que seja o centro do próximo gráfico e utilizá-lo-emos para ajustar os parâmetros de traçado de gráficos.

Mova a retícula para a posição indicada.

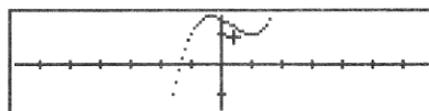
▶ (pressione 4 vezes)

▲ (pressione 9 vezes)



Digitalize o ponto.

INS



Retorne para a apresentação da pilha no visor.

ON

```
3:
2:
1: (.4,1.8)
STEP RECAL FMIN FMAX INDEF DRAW
```

As coordenadas do ponto digitalizado, representados por um número complexo, estão no nível 1.

Redefina o centro do gráfico, utilizando **CENTR** na próxima linha do menu.

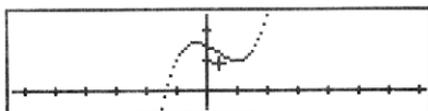
NEXT **CENTR**

```
3:
2:
1:
FPAR RES AXES CENTR *W *H
```

As coordenadas são tomadas da pilha e utilizadas para ajustar os parâmetros de traçado de gráficos. Diversamente de ***H**, **CENTR** não muda a escala.

Tente traçar outro gráfico.

PREV DRAW

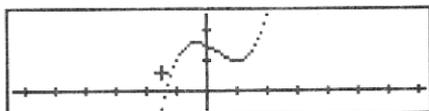


Agora, amplie uma parte interessante do gráfico. Você poderia utilizar ***H** novamente, com um fator de escala *fracionário*. (Por exemplo, um fator de escala .5 retornaria a escala do eixo vertical ao seu valor original.) Mas existe uma maneira mais flexível de ampliar-se uma parte de um gráfico.

Redefinindo os Extremos do Gráfico

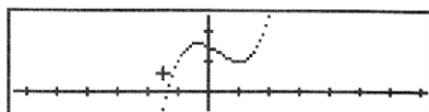
Desta vez, você digitalizará dois pontos: um para indicar o canto esquerdo inferior do novo gráfico e um para o canto direito superior, para ampliar a porção delimitada do gráfico.

Mova a retícula para o canto inferior esquerdo desejado.

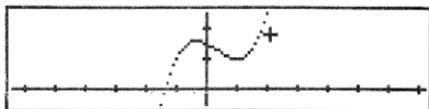


Digitalize o ponto.

INS

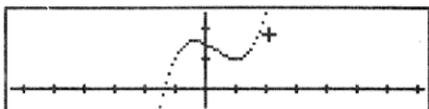


Mova a retícula para o canto superior direito desejado.



Digitalize o ponto.

INS



Retorne para a apresentação da pilha no visor.

ON



As coordenadas do canto inferior esquerdo, representadas por um número complexo, estão no nível 2. As coordenadas do canto superior direito estão no nível 1. (Suas coordenadas podem diferir ligeiramente das da ilustração.)

Redefina o canto superior direito do gráfico, utilizando `PMAX` (*plot maxima = máximo do gráfico*).

PMAX



As coordenadas são tomadas da pilha e utilizadas para ajustar os parâmetros de traçado de gráficos.

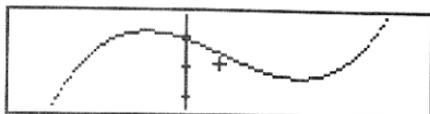
Redefina o canto inferior esquerdo do gráfico, utilizando `PMIN` (*plot minima = mínimo do gráfico*).

PMIN



Tente traçar outro gráfico.

DRAW



Uma vez que você mudou a altura e a largura do gráfico, tanto a escala vertical como a horizontal foram mudadas.

O traçado do gráfico mostra dois *extremos* no gráfico da expressão — um máximo local e um mínimo local. No próximo capítulo, você utilizará o Solver para encontrar um valor preciso para o mínimo. Para evitar a repetição de todos estes passos para gerar nossos parâmetros correntes de traçado de gráficos, armazene o valor corrente de PPAR em uma variável com um nome diferente. Para recriar este gráfico no próximo capítulo, você restaurará PPAR para seu valor corrente.

Retorne para a apresentação da pilha no visor.

ON

```
3:
2:
1:
-----
STEQ  RCEQ  PMIN  PMAX  INDEF  DRAW
```

Coloque o valor corrente de PPAR na pilha.

NEXT PPAR

```
1: ( (-1.5,1.2)
    (2.1,3.6) X 1 (0,0)
  )
-----
PPAR  RES  AXES  CENTR  XW  XH
```

Para informações sobre os parâmetros de traçado de gráficos e para detalhes sobre traçado de gráficos em geral, ver “PLOT” no Manual de Referência.

Crie a variável PPAR1 que contém os parâmetros de traçado de gráficos correntes.

PPAR 1 STO

```
3:
2:
1:
-----
PPAR  RES  AXES  CENTR  XW  XH
```

Agora, você está pronto para utilizar o Solver a fim de encontrar números precisos para o zero e o mínimo local da expressão.

Traçando Gráficos de Equações

Os exemplos, neste capítulo, eram ambos expressões, mas as mesmas regras e técnicas se aplicam para traçar gráficos de equações. Quando a variável EQ contém uma equação, DRAW traça o gráfico de *cada membro* da equação como uma expressão. Você pode encontrar uma raiz da equação, encontrando onde os dois gráficos se cruzam, porque aí é onde os dois membros da equação têm valores iguais.

O Solver

Este capítulo descreve como encontrar uma raiz e um mínimo da expressão da qual você traçou o gráfico no capítulo anterior. Execute os passos do capítulo anterior, se ainda não o fez, porque você necessitará de alguns dos resultados.

Veja uma descrição completa do Solver, em "SOLVE" no Manual da Referência.

Encontrando um Zero de uma Expressão

O exemplo seguinte admite que a expressão $x^3 - x^2 - x + 3$ é ainda a equação corrente e que você criou a variável PPAR1, como descrito no capítulo anterior. Você traçará o gráfico da expressão novamente, digitalizará uma estimativa para um zero da expressão e, então, utilizará o Solver para encontrar um valor mais preciso para o zero.

Antes de iniciar estes exemplos, apague a pilha, selecione modo angular radianos e modo de apresentação de números FIX 2.

CLEAR
 MODE RAD
 2 FIX

```

0:
2:
1:
STD  FIX=  SCI  ENG  DEG  RAD=
  
```

Elimine ("purge") o PPAR existente para assegurar que o próximo gráfico utilizará os parâmetros padrão de traçado de gráficos.

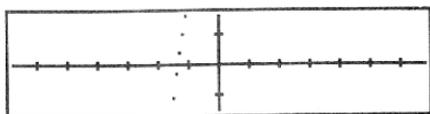
PLOT NEXT
 PPAR PURGE

```

0:
2:
1:
PPAR  RES  ANES  CENTR  %W  %H
  
```

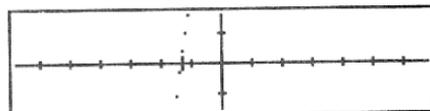
Agora, trace o gráfico da expressão.

PREV **DRAW**



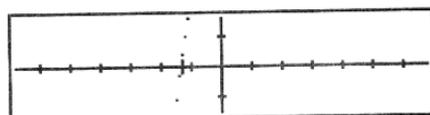
Este gráfico mostra um *zero* da expressão — um valor de X para o qual o valor da expressão é zero. O zero é localizado onde o gráfico da expressão cruza o eixo horizontal.

Mova a retícula para a intersecção aproximada do gráfico com o eixo horizontal. (Utilize **▲**, **▼**, **◀** e **▶** para mover a retícula.)



Digitalize esta estimativa para a raiz.

INS



Você utilizará este ponto como uma estimativa para encontrar o zero exato da expressão. (Caso a expressão tenha mais do que uma raiz, a estimativa indica a que você quer.)

Retorne para a apresentação da pilha no visor.

ON

```
3:
2:
1: (-1.30,0.00)
STEQ RCEQ FMIN FMAX UNDEF DRAW
```

As coordenadas do ponto digitalizado, representadas por um número complexo, estão no nível 1. (Suas coordenadas podem diferir ligeiramente das da ilustração.)

Selecione o menu Solver.

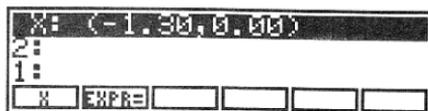
SOLV **SOLVR**

```
3:
2:
1: (-1.30,0.00)
X EXPR=
```

O menu Solver mostra todas as variáveis na equação corrente (somente X neste exemplo).

Armazene a estimativa digitalizada na variável X.

X

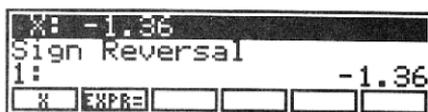


X: (-1.36, 0.00)
2:
1:
X EXPRE

Embora o ponto digitalizado contenha duas coordenadas, o Solver utilizará apenas a primeira coordenada como uma estimativa.

Agora encontre X.

X

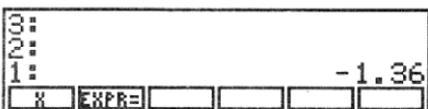


X: -1.36
Sign Reversal
1: -1.36
X EXPRE

A mensagem SIGN REVERSAL (inversão de sinal) indica que o Solver encontrou uma solução aproximada, correta até 12 dígitos. Se o Solver encontrasse uma solução exata, apresentaria a mensagem ZERO. Estas mensagens, denominadas *mensagens de qualificação*, são discutidas em "SOLVE", no Manual de Referência.

Retorne para a apresentação da pilha do visor.

ON



0:
1:
1: -1.36
X EXPRE

Encontrando um Mínimo ou Máximo

Para encontrar o zero de uma expressão, o Solver faz uma amostragem de pontos no gráfico, iniciando com sua estimativa, e tenta encontrar pontos cada vez mais próximos do eixo x . Se sua estimativa estiver bem próxima de um *mínimo local positivo* ou *máximo local negativo*, não existem pontos na proximidade que estejam mais próximos do eixo x . Neste caso, o Solver encontra aquele extremo (mínimo ou máximo) ao invés de um zero. (Geralmente o Solver não ficará "preso" em um extremo, a menos que sua estimativa o force para lá.)

Olhe para o gráfico que você fez no último capítulo, à página 96. Ele mostra que a expressão tem um mínimo local positivo e um máximo local positivo. O Solver pode encontrar o mínimo, porque localmente é o ponto *mais próximo* do eixo x ; mas o Solver não pode encontrar o máximo, porque localmente é o ponto *mais distante* do eixo x .

Nesta seção, você traçará o gráfico da expressão, utilizando os parâmetros de traçado de gráficos armazenados na variável PPAR1 e, então, digitalizará três pontos para estimar o mínimo e utilizará o Solver, a fim de encontrar um mínimo mais preciso.

Recupere a lista armazenada em PPAR1 para a pilha.

USER PPAR1

```
1: { (-1.50,1.20)
      (2.10,3.60) X 1.00
      (0.00,0.00) }
% PPAR PPAR1 EQ PSUM COT
```

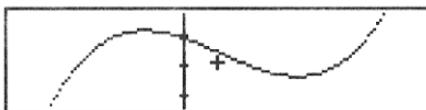
Restabeleça a variável PPAR para os valores armazenados em PPAR1.

' PPAR STO

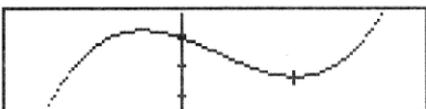
```
3:
2:
1: -1.36
% PPAR PPAR1 EQ PSUM COT
```

Trace o gráfico da expressão.

■ PLOT DRAW

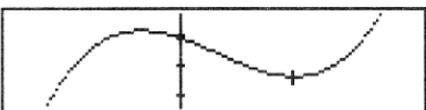


Mova a retícula para a posição imediatamente à esquerda do mínimo.

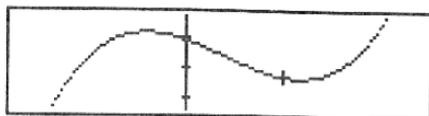


Digitalize o ponto.

INS

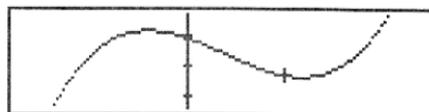


Mova a retícula para a posição imediatamente à esquerda do mínimo.



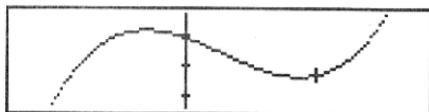
Digitalize o ponto.

INS



Mova a retícula para a posição imediatamente à direita do mínimo e digitalize o ponto.

INS



Volte para a apresentação da pilha no visor.

ON

```
3: (0.99,1.97)
2: (0.86,2.05)
1: (1.15,2.05)
STEQ RCEQ PMIN PMAX INDEP DRAW
```

Os três pontos estão nos níveis 1, 2 e 3. (Seus pontos podem diferir ligeiramente dos da ilustração.)

Agora, combine as três estimativas em uma lista. Fazendo isso, você pode manipular as três estimativas como um único objeto. Esta é uma utilização típica para listas — combinar diversos objetos em um.

LIST 3 →LIST

```
1: { (0.99,1.97)
      (0.86,2.05)
      (1.15,2.05) }
→LIST LIST→ PUT GET PUTI GETI
```

Selecione o menu Solver.

SOLV SOLVR

```
1: { (0.99,1.97)
      (0.86,2.05)
      (1.15,2.05) }
X EXP=
```

O menu Solver mostra todas as variáveis na equação corrente (somente X neste exemplo).

Armazene a lista de pontos na variável X.

X

```
X: (0.99,1.97) (0.8..
2:
1: -1.36
X  EXPR=
```

A lista de pontos é retirada da pilha e armazenada na variável X como estimativas iniciais.

Calcule X.

X

```
X: 1.00
Extremum
1: 1.00
X  EXPR=
```

A mensagem EXTREMUM indica que o Solver encontrou um ponto extremo da expressão.

Volte para a apresentação da pilha no visor.

ON

```
3:
2: -1.36
1: 1.00
X  EXPR=
```

Calcule o valor extremo.

EXPR=

```
EXPR=2.000
2: 1.00
1: 2.00
X  EXPR=
```

O valor mínimo é 2.

Valor do Dinheiro no Tempo

Esta seção mostra como utilizar o Solver com cálculos de valor do dinheiro no tempo (TVM = Time Value of Money). Para um número n de períodos, uma taxa de juros periódica de $i\%$, pagamento $\$pmt$, valor presente $\$pv$ e um valor futuro de $\$fv$ a fórmula de TVM é:

$$(1 - sppv) \times pmt \times (100/i) + pv = -fv \times sppv$$

Onde

sppv (single payment present value - valor presente

$$\text{para pagamento \u00fanico} = (1 + i/100)^{-n}$$

$$= \exp(-n \times \ln(1 + i/100)).$$

Esta express\u00e3o pressup\u00f5e que os pagamentos s\u00e3o efetuados ao final de cada per\u00edodo.

Aqui est\u00e3o os principais passos que voc\u00ea executar\u00e1:

1. digite a express\u00e3o para *sppv* e armazene-a na vari\u00e1vel SPPV.
2. digite a equa\u00e7\u00e3o e armazene-a na vari\u00e1vel TVM.
3. fa\u00e7a TVM a equa\u00e7\u00e3o corrente.
4. utilize o Solver para calcular qualquer uma das cinco vari\u00e1veis *n*, *i*, *pmt*, *pv*, ou *fv*, para valores conhecidos das outras quatro vari\u00e1veis.

Antes de iniciar, apague a pilha e selecione o modo de apresenta\u00e7\u00e3o de n\u00fameros FIX 2.

CLEAR
MODE 2 **FIX**

```
3:  
2:  
1:  
STO FIX SCI ENG DEG RAD
```

Passo 1: Digite a express\u00e3o para *sppv* e armazene-a na vari\u00e1vel SPPV.

Digite a express\u00e3o para *sppv*.

' **LOGS** **EXP** **-** **N** **x** **LNP1**
I **+** **100** **ENTER**

```
2:  
1: 'EXP(-N*LNP1(I/100))  
LOG RLOG LN EXP LNP1 EXPM
```

Esta express\u00e3o tira partido da maior precis\u00e3o de LNP1 para calcular $\ln(1 + i/100)$.

Crie a vari\u00e1vel SPPV e verifique o menu USER.

' **SPPV** **STO**
USER

```
3:  
2:  
1:  
SPPV X PPAR PPARL EQ PSUM
```

Passo 2: Digite a equação e armazene-a na variável TVM.

Digite a equação para TVM.

' (1 - SPPV) × PMT ×
 100 ÷ I + PV = - FV × SPPV
 ENTER



Crie a variável TVM.

' TVM STO



O menu USER mostra um novo rótulo para TVM.

Passo 3: Faça TVM a equação corrente.

Digite o nome TVM.

' TVM



Armazene o nome TVM na variável EQ.

SOLV STEQ



Passo 4: Utilize o Solver para calcular quaisquer das cinco variáveis n , i , pmt , pv , ou fv , para valores conhecidos das outras quatro variáveis.

Selecione o menu Solver.

SOLVR



Todas as variáveis em TVM e SPPV aparecem no menu. (As variáveis em SPPV aparecem porque a equação corrente, TVM, contém SPPV.)

Dados os valores $N = 30 \times 12$, $I = 11.5/12$, $PMT = -630$, e $FV = 0$, calcule PV. (PMT tem um valor negativo porque dinheiro pago é um número negativo, enquanto dinheiro recebido é um número positivo.)

Primeiro atribua o valor a N.

30 12

```
NR: 30.00
2:
1:
N I PMT PV FV LEFT=
```

Atribua o valor a I.

11.5 12

```
I: 0.96
2:
1:
N I PMT PV FV LEFT=
```

Atribua o valor a PMT.

-630

```
PMT: -630.00
2:
1:
N I PMT PV FV LEFT=
```

Atribua o valor a FV.

0

```
FV: 0.00
2:
1:
N I PMT PV FV LEFT=
```

Agora calcule PV.

```
PV: 63617.64
zero
1: 63617.64
N I PMT PV FV LEFT=
```

A mensagem ZERO indica que o valor calculado satisfaz exatamente a equação corrente.

Soluções Simbólicas

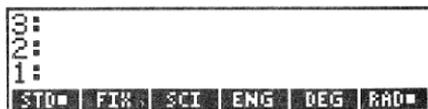
Este capítulo descreve dois métodos para encontrar-se soluções simbólicas. Existe um método simples para resolver-se uma equação quadrática, calculando-se a expressão linear que representa ambos os zeros. Existe também um método mais versátil que fornece uma solução simbólica para uma variável em equações mais gerais.

Cada um dos métodos se aplica tanto com expressões como com equações. O zero de uma expressão $f(x)$ é o mesmo que a raiz da equação $f(x) = 0$, e a raiz da equação $f(x) = g(x)$ é o mesmo que o zero da expressão $f(x) - g(x)$.

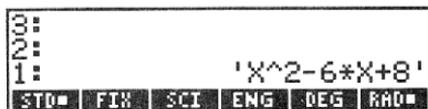
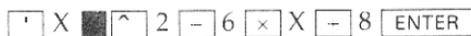
Encontrando os Zeros de uma Expressão Quadrática

Você pode encontrar ambos os zeros de uma expressão quadrática, sem traçar o gráfico ou efetuar estimativas. O seguinte exemplo resolve $x^2 - 6x + 8$.

Antes de iniciar o exemplo, apague a pilha e selecione o formato de apresentação de números STD.



Coloque a expressão na pilha.



Coloque o nome X na pilha, indicando a incógnita.

```
3:
2:
1:
'X^2-6*X+8'
'X'
```

STD= FIX SCI ENG DEG RAD=

Calcule os zeros, utilizando QUAD (quadrática) no menu SOLVE.

```
3:
2:
1:
'(6+s1*2)/2'
```

STEQ RCEQ SOLVR ISOL QUAD SHOW

Esta expressão representa as duas soluções para a expressão quadrática. A variável s1 representa um sinal arbitrário, ou +1 ou -1, e cada valor de s1 corresponde a um zero da expressão.

Armazene a expressão como equação corrente.

```
3:
2:
1:
'X^2-6*X+8'
```

STEQ RCEQ SOLVR ISOL QUAD SHOW

Apresente o menu Solver.

```
3:
2:
1:
s1 EXPR=
```

s1 é a única variável na equação corrente.

Primeiro faça s1 ser um sinal positivo.

1

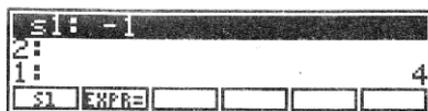
```
s1: 1
2:
1:
s1 EXPR=
```

Devolva uma das soluções ao nível 1.

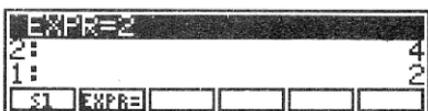
```
EXPR=4
2:
1:
s1 EXPR=
```

Agora faça s1 ser um sinal negativo.

-1



Devolva a segunda solução ao nível 1.



As duas raízes de $x^2 - 6x + 8$ são $x = 4$ e $x = 2$.

Isolando uma Variável

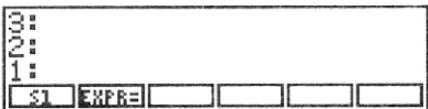
A HP-28S pode isolar uma única ocorrência de uma variável em uma equação, devolvendo uma expressão que representa a solução simbólica da equação. Em outras palavras, se x é a incógnita e a , b e c são as outras variáveis na equação, isolar x produz uma expressão em a , b e c , tal que a equação é satisfeita quando x tem o valor da expressão.

Para o primeiro exemplo, isole x na equação.

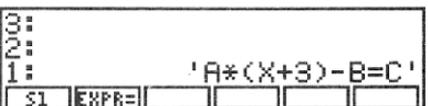
$$a(x + 3) - b = c.$$

Este exemplo é simples porque existe somente uma ocorrência de x . Exemplos mais à frente mostram como manipular equação para produzir uma única ocorrência de x .

Apague a pilha.

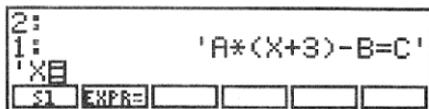


Coloque a equação na pilha.



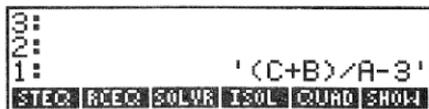
Especifique a variável que você deseja isolar.

X



Isolar x , utilizando ISOL (*isolar*) no menu SOLVE.

SOLV ISOL



A expressão devolvida representa uma solução simbólica da equação para x , isto é, a equação

$$a(x + 3) - b = c$$

é satisfeita quando $x = (c + b) / a - 3$.

Desenvolvendo e Agrupando

Se x ocorre mais de uma vez, você precisa manipular a equação para eliminar todas, exceto uma, ocorrências x . O exemplo a seguir mostra como isolar x na equação

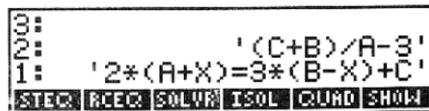
$$2(a + x) = 3(b - x) + c.$$

A estratégia para este exemplo é desenvolver (“expand”) a equação, subtrair, de ambos os membros da equação, o termo contendo x de um dos membros; agrupar (“collect”) os termos da equação, para cancelar os termos contendo x de um dos membros e obter um único termo em x no outro membro e, então, isolá-lo.

Coloque a equação na pilha.

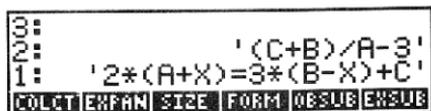
2 × (A + X) = 3 × (B

- X) + C ENTER



Selecione o menu ALGEBRA.

ALGEBRA



Neste exemplo, você utilizará **EXPAN** (*expand = desenvolver*) e **COLCT** (*collect = agrupar*) para manipular a equação. No próximo exemplo, você utilizará **FORM** (*formar expressão algébrica*) para manipular uma equação. Todos os comandos no menu ALGEBRA são descritos brevemente no apêndice D, “Diagramas dos Menus”. Para descrições completas, ver “ALGEBRA” no Manual de Referência. Em adição, FORM, um poderoso editor algébrico, tem sua própria seção “ALGEBRA (FORM)”, no Manual de Referência.

Desenvolva (“expand”) ambos os membros da equação.

EXPAN

```

3:
2:      '(C+B)/A-3'
1:      '2*A+2*X=3*B-3*X+C'
COLCT EXPAN SIZE FORM OESUB ENSUB

```

Para subtrair de ambos os membros da equação o termo em x do membro esquerdo da equação ($2x$), primeiro coloque os termos em x do membro esquerdo na pilha.

2 X

```

3:      '(C+B)/A-3'
2:      '2*A+2*X=3*B-3*X+C'
1:      '2*X'
COLCT EXPAN SIZE FORM OESUB ENSUB

```

Então subtraia $2x$ de ambos os membros.

-

```

2:      '(C+B)/A-3'
1:      '2*A+2*X-2*X=3*B-3*X'
      '+C-2*X'
COLCT EXPAN SIZE FORM OESUB ENSUB

```

Agrupe (“collect”) a equação.

COLCT

```

3:
2:      '(C+B)/A-3'
1:      '2*A=3*B+C-5*X'
COLCT EXPAN SIZE FORM OESUB ENSUB

```

Cada membro é agrupado independentemente e os termos em x são cancelados no membro esquerdo.

Agora você pode isolar x na equação. Especifique a variável que você deseja isolar.

X

```

2:      '(C+B)/A-3'
1:      '2*A=3*B+C-5*X'
      'X'
COLCT EXPAN SIZE FORM OESUB ENSUB

```

Isole x . O comando ISOL aparece na segunda linha do menu ALGEBRA, bem como do menu SOLVE.

NEXT ISOL

```
3:
2:      '(C+B)/A-3'
1:      '(3*B+C-2*A)/5'
TAYLR ISOL QUAD SHOW OBJE GET EXE GET
```

A expressão devolvida representa uma solução simbólica da equação para x , isto é, a equação

$$2(a + x) = 3(b - x) + c$$

é satisfeita quando $x = (3b + c - 2a)/5$.

Utilizando FORM

Se existem múltiplas ocorrências de x e se qualquer ocorrência tem um coeficiente simbólico, o comando COLCT não combinará os coeficientes. O exemplo, a seguir, isola x na equação

$$a(x + b) + 2x = c,$$

onde x ocorre mais do que uma vez e tem um coeficiente simbólico a . A estratégia é desenvolver a equação, utilizar FORM para agrupar os coeficientes de x e, então, isolar x .

Coloque a equação na pilha.

' A x (X + B) + 2 x X = C ENTER

```
3:      '(C+B)/A-3'
2:      '(3*B+C-2*A)/5'
1:      'A*(X+B)+2*X=C'
TAYLR ISOL QUAD SHOW OBJE GET EXE GET
```

Desenvolva a equação.

NEXT EXPAN

```
3:      '(C+B)/A-3'
2:      '(3*B+C-2*A)/5'
1:      'A*X+A*B+2*X=C'
COLCT EXPAN SIZE FORM OBJE SUB EXE SUB
```

Agora utilize FORM para agrupar os coeficientes de x .

FORM

```
((((X)+A*B))+2*X)=C)
COLCT EXPAN LEVEL EXE GET [+ ] [+ ]
```

A operação normal da calculadora é suspensa enquanto FORM está ativo. No modo FORM, o visor apresenta a equação com todas as subexpressões delimitadas por parênteses. Você utilizará FORM para manipular subexpressões dentro da equação.

O objetivo é combinar $(A * X)$ e $(2 * X)$ em um único termo $((A+2) * X)$. Existem três passos necessários, mostrados abaixo, como você poderia escrevê-los em papel. A forma corrente da equação é:

$$(ax + ab) + 2x = c$$

O primeiro passo é comutar ax e ab , resultando em:

$$(ab + ax) + 2x = c$$

O segundo passo é associar ax e $2x$, resultando em:

$$ab + (ax + 2x) = c$$

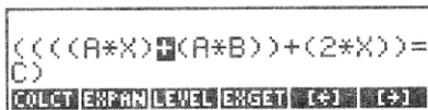
O terceiro passo é combinar ax e $2x$, resultando em :

$$ab + (a + 2) x = c$$

Passo 1: Comutar ax e ab .

Mova o cursor (o caractere ou caracteres em inverso) para +.

[→] [→] [→]



A posição do cursor determina sobre qual expressão você está atuando. Aqui você deseja atuar na subexpressão $((A * X) + (A * B))$ para comutar os argumentos de + (os argumentos da operação soma).

Apresente a primeira linha de manipulações para +.

NEXT

```
((((A*B)+(A*X)))+(2*X))=
C)
-0  ↔  ←M  M→  ←A  A→
```

As manipulações que aparecem quando você pressiona **NEXT** são específicas à função ou variável indicada pelo cursor; estas manipulações são específicas a +.

Comute os argumentos de +, utilizando **↔** (comutar).

↔

```
((((A*X)+(A*B)))+(2*X))=
C)
-0  ↔  ←M  M→  ←A  A→
```

Retorne ao menu FORM principal.

ENTER

```
((((A*B)+(A*X)))+(2*X))=
C)
COLT EXPAN LEVEL ERGET ←) →)
```

Passo 2: Associe ax e $2x$.

Mova o cursor para o segundo +.

[→] [→] [→] [→]

```
((((A*B)+(A*X)))+(2*X))=
C)
COLT EXPAN LEVEL ERGET ←) →)
```

Aqui você deseja agir na subexpressão

$((A*B) + (A*X)) + (2*X)$

para associar os termos $(A*X)$ e $(2*X)$ em uma única subexpressão.

Apresente a primeira linha de manipulações para +.

NEXT

```
((((A*B)+(A*X)))+(2*X))=
C)
-0  ↔  ←M  M→  ←A  A→
```

Aqui estão as mesmas manipulações porque o cursor novamente indicou uma subexpressão aditiva.

Associe os termos $(A * X)$ e $(2 * X)$ na subexpressão $((A * X) + (2 * X))$, utilizando \rightarrow (*associar à direita*).

\rightarrow

$$(((A * B) + ((A * X) + (2 * X))) = C)$$

Volte para o menu FORM principal.

ENTER

$$(((A * B) + ((A * X) + (2 * X))) = C)$$

Passo 3: Combine ax e $2x$.

Mova o cursor para o segundo $+$.

\rightarrow \rightarrow \rightarrow \rightarrow

$$(((A * B) + ((A * X) + (2 * X))) = C)$$

Aqui você deseja atuar na subexpressão $((A * X) + (2 * X))$ para combinar os coeficientes de X .

Apresente a primeira linha de manipulações para $+$.

NEXT

$$(((A * B) + ((A * X) + (2 * X))) = C)$$

Combine os coeficientes de X , utilizando \rightarrow (*merge right = combinar à direita*).

\rightarrow

$$(((A * B) + ((A + 2) * X)) = C)$$

Deste modo atingimos o objetivo de combinar-se $(A * X)$ e $(2 * X)$ em um único termo $((A + 2 * X))$.

Saia de FORM e devolva a equação modificada à pilha.

ON

$$\begin{array}{l} 3: \quad '(C+B)/A-3' \\ 2: \quad '(3*B+C-2*A)/5' \\ 1: \quad 'A*B+(A+2)*X=C' \end{array}$$

Agora que x ocorre somente uma vez na equação, você pode isolar x .

Especifique a variável que você deseja isolar.

X

```
2:      '(3*B+C-2*A)/5'  
1:      'A*B+(A+2)*X=C'  
'X'  
COLT  EXPAN  SIZE  FORM  ODSUB  ERGUS
```

Isole x .

NEXT ISOL

```
3:      '(C+B)/A-3'  
2:      '(3*B+C-2*A)/5'  
1:      '(C-A*B)/(A+2)'  
TAYLR  ISOL  QUAD  SHOW  ODSUB  ERGUS
```

A expressão devolvida pela calculadora representa uma solução simbólica da equação para x , isto é, a equação

$$a(x + b) + 2x = c$$

é satisfeita quando $x = (c - ab) / (a + 2)$.

Cálculo

Você pode diferenciar simbolicamente quase que qualquer expressão para a qual exista uma derivada. A integração é mais restrita: você pode computar uma *integral numérica definida* para qualquer expressão, mas uma *integral simbólica exata* somente para um polinômio.

Este capítulo contém exemplos simples de cálculo de derivadas, integrais indefinidas e integrais definidas para expressões. Para mais informações sobre cálculo diferencial e integral na HP-28S, ver “Cálculo” no Manual de Referência.

Diferenciando uma Expressão

Você pode diferenciar uma expressão passo-a-passo, observando como a calculadora aplica as regras de diferenciação, ou você pode diferenciar uma expressão de uma só vez. Os resultados finais são idênticos. Nesta seção, você irá diferenciar uma expressão duas vezes, primeiro passo-a-passo e, então, de uma só vez.

Diferenciação Passo-a-Passo

Para diferenciar passo-a-passo, digite a derivada como uma expressão. Por exemplo, calcule:

$$\frac{d}{dx} \tan(x^2 + 1)$$

Antes de iniciar o exemplo, apague a pilha, selecione modo angular radianos e selecione modo de apresentação de números STD.

CLEAR
MODE **RAD**
STD

```
3:
2:
1:
STD= FIX SCI ENG DEG RAD=
```

Elimine a variável X (se ela existir).

X **PURGE**

```
3:
2:
1:
STD= FIX SCI ENG DEG RAD=
```

Agora inicie a expressão para a derivada, começando com a variável de diferenciação.

d/dx **X** **(**

```
2:
1:
' dX(
STD= FIX SCI ENG DEG RAD=
```

A seguir, digite a expressão a ser diferenciada.

TRIG **TAN** **X** **^** **2** **+** **1** **ENTER**

```
3:
2:
1:
' dX(TAN(X^2+1))'
SIN ASIN COS ACOS TAN ATAN
```

Esta expressão representa a derivada, com respeito a x, de $\tan(x^2 + 1)$.

Processe a expressão uma vez.

EVAL

```
2:
1:
'(1+SQ(TAN(X^2+1))) *
dX(X^2+1)'
SIN ASIN COS ACOS TAN ATAN
```

O resultado reflete a regra da cadeia da diferenciação:

$$\frac{d}{dx} \tan(x^2 + 1) = \frac{d}{d(x^2 + 1)} \tan(x^2 + 1) \times \frac{d}{dx} (x^2 + 1)$$

A derivada da função tangente foi processada. A seguir você processará a derivada de $x^2 + 1$.

Processe a expressão uma segunda vez.

EVAL

2:
1: '(1+SQ(TAN(X^2+1))) *
∂X(X^2)'
SIN ASIN COS ACOS TAN ATAN

O resultado reflete a derivada de uma soma:

$$\frac{d}{dx} (x^2 + 1) = \frac{d}{dx} x^2 + \frac{d}{dx} 1$$

A derivada de 1 é 0, de tal forma que este termo desaparece. A seguir, você processará a derivada de x^2 .

Processe a expressão uma terceira vez.

EVAL

2:
1: '(1+SQ(TAN(X^2+1))) *
(2*X) * 2 * X^(2-1)'
SIN ASIN COS ACOS TAN ATAN

O resultado novamente reflete a regra da cadeia:

$$\frac{d}{dx} x^2 = \frac{d}{dx} (x^2) \times \frac{d}{dx} x$$

A derivada de x^2 foi processada. Finalmente processe a derivada do próprio x .

Processe a expressão uma quarta vez.

EVAL

2:
1: '(1+SQ(TAN(X^2+1))) *
(2*X)'
SIN ASIN COS ACOS TAN ATAN

Aqui a derivada está totalmente processada.

Diferenciação Completa

Para diferenciar uma expressão de uma só vez, execute a diferenciação como uma operação na pilha. Novamente, suponha que você deseje calcular:

$$\frac{d}{dx} \tan(x^2 + 1)$$

Coloque a expressão a ser diferenciada na pilha.

CLEAR X **TAN** X **^** 2 **+** 1 **ENTER**

```
3:
2:
1: 'TAN(X^2+1)'
SIN ASIN COS ACOS TAN ATAN
```

Especifique a variável de diferenciação.

X **ENTER**

```
3:
2: 'TAN(X^2+1)'
1: 'X'
SIN ASIN COS ACOS TAN ATAN
```

Diferencie a expressão.

d/dx

```
2:
1: '(1+SQ(TAN(X^2+1)))*(2*X)''
SIN ASIN COS ACOS TAN ATAN
```

A derivada totalmente processada é devolvida ao nível 1.

Integrando uma Expressão

A HP-28S calcula a integral indefinida de uma expressão por *integração simbólica*, a qual devolve uma expressão como resultado. Este método devolve o resultado exato somente para expressões polinomiais. (Para outras expressões, a HP-28S integra uma série de Taylor como uma aproximação. Ver "Cálculo", no Manual de Referência, para detalhes.) O primeiro exemplo abaixo demonstra uma integração simbólica.

Em contraste, integrais definidas são calculadas por *integração numérica*, a qual devolve resultados numéricos. Este método se aplica a qualquer expressão que é "bem comportada" no sentido matemático. O segundo exemplo abaixo demonstra integração numérica.

Integração Simbólica de Polinômio

Neste exemplo, você integrará simbolicamente o polinômio.

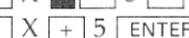
$$8x^3 + 9x^2 + 2x + 5.$$

Apague a pilha.



```
3:
2:
1:
SIN ASIN COS ACOS TAN ATAN
```

Coloque o polinômio na pilha.

  X  3  9  X  2
 2  X  5 

```
3:
2:
1: '8*X^3+9*X^2+2*X+5'
SIN ASIN COS ACOS TAN ATAN
```

Especifique a variável de integração.

 X 

```
3:
2: '8*X^3+9*X^2+2*X+5'
1: 'X'
SIN ASIN COS ACOS TAN ATAN
```

Especifique o grau do polinômio.

3 

```
3: '8*X^3+9*X^2+2*X+5'
2: 'X'
1: 3
SIN ASIN COS ACOS TAN ATAN
```

Integre o polinômio.



```
2:
1: '5*X+X^2+3*X^3+2*X^4'
SIN ASIN COS ACOS TAN ATAN
```

Aguarde o anúncio ((•)) desaparecer, indicando que a integração está concluída. A integral é devolvida ao nível 1.

Integração Numérica de Expressões

Neste exemplo, você encontrará o valor numérico para a integral.

$$\exp(x^3 + 2x^2 - x + 4) dx$$

Apague a pilha.

CLEAR

```
3:
2:
1:
[SIN] [ASIN] [COS] [ACOS] [TAN] [ATAN]
```

Coloque a expressão na pilha.

' **LOGS** **EXP** **X** **^** **3** **+** **2**
x **X** **^** **2** **-** **X** **+** **4** **ENTER**

```
3:
2:
1: 'EXP(X^3+2*X^2-X+4)'  
[LOG] [ALOG] [LN] [EXP] [LNPI] [EXPPI]
```

Digite a variável e os limites de integração. Você os entrará como objetos, dentro de um objeto lista. (Este é um típico uso de uma lista — combinar diversos objetos de forma que você possa manipulá-los como um único objeto.)

[**X** **SPACE** **0** **SPACE** **1** **ENTER**

```
3:
2: 'EXP(X^3+2*X^2-X+4)'  
1: { X 0 1 }  
[LOG] [ALOG] [LN] [EXP] [LNPI] [EXPPI]
```

X é a variável de integração, 0 e 1 são os limites de integração.

A seguir, digite a precisão que você requer.

Se a expressão incluiu constantes oriundas de dados empíricos, especifique a precisão das constantes. Por exemplo, se as constantes são precisas até três casas decimais, especifique uma precisão de .001.

Neste exemplo, você está integrando uma expressão sem constantes empíricas, assim você poderia especificar precisão de 12 dígitos. Entretanto, quanto maior for a precisão, mais demorado será o processo iterativo de integração numérica. Assim, neste exemplo, você especificará uma precisão de .00001.

1E - 5

```
3: 'EXP(X^3+2*X^2-X+4)'  
2: ( X 0 1 )  
1: .00001  
LOG  ALOG  LN  EXP  LNPI  EXPM
```

Encontre a integral.

```
3:  
2: 103.117678153  
1: 1.03086911923E-3  
LOG  ALOG  LN  EXP  LNPI  EXPM
```

A integral estimada é devolvida ao nível 2 e o limite de erro é devolvido ao nível 1.

O valor da integral é $103.118 - .001$. Note que o limite de erro devolvido é, aproximadamente, o produto da integral estimada e a precisão que você especificou.

Vetores e Matrizes

A HP-28S opera com dois tipos de arranjos (arrays): *vetores*, os quais são arranjos uni-dimensionais e *matrizes*, os quais são arranjos bi-dimensionais. Você pode entrar vetores e matrizes como objetos individuais, denominados *objetos arranjo*, e calcular com os mesmos tão facilmente quanto com números.

Este capítulo apresenta cálculos básicos com arranjos, utilizando arranjos reais — vetores e matrizes cujos elementos são números reais. Você também pode efetuar cálculos com arranjos cujos elementos são números complexos.

Todos os comandos no menu ARRAY são descritos brevemente no apêndice D, “Diagramas dos Menus”. Para descrições completas, ver “ARRAY” no Manual de Referência.

Vetores

Esta seção demonstra aritmética vetorial, o produto vetorial e o produto escalar.

Entrando um Vetor

Antes de iniciar estes exemplos, apague a pilha e selecione o modo de apresentação de números STD.

CLEAR
 MODE STD

| |
|---|
| 3: |
| 2: |
| 1: |
| STD <input type="checkbox"/> FIX <input type="checkbox"/> SCI <input type="checkbox"/> ENG <input type="checkbox"/> DEG <input type="checkbox"/> RAD <input type="checkbox"/> |

Digite o vetor [2 3 4]. Você pode utilizar vírgulas ou espaços para separar 2 do 3 e 3 do 4.

2,3,4

```
3:
2:
1: [ 2 3 4 ]
STO= FIX SCI ENG DEG RAD=
```

Multiplicando e Dividindo um Vetor por um Número

Multiplique o vetor por 15.

15

```
3:
2:
1: [ 30 45 60 ]
STO= FIX SCI ENG DEG RAD=
```

A ordem dos argumentos não faz diferença à multiplicação, da mesma forma que não faz diferença quando você multiplica dois números. Entretanto, para a divisão, o vetor precisa estar no nível 2 e o número no nível 1.

Divida o vetor por 5.

5

```
3:
2:
1: [ 6 9 12 ]
STO= FIX SCI ENG DEG RAD=
```

Adicionando e Subtraindo Vetores

Você pode adicionar e subtrair vetores da mesma forma que você adiciona e subtrai números, desde que os vetores tenham o mesmo número de elementos. A ordem dos argumentos é importante à subtração, da mesma forma que é importante quando você subtrai um número de outro.

Para este exemplo, subtraia o vetor [-10 20 30].

-10,20,30

```
3:
2:
1: [ 16 -11 -18 ]
STO= FIX SCI ENG DEG RAD=
```

Calculando o Produto Vetorial

Calcule o produto vetorial do vetor no nível 1 com o vetor $[2 \ -2 \ 1]$. (O produto vetorial é definido somente para vetores com dois e três elementos.)

Digite o vetor.

2, -2, 1

```
3:
1: [ 16 -11 -18 ]
[2, -2, 1]
STO= FIX SCI ENG DEG RAD=
```

Calcule o produto vetorial, utilizando CROSS na terceira linha do menu ARRAY.

ARRAY

```
3:
2:
1: [ -47 -52 -10 ]
CROSS DOT DET ABS RNRM CNRM
```

Calculando o Produto Escalar

Encontre o produto escalar do vetor no nível 1 com o vetor $[5 \ 7 \ 2]$. (Os dois vetores precisam ter o mesmo número de elementos.)

Digite o vetor.

5,7,2

```
3:
1: [ -47 -52 -10 ]
[5,7,2]
CROSS DOT DET ABS RNRM CNRM
```

Calcule o produto escalar.

```
3:
2:
1: -619
CROSS DOT DET ABS RNRM CNRM
```

Matrizes

Esta seção descreve como inverter uma matriz e como encontrar o determinante de uma matriz. Ambos os cálculos são restritos a matrizes *quadradas* — aquelas com o mesmo número de linhas e colunas.

Os cálculos que você executou com vetores também se aplicam a matrizes (com exceção dos produtos escalar e vetorial). Você pode multiplicar ou dividir uma matriz por um número e pode adicionar ou subtrair duas matrizes (desde que as matrizes tenham as mesmas dimensões).

Entrando uma Matriz

Digite a seguinte matriz:

$$\begin{bmatrix} 1 & 2 & 3 \\ 1 & 3 & 3 \\ 1 & 2 & 4 \end{bmatrix}$$

Inicie a matriz.

[]

```
2:
1:
[ ] -619
[CROSS] [DOT] [DET] [ABS] [RNRM] [CNRM]
```

Entre cada linha da matriz como um vetor separado.

[] 1,2,3

[] 1,3,3

[] 1,2,4 [ENTER]

```
1: [[ 1 2 3 ]
    [ 1 3 3 ]
    [ 1 2 4 ]]
[CROSS] [DOT] [DET] [ABS] [RNRM] [CNRM]
```

Vendo uma Matriz Grande

Quando uma matriz tem muitos elementos ou elementos não inteiros, pode ocorrer que você não veja a matriz inteira de uma só vez. Para ver uma matriz grande, utilize **[EDIT]** (se a matriz está no nível 1) ou **n [VISIT]** (se a matriz está no nível n) para colocar a matriz na linha de comando. Você pode então utilizar as teclas do menu do cursor para apresentar no visor qualquer parte da matriz. Para detalhes, ver "Editando Objetos Existentes", no capítulo 18.

Invertendo uma Matriz

Como a matriz no nível 1 é quadrada, você pode encontrar sua inversa.

 M^{-1}

```
1: [[ 6 -2 -3 ]
    [ -1 1 0 ]
    [ -1 0 1 ]]
CROSS DOT DET ABS RNAME CNAME
```

Calculando o Determinante

Como a matriz no nível 1 é quadrada, você pode encontrar seu determinante.

DET

```
3:
2: -619
1: 1
CROSS DOT DET ABS RNAME CNAME
```

Multiplicando Dois Arranjos

Você pode utilizar a função \times para multiplicar duas matrizes ou uma matriz e um vetor. (Utilize **CROSS** ou **DOT** para multiplicar dois vetores, como descrito acima.)

Multiplicando Duas Matrizes

A ordem dos argumentos é importante quando se está multiplicando duas matrizes. O número de *colunas* na matriz no nível 2 precisa ser igual ao número de *linhas* na matriz no nível 1. Por exemplo, você pode calcular o seguinte produto matricial.

$$\begin{bmatrix} 2 & 2 \\ 4 & 1 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} 2 & 2 & 1 & 4 \\ 3 & 4 & 2 & 1 \end{bmatrix}$$

Para calcular este produto matricial:

Entre a primeira matriz.

$\boxed{1}$ $\boxed{[$ $\boxed{2,2}$
 $\boxed{1}$ $\boxed{4,1}$
 $\boxed{1}$ $\boxed{2,3}$ $\boxed{\text{ENTER}}$

```
1: [[ 2 2 ]  
    [ 4 1 ]  
    [ 2 3 ]]  
CROSS DOT DET ABS RNRM CNRM
```

Digite a segunda matriz.

$\boxed{1}$ $\boxed{[$ $\boxed{2,2,1,4}$
 $\boxed{1}$ $\boxed{3,4,2,1}$

```
1: [[ 2 2 ]  
    [ 4 1 ]  
[[2,2,1,4][3,4,2,1]  
CROSS DOT DET ABS RNRM CNRM
```

Multiplique as matrizes.

$\boxed{\times}$

```
1: [[ 10 12 6 10 ]  
    [ 11 12 6 17 ]  
    [ 13 16 8 11 ]]  
CROSS DOT DET ABS RNRM CNRM
```

Multiplicando uma Matriz e um Vetor

A ordem dos argumentos é importante quando se está multiplicando uma matriz e um vetor. A matriz precisa estar no nível 2 e o vetor no nível 1. O número de *colunas* na matriz precisa ser igual ao número de *elementos* no vetor.

Para o exemplo a seguir, multiplique a matriz correntemente no nível 1 pelo vetor $[3 \ 1 \ 1 \ 2]$.

Digite o vetor.

$\boxed{1}$ $\boxed{[$ $\boxed{3,1,1,2}$

```
1: [[ 10 12 6 10 ]  
    [ 11 12 6 17 ]  
[3,1,1,2]  
CROSS DOT DET ABS RNRM CNRM
```

Multiplique a matriz e o vetor.

$\boxed{\times}$

```
3: 7619  
1: 1  
1: [ 68 85 85 ]  
CROSS DOT DET ABS RNRM CNRM
```

Resolvendo um Sistema de Equações Lineares

Para resolver um sistema de n equações lineares com n variáveis, utilize um vetor *constante* com n elementos, uma matriz de *coeficientes* $n \times n$ e a divisão (\div). O vetor constante contém os valores constantes das equações. O coeficiente de matrizes contém os coeficientes das variáveis.

O exemplo, a seguir, mostra como resolver o sistema de três equações, linearmente independentes, em três variáveis. Suponha que as equações sejam:

$$\begin{aligned}3x + y + 2z &= 13 \\x + y - 8z &= -1 \\-x + 2y + 5z &= 13\end{aligned}$$

Entre o vetor constante.

$\boxed{13, -1, 13}$ $\boxed{\text{ENTER}}$

```
3:
2:      [ 68 85 85 ]
1:      [ 13 -1 13 ]
CROSS DOT DET ABS RNRM CNRM
```

Digite a matriz coeficiente.

$\boxed{3, 1, 2}$
 $\boxed{1, 1, -8}$
 $\boxed{-1, 2, 5}$

```
2:      [ 68 85 85 ]
1:      [ 13 -1 13 ]
[[3, 1, 2][1, 1, -8][-1, 2, 5]
CROSS DOT DET ABS RNRM CNRM
```

Resolva o sistema de equações.

$\boxed{\div}$

```
3:
2:      [ 68 85 85 ]
1:      [ 2 5 1 ]
CROSS DOT DET ABS RNRM CNRM
```

Os valores no vetor solução são os valores das variáveis que satisfazem as equações:

$$x = 2, y = 5, z = 1$$

Para resolver sistemas de equações sub-determinados, sobre-determinados, ou quase singulares, ver "ARRAY" no Manual de Referência.

Estatística

Este capítulo descreve como entrar dados estatísticos e como efetuar cálculos estatísticos com amostras simples e amostras de pares de dados, utilizando comandos no menu STAT. Todos os comandos no menu STAT são descritos brevemente no apêndice D, "Diagramas dos Menus". Para descrições completas, ver "STAT" no Manual de Referência.

A tabela, a seguir, relaciona a variação dos preços ao consumidor (CPI), a variação do índice de preços dos produtores (PPI) e a taxa de desemprego (UR), todas em porcentagem, para os Estados Unidos, para um período de cinco anos. Entre estes dados e efetue cálculos estatísticos a partir deles.

Dados para Exemplo Estatístico

| Ano | CPI | PPI | UR |
|------|------|------|-----|
| 1975 | 9.1 | 9.2 | 8.5 |
| 1976 | 5.8 | 4.6 | 7.7 |
| 1977 | 6.5 | 6.1 | 7.0 |
| 1978 | 7.6 | 7.8 | 6.0 |
| 1979 | 11.5 | 19.3 | 5.8 |

Entrando Dados

Dados estatísticos são armazenados em uma matriz estatística denominada Σ DAT — uma matriz ordinária com um nome especial. Cada linha da matriz contém um ponto dos dados, os quais, neste exemplo, constituem os valores de CPI, PPI e UR para um ano.

Antes de você iniciar, apague a pilha e selecione o modo de apresentação de números FIX 2.

CLEAR
MODE 2 FIX

```
3:  
2:  
1:  
STD FIX SCI ENG DEG RAD
```

Apague quaisquer dados estatísticos anteriores, utilizando **CLΣ** (*clear statistics* = *apague estatísticas*) no menu **STAT**. (Qualquer Σ DAT existente é eliminada.)

STAT CLΣ

```
3:  
2:  
1:  
Σ+ Σ- NE CLΣ STOΣ RCLΣ
```

Digite os dados para 1975.

1 9.1,9.2,8.5

```
2:  
1:  
[9.1,9.2,8.5]  
Σ+ Σ- NE CLΣ STOΣ RCLΣ
```

Armazene estes dados em Σ DAT.

Σ+

```
3:  
2:  
1:  
Σ+ Σ- NE CLΣ STOΣ RCLΣ
```

Uma nova matriz denominada Σ DAT é automaticamente criada. O ponto de dados correspondentes a 1975 é a primeira linha de Σ DAT.

Entre os dados para 1976.

1 5.8,4.6,7.7 **Σ+**

```
3:  
2:  
1:  
Σ+ Σ- NE CLΣ STOΣ RCLΣ
```

Os dados correspondentes a 1976 são adicionados a Σ DAT, formando a segunda linha da matriz estatística.

Entre os dados para 1977.

$\Sigma+$

| | |
|-----------|------------------------------------|
| 3: | |
| 2: | |
| 1: | |
| $\Sigma+$ | $\Sigma-$ NE CLR STO Σ RCLZ |

Os dados para 1977 são adicionados a Σ DAT, formando a terceira linha da matriz estatística.

Editando Dados

Se você cometer um erro enquanto digita os dados, e se você perceber seu erro antes de pressionar $\Sigma+$, você pode simplesmente editar a linha de comando. Suponha que você desconfie que cometeu um erro, entrando dados para 1976. Você pode devolver dados à pilha, editar aqueles que contêm erros e devolver os dados para Σ DAT.

Remova o ponto de dados referentes a 1977 (a última linha em Σ DAT) e coloque-o na pilha.

$\Sigma-$

| | |
|-----------|------------------------------------|
| 3: | |
| 2: | |
| 1: | [6.50 6.10 7.00] |
| $\Sigma+$ | $\Sigma-$ NE CLR STO Σ RCLZ |

Remova o ponto de dados referentes a 1976 (a última linha em Σ DAT) e coloque-o na pilha.

$\Sigma-$

| | |
|-----------|------------------------------------|
| 3: | |
| 2: | [6.50 6.10 7.00] |
| 1: | [5.80 4.60 7.70] |
| $\Sigma+$ | $\Sigma-$ NE CLR STO Σ RCLZ |

Se você descobrir que cometeu um erro com este ponto de dados, pressione para devolvê-lo à linha de comando, edite-o e pressione para colocá-lo corrigido de volta à pilha. (Ver "Editando Objetos Existentes", no capítulo 18.)

Devolva o ponto de dados corrigido, referentes a 1976 para Σ DAT.

$\Sigma+$

| | |
|-----------|------------------------------------|
| 3: | |
| 2: | |
| 1: | [6.50 6.10 7.00] |
| $\Sigma+$ | $\Sigma-$ NE CLR STO Σ RCLZ |

Devolva o ponto de dados referentes a 1977 para Σ DAT.

$\Sigma+$

```
0:
1:
2:
3:
4:
5:
6:
7:
8:
9:
 $\Sigma+$   $\Sigma-$   $N\Sigma$   $CL\Sigma$   $STO\Sigma$   $RCL\Sigma$ 
```

Agora entre o restante dos dados (para 1978 e 1979) e verifique se você entrou todos os cinco pontos de dados.

7.6,7.8,6 $\Sigma+$

11.5,19.3,5.8 $\Sigma+$

$N\Sigma$

```
0:
1:
2:
3:
4:
5:
6:
7:
8:
9:
5.00
 $\Sigma+$   $\Sigma-$   $N\Sigma$   $CL\Sigma$   $STO\Sigma$   $RCL\Sigma$ 
```

Estatística com uma Única Amostra

Nesta seção, você calculará a média, desvio padrão e a variância de CPI, PPI e UR. Os dados para CPI estão contidos na primeira coluna de Σ DAT, os dados para PPI na segunda coluna, e os dados para UR na terceira coluna.

Apresente a segunda linha do menu STAT.

NEXT

```
0:
1:
2:
3:
4:
5:
6:
7:
8:
9:
5.00
TOT MEAN SDEV VAR MAXE MINX
```

Aqui estão os comandos para média, desvio padrão e variância.

Calculando a Média

Calcule a média.

MEAN

```
0:
1:
2:
3:
4:
5:
6:
7:
8:
9:
5.00
[ 8.10 9.40 7.00 ]
TOT MEAN SDEV VAR MAXE MINX
```

A média do CPI é 8.1, do PPI é 9.4 e do UR é 7.

Calculando o Desvio Padrão

Calcule o desvio padrão.

SDEV

| | | | | | |
|-----|------|------|------|------|-----|
| 3: | | | 5.00 | | |
| 2: | [| 8.10 | 9.40 | 7.00 |] |
| 1: | [| 2.27 | 5.80 | 1.14 |] |
| TOT | MEAN | SDEV | VAR | MAX | MIN |

O desvio padrão da amostra é 2.27 para CPI, 5.8 para PPI e 1.14 para UR.

Calculando a Variância

Calcule a variância.

VAR

| | | | | | |
|-----|------|------|-------|------|-----|
| 3: | [| 8.10 | 9.40 | 7.00 |] |
| 2: | [| 2.27 | 5.80 | 1.14 |] |
| 1: | [| 5.17 | 33.64 | 1.30 |] |
| TOT | MEAN | SDEV | VAR | MAX | MIN |

A variância da amostra é 5.17 para CPI, 33.64 para PPI e 1.3 para UR.

Estatísticas com Dados Emparelhados

Nesta seção, você calculará a correlação e covariância de CPI e PPI, utilizando então um modelo de regressão linear, para projetar valores de PPI a partir de valores de CPI.

Apresente a terceira linha do menu STAT.

NEXT

| | | | | | |
|-----|------|------|-------|------|---|
| 3: | [| 8.10 | 9.40 | 7.00 |] |
| 2: | [| 2.27 | 5.80 | 1.14 |] |
| 1: | [| 5.17 | 33.64 | 1.30 |] |
| COL | CORR | COV | LR | PRED | |

Aqui estão os comandos para correlação, covariância, regressão linear e valor projetado.

Especificando um Par de Colunas

Antes de executar estatística com dados emparelhados, especifique quais colunas da matriz estatística Σ DAT contém os dados dependentes e independentes. Neste exemplo, você quer CPI (na coluna 1) como os dados independentes e PPI (na coluna 2) como os dados dependentes.

Especifique colunas 1 e 2 como dados independentes e dependentes.

1,2 COLΣ

| | | | | | |
|----|------|------|-------|------|-------|
| 3: | [| 8.10 | 9.40 | 7.00 |] |
| 2: | [| 2.27 | 5.80 | 1.14 |] |
| 1: | [| 5.17 | 33.64 | 1.30 |] |
| | COLΣ | CORR | COV | LR | PREDV |

Os números 1 e 2 são armazenados em uma lista denominada Σ PAR, que é uma lista comum com um nome especial. Os comandos que executam estatísticas com dados emparelhados buscam informação em Σ PAR.

Se você não especificar as colunas contendo os dados independentes e dependentes, a calculadora utilizará as colunas 1 e 2. Neste exemplo, você não necessita especificar as colunas, mas lembre-se de executar **COLΣ**, se seus dados independentes e dependentes não estão contidos nas colunas 1 e 2.

Calculando a Correlação

Calcule a correlação.

CORR

| | | | | | |
|----|------|------|-------|------|-------|
| 3: | [| 2.27 | 5.80 | 1.14 |] |
| 2: | [| 5.17 | 33.64 | 1.30 |] |
| 1: | | | | 0.96 | |
| | COLΣ | CORR | COV | LR | PREDV |

A correlação entre CPI e PPI é 0.96.

Calculando a Covariância

Calcule a covariância da amostra.

COV

| | | | | | |
|----|------|------|-------|-------|-------|
| 3: | [| 5.17 | 33.64 | 1.30 |] |
| 2: | | | | 0.96 | |
| 1: | | | | 12.65 | |
| | COLΣ | CORR | COV | LR | PREDV |

A covariância da amostra entre a CPI e PPI é 12.65.

Calculando a Regressão Linear

Calcule a reta que melhor ajuste os dados de CPI e PPI.

LR

| | | | | |
|------|--------|-----|----|-------|
| 3: | 12.65 | | | |
| 2: | -10.43 | | | |
| 1: | 2.45 | | | |
| COLZ | CORR | COV | LR | PREDV |

A reta tem uma inclinação (coeficiente angular) de 2.45 (coeficiente de CPI) e intercepta o eixo PPI (coeficiente linear) em -10.43. Esses valores são também armazenados na lista Σ PAR.

Efetuando Projeções

Suponha que você deseje calcular as projeções para PPI quando CPI tem valores de 6 e 7. O valor projetado pode ser calculado dos coeficientes linear e angular armazenados em Σ PAR.

Calcule o valor projetado para PPI quando CPI tem valor 6.

6 PREDV

| | | | | |
|------|--------|-----|----|-------|
| 3: | -10.43 | | | |
| 2: | 2.45 | | | |
| 1: | 4.26 | | | |
| COLZ | CORR | COV | LR | PREDV |

O valor projetado é 4.26.

Calcule o valor projetado para PPI quando CPI tem valor 7.

7 PREDV

| | | | | |
|------|------|-----|----|-------|
| 3: | 2.45 | | | |
| 2: | 4.26 | | | |
| 1: | 6.71 | | | |
| COLZ | CORR | COV | LR | PREDV |

O valor projetado é 6.71

Aritmética Binária

Este capítulo descreve como efetuar operações aritméticas com inteiros binários. Cada inteiro binário contém de 1 a 64 bits e representa um número binário sem sinal. Para facilitar a entrada de números binários e leitura dos resultados, você pode escolher uma base decimal, hexadecimal, octal ou binária. Entretanto, esta escolha não afeta a representação interna de inteiros binários e os comandos atuam em inteiros binários bit-a-bit.

Todos os comandos no menu BINARY são descritos no apêndice D, "Diagramas dos Menus". Para descrições completas, ver "BINARY" no Manual de Referência.

Selecionando o Tamanho de Palavra

O tamanho da palavra corrente afeta o comprimento dos inteiros binários desenvolvidos por comandos e a apresentação de inteiros binários na pilha. O tamanho de palavra pode ter um intervalo de 1 a 64 bits, com um tamanho de palavra padrão de 64 bits. Suponha que você deseje um tamanho de palavra de 16.

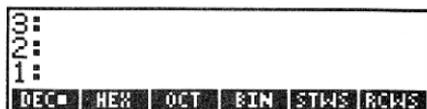
Antes de você iniciar o exemplo, apague a pilha e apresente o menu BINARY.

CLEAR BINARY

| |
|---|
| 3: |
| 2: |
| 1: |
| DEC <input checked="" type="checkbox"/> HEX <input type="checkbox"/> OCT <input type="checkbox"/> BIN <input type="checkbox"/> STWS <input type="checkbox"/> RCWS |

Especifique um tamanho de palavra de 16 bits.

16 STWS



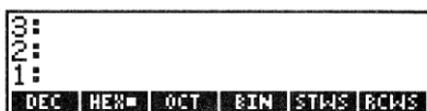
Agora se você digitar um inteiro binário maior do que 16 bits, somente os 16 bits menos significativos serão apresentados.

Selecionando a Base

A base corrente afeta como os inteiros binários são apresentados na pilha. As escolhas para a base são decimal, hexadecimal, octal e binária, sendo a seleção padrão base decimal.

Suponha que você deseje base hexadecimal.

HEX

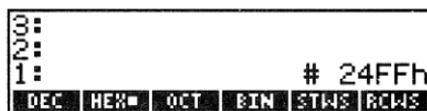


O rótulo para **HEX** agora inclui um pequeno quadrado, indicando que a base corrente é HEX.

Entrando Inteiros Binários

Entre o endereço $24FF_{16}$.

24FF ENTER



O “h” minúsculo é um *marcador de base*, indicando que a base corrente é HEX. Quando você entra um número, não necessita digitar o marcador de base, a menos que o número não esteja na base corrente.

Verifique como este inteiro binário é representado em outras bases. Você não necessita mudar o inteiro binário, somente o modo corrente.

Mude para base DEC.

DEC

```
3:
2:
1: # 9471d
DEC= HEX= OCT= BIN STWS RCWS
```

Mude para base OCT.

OCT

```
3:
2:
1: # 22377o
DEC= HEX= OCT= BIN STWS RCWS
```

Mude para base BIN.

BIN

```
3:
2:
1: # 100100111111111b
DEC= HEX= OCT= BIN= STWS RCWS
```

Volte à base HEX.

HEX

```
3:
2:
1: # 24FFh
DEC= HEX= OCT= BIN STWS RCWS
```

Calculando com Inteiros Binários

Calcule o endereço $1FO_{16}$ menos do que o endereço dado.

1FO -

```
3:
2:
1: # 230Fh
DEC= HEX= OCT= BIN STWS RCWS
```

A diferença é devolvida ao nível 1 da mesma forma que para outros números.

Você pode combinar inteiros binários e números reais em seus cálculos. Um real inteiro normal (entrado sem o delimitador #) é interpretado na base 10, independentemente da base binária inteira corrente.

Por exemplo, calcule o endereço 27_{10} menos do que o endereço dado.

27 -

```
3:
2:
1: # 22F7h
DEC= HEX= OCT= BIN STWS RCWS
```

A diferença, expressa como um inteiro binário, é devolvida ao nível 1.

Conversão de Unidades

Este capítulo contém exemplos de conversão de unidades — converter o valor numérico de uma medida física de um sistema de unidades para outro. Para informações detalhadas, ver “UNITS” no Manual de Referência.

O Catálogo UNITS

O catálogo UNITS relaciona alfabeticamente todas as unidades pré-programadas na HP-28S. Você o utilizará para verificar a grafia correta e a definição de unidades.

Primeiro apague a pilha e selecione modo de apresentação de números STD.

CLEAR **MODE** **STD**

```
3:
2:
1:
STD  FIX  SCI  ENG  DEG  RAD
```

Inicie o catálogo UNITS.

UNITS

```
3  312
100
m^2
NEXT  PREV  FETCH  QUIT
```

A primeira unidade é “are”, abreviado “a”. Esta é uma unidade de área equivalente a 100 metros quadrados.

Tente percorrer o catálogo para frente e para trás, mantendo pressionadas as teclas do menu **NEXT** e **PREV** (não as teclas permanentes no teclado).

Você pode deslocar-se para a primeira unidade que começa com uma letra específica, pressionando a tecla correspondente àquela letra.

S

```
s second
1
s
NEXT PREV          FETCH QUIT
```

O item referente a "second" (segundo) mostra que a abreviação correta é "s" e o valor é 1 segundo. "Second" é definida em termos dela mesma porque é uma unidade fundamental.

Assegure-se de que você utiliza as abreviações exatamente como elas aparecem no catálogo UNITS. Por exemplo, a HP-28S reconhece o "s" minúsculo como segundos, mas não o "S" maiúsculo.

A seguir, verifique o item referente a "day"(dia).

D

```
d day
86400
s
NEXT PREV          FETCH QUIT
```

Este item mostra que a abreviação correta é "d" e o valor é 86,400 segundos.

A seguir, procure a unidade "foot" (pé).

F

```
F farad
1
A^2*s^4/kg*m^2
NEXT PREV          FETCH QUIT
```

O catálogo mostra o item referente a "farad". Siga adiante sete itens.

```
NEXT NEXT NEXT NEXT
NEXT NEXT NEXT
```

```
ft int'l foot
.3048
m
NEXT PREV          FETCH QUIT
```

O catálogo mostra o item referente a "international foot" (pé internacional). Existem duas versões de "foot" no catálogo; a próxima unidade é "survey foot" (utilizado em agrimensura).

Você pode copiar a abreviação "pé internacional" para a linha de comando.

FETCH

```
2:
1:
ft
STO= FIX SCI ENG DEG RAD=
```

O visor normal volta a aparecer e a linha de comando mostra a abreviação da unidade.

Os exemplos, neste capítulo, mostram-lhe como digitar unidades diretamente, mas você pode utilizar **UNITS** e **FETCH**, se você preferir.

Apague a linha de comando.

ON

```
3:
2:
1:
STD= FIX SCI ENG DEG RAD=
```

Convertendo Unidades

Primeiro converta 15 °C para graus Fahrenheit.

Coloque o valor numérico na pilha.

15 **ENTER**

```
3:
2:
1: 15
STD= FIX SCI ENG DEG RAD=
```

Entre a abreviação de unidade para “graus Celsius”.

° C **ENTER**

```
3:
2:
1: 15 °C
STD= FIX SCI ENG DEG RAD=
```

A abreviação de unidade é convertida para um nome.

Entre a abreviação de unidade para “graus Fahrenheit”.

° F **ENTER**

```
3:
2:
1: 15 °C °F
STD= FIX SCI ENG DEG RAD=
```

A abreviação da unidade é convertida para um nome.

Converta o valor numérico da unidade antiga para a nova unidade.

CONVERT

```
3:
2:
1: 15 °F
STD= FIX SCI ENG DEG RAD=
```

O resultado mostra que 15 °C corresponde a 59 °F.

Para o exemplo a seguir, converta 40 polegadas em milímetros. Desta vez, você fará  **CONVERT** executar ENTER automaticamente por você.

Apague a pilha e entre o valor numérico.

 **CLEAR**

40 **ENTER**

```
3:
2:
1: 40
STD= FIX SCI ENG DEG RAD=
```

Entre a unidade para “polegadas” (inches).

LC in **ENTER**

```
3:
2:
1: 40
   'in'
STD= FIX SCI ENG DEG RAD=
```

Digite a unidade para “milímetros” e converta as unidades.

Você não encontrará “milímetro” no catálogo UNITS. Esta é considerada uma unidade *prefixada* — a unidade “m” (para metro) prefixada por “m” (mili ou um milésimo). Similarmente, “km” é uma unidade prefixada significando quilômetros e “ms” é uma unidade prefixada significando milissegundos. Uma lista completa de prefixos aparece em “UNITS”, no Manual de Referência.

LC mm  **CONVERT**

```
3:
2:
1: 1016
   'mm'
STD= FIX SCI ENG DEG RAD=
```

O resultado mostra que 40 polegadas correspondem a 1016 milímetros.

Convertendo Cadeias de Unidades

Cadeias (strings) são objetos que contêm caracteres. Você pode utilizar *cadeias de unidades*, para definir unidades mais complicadas do que aquelas utilizadas até agora.

Uma cadeia de unidades pode representar uma unidade elevada a uma potência, tal como “ft^2”, ou o produto de unidades, tal como “ft*lb”, ou qualquer combinação de potências e produtos de unidades.

Uma cadeia de unidades também pode representar um quociente de unidades, tal como "m/s". Entretanto, o símbolo / não pode aparecer mais do que uma vez. Assegure-se de agrupar todas as unidades diretas antes do símbolo / e todas as unidades inversas após o símbolo /. Por exemplo, "pés por segundo por segundo" é representado por "ft/s ^ 2".

Para o próximo exemplo, converta 1 milha por hora em pés por segundo.

Apague a pilha e entre o valor numérico.

■ CLEAR
1 ENTER

```
3:
2:
1: 1
STD FIX SCI ENG DEG RAD
```

Entre a unidade para "milhas por hora".

LC mph ENTER

```
3:
2:
1: 'mph' 1
STD FIX SCI ENG DEG RAD
```

Digite a unidade para "pés por segundos".

Não existe unidade pré-programada para "pés por segundo", de forma que você terá que utilizar uma cadeia de unidades.

■ " LC ft + s

```
2:
1: 'mph' 1
"ft / s"
STD FIX SCI ENG DEG RAD
```

O modo de entrada alfanumérico estava ativado (como indicado pela forma do cursor) quando você iniciou a digitação da cadeia. Em modo alfanumérico, todos os comandos são escritos na linha de comando, de modo que você terá que pressionar ENTER para completar a cadeia.

ENTER

```
3:
2:
1: "ft / s" 'mph' 1
STD FIX SCI ENG DEG RAD
```

Converta o valor numérico de unidades antigas para novas unidades.

■ CONVERT

```
3:
2: 1.466666666667
1: "ft / s"
STD FIX SCI ENG DEG RAD
```

O resultado mostra que uma milha por hora corresponde a 1.466666666667 pés por segundo.

A seguir, converta 10 pés cúbicos em galões.

Apague a pilha e entre o valor numérico.

■ CLEAR
10 ENTER

```
0:  
2:  
1: 10  
STO= FIX SCI ENG DEG RAD=
```

Entre a cadeia de unidades para “pés cúbicos”.

■ " LC ft **■** ^ 3 ENTER

```
0:  
2:  
1: "ft ^ 3" 10  
STO= FIX SCI ENG DEG RAD=
```

Digite a unidade para “US gallon” (galões americanos) e efetue a conversão.

LC gal **■** CONVERT

```
0:  
2: 74.8051948052  
1: 'gal'  
STO= FIX SCI ENG DEG RAD=
```

O resultado mostra que 10 pés cúbicos correspondem a 74.8051948052 galões.

Verificando se as Unidades estão Corretas

Utilizar unidades incorretas pode levar a resultados numéricos inesperados ou a uma mensagem de erro `INCONSISTENT UNITS`. A solução, em quaisquer dos casos, é verificar o catálogo `UNITS` ou a seção “UNITS” do Manual de Referência.

Resultados numéricos inesperados podem ocorrer se você utilizar uma unidade com as dimensões corretas, mas um valor numérico incorreto. Por exemplo, se você converter um acre em “ft ^ 2”, o resultado é maior do que 43,560. Isto ocorre porque existem duas unidades “foot”, “ft” (pé internacional) e “ftUS” (“survey foot” — utilizado em agrimensura). Convertendo um acre em “ftUS ^ 2” resulta exatamente em 43,560.

Um erro `INCONSISTENT UNITS` ocorre se você utilizar uma unidade com dimensões incorretas. Por exemplo, isto ocorre se você utilizar “lb” (libra) como uma unidade de força. A unidade de força correta é “lbf” (libra-força).

Funções do Usuário para Conversão de Unidades

Se você efetua conversões de unidades específicas com frequência, você pode escrever funções do usuário para estas conversões. Nesta seção, você escreverá funções do usuário $O \rightarrow G$ e $G \rightarrow O$ que convertem entre onças e gramas; uma vez que elas são funções do usuário, você pode utilizá-las em sintaxe RPN ou algébrica.

Lembre-se de que funções do usuário precisam preencher dois requisitos:

- indicar seus argumentos explicitamente.
- devolver exatamente um resultado.

Primeiro escreva $O \rightarrow G$.

Inicie o programa e indique o argumento.

◀ █ → SPACE LC x

```
2: 74.8051948052
1: 'gal'
◀ → x █
STD= FIX SCI ENG DEG RAD=
```

A seta à direita indica que o nome a seguir é uma *variável local*, que existe somente dentro deste programa.

Defina a conversão.

◀ x ' oz ' ' g ' █ CONVERT
DROP ENTER

```
2: 'gal'
1: ◀ → x ◀ x 'oz' 'g'
CONVERT DROP » »
STD= FIX SCI ENG DEG RAD=
```

Os delimitadores de encerramento são colocados automaticamente.

Este programa significa: tome um argumento da pilha (em sintaxe RPN) ou da expressão (em sintaxe algébrica) e dê-lhe o nome *x*; converta *x* de onças para gramas; e elimine (drop) a unidade grama da pilha.

Armazene o programa na variável $O \rightarrow G$.

' O █ → G STO

```
3:
2: 74.8051948052
1: 'gal'
STD= FIX SCI ENG DEG RAD=
```

Agora escreva G → O.

Inicie o programa e indique o argumento.

◀ █ → SPACE LC x

```
2: 74.8051948052
1: 'gal'
◀ → x █
STO FIX SCI ENG DEG RAD
```

Defina a conversão.

◀ x ' g ' oz ' █ CONVERT
DROP ENTER

```
2: 'gal'
1: ◀ → x ◀ x 'g' 'oz'
CONVERT DROP » »
STO FIX SCI ENG DEG RAD
```

Este programa significa: tome um argumento da pilha (em sintaxe RPN) ou da expressão (em sintaxe algébrica) e dê-lhe o nome *x*; converta *x* de gramas para onças; e elimine (drop) a unidade onça da pilha.

Armazene o programa na variável G → O.

' G █ → O STO

```
3:
2: 74.8051948052
1: 'gal'
STO FIX SCI ENG DEG RAD
```

Para testar as conversões, verifique quantos gramas são uma onça e, então, converta aquele resultado de volta para onças. O resultado deveria ser 1 novamente.

Converta uma onça para gramas.

1 USER O → G

```
3: 74.8051948052
2: 'gal'
1: 28.349523125
G→O O→G ΣPAR ΣDAT SI PV
```

Existem em torno de 28 gramas em 1 onça. Agora converta este resultado de volta para onças.

G → O

```
3: 74.8051948052
2: 'gal'
1: 1
G→O O→G ΣPAR ΣDAT SI PV
```

As conversões são inversas, como deveriam ser.

Imprimindo

Este capítulo descreve alguns comandos básicos para utilizar a sua HP-28S com uma impressora HP 82240A. Refira-se ao manual da impressora, para instruções sobre como posicionar a impressora relativamente à HP-28S e como ligá-la.

Todos os comandos no menu PRINT são descritos, brevemente, no apêndice D, "Digramas dos Menus". Para descrições completas, ver "PRINT" no Manual de Referência.

Imprimindo o Visor

Você pode imprimir uma imagem do visor como segue.

1. Pressione e mantenha pressionada .
2. Pressione (a tecla com "PRINT" escrito acima dela).
3. Solte .

Esta seqüência de teclas é equivalente, no teclado, ao comando PRLCD (*print LCD = imprima visor*, encontrada na primeira linha do menu PRINT). Você pode utilizar esta seqüência de teclas, para imprimir o visor praticamente a qualquer tempo, sem perturbar a operação da calculadora.

Se você desejar um programa que imprima o visor, simplesmente inclua o comando PRLCD, encontrado no menu PRINT.

Apague a pilha e apresente o menu PRINT.



- PR1 (*print 1 = imprima 1*) imprime o objeto no nível 1.
- PRST (*print stack = imprima a pilha*) imprime todos os objetos da pilha.
- PRVAR (*print variable = imprima a variável*) imprime o nome e conteúdo de uma variável.
- PRLCD (*print LCD = imprima o visor*) imprime o visor.
- CR (*carriage right = carro à direita*) imprime uma linha em branco.
- TRAC (*trace on / off = liga / desliga monitoração de impressão*) liga ou desliga modo de monitoração de impressão (Trace).

Imprimindo um Registro Contínuo

Para imprimir um registro contínuo em seus cálculos, ligue o modo de monitoração de impressão (Trace).

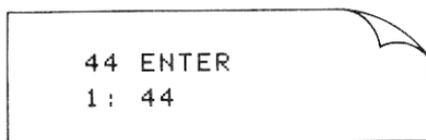
TRAC



Um quadrado aparece no rótulo de menu **TRAC**, para indicar que o modo de monitoração está ligado.

Agora veja o que ocorre quando você soma dois números — por exemplo, 44 e 72. Primeiro coloque 44 na pilha.

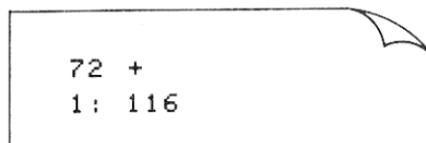
44



A entrada e o resultado do nível 1 são impressos.

Agora adicione 72.

72



Novamente, a entrada e o resultado do nível 1 são impressos.

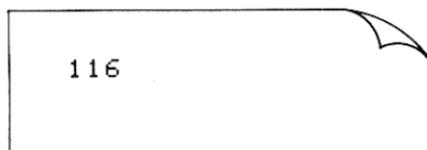
Desligue o modo de monitoração de impressão.



Imprimindo o Nível 1

Ao invés de imprimir todos os resultados utilizando o modo de monitoração de impressão (Trace), você pode imprimir, seletivamente, resultados utilizando PR1.

PR1



O resultado permanece no nível 1, inalterado.

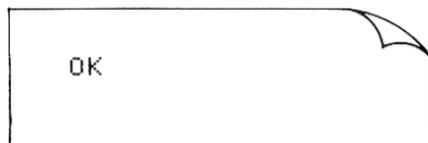
Você pode imprimir uma mensagem, colocando uma cadeia no nível 1. Para imprimir a mensagem "OK", primeiro coloque a cadeia na pilha.

█ " OK ENTER



Agora imprima a mensagem.

PR1

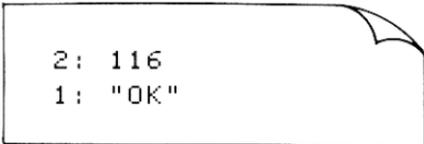


Somente o conteúdo da cadeia é impresso, sem as aspas.

Imprimindo a Pilha

Você pode imprimir todos os objetos da pilha, utilizando PRST.

PRST



```
2: 116
1: "OK"
```

O conteúdo da pilha permanece inalterado.

Imprimindo uma Variável

Você pode imprimir o nome e conteúdo de uma variável, sem trazer a variável para a pilha. Para demonstrar, armazene a cadeia "OK" em uma variável denominada "A", então imprima a variável A.

Crie a variável A com o valor "OK".

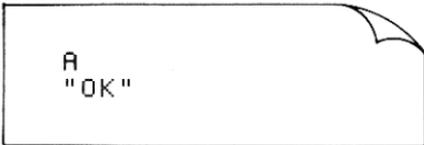
A STO



```
3:
2:
1: 116
PR1 PRST PRVAR PRLCD CR TRAC
```

Imprima o nome e valor da variável.

A PRVAR



```
A
"OK"
```

O nome da variável é eliminado da pilha.

Parte 2

Sumário dos Recursos da Calculadora

| | | |
|---------------|------------|--|
| Página | 154 | 16: Objetos |
| | 164 | 17: Operações, Comandos e Funções |
| | 166 | 18: A Linha de Comando |
| | 176 | 19: A Pilha |
| | 182 | 20: Memória |
| | 192 | 21: Menus |
| | 196 | 22: Catálogo de Comandos |
| | 198 | 23: Processamento |
| | 205 | 24: Modos |
| | 215 | 25: Operações de Sistema |

Objetos

A parte 1 deste manual contém exemplos de 9 tipos básicos de objetos na HP-28S. Objetos são as entidades básicas na calculadora — as entidades que você cria para formular problemas e manipulá-los para encontrar soluções.

A finalidade da maioria dos tipos de objetos é economizar trabalho a você, oferecendo tipos de dados específicos. Por exemplo, imagine se você utilizasse números reais para representar arranjos, de alguma forma tendo que saber onde está cada elemento, em cada arranjo e escrevendo programas para efetuar cálculos aritméticos com estes arranjos. É mais simples entrar os números num objeto arranjo, o qual você pode manipular como uma única entidade e executar cálculos através da utilização das funções aritméticas normais.

Entretanto, a razão para a existência de múltiplos tipos de objetos é mais ampla do que somente múltiplos tipos de dados. Os recursos simbólicos e programáveis da calculadora são baseados em objetos simbólicos (nomes e algébricos) e objetos programa. Estes objetos não são somente dados; eles podem ser processados para produzir um resultado. (Processamento de objetos é discutido no capítulo 23.)

Utilizando o conceito simples de tipos de objetos para fundamentar múltiplos tipos de dados, operações simbólicas e programação, a HP-28S minimiza as regras que você necessita lembrar. Objetos são digitados na linha de comando, colocados na pilha ou armazenados em variáveis exatamente da mesma maneira, não importando o tipo de objeto.

Este capítulo sumariza o que você aprendeu sobre cada tipo de objeto, oferece informações mais detalhadas e sugere usos adicionais.

Números Reais

Números reais representam números maiores que -10^{500} e menores que 10^{500} . Eles são armazenados internamente como uma *mantissa* entre 1 e 9.999999999999, um sinal (positivo ou negativo) para a mantissa, um *expoente* entre 0 e 499, e um sinal para o expoente.

Formato Horas-Minutos-Segundos. Você pode utilizar os comandos HMS+ e HMS – para adicionar ou subtrair números expressos em horas, minutos e segundos (ou graus, minutos e segundos). Para qualquer cálculo além de adição e subtração, primeiro utilize HMS →, para converter os números do formato HMS para o formato graus decimais. (Ver “TRIG”, no Manual de Referência, para detalhes.)

Números Complexos

Objetos número complexo são pares ordenados de números reais que representam a *parte real* e a *parte imaginária* de um número complexo ou as coordenadas de um ponto em um plano.

Coordenadas Retangulares e Polares. Nos capítulos 7 e 8, você utilizou números complexos para traçar gráficos e digitalizar; cada número complexo representava *coordenadas retangulares* — isto é, distâncias ao longo de eixos perpendiculares.

O capítulo 6 descreveu *coordenadas polares* — uma distância radial e um ângulo — e utilizou os comandos R → P e P → R para converter entre coordenadas polares e retangulares. Você pode utilizar coordenadas polares para digitar coordenadas e apresentar resultados no visor, mas precisa utilizar coordenadas retangulares para cálculos. A função do usuário PSUM, descrita à página 86, adiciona pontos em coordenadas polares convertendo-os para retangulares, efetuando a soma e convertendo de volta a coordenadas polares.

Em Objetos Algébricos. Quando você digita um número complexo em um objeto algébrico, você pode necessitar dois pares de parênteses, como na expressão 'SIN((0,1))'. O par de parênteses externos é requerido pela função SIN(), enquanto o par de parênteses internos é o delimitador para números complexos.

Inteiros Binários

Inteiros binários representam uma seqüência de bits. Cujo comprimento de 1 a 64 bits, depende do tamanho de palavra corrente. A base binária inteira corrente determina como inteiros binários são apresentados, mas não tem efeito na sua representação interna.

Inteiros Grandes. Utilizando inteiros binários em modo de base decimal, você pode expressar um inteiro positivo de 19 dígitos exatamente; isto é, 7 dígitos a mais do que você pode expressar exatamente, utilizando números reais.

Exemplo de Programação. Os programas sob o título “Apresentando no Visor um Inteiro Binário”, à página 257, trabalham juntos, para apresentar um número inteiro binário em todas as quatro bases.

Preservando o Estado. O comando RCLF (*recall flags = recupera indicadores*) devolve um inteiro binário, representando o estado de todos os 64 flags do usuário; o comando STOF (*store flags = armazena indicadores*) define os “flags” do usuário de acordo com um argumento inteiro binário. Estes comandos são demonstrados sob o título “PRESERVE (Salva e Restabelece o Estado Anterior)”, um dos programas em “Apresentando no Visor um Inteiro Binário”, descrito acima.

Cadeias

Uma cadeia é constituída de uma seqüência de caracteres. A parte 1 mostrou os seguintes usos para cadeias.

- no capítulo 14, “Conversão de Unidades”, você utilizou cadeias para representar uma combinação de produtos e potências de unidades.
- no capítulo 15, “Imprimindo”, você entrou uma mensagem com uma cadeia para imprimi-la. Você também pode apresentar mensagens no visor, utilizando o comando DISP; descrito no capítulo 27, “Programas Interativos”.

Mais freqüentemente uma cadeia representa texto, mas cada caractere pode também representar um valor numérico de 0 até 255. Os comandos CHR (*caractere*) e NUM (*número do caractere*) convertem caracteres para seus valores numéricos e vice-versa.

Caracteres que Não Pertencem ao Teclado. Você pode apresentar no visor caracteres que não aparecem no teclado da HP-28S, entrando um valor numérico e executando CHR. Existem também caracteres que você não pode apresentar no visor, mas pode imprimir. Para uma lista de todos os caracteres, ver "STRING" no Manual de Referência.

Cadeias Gráficas. O comando LCD \rightarrow (*LCD to string = visor para cadeia*) devolve uma cadeia gráfica que representa a imagem correntemente no visor; o comando \rightarrow LCD (*cadeia para visor*) apresenta a imagem representada por um argumento cadeia-gráfica. Para detalhes sobre estes comandos, ver o Manual de Referência.

Manipulações de Cadeias. Os programas sob o título "Apresentando no Visor um Inteiro Binário", à página 257, mostram como converter um objeto para a forma de cadeia, contar o número de caracteres e juntar duas cadeias.

Arranjos (Arrays)

Arranjos podem ser uni-dimensionais (denominados *vetores*) ou bi-dimensionais (denominados *matrizes*) e podem ser constituídos de números reais ou complexos. O capítulo 11, "Vetores e Matrizes", mostra os cálculos básicos com arranjos. A parte 1 incluiu os seguintes usos adicionais para arranjos:

- o capítulo 11 mostra como resolver um sistema de n equações lineares em n incógnitas, utilizando um vetor constante de n elementos e uma matriz de coeficientes de $n \times n$. Para detalhes sobre este processo e sua precisão, ver "ARRAY" no Manual de Referência.
- no capítulo 12, "Estatística", os dados estatísticos que você entrou foram armazenados na matriz estatística corrente Σ DAT .

Em Sintaxe Algébrica. Se um arranjo está armazenado em uma variável, você pode referir-se a elementos no arranjo, utilizando o nome da variável como uma função. Por exemplo, você poderia representar a soma do terceiro e quinto elementos de um Vetor V como ' $V(3)+V(5)$ '.

Manipulações de Arranjos. Os programas sob o título "Estatísticas com Somatórios", à página 262, e "Mediana de Dados Estatísticos", à página 270, demonstram diversos exemplos de manipulações de arranjos.

Listas

Listas são seqüências de objetos. Elas constituem o método mais geral de combinar-se diversos objetos em um. A parte 1 mostrou as seguintes utilizações para listas:

- no capítulo 4, "Repetindo um Cálculo", o comando PATH (trilha) devolveia uma lista de nomes de diretórios, do diretório HOME até o diretório corrente.
- no capítulo 7, "Traçando Gráficos", a variável lista PPAR continha parâmetros utilizados por DRAW.
- no capítulo 8, "O Solver", você forneceu uma lista contendo três pontos digitalizados como uma estimativa.
- no capítulo 10, "Cálculo", você especificou a variável de integração e os limites inferior e superior de integração, combinando-os em uma lista.
- no capítulo 12, "Estatística", a lista de variáveis ΣPAR continha parâmetros para estatística com dados emparelhados.

Em Sintaxe Algébrica. Se uma lista está armazenada em uma variável, você pode referir-se aos elementos da lista, utilizando o nome da variável como uma função. Por exemplo, você poderia representar a soma do terceiro e quinto elementos da lista L como $L(3)+L(5)$.

Listas e a Pilha. O programa MEDIAN (mediana), à página 273, mostra como colocar os elementos de uma lista na pilha e combinar objetos da pilha em uma lista.

Classificando uma Lista. O programa SORT (classifica), à página 270, mostra como classificar os elementos de uma lista.

Extraindo Elementos de uma Lista. O programa LMED, à página 272, mostra como extrair elementos de uma lista.

Nomes

Nomes são uma seqüência de caracteres, utilizados para atribuir nomes a outros objetos. Eles podem conter até 127 caracteres embora considerações práticas sugiram que os nomes não tenham mais do que 5 ou 6 caracteres.

Os caracteres permitidos, disponíveis no teclado, são letras, dígitos e os caracteres $\{ \} \pi \div \mu \cdot$. O primeiro caractere não pode ser um dígito. Os seguintes caracteres não podem ser incluídos em nomes:

- caracteres que separam objetos: delimitadores ($\# \ [\] \ " \ ' \ (\) \ (\) \ \& \ \& \ ,$ espaço, ponto ou vírgula).
- símbolos de operadores algébricos ($+ \ - \ * \ / \ ^ \ \sqrt{\ } \ = \ < \ > \ \leq \ \geq \ \neq \ \approx \ \int$).

A calculadora determina se um nome é global ou local quando a linha de comando é processada: se o nome é utilizado por uma estrutura de programa para criar uma variável local, o nome é local dentro daquela estrutura de programa; caso contrário, o nome é global.

Nomes Locais. Na parte 1, você escreveu funções do usuário que criaram variáveis locais. Este manual utilizou letras minúsculas para os nomes locais para ajudá-lo a distingui-los de nomes globais. É importante lembrar que foi o comando \rightarrow que fez com que os nomes fossem locais, não as letras minúsculas. Se você atribuir a uma variável local os nomes e ou i , sua definição local substitui a definição padrão.

Nomes Globais. Todos os nomes na parte 1 eram globais. Exemplos incluem:

- nomes para variáveis globais (variáveis numéricas utilizadas para traçado de gráfico ou para o Solver, todas as variáveis no menu USER).
- nomes de diretórios.
- nomes utilizados simbolicamente, sem referência a valores específicos (aritmética simbólica, soluções simbólicas e cálculo diferencial integral).

Nomes de comando incluindo ϵ , i e π não podem ser utilizados como nomes globais. Em adição, os nomes a seguir são reservados para usos específicos.

- EQ se refere à equação corrente utilizada pelo Solver e os comando do PLOT.
- ΣPAR se refere a uma lista de parâmetros utilizados por comandos estatísticos.
- PPAR se refere a uma lista de parâmetros utilizados por comandos de traçado de gráfico.
- ΣDAT se refere ao arranjo estatístico corrente.
- s1, s2 e assim por diante, são criados por ISOL e QUAD para representar sinais arbitrários obtidos em soluções simbólicas.
- n1, n2 e assim por diante, são criados por ISOL para representar inteiros arbitrários obtidos em soluções simbólicas.
- Nomes começando com “der” se referem a derivadas definidas pelo usuário.

Você pode utilizar quaisquer destes nomes para suas próprias finalidades, mas lembre-se de que certos comandos utilizam estes nomes como argumentos implícitos.

Programas

Programas são seqüências de objetos e comandos. Cada programa é, essencialmente, uma linha de comando transformada em um objeto; quando você circunda o conteúdo da linha de comando por delimitadores de programa, você indica que deseja salvar o conteúdo para posterior execução.

Comandos especiais de programa aparecem nos menus PROGRAM BRANCH (desvio de programa), PROGRAM CONTROL (controle de programa) e PROGRAM TEST (teste de programa). Estes menus são descritos no Manual de Referência, juntamente com o tópico geral “Programas”.

Você escreveu cinco programas na parte 1:

- no capítulo 3, você escreveu um programa para modificar nomes de variáveis, e armazenou-o na variável RENAME.
- no capítulo 5, você escreveu um programa para a função cotangente e armazenou-o na variável COT.
- no capítulo 6, você escreveu um programa para somar coordenadas polares e armazenou-o na variável PSUM.
- no capítulo 14, você escreveu programas para converter entre onças e gramas, e armazenou-os nas variáveis $O \rightarrow G$ e $G \rightarrow O$

Funções do Usuário. Os programas COT, PSUM, O → G e G → O são funções do usuário — eles começam com o comando → e um ou mais nomes os quais juntos definem uma ou mais variáveis locais, seguidas por uma expressão ou programa. Quando a função do usuário é armazenada em uma variável, você pode utilizar o nome da variável em expressões ou equações como você utilizaria uma função pré-programada da máquina.

Estruturas de Programa. O comando → seguido por nomes e uma expressão ou programa é denominado *estrutura de variável local*, que é um tipo de estrutura de programa. Existem também estruturas de programa para desvios (tais como IF ... THEN ... ELSE ... END) e “looping” (tais como DO ... UNTIL ... END). Ver capítulo 26, “Estruturas de Programa”, para descrições. Em adição, o capítulo 28, “Exemplos de Programação”, contém 20 programas que demonstram cada estrutura de programa juntamente com uma variedade de técnicas de programação.

Programas Sem Nome. Programas não necessitam ser armazenados em variáveis para serem úteis; para exemplos, ver “Desenvolvendo e Agrupando Completamente”, à página 253, e “Apresentando um Inteiro Binário”, à página 257.

Algébricos

Algébricos se constituem de uma ou mais funções e os argumentos das funções; os argumentos podem ser números, nomes, ou subexpressões. Algébricos são escritos e apresentados em sintaxe algébrica, uma forma similar à notação matemática escrita. Existem dois tipos de algébricos: expressões e equações.

Expressões

Na parte 1, você utilizou expressões de três formas diferentes: como dados, como funções e como equações implícitas.

Expressões como Dados. Quando você calcula com expressões, tal como adicionar duas expressões, elevar uma expressão ao quadrado ou diferenciar uma expressão, o resultado é outra expressão. Nestes casos, as expressões atuam como dados a serem manipulados, independentemente de quaisquer valores atribuídos às variáveis.

Expressões como Funções. No capítulo 4, você criou a expressão RTOT e, utilizando o Solver, atribuiu valores às variáveis e, então, processou RTOT para calcular o resultado desejado. Neste caso, a expressão atuou como uma função, que dados os valores de entrada, produziu um resultado.

Expressões como Equações Implícitas. No capítulo 8, você utilizou o Solver para encontrar o zero *numérico* de uma expressão, isto é, o valor numérico da variável independente para o qual a expressão tenha o valor 0. No capítulo 9, você utilizou QUAD para encontrar um zero *simbólico*, isto é, uma expressão que, colocada no lugar da variável independente, faria com que a expressão original tivesse valor 0.

Em ambos os casos, a expressão $f(x)$ atua como a equação $f(x) = 0$ porque o zero da expressão é o mesmo que a raiz da equação.

Equações

Equações se constituem de duas expressões relacionadas pelo sinal de igual (=). Em matemática, existem dois usos para o sinal de igual:

- para indicar uma proposição, tal como " $x^2 = 4$ " ou " $x^2 + y^2 = 1$ ". Aqui a equação é válida somente para alguns valores das variáveis.
- para indicar uma identidade ou definição, tal como " $\sin 2x = 2 \sin x \cos x$ " ou " $y = 3x^2 + 2x + 5$ ". Aqui a equação é válida para todos os valores das variáveis.

Na HP-28S, equações são utilizadas apenas para proposições; para efetuar-se uma definição tal como " $y = 3x^2 + 2x + 5$ ", a expressão ' $3**X^2+2**X+5$ ' é armazenada em uma variável denominada Y.

Em "Valor do Dinheiro no Tempo" à página 103, TVM e SPPV são expressos matematicamente como equações. A equação TVM, que é válida somente para certos valores e suas variáveis, é entrada como uma equação; mas SPPV, cujo valor é definido pelos valores de suas variáveis, é criada como uma variável.

Equações como Dados. Quando você calcula com equações, tal como adicionar duas equações ou elevar uma equação ao quadrado, ou diferenciar uma equação, o resultado é outra equação. Cada membro da equação é tratado independentemente — cada membro é uma expressão tratada como dados. A equação mantém sua natureza proposicional, onde ela é válida somente para alguns valores de suas variáveis.

Resolvendo Equações. Quando você resolve uma equação numericamente, como fez em “Valor do Dinheiro no Tempo”, você encontra o valor da variável independente que satisfaça a igualdade. Similarmente, quando você resolve uma equação simbolicamente, como fez em “Isolando uma Variável”, à página 109, você encontra uma expressão que, se colocada no lugar da variável independente, satisfaria a equação.

Constantes Simbólicas

Algébricos podem incluir as seguintes constantes simbólicas. Estas parecem nomes, mas na realidade são funções.

- MINR (*mínimo real*) representa o menor número real positivo. Seu valor numérico é 1.00000000000E-499.
- MAXR (*máximo real*) representa o maior número real positivo. Seu valor numérico é 9.99999999999E499.
- e representa a base dos logaritmos naturais. Seu valor numérico na HP-28S é 2.71828182846.
- π representa a relação da circunferência para o diâmetro do círculo. Seu valor numérico, na HP-28S, é 3.14159265359.
- i representa o número imaginário $\sqrt{-1}$. Seu valor numérico é (0,1).

Em modo Constantes Numéricas ou modo Resultado Numérico, o processamento de constantes simbólicas devolve seus valores numéricos; em outros casos, o processamento devolve sua forma simbólica. (Modo Constantes e modo Resultado são descritos no capítulo 24.)

Operações, Comandos e Funções

Cada procedimento pré-programado na HP-28S pode ser classificado como uma operação, um comando, uma função ou uma função analítica.

- Uma *operação* é qualquer procedimento pré-programado na calculadora.
- Um *comando* é uma operação programável.
- Uma *função* é um comando permitido em algébricos.
- Uma *função analítica* é uma função para a qual a HP-28S fornece uma derivada e uma inversa.

Procedimentos pré-programados são, usualmente, caracterizados por sua capacidade mais alta. Por exemplo, SWAP e IP são ambos comandos, mas caracterizamos SWAP como um comando e IP como uma função. A seguinte tabela mostra exemplos de cada tipo.

Operações

| Operações Não-Programáveis | Comandos | | |
|--|--|---|--|
| | RPN Comandos | Funções | |
| | | Não-Analíticas | Analíticas |
| INS NEXT EDIT VIEW ↓ ENTER EEX COMMAND UNDO CONT ON | SWAP DROP LAST RCL PURGE ∫ STO EVAL CLEAR CONVERT | ABS ∂ IP MAX OR %CH R → D R → P XPON ≠ | ASIN EXP INV LN NEG SIN SINH SQ + = |

O índice de operações, à página 381 do Manual de Referência, identifica cada procedimento pré-programado como uma operação, um comando, uma função ou uma função analítica. Como uma indicação, seguem alguns comentários gerais sobre cada tipo.

- A maioria das operações não programáveis pode ser executada somente pressionando-se uma tecla. Entretanto, existem equivalentes programáveis para algumas operações: por exemplo, a operação $\boxed{\text{TRIG}}$ (para selecionar o menu TRIG) pode ser efetuada em um programa, executando-se $\boxed{21}$ MENU, e a operação $\boxed{\text{RAD}}$ (para selecionar modo angular radiano) pode ser efetuada executando $\boxed{60}$ FS.
- A maioria dos comandos RPN envolve manipulação da pilha ou alteração da memória do usuário, ao invés de calcular valores matemáticos.
- A maioria das funções não analíticas são cálculos matemáticos sem inversos, isto é, elas devolvem algumas características dos argumentos, mas os argumentos não podem ser reconstruídos dos resultados. Exemplos incluem a parte inteira e parte fracionária, valor absoluto e sinal.
- Em matemática, uma função de variáveis complexas é *analítica* se ela pode ser expressa como uma série de potência em todos os pontos de seu domínio; neste caso, ela tem uma inversa e uma derivada. A HP-28S faz poucas exceções a esta definição. Por exemplo, nenhuma derivada é fornecida para a função $\frac{1}{z}$, embora uma fosse possível; uma derivada é dada para a função ABS embora a função seja não analítica no ponto $0 + 0i$.

Cada procedimento pré-programado está disponível em uma tecla, seja no teclado, ou seja, em um menu. Quando você pressiona uma tecla, o resultado exato depende do tipo de procedimento e do *modo de entrada*, como será discutido no próximo capítulo.

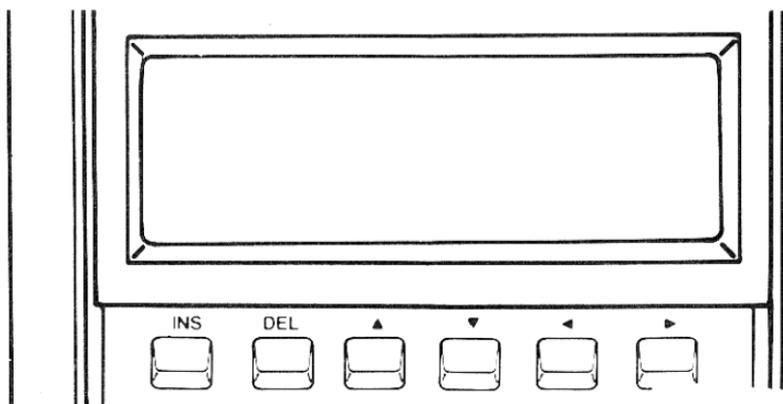
A Linha de Comando

A linha de comando aceita qualquer número de caracteres representando objetos em forma de texto. Ela aparece na parte inferior do visor (imediatamente acima dos rótulos de menu, se presentes) quando você inicia a digitação de um objeto ou quando você utiliza  ou  , para editar o conteúdo de um objeto existente.

A linha de comando pode conter mais do que uma linha de texto. Se você entrar mais do que 23 caracteres em uma linha, os caracteres “rolam” para fora do visor, à esquerda. Reticências (...) aparecem na posição do caractere mais à esquerda, para indicar os caracteres não apresentados. Se você tentar mover o cursor além do extremo esquerdo do visor, os caracteres mais à esquerda “rolam” de volta para o visor, e caracteres “rolam” para fora do visor à direita. Reticências aparecem ao lado direito do visor. Quando a linha de comando contém múltiplas linhas de texto, todas as linhas “rolam” à esquerda e à direita juntas.

O Menu do Cursor

O menu do cursor é um menu especial de operações de edição. Ele está ativo sempre que a linha de comando está presente e não existem rótulos de menu visíveis. O menu do cursor contém teclas prefixadas e não prefixadas. As teclas não prefixadas estão rotuladas em branco, acima da tecla do menu correspondente, como ilustrado.



Se você pressionar e mantiver pressionada uma tecla do menu do cursor não prefixada (exceto **INS**), a operação é repetida até que você solte a tecla.

Tecla

Descrição

- | | |
|------------|---|
| INS | Muda entre modo Substituição ("Replace") e modo Inserção ("Insert"). Em modo Substituição, novos caracteres substituem os caracteres existentes. Em modo Inserção, novos caracteres são inseridos entre os caracteres existentes. |
| DEL | Elimina o caractere na posição do cursor. |
| ▲ | Move o cursor uma linha para cima. |
| ▼ | Move o cursor uma linha para baixo. |
| ◀ | Move o cursor à esquerda uma posição. |
| ▶ | Move o cursor à direita uma posição. |

As teclas do menu do cursor prefixadas (exceto **INS**) são equivalentes a repetições das operações não prefixadas.

| Tecla | Descrição |
|--|---|
|  INS | Elimina todos os caracteres à esquerda do cursor. |
|  DEL | Elimina o caractere na posição do cursor e todos os caracteres à direita. |
|  ▲ | Move o cursor para a linha superior da linha de comando. |
|  ▼ | Move o cursor para a linha inferior da linha de comando. |
|  ◀ | Move o cursor para a extrema esquerda da linha de comando. |
|  ▶ | Move o cursor para a extrema direita da linha de comando. |

Algumas Teclas de Entrada

As teclas abaixo são úteis quando você está entrando objetos na linha de comando.

| Tecla | Descrição |
|--|---|
|  ◀▶ | Liga/Desliga Menu do Cursor. Quando o menu do cursor não está ativo, seleciona o menu do cursor. Quando o menu do cursor está ativo, seleciona o menu anterior. |
|  CHS | Troca Sinal. Quando o cursor está posicionado em um número, troca o sinal do número. Quando o cursor não está posicionado em um número, escreve um sinal de menos. (Se a linha de comando não está presente, executa o comando NEG.) |
|  EEX | Entra Expoente. Quando o cursor está posicionado em um número sem um expoente, escreve o caractere E após o número. Quando o cursor está posicionado em um número com expoente, posiciona o cursor após o E. Se o cursor não está posicionado em um número, escreve 1E. |
|  ◀ | Retrocesso. Elimina o caractere à esquerda do cursor, movendo o cursor (e quaisquer caracteres à direita) um espaço à esquerda. Se você pressionar e mantiver pressionado  , a ação é repetida até que você solte a tecla. |
|  LC | Letras Minúsculas. Troca entre o modo Letras Maiúsculas e Letras Minúsculas. Quando a linha de comando é criada, modo Letras Maiúsculas está ativado —  até  escreve Â até Z . Em modo de Letras Minúsculas,  até  escreve a até z |

■ **MENUS**

Chave do Menu. Liga e desliga a chave do menu. Quando a chave do menu está ligada, as posições prefixadas e não prefixadas são trocadas para as três linhas superiores do teclado da esquerda (tecla de letras **A** até **R**). Você não necessita pressionar ■ antes de **ARRAY** até **UNITS**, mas você precisa pressionar ■ antes das letras A até R.

ON

Atenção. Cancela linha de comando.

Delimitadores de Objetos e Separadores

Para entrar mais que um objeto ou comando em uma mesma linha de comando, você deve separá-los por um dos seguintes meios:

- um delimitador de objeto: () [] { } # " ' « ».
- um espaço ou uma nova linha. Pressionando ■ **NEWLINE** insere um caractere “nova linha” (muda de linha) na linha de comando, na posição de cursor. Caracteres de “nova linha” são equivalentes a espaços quando a linha de comando é executada.
- uma vírgula (admitindo que você não tenha selecionado a vírgula para servir como ponto decimal).

Modos de Entrada

Para tornar mais fácil a entrada de objetos, existem três modos de entrada — *Imediato*, *Algébrico* e *Alfanumérico* — para entrar diferentes tipos de objetos. Lembre-se das distinções feitas no capítulo anterior, “Operações, Comandos e Funções”:

- *operações* não são programáveis.
- *comandos* podem aparecer em programas, mas não em algébricos.
- *funções* (analíticas e não analíticas) e *nomes* podem aparecer em programas ou algébricos.

A calculadora reconhece estas distinções à medida que você entra objetos na linha de comando. Pressionar uma tecla de operação (tal como **ENTER**), sempre causa execução da operação. O modo de entrada corrente afeta, primordialmente, o que ocorre quando você pressiona uma tecla de comando (tal como **STO**), uma tecla de função (tal como **+**), ou uma tecla do menu **USER**.

Modo de Entrada Imediato. Este modo é para a entrada de números, listas e arranjos. Em modo de entrada Imediato:

- pressionar uma tecla de comando executa a linha de comando e, então, executa o comando.
- pressionar uma tecla de função executa a linha de comando e, então, executa a função.
- pressionar uma tecla do menu **USER** executa a linha de comando e, então, processa o nome correspondente.

Modo de Entrada Algébrico. Este modo é para a entrada de nomes e algébricos. Se você começar a linha de comando pressionando **'**, o modo de entrada Algébrico é, automaticamente, ativado. Neste modo:

- pressionar uma tecla de comando executa a linha de comando e, então, executa o comando.
- pressionar uma tecla de função escreve o nome da função na linha de comando. Se a função exige seus argumentos entre parênteses, a abertura de parênteses está incluída.
- pressionar uma tecla do menu **USER** escreve o nome correspondente na linha de comando.

Modo de Entrada Alfanumérico. Este modo é para entrada de cadeias e programas. Pressionar **▀** **"** ou **«**, automaticamente, ativa modo de entrada Alfanumérico e liga o anúncio α . Neste modo:

- pressionar uma tecla de comando escreve o nome do comando na linha de comando.
- pressionar uma tecla de função escreve o nome da função na linha de comando.
- pressionar uma tecla do menu **USER** escreve o nome correspondente na linha de comando.

Se o cursor está posicionado no fim da linha de comando, ou se o modo de Inserção (Insert) está ativado, espaços são incluídos, quando necessário, para manter comandos sucessivos separados.

Exceções

Para permitir que você selecione um modo enquanto utiliza a linha de comando nos modos de entrada Imediato ou Algébrico, as seguintes teclas de comando executam seus comandos sem perturbar a linha de comando.

- `STD` , `DEG` e `RAD` no menu `MODE`.
- `DEC` , `HEX` , `OCT` e `BIN` no menu `BINARY`.

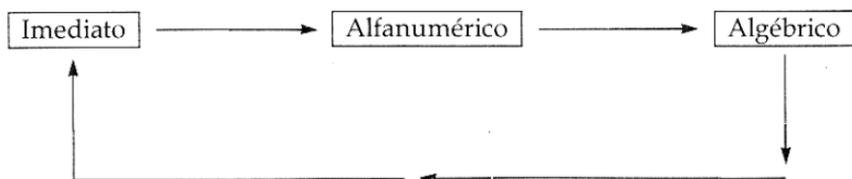
Uma vez que os comandos a seguir somente fazem sentido em um programa, pressionar uma destas teclas, sempre escreve o nome do comando na linha de comando.

- `HALT` no menu `PROGRAM CONTROL` (Controle de Programa).
- Todas as teclas no menu `PROGRAM BRANCH` (Desvio de Programa).

Para ajudar a prevenir a perda acidental de variáveis, pressionar `CLUSR` (no menu `MEMORY`) sempre escreve `CLUSR` na linha de comando. Você precisa pressionar `ENTER` para executar o comando.

Seleção Manual de Modos de Entrada

A calculadora, automaticamente, muda entre modos de entrada Imediato e Algébrico cada vez que você pressiona `'`, para iniciar ou terminar um nome ou um algébrico. Ela também muda para modo de entrada Alfanumérico quando você pressiona `▀` ou `«`. Você pode, manualmente, selecionar o modo de entrada, pressionando a tecla `α`. Fazendo isso, você muda o modo de entrada no ciclo ilustrado abaixo.



Seleção Manual de Modos de Entrada

Assim você pode mudar para qualquer modo de entrada, pressionando α , uma ou duas vezes. Abaixo estão alguns exemplos da utilização da tecla α .

- Suponha que você queira escrever um programa que executará somente uma ou duas vezes. Pressione α para selecionar modo de entrada Alfanumérico; digite o programa sem delimitadores de programa; pressione ENTER para executar o programa; pressione \blacksquare COMMAND para retornar o programa à linha de comando; pressione ENTER para executar o programa novamente.
- Suponha que você queira eliminar diversas variáveis de uma só vez. Pressione [] para iniciar uma lista; pressione α para selecionar modo de entrada Alfanumérico; pressione as teclas do menu USER para as variáveis a serem eliminadas; pressione ENTER para colocar a lista na pilha; pressione \blacksquare PURGE
- Suponha que você esteja digitando um programa e queira utilizar o caractere \rightarrow em um nome. Uma vez que o modo de entrada Alfanumérico está ativado, pressionar \blacksquare \rightarrow iria escrever o comando " \rightarrow " circundado por espaços. Pressione α para selecionar modo de entrada Algébrico; pressione \blacksquare \rightarrow ; pressione α α para retornar ao modo de entrada Alfanumérico.

Como o Cursor Indica os Modos

A aparição do cursor indica o modo de entrada e a escolha corrente do modo de Inserção (Insert) ou de Substituição (Replace). A tabela abaixo mostra as seis possíveis combinações de modos de entrada e modo de Inserção ou Substituição.

| | Modo de Inserção | Modo de Substituição |
|-------------------------------------|------------------|----------------------|
| Modo de entrada imediato | \diamond | \square |
| Modo de entrada algébrico | \dagger | [] |
| Modo de entrada alfanumérico | \blacklozenge | [] |

Executando a Linha de Comando

Quando você pressiona `ENTER` (ou uma tecla que executa ENTER no modo de entrada corrente), a calculadora faz o seguinte:

1. o anúncio ocupado ((•)) é ligado.
2. se UNDO está ativado, uma cópia do conteúdo corrente da pilha é armazenada.
3. uma busca é efetuada na cadeia de texto, na linha de comando, para identificar delimitadores e separadores de objetos, e então quebradas nas correspondentes subcadeias.
4. cada subcadeia de texto é testada contra as regras de sintaxe, para identificar seu tipo de objeto.
5. se COMMAND está ativado, uma cópia da linha de comando é armazenada na pilha de comando.
6. a linha de comando é executada.
7. o anúncio ocupado ((•)) é desligado.

Se uma subcadeia não passa no teste de sintaxe no passo 4, os passos 5 e 6 não são executados. Ao invés disso, `SYNTAX ERROR` é apresentada no visor e o texto incorreto é salientado em caracteres inversos, seguidos pelo cursor. Se o erro resultou de sintaxe incompleta, o cursor é posicionado no fim da linha.

Editando Objetos Existentes

Você pode devolver um objeto existente à linha de comando, vê-lo ou editá-lo, utilizando operações de linha de comando e substituir o objeto original com o objeto modificado se desejado.

| Tecla | Descrição |
|---------------------------|--|
| <code>EDIT</code> | Editar Nível 1. Devolve o objeto do nível 1 para a linha de comando. |
| <code>n VISIT</code> | Editar Nível n. Devolve o objeto do nível <i>n</i> para a linha de comando. |
| <code>'name' VISIT</code> | Editar uma Variável. Devolve o conteúdo da variável especificado para a linha de comando. |

O menu do cursor e o modo de entrada Alfanumérico são ativados. O objeto original, se visível, é realçado para lembrá-lo de que você está editando aquele objeto e que a cópia original ainda está preservada.

Quando você terminar de ver ou editar o objeto, você pode:

- pressionar **ON** para cancelar a edição, apagar a linha de comando e deixar o objeto original inalterado.
- pressionar **ENTER** (ou uma tecla que execute ENTER) para substituir o objeto original.

Se o menu do cursor ainda estiver ativo quando você completar a edição, o menu anterior é restabelecido.

Recuperando Linhas de Comando

A HP-28S salva o conteúdo das últimas quatro linhas de comando que você executou. Pressionar **COMMAND** uma vez, devolve a última linha de comando executada (substituindo a linha de comando corrente, se ela existir); pressionar **COMMAND** uma segunda vez, devolve a penúltima linha de comando; e assim por diante. Se você pressionar **COMMAND** mais do que quatro vezes, a seqüência recomeça com a linha de comando mais recente.

Algumas utilizações para **COMMNAD** aparecem sob o título “Se Você Executar a Função Errada”, à página 47 e “Seleção Manual de Modos de Entrada”, à página 171.

Você pode desativar este recurso, pressionando **CMD** no menu MODE. O quadrado desaparece do rótulo do menu, indicando que linhas de comando não serão salvas. Para ativar este recurso novamente, pressione **CMD** uma segunda vez.

A Linha de Comando como uma Cadeia

O texto que você digita na linha de comando é equivalente ao conteúdo de um objeto cadeia, isto é, uma seqüência de caracteres. Você pode executar através de programa uma linha de comando, entrando o texto em uma cadeia e executando STR→ (*string-to-objects* = *cadeia-para-objetos*). Esta técnica é útil para armazenar programas em forma de texto, que é uma forma mais compacta do que a forma objeto. Quaisquer nomes locais que também existam quando STR→ é executado serão reconhecidos na linha de comando.

A Pilha

Este capítulo revê o que você aprendeu sobre a pilha e descreve comandos para manipulação de objetos na pilha. Também são descritos, brevemente, o uso de variáveis locais, para simplificar manipulações na pilha.

Revisão de Conceitos da Pilha

A pilha é uma seqüência de *níveis* numerados, cada um contendo um objeto. Os objetos que você digita na linha de comando são colocados na pilha quando você executa ENTER. O primeiro objeto na linha de comando é o primeiro objeto colocado na pilha. Cada objeto é colocado no nível 1, elevando outros objetos para o próximo nível mais alto na pilha. A pilha pode crescer indefinidamente (dentro dos limites da memória da calculadora), de forma que você não precisa pensar quantos objetos estão na pilha antes de entrar mais objetos.

Em geral, um comando remove objetos de entrada (denominados *argumentos*) da pilha e os substitui por objetos de saída (denominados *resultados*), na pilha. Por exemplo, a função + remove dois argumentos dos níveis 1 e 2, substituindo-os pela sua soma no nível 1.

Os argumentos precisam estar presentes na pilha antes do comando ser executado. Este tipo de lógica, onde o comando vem após os argumentos, é chamado *lógica de pilha* ("stack logic"), *lógica pós-fixada* ou RPN — *Reverse Polish Notation* = *Notação Polonesa Reversa*, em honra ao matemático polonês Jan Lukasiewicz (1878 - 1956).

Os resultados de um comando são disponíveis como argumentos para o próximo comando. Se você não está pronto para utilizar os resultados ainda, simplesmente deixe-os na pilha — eles estarão disponíveis quando você estiver pronto para utilizá-los.

Objetos deixam a pilha através do nível 1 e os objetos remanescentes na pilha descem cada um para um nível mais baixo. É melhor “deixar cair” (drop), ou seja, eliminar um objeto ou armazená-lo em uma variável, quando você não o necessita na pilha; isto torna mais fácil acompanhar na pilha os objetos que você necessita. Similarmente, é melhor apagar a pilha inteira quando você inicia um problema, assim saberá que os objetos na pilha são pertinentes ao seu problema.

Vendo a Pilha

Normalmente, você vê somente os primeiros objetos da pilha. Se o objeto do nível 1 é grande, você vê somente a primeira parte dele. As operações  e  permitem que você veja a primeira parte de qualquer objeto na pilha e de todo o objeto no nível 1.

Estas operações movem a “janela” através da qual você vê a pilha. O tamanho desta janela pode variar de uma a quatro linhas do visor, dependendo da presença de um menu, da linha de comando ou ambos.

Tecla

Descrição



Move a janela para cima (na direção dos níveis mais altos da pilha).



Move a janela para baixo (na direção do fim do objeto no nível 1).

Ver a pilha não tem nenhum efeito no seu conteúdo, na linha de comando ou na ação dos comandos.

Manipulando a Pilha

Na parte 1, você utilizou alguns comandos básicos para manipulação da pilha: CLEAR (para apagar a pilha), DROP (para “deixar cair” — eliminar — o objeto no nível 1) e SWAP (para trocar a ordem dos objetos nos níveis 1 e 2). Esta seção, brevemente, descreve todos os comandos para mover, copiar e “deixar cair” objetos da pilha; para detalhes, ver “STACK” no Manual de Referência.

Movendo Objetos da Pilha. Estes comandos rearranjam os objetos na pilha, o número de objetos não muda. Comandos precedidos por “*n*” requerem um número real como argumento.

| Comando | Descrição |
|----------------|---|
| SWAP | Movê o objeto no nível 2 para o nível 1. |
| ROT | Movê o objeto no nível 3 para o nível 1. |
| <i>n</i> ROLL | Movê objeto no nível <i>n</i> para o nível 1. |
| <i>n</i> ROLLD | Movê objeto no nível 1 para <i>n</i> . |

Os nomes de comando ROT (*rotate* = *rolar*), ROLL e ROLLD (*roll down* = *rola para baixo*) são descritivos dos movimentos dos objetos como um bloco. ROT move o objeto no nível 3 para o nível 1, rolando um bloco de três objetos; ROLL e ROLLD rolam blocos de *n* objetos.

Copiando Objetos da Pilha. Estes comandos devolvem uma cópia de um ou mais objetos da pilha. Copiar somente um objeto, devolve a cópia ao nível 1 e eleva os outros objetos na pilha (incluindo o objeto original) para um nível mais alto. Quando você copia mais do que um objeto, eles são copiados como um bloco, de uma maneira similar. Comandos precedidos por “*n*” requerem um número real como argumento.

| Comando | Descrição |
|----------------|---|
| DUP | Copia o objeto no nível 1. (Quando não há linha de comando, você pode executar DUP pressionando <input type="text" value="ENTER"/> .) |
| OVER | Copia o objeto no nível 2. |
| <i>n</i> PICK | Copia o objeto no nível <i>n</i> . |
| DUP2 | Copia os objetos nos níveis 1 e 2. |
| <i>n</i> DUPN | Copia os objetos nos níveis 1 até <i>n</i> . |

Deixando Cair Objetos da Pilha. Estes comandos eliminam um ou mais objetos da pilha. Os objetos remanescentes na pilha descem para um nível mais baixo. Comandos precedidos por "n" requerem um número real como argumento.

| Comando | Descrição |
|----------------|--|
| DROP | Elimina o objeto no nível 1. |
| DROP2 | Elimina os objetos nos níveis 1 e 2. |
| n DROPN | Elimina os objetos nos níveis 1 até n. |
| CLEAR | Elimina todos os objetos. |

Variáveis Locais

Na parte 1, você escreveu algumas funções do usuário — programas que definem variáveis locais e utilizam-nas em uma única expressão ou programa. Funções do usuário podem ser incluídas em Algébricos, da mesma forma que funções que são parte intrínseca da calculadora.

A utilização de variáveis locais reduz a necessidade de manipulações da pilha. Quando você cria variáveis locais, seus valores são removidos da pilha. Você pode, então, referir-se a eles pelo nome ao invés de encontrá-los na pilha.

Variáveis locais têm aplicações adicionais além de funções do usuário. Quase todos os exemplos de programação, no capítulo 28, utilizam variáveis locais. De particular interesse são as "Funções Caixa" ("Box Functions"), à página 241; "MULTI (Execução Múltipla)", à página 253; "PRESERVE (Salva e Restabelece o Estado Anterior)", à página 258 e "SORT (Classifica uma Lista)" à página 270.

Recuperando os Últimos Argumentos

A HP-28S salva os argumentos do último comando executado. Dependendo do comando, um, dois ou três objetos podem ser salvos. (Se um comando não tem argumentos, os argumentos salvos anteriormente são preservados.) O comando LAST devolve os argumentos salvos para os níveis que eles ocupavam originalmente, na pilha.

N.T. Utilizamos a expressão "deixar cair" como tradução para o comando DROP por representar exatamente a idéia de como ele elimina os objetos da pilha, isto é, o último a ser entrado "cai" (é eliminado) e todos os demais "caem" uma posição.

Se você necessitar exatamente dos mesmos argumentos para dois ou mais comandos em seqüência, você pode executar LAST, para devolver cópias dos argumentos para a pilha, para o próximo comando. Se os comandos não requerem exatamente os mesmos argumentos, ou se os comandos não estão em seqüência, é mais fácil utilizar variáveis locais.

Você pode desativar LAST, isto é, deixar de salvar os argumentos, pressionando LAST no menu MODE. O quadrado desaparece do rótulo do menu, indicando que os argumentos não serão salvos. De um modo geral, esta prática não é recomendada, uma vez que a calculadora utiliza os argumentos salvos para recuperação, quando um erro ocorre. Entretanto, se um comando ou programa não pode ser executado por memória insuficiente, você pode tentar execução com LAST desativado. Quando você terminar, assegure-se de ativar LAST novamente, pressionando LAST uma segunda vez.

Restaurando a Pilha

Cada vez que você pressiona **ENTER** (ou uma tecla que executa ENTER) a HP-28S primeiro salva uma cópia da pilha e, então, executa as ações especificadas. Se você não está satisfeito com os resultados, você pode restaurar a pilha salva, pressionando **UNDO**. Note que UNDO (desfazer) afeta somente a pilha — ele não desfaz modificações para flags ou variáveis do usuário. Para um exemplo utilizando **UNDO**, ver “Se Você Executar a Função Errada”, à página 47.

Você pode desativar este recurso pressionando **UNDO** no menu MODE. O quadrado desaparece do rótulo do menu, indicando que a pilha não será salva. Para ativar este recurso novamente, pressione **UNDO** uma segunda vez.

A Pilha como uma Lista

O conteúdo da pilha é equivalente ao conteúdo de uma lista, isto é, uma seqüência de objetos. Você pode colocar todos os objetos da pilha em uma única lista, executando `DEPTH →LIST`. O comando `DEPTH` devolve o número de objetos na pilha e o comando `→LIST` (*stack to list = da pilha para a lista*) combina o número especificado de objetos em uma lista.

Uma lista é “aberta” na pilha, mais freqüentemente, pelo comando `LIST →` (*da lista para a pilha*). Após os elementos serem manipulados na pilha, eles podem ser recombinaados em uma lista pelo comando `→LIST`. Para exemplos destes comandos, ver “`MEDIAN` (Mediana de Dados Estatísticos)”, à página 273.

Memória

A memória é utilizada para uma série de finalidades na HP-28S, incluindo a linha de comando, a pilha, memória do usuário, recursos de recuperação e o sistema operacional. A linha de comando e a pilha são descritas nos capítulos 18 e 19. Este capítulo discute primordialmente a memória do usuário, incluindo diretórios; também discute condições de pouca memória e seus efeitos nos recursos de recuperação e no sistema operacional.

Memória do Usuário

A memória do usuário pode conter variáveis e pode conter diretórios para organizar as variáveis.

Variáveis Globais

Uma variável é a combinação de um objeto nome e qualquer outro objeto. O objeto nome representa o nome da variável; o outro objeto é o valor ou o conteúdo da variável.

Variáveis *globais* são aquelas que são armazenadas na memória do usuário. Existem também variáveis *locais*, que são criadas por estruturas de programas e existem somente durante a execução das estruturas de programa. Variáveis locais são, fundamentalmente, um substituto para manipulações da pilha e são descritas no capítulo 19, "A Pilha". No capítulo presente, o termo "variáveis" indica variáveis globais.

O conteúdo de uma variável pode ser qualquer tipo de objeto. Na parte 1, você criou variáveis numéricas, variáveis programas, variáveis algébricas, variáveis lista e variáveis arranjo. Você até mesmo criou variáveis nome onde o conteúdo de uma variável era o nome de outra variável.

Você utilizou os comandos a seguir para criar, recuperar e eliminar variáveis. Estes comandos tratam todas as variáveis da mesma forma, independentemente de seu conteúdo.

| Comando | Descrição |
|----------------|---|
| STO | Cria uma variável com o valor e nome especificados. |
| RCL | Recupera o conteúdo da variável especificada. |
| PURGE | Elimina uma ou mais variáveis especificadas. |

Diretórios

No capítulo 4, “Repetindo um Cálculo”, você utilizou o Solver para calcular a resistência total de dois circuitos série-paralelo, com dois conjuntos de valores de resistência que poderiam ser aplicados a cada um dos circuitos. Segue uma revisão dos conceitos que você aprendeu.

Existem duas motivações principais para criar-se diretórios.

- Para *agrupar* as variáveis para uma aplicação ou tópico em particular. Você criou o diretório EE para seus problemas de engenharia elétrica de forma que, quando você estiver trabalhando nesses problemas, você pode centrar a sua atenção nas variáveis relevantes. Também igualmente importante é o fato de que, quando você estiver trabalhando em outros problemas, as variáveis de engenharia elétrica estarão “escondidas” dentro do diretório EE.
- Para *manter separados* conjuntos de variáveis que utilizam os mesmos nomes. Você criou os diretórios SP1 e SP2 (*série-paralelo-1* e *série-paralelo-2*) dentro de EE para conter valores diferentes das variáveis R1, R2 e R3. Você pode mudar de um conjunto de valores para o outro simplesmente mudando de diretórios.

Criando um Diretório. Para criar um diretório, você entra um nome e executa CRDIR (*criar diretório*). O nome do diretório aparece no menu USER. O novo diretório é denominado um *subdiretório*, e o diretório que o contém é denominado seu *diretório-pai*.

O Diretório Corrente. Inicialmente, o único diretório que existe é o diretório permanente HOME. Após criar outros diretórios, você pode escolher qual é o *diretório corrente*, isto é, qual conjunto de variáveis aparece no menu USER.

Para escolher o diretório corrente, você processa seu nome. Por exemplo, se você acabou de criar um diretório, você o torna diretório corrente pressionando a tecla apropriada no menu USER.

Quase todos os comandos que utilizam variáveis funcionam somente no diretório corrente, uma vez que a finalidade de múltiplos diretórios é controlar quais variáveis estão disponíveis. Você pode alterar uma variável somente se ela estiver no diretório corrente.

Os seguintes comandos no menu MEMORY atuam no diretório corrente.

| Comando | Descrição |
|----------------|--|
| VARS | Devolve uma lista de nomes de todas as variáveis e diretórios ao diretório corrente. |
| ORDER | Reordena variáveis e diretórios no diretório corrente conforme especificado por uma lista. |
| CLUSR | Elimina todas as variáveis e diretórios vazios no diretório corrente. |

A Trilha Corrente. Você pode verificar onde está na estrutura de diretório, executando o comando PATH (trilha). Ele devolve uma lista, especificando a seqüência de diretórios desde o diretório HOME até o diretório corrente.

Em alguns casos, a calculadora busca não somente o diretório corrente, mas toda a trilha corrente. A busca começa no diretório corrente; se a variável não for encontrada, a busca continua no diretório-pai; este processo continua de volta até o diretório HOME.

Isto ocorre no processamento de nomes — afinal de contas, você nunca poderia retornar a um diretório-pai se não conseguisse processar seu nome. Processamento de nomes ocorre quando você digita um nome sem aspas, quando traça um gráfico ou utiliza o Solver, quando processa algébricos na pilha, etc.

Outros comandos que buscam a trilha corrente são RCL e PRVAR (*imprima variáveis*). Note que nenhuma das ações que buscam a trilha corrente podem alterar a variável.

Uma vez que o diretório HOME está sempre na trilha corrente, a calculadora pode sempre encontrar variáveis no diretório HOME. Você pode decidir limitar o conteúdo do diretório HOME a subdiretórios e àquelas variáveis que você deseja sempre disponíveis.

Estrutura de Diretório. Os diagramas abaixo mostram a estrutura de diretório que você criou no capítulo 4. No primeiro diagrama, HOME é o diretório corrente; no segundo, EE; e no terceiro, SP2. Cada diagrama utiliza os seguintes símbolos:

Símbolos Utilizados no Diagrama de Diretórios

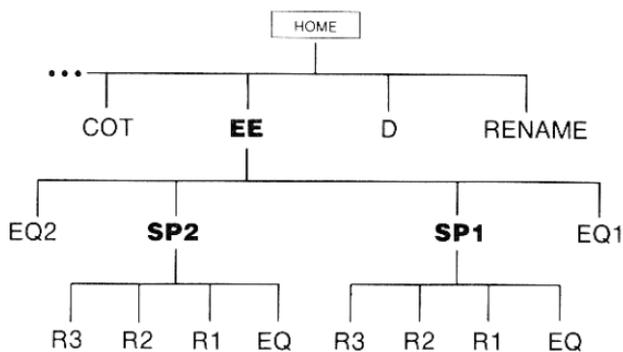
nome Nome de um diretório.

nome Nome do diretório corrente.

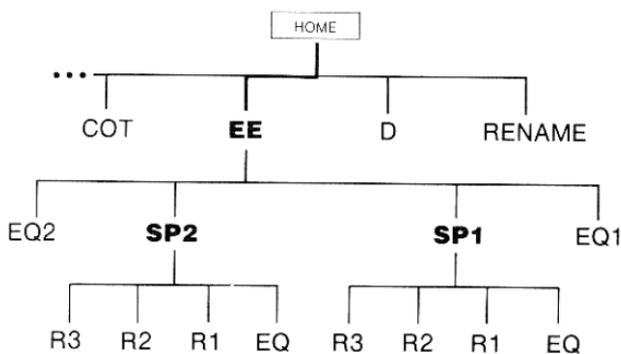
nome Um nome no diretório corrente. Estes nomes aparecem no menu USER. As variáveis correspondentes podem ser alteradas.

A trilha corrente.

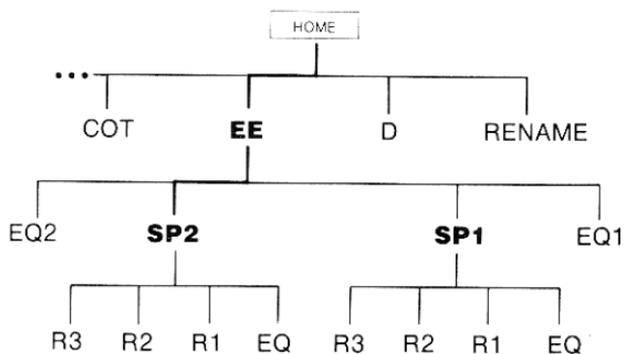
nome Um nome na trilha corrente. Estes nomes podem ser encontrados somente por processamento, RCL e PRVARs. As variáveis correspondentes não podem ser alteradas.



Diretório Corrente é HOME



Diretório Corrente é EE



Diretório Corrente é SP2

Eliminando um Diretório. Você pode eliminar um diretório *vazio*, da mesma forma que você faria com uma variável: mude para o diretório que contém o diretório a ser eliminado, coloque o nome do diretório na pilha e execute PURGE.

Se o diretório a ser eliminado contém variáveis ou subdiretórios, você precisa eliminar as variáveis ou subdiretórios antes que possa eliminar o diretório. Abaixo está um procedimento geral.

1. Mude para o diretório a ser eliminado.
2. Execute CLUSR para apagar o diretório.
3. Mude para o diretório-pai.
4. Elimine o diretório.

Se um erro NON-EMPTY DIRECTORY (diretório não vazio) ocorrer no passo dois, o diretório contém um subdiretório que não está vazio. Neste caso, você precisa executar os passos 1, 2 e 3 para apagar o subdiretório. Você pode então continuar com passos 2 a 4, para eliminar o diretório.

Movendo-se para Cima e para Baixo na Estrutura de Diretórios. O capítulo 28 inclui programas para mover-se para cima na estrutura de diretórios (mudando para um diretório pai) ou mover-se para baixo (mudando para um subdiretório). Ver "Mudando de Diretórios", à página 275.

Recursos para Recuperação

A HP-28S, automaticamente, salva cópias das linhas de comando, dos argumentos e da pilha. Estas cópias permitem que você se recupere de um erro — voltar ao ponto em que você estava antes do erro. Você pode, então, refazer o cálculo sem voltar ao começo. As cópias das linhas de comando e argumentos são também práticas para repetir-se cálculos.

Essas cópias podem consumir uma quantidade significativa de memória. Para cada um destes recursos de recuperação — linhas de comando, a pilha e argumentos — você pode decidir ativar ou desativar cada um deles. As operações para ativar ou desativar recursos de recuperação aparecem no menu MODE.

De um modo geral, é melhor deixar esses recursos ativados. Se é muito pouca a memória disponível e grandes objetos foram salvos pelos recursos de recuperação, seguramente você pode ganhar algum espaço de memória, desativando e reativando cada recurso, desta forma apagando os objetos armazenados.

Pouca Memória

A HP-28S contém 32 Kbytes de memória do usuário, dos quais cerca de 400 bytes são reservados para uso do sistema. Virtualmente, toda a operação da HP-28S requer a utilização de uma quantidade de memória — mesmo interpretar a linha de comando. A quantidade de memória utilizada por alguns dos comandos de Álgebra (COLCT, EXPAN, TAYLR) aumenta rapidamente à medida que seus argumentos se tornam mais complicados. Tente deixar pelo menos uns poucos milhares de bytes livres para uso dinâmico do sistema.

Você pode verificar a quantidade de memória disponível executando MEM, encontrado no menu MEMORY.

Uma vez que o sistema operacional da HP-28S compartilha memória com objetos do usuário, você pode preencher a memória de tal forma com objetos do usuário que a operação normal da calculadora se torne difícil ou impossível. A HP-28S fornece uma série de advertências sobre pouca memória, as quais são relacionadas abaixo em ordem de importância crescente.

Memória Insuficiente. Se não existe memória disponível para executar-se um comando, ocorre um erro de `INSUFFICIENT MEMORY`. Se LAST está ativado, os argumentos originais são restaurados na pilha. Se LAST está desativado, os argumentos são perdidos.

Sem Espaço para UNDO (desfazer). Se não existe memória disponível suficiente para salvar uma cópia da pilha, ocorre um erro `NO ROOM FOR UNDO` (*sem memória para desfazer*). O recurso UNDO é automaticamente desativado; para reativar UNDO, pressione `UNDO` no menu MODE.

Sem Espaço para ENTER. Se não houver memória disponível para processar a linha de comando, a calculadora apaga a linha de comando e apresenta no visor `NO ROOM TO ENTER`. Uma cópia da linha de comando não executada é salva na pilha de comando se a pilha de comando estiver ativada.

Se você tentar editar um objeto existente, utilizando EDIT ou VISIT e uma cópia da linha de comando não executada está salva na pilha de comando, elimine a cópia original do objeto, pressione **COMMAND** para recuperar a linha de comando contendo o objeto editado, e pressione **ENTER** para entrar a versão editada.

Pouca Memória! Se menos do que 128 bytes de memória livre permanecerem, LOW MEMORY (Pouca Memória) aparece e desaparece uma vez, na linha superior do visor. Esta mesma mensagem aparecerá instantaneamente a cada tecla pressionada até que haja memória adicional disponível. Apague da memória objetos que você não necessita, antes de continuar seus cálculos.

Sem Espaço para Mostrar Pilha. Algumas vezes é possível para a HP-28S completar todas as operações pendentes e não restar memória livre suficiente para apresentação normal da pilha no visor. Neste caso, a calculadora apresenta na linha superior do visor NO ROOM TO SHOW STACK (Sem Espaço para Mostrar Pilha). Estas linhas do visor, que normalmente iriam apresentar objetos da pilha, agora mostram estes objetos somente por tipo. Por exemplo, REAL NUMBER, ALGEBRAIC, etc.

A quantidade de memória necessária para apresentar um objeto pilha varia com o tipo de objeto — algébricos usualmente requerem mais memória. Apague um ou mais objetos da memória, ou armazene um objeto pilha como uma variável, de forma que ele não tenha que ser apresentado no visor.

Memória Cheia. No caso extremo de pouca memória, a memória é insuficiente para a calculadora fazer qualquer coisa — apresentar a pilha, mostrar rótulos de menu, construir uma linha de comando, etc. Nesta situação, você *precisa* apagar alguma quantidade de memória antes de continuar. Um procedimento especial OUT OF MEMORY (Memória Cheia) é ativado, e cria o visor:

```
Out of Memory
Purge?
Command Stack
YES NO
```

A calculadora irá, seqüencialmente, avisá-lo para apagar:

1. a pilha COMMAND (se ativada).
2. a pilha UNDO (se ativada).
3. LAST (últimos) argumentos (se ativado).
4. o menu personalizado (se houver).
5. a pilha.
6. cada variável no diretório HOME.

Para cada item que você deseja eliminar, pressione a tecla do menu **YES** ; para aquelas que você deseja manter, pressione **NO** .

Após pressionar **YES** pelo menos uma vez, você pode tentar terminar o procedimento **OUT OF MEMORY** pressionando **ON** . Se houver memória suficiente, a calculadora volta ao visor normal; se não, a calculadora emite um sinal audível e continua a seqüência de eliminação (Purge). Após completar um ciclo através das escolhas, o procedimento **OUT OF MEMORY** tenta retornar à operação normal. Se ainda não houver memória livre suficiente, o procedimento se reinicia com a seqüência de escolha para a eliminação.

Se você pressionar **YES** para um diretório vazio, ele é eliminado. Se você pressionar **YES** para um diretório que contém variáveis, as variáveis naquele diretório são apresentadas no visor.

Maximizando o Desempenho

De quando em quando a calculadora efetua uma "arrumação na casa" para melhor utilizar a memória. Geralmente, este processo somente é notado como pequenas pausas, por exemplo, durante o traçado de gráficos; entretanto, quando a memória está quase cheia e a pilha contém centenas de objetos, a calculadora pode responder muito lentamente, até mesmo a operações simples tal como a seleção de um menu.

Esta seção contém sugestões para maximizar a velocidade (reduzindo-se a quantidade de arrumação necessária na casa) e a memória disponível (através do aumento da efetividade da arrumação na casa).

Para Maximizar a Velocidade:

- não coloque mais do que algumas centenas de objetos na pilha.
- não deixe listas grandes (mais do que algumas centenas de objetos) na pilha; armazene-os na memória do usuário.

Para Maximizar a Memória Disponível:



Nota

O procedimento a seguir apaga a pilha, dados de recuperação (COMMAND, UNDO, LAST), o menu personalizado corrente (CUSTOM) e quaisquer programas suspensos.

1. elimine da memória do usuário variáveis e diretórios que você não quer.
2. armazene na memória do usuário quaisquer objetos da pilha que você deseja salvar.
3. execute uma parada de sistema (System Halt) (ON ▲).

O diretório corrente agora é HOME.

Para Minimizar Utilização de Memória para Cálculos com Arranjos: armazene arranjos em variáveis e refira-se a eles pelo nome; evite utilizá-los na pilha. Abaixo sugerimos uma estratégia abrangente para fazer-se isso.

1. Planeje antecipadamente de quantas variáveis você necessitará, incluindo os resultados intermediários.
2. Crie *pequenos* arranjos do tipo correto (reais ou complexos, vetores ou matrizes) e armazene-os em variáveis e, então, utilize RDM para ajustar seus tamanhos.
3. Execute os cálculos, utilizando os comandos para aritmética de armazenamento no menu STORE.
4. Para agir sobre elementos individuais, utilize GET, GETI, PUT, PUTI *com o nome da variável*, ou utilize sintaxe algébrica tal como 'A(5,6)' EVAL e 'B(3)' STO; não devolva o arranjo inteiro à pilha.

Menus

Cada operação, comando e função na HP-285 está disponível no teclado ou em um menu. Quando você seleciona um menu, seis *rótulos de menu* aparecem na linha inferior do visor. Estes rótulos constituem uma *linha de menu*, que indica as definições correntes das seis *teclas de menu*, na parte superior do teclado. (O menu Cursor é uma exceção; suas definições são impressas em branco acima das teclas do menu.)

Em adição às teclas que selecionam menus específicos (tais como  ARRAY ou  TRIG), as seguintes teclas controlam operações do menu.

| Tecla | Descrição |
|--|---|
|  | Liga/Desliga Menu do Cursor. Quando o menu do cursor não está ativado, seleciona o menu do cursor. Quando o menu do cursor está ativado, seleciona o menu anterior. |
|  | Último Menu Personalizado. Apresenta o último menu personalizado (custom) criado pelo comando MENU. |
|  | Próxima Linha do Menu. Apresenta a próxima linha de rótulos de menu. Se a última linha está no visor, apresenta primeira linha. |
|  | Linha Anterior do Menu. Apresenta linha anterior de rótulos de menu. Se a primeira linha está no visor, apresenta a última linha. |
|  | Chave do Menu. Liga e desliga a chave do menu. Quando a chave do menu está ligada, as posições prefixadas e não prefixadas são trocadas, para as três linhas superiores do teclado da esquerda (tecla de letras  até ). Quando a chave do menu está ligada, pressionar  seleciona o menu ARRAY e pressionar  escreve a letra A. |

Menus de Comandos

Os menus abaixo contêm teclas para operações pré-programadas da calculadora, a maioria das quais são comandos programáveis. Para uma breve descrição dos comandos em cada menu, ver apêndice D, "Diagramas dos Menus". O Manual de Referência cobre estes menus em ordem alfabética e descreve-os em detalhes.

A ação das teclas, nestes menus, depende do modo de entrada descrito à página 169.

| Menu | Descrição |
|--------------------|---|
| ALGEBRA | Comandos de álgebra. |
| ARRAY | Comandos de vetores e matrizes. |
| BINARY | Aritmética com inteiros, conversões de base, manipulações de bit. |
| COMPLEX | Comandos de números complexos. |
| LIST | Comandos de lista. |
| LOGS | Funções logarítmicas, exponenciais e hiperbólicas. |
| MEMORY | Memória do usuário, diretórios. |
| MODE | Visor, ângulos, modos de recuperação. |
| PLOT | Comandos de traçado de gráficos. |
| PRINT | Comandos de impressão. |
| PROGRAM BRANCH | Estruturas de desvio de programa. |
| PROGRAM CONTROL | Controle de programa, interrupção e operações de passo único. |
| PROGRAM TEST | Flags, testes lógicos. |
| REAL | Comandos de números reais. |
| SOLVE | Comandos de soluções simbólicas e numéricas, o Solver. |

| Menu | Descrição |
|-------------|---|
| STACK | Manipulação da pilha. |
| STAT | Comandos de estatística e probabilidade. |
| STORE | Aritmética nos registros. |
| STRING | Cadeias de caracteres. |
| TRIG | Funções trigonométricas, conversões de coordenadas e ângulos. |

Menus de Operações

Os seguintes menus oferecem operações não programáveis.

| Menu | Descrição |
|-------------|--|
| Cursor | Para edição da linha de comando. Descrito no capítulo 18. |
| CATALOG | Catálogo de comando, incluindo submenu USAGE. Descrito no capítulo 22. |
| UNITS | Unidades disponíveis para conversão. Descrito no capítulo 14. |

Menus de Variáveis

| Menu | Descrição |
|-------------|--|
| Solver | Armazena valores e calcula as variáveis na equação corrente. Aparência distinta (letras pretas contra rótulos de menu brancos) indica sua ação distinta. |
| USER | Apresenta variáveis e subdiretórios no diretório corrente. A ação das teclas depende do modo de entrada descrito à página 169. |

Menus Personalizados

O comando MENU pode criar um menu personalizado de uma lista de nomes e comandos. O menu personalizado (CUSTOM) pode ser similar ao menu Solver ou ao menu USER.

- Se o primeiro elemento na lista for o comando STO, seguido por uma seqüência de nomes, MENU cria um menu para *entrada personalizada* (Custom Input). Este menu parece e atua como menu Solver: pressionar uma tecla do menu faz com que o valor seja tomado da pilha e armazenado na variável correspondente. Para um exemplo, ver capítulo 27, "Programas Interativos"
- Se a lista contém uma seqüência de nomes e comandos (o primeiro elemento sendo diferente de STO), MENU cria um menu *personalizado do usuário* (Custom User). Este menu atua como um híbrido do menu USER e um menu de comando. Para um exemplo, ver "Mudando de Diretórios", à página 275.

Catálogo de Comandos

No capítulo 1, você utilizou o catálogo de comandos para verificar a grafia correta de alguns comandos e para verificar várias combinações de argumentos para a função + . Este capítulo revê as operações disponíveis no catálogo, incluindo o menu USAGE que mostra as combinações corretas de argumentos.

Pressionar  apresenta no visor o comando ABORT, que é o primeiro comando por ordem alfabética e o menu CATALOG.

| Tecla | Descrição |
|-------|--|
| NEXT | Avança o catálogo para o próximo comando. Se você pressionar e mantiver pressionada essa tecla, o catálogo avança, repetidamente, até que você solte a tecla. |
| PREV | Movê o catálogo de volta para o comando anterior. Se você pressionar e mantiver pressionada esta tecla, o catálogo se move para trás, repetidamente, até que você solte a tecla. |
| USE | Ativa a apresentação no visor do menu USAGE (ver abaixo), mostrando os argumentos da pilha utilizados pelo comando. |
| FETCH | Sai do catálogo e escreve o nome do comando na linha de comando. |
| QUIT | Sai do catálogo, deixando a linha de comando inalterada. |

Você pode sair do catálogo e apagar qualquer linha de comando corrente pressionando .

Encontrando um Comando

Você pode utilizar as teclas do teclado da esquerda, para posicionar o catálogo em um caractere específico.

- Ao pressionar uma tecla alfabética, no teclado da esquerda, o catálogo se posiciona no primeiro comando que começa com aquela letra. Se não existem comandos iniciados com aquela letra, o catálogo se move para o primeiro comando iniciado com a letra anterior.
- Pressionar uma tecla de um caractere não alfabético (tal como Σ) move o catálogo para o primeiro comando que se inicia com aquele caractere. Se não existem comandos começados com aquele caractere, o catálogo se move para +, o primeiro comando que se inicia com um caractere não alfabético.
- Pressionar $\boxed{\text{MENUS}}$ move o catálogo para \rightarrow STR, o último elemento do catálogo.

Verificando a Utilização de um Comando

Você pode verificar os tipos de argumentos corretos da pilha, para o comando correntemente apresentado no visor pelo catálogo. Pressionar **USE** ativa um segundo nível do catálogo denominado o menu **USAGE**, que mostra todas as combinações de argumentos para o comando. Se o comando aceita mais de uma combinação de argumentos, as teclas de menu abaixo aparecem. (Se o comando aceita somente uma combinação de argumentos, os rótulos **NEXT** e **PREV** não aparecem.)

| Tecla | Descrição |
|-------------|--|
| NEXT | Apresenta no visor a próxima combinação de argumentos. |
| PREV | Apresenta no visor a combinação anterior de argumentos. |
| QUIT | Volta para o catálogo principal, com o comando corrente apresentado no visor. Você pode então mover-se através do catálogo para outros comandos ou sair, pressionando QUIT novamente. |

Você pode sair tanto de **USAGE** como do catálogo principal e apagar qualquer linha de comando corrente, pressionando $\boxed{\text{ON}}$.

Processamento

Todas as operações da calculadora, desde simples cálculos no teclado até programas complicados, envolvem processamento. Alguns exemplos:

- quando você digita um ou mais objetos na linha de comando e pressiona `ENTER`, a linha de comando é traduzida em um programa, o qual é então processado.
- quando você pressiona uma tecla no menu USER em modo de entrada Imediato, o nome correspondente é processado.
- quando você executa diferenciação passo-a-passo, você pressiona `EVAL` para processar a expressão no nível 1.
- quando você utiliza o Solver para encontrar soluções numéricas, o procedimento armazenado na variável EQ é repetidamente processado.

É mais fácil entender as operações da calculadora em termos de *adiando o processamento* e *causando o processamento*. Embora o termo “adiando o processamento” seja novo, o processo é familiar: sempre que você entra um nome ou algébrico entre aspas, os delimitadores do objeto indicam que você deseja adiar o processamento do objeto, isto é, que você deseja que o objeto vá para a pilha.

O adiamento do processamento é a base para a programação de qualquer equipamento computacional, uma vez que de outra forma um programa seria executado tão logo você o escrevesse. A HP-28S estende o conceito de uma maneira uniforme para permitir operações simbólicas, isto é, você pode utilizar nomes e algébricos como dados para cálculos simbólicos. Por exemplo, você decide quando quer processar uma expressão. Você pode diferenciá-la, resolvê-la simbolicamente ou substituir suas variáveis, etc. Você pode também, é claro, calcular seu valor numérico.

Este capítulo descreve o que ocorre quando você processa os vários tipos de objetos. Como uma introdução geral, considere as seguintes *classes de objetos*:

- *objetos classe-dados*. Esta classe compreende números reais, números complexos, binários inteiros, cadeias, arranjos e listas. O “valor” de um objeto dados é exatamente o que ele contém.
- *objetos classe-nome*. Esta classe compreende nomes globais e nomes locais. O “valor” de um nome é geralmente o conteúdo de uma variável.
- *objetos classe-procedimento*. Esta classe compreende algébricos e programas. O “valor” de um procedimento é o resultado de quaisquer processos que ele define.

A grosso modo, estas classes definem o que ocorre quando você processa um objeto: ele se devolve, ou o conteúdo de uma variável, ou o resultado de um processo. Todavia, não é assim tão simples e maiores detalhes serão fornecidos abaixo para cada classe de objeto.

Objetos Classe-Dados

Esta é a classe mais simples de objetos. O processamento de qualquer objeto classe-dados devolve o mesmo objeto.

Note que listas são objetos dados para todas as finalidades, uma vez que eles podem conter qualquer tipo de objeto. Considere uma lista de nomes: os nomes estão protegidos de processamento pela lista e não podem ser processados até que sejam removidos da lista.

Objetos Classe-Nome

Geralmente, o “valor” de um nome é o conteúdo de uma variável. Processamento de nomes locais é simples e é descrito em primeiro lugar, seguido pelo processamento de nomes globais.

Processamento de Nomes Locais

Como foi descrito no capítulo 19, o uso de variáveis locais simplifica manipulações na pilha. A finalidade de variáveis locais é (1) remover o conteúdo da variável da pilha para que ele não atrapalhe e (2) devolver uma cópia do conteúdo da variável sempre que você a necessite. Conseqüentemente, o processamento de um nome local sempre devolve à pilha o conteúdo da variável local correspondente.

Processamento de Nomes Globais

Em geral, o processamento de um nome global causa o processamento do conteúdo das variáveis globais correspondentes. Em outras palavras, o processamento de um nome global tem o mesmo efeito que o processamento do objeto que ele representa.

Existem duas exceções a esta regra geral:

- se não existe nenhuma variável com o nome especificado, o nome é devolvido à pilha. Um nome indefinido, utilizado como uma variável, é denominado uma *variável formal*.
- se o conteúdo da variável especificada é um algébrico, ele *não é processado*. A calculadora evita processamento destes objetos, de modo que você possa continuar efetuando cálculos simbólicos. Se você quer executar o processamento, execute o comando EVAL com o algébrico no nível 1. (Para processar um algébrico repetidamente até que ele produza um resultado numérico, execute ↗NUM.

Se a variável contém um objeto classe-dados, o processamento do nome da variável é equivalente a, simplesmente, recuperar o conteúdo da variável. Entretanto, o processamento do nome de uma variável pode levar a uma longa cadeia de processamento. Por exemplo, se uma variável contém um nome, e este nome é o nome de uma segunda variável, e a segunda variável contém um nome e aquele nome é o nome de uma terceira variável, então o processamento do nome da primeira variável, em última análise, causa o processamento do conteúdo da terceira variável.



Nota

Não crie uma variável cujo valor é seu próprio nome, tal como uma variável denominada X que contém o nome 'X'. O processamento de tal variável causa um "loop" infinito. Para interromper um "loop" infinito, você precisa executar uma parada de sistema (ON ▲), que também apaga a pilha.

Da mesma forma, não crie variáveis que façam referência uma a outra de uma forma circular. O processamento de uma variável, incluída em uma definição circular, também causa um "loop" infinito.

Objetos Classe-Procedimento

Geralmente, o "valor" de um procedimento é o resultado de quaisquer processos que ele define. Programas são os objetos classe-procedimento mais gerais, assim eles são descritos primeiro, seguidos por algébricos.

Processamento de Programas

Um programa é uma seqüência de objetos e comandos. Este manual utiliza os termos "processar um programa" e "executar um programa" alternativamente. Em geral, o processamento de um programa toma o conteúdo do programa em ordem, colocando cada objeto na pilha e executando cada comando. Existem dois pontos adicionais a serem lembrados:

- nomes não colocados entre aspas são processados, enquanto nomes entre aspas vão para a pilha. Nomes são colocados entre aspas expressamente para adiar processamento, como discutido à página 57.
- estruturas de programa são executadas de acordo com suas próprias regras. Na parte 1, você escreveu diversas funções de usuário, as quais continham uma *estrutura de variável local*. Estruturas de programa são descritas no capítulo 26.

As regras para processar-se nomes e programas levam a uma das idéias fundamentais para programar-se a HP-28S. Para esta discussão, "programa" significa programa armazenado em uma variável e "nome de um programa" significa nome da variável que contém um programa.

A idéia fundamental é chamada *programação estruturada*. Isto significa que uma tarefa complicada é dividida em sub-tarefas, e um programa é escrito para cada sub-tarefa. O programa principal pode ser agora relativamente simples, refletindo a lógica global da tarefa. Ele pode executar cada sub-tarefa, simplesmente incluindo o nome sem aspas do programa para aquela sub-tarefa. Se uma sub-tarefa é executada mais de uma vez, o nome sem aspas pode ser incluído mais do que uma vez. Se outros programas principais usam a mesma sub-tarefa, eles podem executar esta sub-tarefa da mesma forma.

Programação estruturada é demonstrada em "Desenvolvendo e Agrupando Completamente" à página 253, "Apresentando no Visor um Inteiro Binário", à página 257, e "Mediana de Dados Estatísticos", à página 270.

Processamento de Algébricos

Cada algébrico é equivalente a um programa que contém somente nomes sem aspas e funções. O processamento de um algébrico produz o mesmo resultado que o processamento do programa correspondente; nomes sem aspas são processados e funções são executadas. Este tópico é também discutido em "Processamento de Objetos Algébricos", no Manual de Referência.

O resultado do processamento de um nome depende da existência de uma variável com aquele nome, como descrito em "Processamento de Nomes Globais" acima. Alguns exemplos.

- se um nome faz referência a uma função do usuário, você pode utilizar o nome da função do usuário da mesma forma que o de uma função pré-programada. O processamento do algébrico causa a execução da função do usuário. Os argumentos da função do usuário, entre parênteses e seguidos pelo nome da função do usuário, são parte do algébrico.

- se um nome faz referência a um programa que não utiliza argumentos da pilha e devolve exatamente um resultado, você pode utilizar o nome do programa para fazer referência (indiretamente) ao resultado. O processamento do algébrico causa a execução do programa, assim, na realidade, o nome do programa é substituído pelo resultado. Para um exemplo, ver “Estatísticas com Somatórios” à página 262.
- se um nome se refere a um segundo algébrico, o processamento do primeiro algébrico *não* causa o processamento do segundo algébrico. Ao invés disso, o segundo algébrico, efetivamente, substitui seu nome no primeiro algébrico.

Um caso especial entre as funções é a função “=”, que distingue equações de expressões. Dependendo do modo Resultado (Simbólico ou Numérico), a execução de = devolve uma equação ou um resultado numérico.

- Em modo de Resultado Simbólico, o processamento de uma equação produz uma nova equação. O novo membro esquerdo da expressão é o resultado do processamento do membro esquerdo da expressão original. O membro direito da expressão é o resultado do processamento do membro direito da expressão original.
- Em modo de Resultado Numérico, o processamento de uma equação produz a diferença numérica entre o membro esquerdo da expressão original (numericamente processado) e o membro esquerdo da expressão original (numericamente processado).

A próxima seção descreve os modos Resultado com mais detalhes.

Processamento de Funções

Quando uma função é processada, sua ação depende do modo Resultado corrente, o qual pode ser simbólico ou numérico. Esses modos são também descritos no próximo capítulo, “Modos”.

Modo de Resultado Simbólico. Este é o caso padrão, onde uma função aceita argumentos simbólicos e devolve resultados simbólicos. A ação de funções no modo de Resultado Simbólico é evidente quando você calcula com nomes e expressões, para criar expressões maiores.

Modo Resultado Numérico. Esta alternativa é utilizada em traçados de gráficos e pelo Solver. Sua finalidade é assegurar um resultado numérico da função. Neste modo, as funções processam repetidamente os argumentos simbólicos, aceitando somente argumentos numéricos e devolvendo resultados numéricos.

Você pode forçar o processamento de um objeto, até que ele devolva um resultado numérico, executando \rightarrow NUM (*para número*); no capítulo 5, você fez isso para devolver um valor numérico para π .



Nota

Em modo de Resultado Numérico, não processe uma variável cujo valor inclui seu próprio nome, tal como uma variável denominada X que contenha a expressão 'X + Y'. O processamento de tal variável causaria um "loop" infinito. Para interromper um "loop" infinito, você precisa executar uma parada de sistema ("system halt") (ON ▲), que também apaga a pilha.

Da mesma forma, não crie variáveis que façam referência uma a outra, numa definição circular. Processamento de uma variável incluída em uma definição circular também causa um "loop" infinito.

Modos

Você pode afetar os resultados de muitas operações através da seleção de um modo. Alguns modos, tal como o modo angular (Graus ou Radianos), podem ser selecionados, pressionando uma tecla de menu. O rótulo de menu para modo inclui um pequeno quadrado quando o modo está selecionado. Por exemplo, o rótulo de menu para modo angular Radianos aparece como `RAD▣` quando aquele modo é selecionado.

A maioria dos modos, tal como o modo Beeper (sinal audível), pode ser selecionada ativando ou desativando um “flag” (indicador) do usuário, utilizando os comandos `SF` (*set flag = ativa indicador*) e `CF` (*clear flag = desativa indicador*). Por exemplo, “flag” 51 controla o modo Beeper, assim você pode desligar o sinal audível (Beeper), executando `51 SF`.

Este capítulo descreve como os modos afetam a operação da calculadora e relaciona os rótulos de menu e “flags” associados. Os anúncios que aparecem, quando um modo é selecionado, também são mostrados. Para cada modo, a escolha apresentada em primeiro lugar é a seleção padrão, ativada logo após Memória Apagada (Memory Lost).

Modos Gerais

Estes modos afetam computações e o sinal audível.

Modo Angular

Este modo determina se números reais representam medidas angulares em graus ou em radianos. Isto afeta os argumentos de funções trigonométricas e os resultados de funções trigonométricas inversas.

Modo Graus (DEG , Flag 60 Desativado). Números reais representam medida angular em graus.

Modo Radianos (RAD , Flag 60 Ativado, (2π)). Números reais representam medida angular em radianos.

Modo Beeper (sinal audível)

Este modo controla se a calculadora emite sons quando um erro ocorre ou BEEP é executado.

Beeper Ligado (Flag 51 Desativado). A calculadora emite sons.

Beeper Desligado (Flag 51 Ativado). A calculadora não emite sons.

Valor Principal (Principal Value)

Uma solução devolvida por ISOL ou QUAD , geralmente, requer sinais arbitrários (+1 ou -1) e inteiros (0,1,2,...), para representar todas as soluções possíveis. Este modo determina se sinais e inteiros arbitrários são incluídos em soluções geradas por ISOL ou QUAD.

Valor Principal Desligado (Flag 34 Desativado). Soluções devolvidas por ISOL e QUAD incluem variáveis s_1, s_2, \dots para sinais arbitrários e n_1, n_2, \dots , para inteiros arbitrários.

Valor Principal Ligado (Flag 34 Ativado). ISOL e QUAD consideram sinais arbitrários como sendo +1 e inteiros arbitrários como sendo zero.

Modo Constantes

Este modo determina se o processamento de uma constante simbólica (E, I, MINR, MAXR, ou π) devolve seu valor numérico. Em modo Resultados Numéricos (Flag 36 desativado), o processamento de uma constante simbólica devolve seu valor numérico, independentemente do modo Constantes.

Constantes Simbólicas (Flag 35 Ativado). O processamento de uma constante simbólica devolve sua forma simbólica.

Constantes Numéricas (Flag 35 Desativado). O processamento de uma constante simbólica devolve seu valor numérico.

Modo Resultados

O modo Resultado corrente afeta o resultado do processamento de uma função, quando seus argumentos são simbólicos.

Resultados Simbólicos (Flag 36 Ativado). Dados os argumentos simbólicos, as funções devolvem resultados simbólicos.

Resultados Numéricos (Flag 36 Desativado). As funções sempre devolvem resultados numéricos. Para isso, as funções processam argumentos simbólicos repetidamente para determinar seus valores numéricos. O processamento de uma constante simbólica devolve seu valor numérico independentemente do modo Constantes.

Modos de Entrada e Apresentação no Visor

Estes modos afetam como os objetos são entrados e apresentados no visor.

Modo de Entrada

O modo de entrada corrente afeta o resultado quando você pressiona um comando, função, ou tecla do menu USER. O modo de entrada automaticamente muda quando você pressiona , , ou ; você também pode mudá-lo manualmente pressionando . A aparência do cursor indica o modo de entrada corrente. Para detalhes, ver capítulo 18, "A Linha de Comando".

Entrada Imediata (Cursor Tem Apenas o Contorno). A linha de comando é executada quando você pressiona um comando, uma função ou uma tecla do menu USER.

Entrada Algébrica (Cursor Parcialmente Preenchido). A linha de comando é executada quando você pressiona uma tecla de comando.

Entrada Alfanumérica (Cursor Totalmente Preenchido, α). A linha de comando é executada quando você pressiona .

Modo de Substituição ou de Inserção

Pressionar  no menu do cursor muda entre modo Substituição e Inserção. A aparência do cursor indica modo de Substituição ou Inserção.

Modo de Substituição (Cursor Quadrado). Novos caracteres substituem os caracteres existentes.

Modo de Inserção (Cursor Em Seta). Novos caracteres são inseridos entre os caracteres existentes.

Letras Maiúsculas ou Minúsculas

Pressionar  troca o modo Letra Maiúscula por Letra Minúscula, ou vice-versa.

Modo Letra Maiúscula. Pressionar uma tecla de letra, escreve a letra em maiúsculo na linha de comando.

Modo Letra Minúscula. Pressionar uma tecla de letra, escreve a letra em minúsculo na linha de comando.

Nível 1 do Visor

Muitos objetos são muito grandes para serem apresentados em uma única linha do visor. Você pode preferir utilizar mais do que uma linha para apresentar o objeto no nível 1, se necessário, ou utilizar somente uma linha, independentemente do tamanho do objeto. Esta escolha afeta a saída impressa em modo de monitoração de impressão.

ML Ligado ( , Flag 45 Ativado). Objetos no nível 1 são apresentados no visor em mais de uma linha se necessário.

ML Desligado ( , Flag 45 Desativado). Objetos no nível 1 são apresentados no visor em apenas uma linha.

Modo Ponto Decimal

A vírgula e o ponto dividem os papéis de marca de raiz (para distinguir a parte inteira de um número, da sua parte fracionária) e separador (para distinguir objetos na linha de comando; o espaço é sempre um separador). Você pode atribuir estes papéis para a vírgula ou o ponto em qualquer ordem.

RDX, Desligado (RDX, , Flag 48 Desativado). O ponto é a marca de raiz (ponto decimal) e a vírgula é um separador.

RDX, Ligado (RDX, , Flag 48 Ativado). A vírgula é a marca de raiz e o ponto é um separador.

Formatos de Números

Estes modos determinam o número de casas decimais apresentadas no visor para números reais. Os comandos FIX, SCI e ENG requerem como argumento um número real n . O modo corrente do formato de número também afeta o comando RND (*arredondar*).

Formato STD (STD). Números reais são apresentados com um ponto decimal ou um expoente somente se necessário.

Formato FIX (FIX). Números reais são apresentados com n casas decimais. Um expoente é apresentado somente se necessário.

Formato SCI (SCI). Números reais são apresentados como uma mantissa, que é menor do que 10 e contém n casas decimais e um expoente.

Formato ENG (ENG). Números reais são apresentados como uma mantissa, que contém $n + 1$ dígitos e um expoente que é um múltiplo de 3.

Base Inteira

Você pode escolher a base utilizada para entrada e apresentação de inteiros binários. A escolha da base não afeta a estrutura interna de inteiros binários, os quais são sempre tratados como uma seqüência de bits.

Base DEC (DEC▪). Inteiros binários entrados sem marcadores de base são interpretados na base 10. Todos os inteiros binários são apresentados na base 10 e mostram um “d” como marcador de base.

Base HEX (HEX▪). Inteiros binários entrados sem marcadores de base são interpretados na base 16. Todos os inteiros binários são apresentados na base 16 e mostram um “h” como marcador de base.

Base OCT (OCT▪). Inteiros binários entrados sem marcadores de base são interpretados na base 8. Todos os inteiros binários são apresentados na base 8 e mostram um “o” como marcador de base.

Base BIN (BIN▪). Inteiros binários entrados sem marcadores de base são interpretados na base 2. Todos os inteiros binários são apresentados na base 2 e mostram um “b” como marcador de base.

Tamanho de Palavra de Inteiros Binários

O tamanho da palavra corrente pode variar de 1 a 64 bits. Ele controla como os inteiros binários são apresentados no visor; além disso, inteiros binários são truncados no tamanho da palavra corrente, quando utilizados como argumentos ou devolvidos como resultados. Para definir o tamanho de palavra como *n*, execute *n STWS* (*store wordsize = armazenar tamanho de palavra*).

Modos de Recuperação

Os modos de recuperação determinam se são efetuadas cópias das linhas de comando, da pilha e dos argumentos para os comandos. Estas cópias podem auxiliá-lo na recuperação, se você cometer um erro.

Modo CMD

Este modo determina se uma cópia da linha de comando é salva quando você pressiona (ou uma tecla que execute ENTER).

CMD Ligado (CMD▪). Linhas de comando são salvas para procedimento de recuperação por .

CMD Desligado (`CMD`). Linhas de comando não são salvas.

Modo UNDO

Este modo determina se uma cópia da pilha é salva quando você pressiona `ENTER` (ou uma tecla que execute ENTER).

UNDO Ligado (`UNDO`). A pilha é salva para procedimento de recuperação por `UNDO`.

UNDO Desligado (`UNDO`). A pilha não é salva.

Modo LAST

Este modo determina se cópias dos argumentos são salvas quando um comando é executado.

LAST Ligado (`LAST` , **Flag 31 Desativado).** Argumentos são salvos para procedimento de recuperação por LAST ou em caso de erro.

LAST Desligado (`LAST` , **Flag 31 Ativado).** Argumentos não são salvos. Se um erro ocorrer, os argumentos do último comando não são devolvidos para a pilha.

Exceções Matemáticas

Certos erros, que podem ocorrer durante cálculos comuns com números reais, são chamados *exceções matemáticas*. Uma exceção pode agir como um erro comum e interromper o cálculo, ou ela pode fornecer um resultado padrão e permitir que o cálculo prossiga.

Ação Sobre Resultado Infinito

Uma exceção de Resultado Infinito ocorre quando um cálculo devolve um resultado infinito. Exemplos incluem o processamento de `'LN(0)'`, `'TAN(90)'`, (em modo angular Graus) ou `'X/0'`.

Erro de Resultado Infinito Ligado (Flag 59 Ativado). Exceções de Resultado Infinito são erros. Elas devolvem a mensagem de erro `INFINIT RESULT` (Resultado Infinito).

Impressão em Modo Monitoração

Você pode automaticamente imprimir um registro de seus cálculos.

Monitoração de Impressão Desligada (`TRAC` , Flag 32 Desativado). Não há impressão automática.

Monitoração de Impressão Ligada (`TRAC` , Flag 32 Ativado). Cada vez que a linha de comando é executada, a calculadora imprime o conteúdo da linha de comando, a operação que causou a execução e o resultado no nível 1.

Modo de Mudança Automática de Linha (Auto CR)

Geralmente, você deseja enviar dados para a impressora e imprimi-los com um único comando. Em outros casos, tal como quando imprimindo gráficos, você deseja acumular dados na impressora sem imprimir. Este modo determina se os comandos de impressão automaticamente causam impressão.

Auto CR (Flag 33 Desativado). Comandos de impressão enviam CR (Carriage Right) ao fim da transmissão, causando impressão dos dados.

Acumular Dados de Impressão (Flag 33 Ativado). Comando de impressão envia dados sem CR, fazendo com que os dados se acumulem no "buffer" da impressora.

Velocidade de Impressão

A calculadora não pode perceber quando a impressora está pronta para receber mais dados, assim ela calcula a taxa na qual dados podem ser transmitidos com segurança. Este modo determina se o cálculo é feito para uma impressora operada à bateria ou para uma que está conectada à rede elétrica.

Velocidade Normal de Impressão (Flag 52 Desativado). A calculadora envia dados a uma taxa adequada para a impressora operada à bateria.

Velocidade de Impressão Rápida (Flag 52 Ativado). A calculadora envia dados a uma taxa adequada para a impressora conectada à rede elétrica.

Espaçamento de Impressão

Este modo determina se linhas em branco são impressas automaticamente.

Impressão com Espaçamento Simples (Flag 47 Desativado). Não são impressas linhas em branco automaticamente.

Impressão com Espaçamento Duplo (Flag 47 Ativado). Uma linha em branco é automaticamente impressa entre cada duas linhas de texto.

Operações de Sistema

Este capítulo descreve combinações especiais de teclas que interrompem operação normal da HP-28S. Estas operações de sistema incluem a impressão do visor, ajuste do contraste do visor, interrupção de programas, inicialização da memória e execução de testes diagnósticos.

Todas as operações de sistema começam com a tecla **ON**. Você pode cancelar qualquer operação de sistema, pressionando **DEL** antes que você solte **ON**.

A tabela abaixo mostra as teclas para operações de sistema, seguidas cada uma por sua descrição.

Operações de Sistema

| Nome | Seqüência de Teclas |
|-------------------------------------|--|
| Impressão do Visor | ON L |
| Controle de Contraste | ON + ou ON - |
| Atenção | ON |
| Parada de Sistema | ON ▲ |
| Inicialização da Memória | ON INS ▶ |
| Teste Contínuo | ON ◀ |
| Teste do Teclado | ON NEXT |
| Cancelamento de Operação de Sistema | ON DEL |

Imprimindo o Visor

Para imprimir a imagem corrente do visor:

1. pressione e mantenha pressionada .
2. pressione (a tecla com "PRINT" acima dela).
3. solte .

Controle de Contraste

Para mudar o contraste do visor:

1. pressione e mantenha pressionada .
2. pressione para aumentar o contraste ou pressione para diminuir o contraste. Enquanto você mantiver pressionado , você pode pressionar ou repetida ou continuamente para encontrar o melhor contraste.
3. solte .

Operações de Apagar

Existem três operações de apagar, dadas abaixo, em ordem de severidade crescente.

Atenção

Para retornar à apresentação normal da pilha no visor, execute Atenção pressionando . Em alguns casos, pode ser necessário pressionar duas vezes. Atenção tem os seguintes efeitos:

- apaga a linha de comando.
- cancela todos os comandos ou execução de procedimentos.
- sai de operações especiais tais como FORM, PLOT e catálogos.
- reinicia operação normal do teclado.

Parada de Sistema

Para interromper um programa que não responde a **ON**, execute Parada de Sistema como segue:

1. pressione e mantenha pressionada **ON**.
2. pressione **▲**.
3. solte **ON**.

Parada de Sistema tem os seguintes efeitos:

- todos os efeitos de Atenção.
- apaga todos os programas suspensos e variáveis locais.
- apaga a pilha.
- apaga itens salvos para procedimento de recuperação (CMD, UNDO, LAST).
- apaga o menu personalizado.
- seleciona HOME como o diretório corrente.
- ativa o menu do cursor.
- seleciona modo de monitoração desligado.

Inicialização da Memória

Para inicializar toda a memória:

1. pressione e mantenha pressionada **ON**.
2. pressione e mantenha pressionada **INS** e **▶**.
3. solte **INS** e **▶**.
4. solte **ON**.

Inicializar memória tem os seguintes efeitos:

- todos os efeitos de Atenção e Parada de Sistema.
- elimina todas as variáveis de usuário e diretórios.
- inicializa todos os “flags” do usuário a seus valores padrão.
- emite sinal audível e apresenta no visor MEMORY LOST (Memória Apagada) na linha 1.

Operações de Teste

Existem duas operações de sistema para testar-se a calculadora. A primeira é um teste contínuo da parte eletrônica, que é executado sem assistência. A segunda é um teste de teclado que requer que você pressione todas as teclas numa seqüência especificada. Ambos os testes executam uma Parada de Sistema.

Teste Contínuo

Para executar o teste contínuo:

1. inicie o teste.
 - a. Pressione e mantenha pressionada .
 - b. Pressione .
 - c. Solte .
2. o visor mostra linhas horizontais e verticais, um visor em branco, um padrão aleatório e então apresenta brevemente o resultado do teste antes de reiniciá-lo novamente.
 - A mensagem `OK-288` indica que a calculadora passou no teste.
 - Uma mensagem tal como `1 FAIL` indica que a calculadora não passou no teste. O número indica a natureza da falha.

Se você interromper o teste pressionando uma tecla, o teste devolve uma mensagem de falha porque ele não espera nenhuma tecla. *Tal mensagem de falha não indica um problema com a calculadora.*

3. saia do teste, executando uma Parada de Sistema.
 - a. Pressione e mantenha pressionada .
 - b. Pressione .
 - c. Solte .

Teste do Teclado

Para executar o teste do teclado:

1. inicie o teste.
 - a. Pressione e mantenha pressionada **ON**.
 - b. Pressione **NEXT**.
 - c. Solte **ON**.
2. a calculadora apresenta a mensagem **KEYBOARD TEST**.
 - a. Teste a primeira linha do teclado da esquerda, pressionando **A** **B** **C** **D** **E** **F**.
 - b. Teste da segunda à sexta linha do teclado da esquerda da mesma forma.
 - c. Teste a primeira linha do teclado da direita, pressionando **INS** **DEL** **▲** **▼** **◀** **▶**.
 - d. Teste da segunda à sétima linha do teclado da direita da mesma forma. (Pressione a tecla **ON** na ordem correta — ela não interromperá o teste.)
3. se você pressionou as teclas na ordem correta e o teclado estiver funcionando adequadamente, a calculadora apresenta no visor **OK-288**. Uma mensagem tal como **1 FAIL** indica que você não seguiu a ordem correta ou a calculadora não passou no teste. O número indica a natureza da falha.
4. pressione **ON**

Parte 3

Programação

| | |
|-------------------|------------------------------------|
| Página 222 | 26: Estruturas de Programa |
| 234 | 27: Programas Interativos |
| 240 | 28: Exemplos de Programação |

Estruturas de Programa

Muitos programas são equivalentes a uma série de cálculos de teclado de execução imediata. Os objetos vão para a pilha e comandos são executados, produzindo o resultado desejado. Estes programas são simplesmente um registro dos objetos e comandos, escritos na mesma ordem que você os executaria no teclado. Entretanto, existem recursos que você pode utilizar em programas que vão além de simples seqüências de teclas.

Por exemplo, na parte 1 você escreveu programas que criaram variáveis locais. O comando especial `→`, seguido por um ou mais nomes, seguido por um procedimento, é denominado uma *estrutura de variável local*. Você não pode executar o comando `→` do teclado; ele precisa aparecer no mesmo programa que os nomes e o procedimento que constituem a estrutura completa do programa.

Este capítulo revê, primeiramente, a estrutura de variável local. Em seguida, descreve estruturas de programa adicionais que executam testes e modificam a execução do programa baseados no resultado. Todos os comandos para estas estruturas de programa aparecem no menu PROGRAM BRANCH. Não deixe de ler o primeiro exemplo sob o título “Estruturas Condicionais”, à página 223, a qual introduz conceitos utilizados no restante do capítulo.

Estrutura de Variável Local

Na parte 1, você escreveu diversas funções do usuário, que foram a aplicação mais importante da estrutura de variável local. Existem dois requisitos para funções do usuário. Elas precisam:

- explicitamente indicar seus argumentos.
- devolver exatamente um resultado.

Por exemplo, a função do usuário COT (do capítulo 5) foi escrita:

```
« + X * INV(TAN(X)) * »
```

Aqui a estrutura de variável local armazena um argumento em uma variável X (satisfazendo o primeiro requisito) e processa a expressão 'INV(TAN(X))' satisfazendo o segundo requisito). A função do usuário O→G (do capítulo 14) incluía um programa ao invés de uma expressão, mas uma vez que o programa devolvia exatamente um resultado, O→G também satisfazia o segundo requisito.

Esses requisitos se aplicam somente a funções do usuário. Mais genericamente, variáveis locais são utilizadas como um substituto para manipulações da pilha. O seguinte exemplo devolve a soma e a diferença de dois números. Uma vez que ele devolve dois resultados, não pode ser uma função do usuário.

```
« + X Y « X Y + X Y - » »
```

Para mais exemplos, ver programas no capítulo 28. Eles utilizam estruturas de variável local para evitar manipulações da pilha com maior frequência do que para criar funções do usuário.

Estruturas Condicionais

Estruturas condicionais permitem a um programa testar uma condição especificada e tomar uma decisão baseada no resultado do teste. Esta seção primeiro dá um exemplo de uma estrutura condicional e utiliza este exemplo para discutir estruturas de programa em geral; finalmente descreve outros tipos de estruturas condicionais.

Suponha que você esteja escrevendo um programa que utiliza a variável x , e você quer calcular $(\sin x)/x$. Surge um problema porque o quociente é indefinido quando $x = 0$. O seguinte exemplo devolve $(\sin x)/x$ se $x \neq 0$, ou devolve 1 se $x = 0$.

```
IF X 0 ≠ THEN X SIN X / ELSE 1 END
```

Eis como esta estrutura funciona quando você executa o programa:

1. o comando IF simplesmente marca o início da estrutura. Ele pode estar em qualquer lugar antes do comando THEN.
2. X é processado.
3. o número 0 vai para a pilha.
4. o comando \neq toma o valor de X e o número 0 como argumentos.
 - Se os argumentos são “não iguais”, \neq devolve 1.
 - Se os argumentos *não* são “não iguais”, \neq devolve 0.
5. o comando THEN toma 1 ou 0 como seu argumento.
 - Se o argumento é 1, THEN processa o programa até o ELSE (nominalmente X SIN X /).
 - Se o argumento é 0, THEN processa o programa de ELSE até END (nominalmente 1).
6. execução do programa continua após o comando END.

Antes de prosseguir, abordando pontos específicos sobre estruturas condicionais, apresentamos algumas informações de caráter geral sobre estruturas de programas.

Comandos de Estrutura de Programa. Os comandos IF, THEN, ELSE e END (SE, ENTÃO, CASO CONTRÁRIO e FIM) são exemplos de *comandos de estrutura de programa*. A ordem e significado desses comandos são similares ao seu uso em linguagem corrente. Você não pode utilizar comandos de estrutura de programa com tanta flexibilidade como outros comandos; eles funcionam apenas nas combinações descritas neste capítulo.

Funções de Teste e Comandos. A função \neq é chamada uma *função de teste*. Dados os dois números, \neq devolve 1 ou 0, indicando se o teste é verdadeiro ou falso. Outras funções de teste são \langle , \leq , \geq , \rangle , e $=$. (Lembre-se de que $=$ é utilizado para equações, não para teste de igualdade.) Dados argumentos simbólicos, funções de teste devolvem um resultado simbólico.

Existem também comandos de teste que sempre devolvem 1 ou 0. Por exemplo, o comando de teste SAME (mesmo) é similar a $==$, mas ele simplesmente testa se os dois objetos são idênticos. Comandos de testes adicionais são disponíveis para trabalhar-se com “flags” (descritos a seguir). Para mais informações sobre funções de teste e comandos, ver “PROGRAM TEST” no Manual de Referência.

“Flags”. Os números 1 e 0 que são devolvidos pelos comandos de teste são chamados “*flags*” da pilha. Como eles representam a verdade ou falsidade de um teste, 1 é chamado um “*flag*” verdadeiro e 0 é chamado um “*flag*” falso.

O termo “*flag*” também se refere aos “*flags*” do usuário incorporados à calculadora. Eles são numerados de 1 a 64; “*flags*” de 31 a 64 têm significados específicos para a calculadora, enquanto que os “*flags*” 1 a 30 podem representar qualquer distinção verdadeira/falsa que você queira. Você pode armazenar um “*flag*” da pilha em um “*flag*” do usuário, uma vez que ambos representam um valor-verdade (1 ou 0). Por exemplo, a seqüência

```
IF A B < THEN 12 SF ELSE 12 CF END
```

ativa “*flag*” 12 se $A < B$, ou desativa “*flag*” 12 se $A \geq B$. Posteriormente você pode testar se o “*flag*” 12 está ativado através da seqüência

```
IF 12 FS? THEN ...
```

a qual devolve o mesmo valor verdadeiro que o teste original $A < B$. A vantagem desta técnica é que o valor-verdade (1 ou 0) do teste original é preservado, mesmo que os valores de A e B tenham mudado. Os comandos para mudar-se e testar “*flags*” do usuário aparecem em “PROGRAM TEST”, no Manual de Referência. Para o restante deste capítulo, “*flag*” se refere a um “*flag*” da pilha.

Condições. Os objetos e comandos entre dois comandos de estrutura de programa são denominados uma *condição*. Cada condição é manipulada como uma entidade única pela estrutura de programa. Uma condição é rotulada pelo seu papel lógico ou pelo comando que a precede. No primeiro exemplo:

- a condição entre IF e THEN ($X \neq$) é chamada de *condição de teste* ou *condição IF*.
- a condição entre THEN e ELSE ($X \text{ SIN } X \nearrow$) é chamada *condição verdadeira* ou *condição THEN*.
- a condição entre ELSE e END (1) é chamada *condição falsa* ou *condição ELSE*.

As condições, no exemplo, representam simples cálculos numéricos, mas você pode incluir qualquer seqüência de objetos e comandos. Com efeito, uma condição é como um subprograma dentro do programa. Se você escrever um programa separado que contém a condição e armazenar este programa em uma variável, você pode utilizar o nome da variável como a condição inteira. Neste caso, uma estrutura com aparência simples tal como

```
IF A THEN B ELSE C END
```

pode representar um complicado processo de decisão, com dois possíveis resultados também complicados, dependendo dos conteúdos, de A, B e C.

IF ... THEN ... ELSE ... END

Utilizando a terminologia que acabamos de definir, o processamento desta estrutura condicional pode ser descrito como segue: a condição IF é processada e devolve um "flag". Se o "flag" é verdadeiro, a condição THEN é processada; se o "flag" é falso, a condição ELSE é processada.

Para outro exemplo dessa estrutura, ver "FIB2 (Números de Fibonacci, Versão "Loop")", à página 248.

IFTE (Função If-Then-Else-End)

O primeiro exemplo, neste capítulo, pode ser escrito em sintaxe algébrica, utilizando a função IFTE:

```
' IFTE (X≠0, SIN (X)/X, 1)'
```

Esta forma é prática para cálculo simbólico. Se você executar a versão da estrutura de programa enquanto X está indefinido, esta forma algébrica é o resultado. Os argumentos para IFTE precisam ser representáveis em sintaxe algébrica; para incluir comando RPN na expressão condicional, você precisa utilizar a forma de estrutura de programa.

A função IFTE é utilizada em "FIB1 (Números de Fibonacci, Versão Recursiva)", à página 247.

IF ... THEN ... END

Se uma condição ELSE não é necessária, isto é, se as escolhas são para fazer-se alguma coisa ou não se fazer nada, você pode omitir ELSE da estrutura de programa. O exemplo a seguir assegura que o objeto no nível 1 é maior que o objeto no nível 2, alternando-os se necessário.

```
IF DUP2 < THEN SWAP END
```

Note o uso de DUP2 para fazer cópias dos objetos. As cópias são então consumidas pela comparação <. Para outro exemplo de IF ... THEN ... END, ver "SORT (Classificar uma Lista)", à página 270.

IFT (Comando If-Then-End)

Você poderia escrever o exemplo anterior utilizando o comando IFT, ao invés da estrutura de programa:

```
DUP2 < « SWAP » IFT
```

A seqüência DUP2 < deixa um "flag" na pilha, o programa « SWAP » vai para a pilha e o comando IFT toma o "flag" e o programa como argumentos. Se o "flag" é verdadeiro, IFT processa o programa; se o "flag" é falso, IFT elimina o programa. O resultado é idêntico à forma de estrutura de programa.

Armadilhas de Erro ("Error Traps")

Em alguns casos, você pode prever que um erro pode ocorrer durante a execução do programa. Normalmente, um erro cancela a execução do programa; mas se você *preparar uma armadilha* para o comando com potencial de erro, colocando-o em uma estrutura especial de programa, a execução pode continuar quando o erro ocorre.

Lembre-se do problema com $(\sin x)/x$ — ele causa erro de Resultado Infinito quando $x = 0$. Outro método para definir $(\sin 0)/0 = 1$ seria:

```
IFERR X SIN X / THEN DROP2 1 END
```

Isto significa: processe a condição IFERR (X SIN X /). Se um erro ocorrer, processe a condição THEN (DROP2 1).

Este exemplo inclui o comando DROP2 para eliminar os dois zeros que causaram o erro. Note que isto pressupõe que LAST está ativado. Se LAST estiver desativado, os zeros não estarão presentes e o comando DROP2 é inadequado. Certifique-se de levar em conta o estado de LAST, quando utilizar armadilhas para erros.

Outro exemplo de IFERR ... THEN ... END aparece em "BDISP (Apresentação de Binário no Visor)", à página 259. Além disso, você pode incluir uma condição ELSE para ser processada somente se um erro não ocorrer, utilizando a forma

```
IFERR ... THEN ... ELSE ... END
```

Estruturas de "Loop" Definido

Estruturas de "loop" contêm uma *condição de "loop"* que é repetidamente processada. Em uma *estrutura de "loop" definido*, o programa especifica antecipadamente quantas vezes processar a condição de "loop". Outro tipo de estrutura de programa, chamada uma *estrutura de "loop" indefinido*, utiliza uma condição de teste para determinar se repete o processamento da condição de "loop". Esta seção descreve estruturas de "loop" definido. Estruturas de "loop" indefinido são descritas à página 231.

INICIE ... PRÓXIMO (START ... NEXT)

O seguinte exemplo emite um som quatro vezes.

```
1 4 START 440 . 1 BEEP NEXT
```

Esta estrutura opera como segue:

1. o comando START toma os valores 1 e 4 da pilha e cria um *contador*. O contador será utilizado para monitorar quantas vezes o "loop" vai ser repetido. O valor 1 especifica o *valor inicial* do contador e o valor 4 especifica seu *valor final*.

2. a condição de "loop" `440 . 1 BEEP` é executada.
3. o comando `NEXT` adiciona 1 ao contador.
4. o valor corrente do contador é comparado ao seu valor final.
 - Se o valor corrente do contador não excede seu valor final (2), (3) e (4) são repetidos.
 - Se o valor corrente do contador excede seu valor final, a estrutura de "loop" definido é completada e a execução do programa continua após o comando `NEXT`.

Neste exemplo, os passos (2), (3) e (4) são repetidos quatro vezes. O contador do "loop" é primeiro incrementado de 1 para 2, depois para 3, depois para 4, depois para 5. Neste ponto, ele excede o valor final 4, assim a estrutura de "loop" definido termina. Note que o passo (1) é executado antes que quaisquer testes sejam feitos, assim a condição de "loop" é sempre processada pelo menos uma vez. Para outro exemplo de `START ... NEXT`, ver "FIB2 (Números de Fibonacci, Versão "Loop")", à página 248.

PARA contador ... PRÓXIMO (FOR counter ... NEXT)

Em muitos casos, é prático utilizar o valor corrente do contador como uma variável na condição do "loop". Para fazer isso, no lugar de `START` coloque `FOR nome`. O contador se torna uma variável local com o nome especificado. Como anteriormente, este manual segue a convenção de escrever nomes locais em letras minúsculas, para ajudá-lo a distingui-los de nomes globais. O exemplo a seguir coloca o quadrado dos primeiros cinco inteiros na pilha.

```
1 5 FOR X X SQ NEXT
```

A seqüência `FOR X` é executada somente uma vez. A seqüência `X SQ` é a condição do "loop", a qual é executada repetidamente.

Os exemplos até aqui especificaram um valor inicial do contador de 1, mas qualquer inteiro é aceitável. Uma vez que você está utilizando o contador como uma variável, defina os valores inicial e final do contador como sendo os valores inicial e final desejados da variável. O exemplo a seguir coloca o quadrado do terceiro até o nono inteiro na pilha.

```
3 9 FOR X X SQ NEXT
```

Para outro exemplo, ver “BDISP (Apresentação de Binário no Visor)”, à página 259.

... increment STEP (... incremento Passo)

O comando NEXT sempre incrementa o contador de 1. Para especificar um incremento diferente, substitua NEXT por n STEP, onde n é o incremento desejado. STEP é comumente utilizado após FOR *contador*, como demonstrado nos exemplos abaixo, mas também pode ser utilizado após START. O exemplo a seguir coloca os primeiros quadrados *ímpares* de 1^2 até 5^2 na pilha.

```
1 5 FOR X X SQ 2 STEP
```

A condição de “loop” $X \text{ SQ } 2$ é executada três vezes. O comando STEP *primei-*ro incrementa o contador de 1 até 3, então para 5, então para 7. Neste ponto, o valor corrente do contador excede o valor final 5, assim a estrutura de “loop” definido termina.

Os exemplos até agora utilizaram valores *crecipientes* do contador. Para valores *decrecipientes* do contador, você pode especificar um incremento negativo. O exemplo a seguir coloca os quadrados inteiros *ímpares* de 5^2 até 1^2 na pilha.

```
5 1 FOR X X SQ -2 STEP
```

A seqüência -2 STEP decrementa o contador de 5 a 3, então a 1 e então a -1. Neste ponto, o valor corrente do contador é menor do que o valor final 1, assim a estrutura de “loop” definido termina.

O programa “SORT (Classifica uma lista)”, à página 270, utiliza -1 STEP para decrementar o contador por 1. Neste caso, STEP altera o valor do contador por 1, como faz NEXT, mas o contador decresce ao invés de crescer.

Estruturas de “Loop” Indefinido

Se você não puder especificar antecipadamente quantas vezes repetirá um “loop”, escreva uma *estrutura de “loop” indefinido* contendo uma condição de “loop” e uma condição de teste. As condições são executadas alternadamente, com o resultado da condição de teste determinando se continua.

Esta seção descreve dois tipos de estrutura de “loop” indefinido. A primeira, DO ... UNTIL ... END (FAÇA ... ATÉ ... FIM), executa a condição de “loop” antes da condição de teste. Conseqüentemente, a condição de “loop” é sempre executada pelo menos uma vez. O segundo tipo, WHILE ... REPEAT ... END (ENQUANTO ... REPITA ... FIM), executa a condição de teste primeiro. Conseqüentemente, em alguns casos, a condição de “loop” nunca é executada.

DO ... UNTIL ... END (FAÇA ... ATÉ ... FIM)

O exemplo a seguir processa um objeto repetidamente, até que o processamento não mude o objeto. Uma vez que o processamento precisa ocorrer pelo menos uma vez, antes que o teste possa ser executado, este exemplo utiliza DO ... UNTIL ... END.

```
DO DUP EVAL UNTIL DUP ROT SAME END
```

Esta estrutura funciona como segue:

1. a condição de “loop” DUP EVAL é executada, deixando o objeto e o resultado processado na pilha.
2. a condição de teste DUP ROT SAME é processada, deixando o resultado processado e um “flag” na pilha. O “flag” indica se o objeto e o resultado processado são os mesmos.
3. o “flag” é tomado da pilha. Seu valor determina se a estrutura de “loop” é repetida.
 - Se o “flag” é falso, os passos (1), (2) e (3) são repetidos.
 - Se o “flag” é verdadeiro, a estrutura de “loop” termina.

Suponha que você deseje processar completamente 'A+B', onde A contém 'P+Q' e P contém 2. O primeiro processamento da condição de "loop" devolve 'A+B' e 'P+Q+B'. Estas expressões não são as mesmas, assim a condição de "loop" é processada uma segunda vez, devolvendo 'P+Q+B' e '2+Q+B'. Estas expressões não são as mesmas, assim a condição de "loop" é processada uma terceira vez, devolvendo '2+Q+B' e '2+Q+B'. Estas expressões são as mesmas, assim a estrutura de "loop" termina.

O efeito deste exemplo é similar ao efeito de \rightarrow NUM, exceto que \rightarrow NUM causa um erro se um nome for indefinido. Para uma versão mais versátil deste exemplo, ver "MULTI (Execução Múltipla)", à página 253.

WHILE ... REPEAT ... END (ENQUANTO ... REPITA ... FIM)

O exemplo a seguir toma qualquer número de vetores da pilha e adiciona-os à matriz estatística corrente. Uma vez que ele necessita testar se o objeto no nível 1 é um vetor *antes* de tentar somá-lo, este exemplo utiliza WHILE ... REPEAT ... END.

```
WHILE DUP TYPE 3 == REPEAT  $\Sigma$ + END
```

Esta estrutura opera como segue:

1. a condição de teste `DUP TYPE 3 ==` é processada, deixando um "flag" na pilha. O "flag" indica se o objeto no nível 2 é um vetor real.
2. o "flag" é tomado da pilha. Seu valor determina se a condição de "loop" é executada.
 - Se o "flag" é verdadeiro, a condição de "loop" Σ + é executada, adicionando o vetor à matriz estatística corrente e os passos (1) e (2) são repetidos.
 - Se o "flag" é falso, a estrutura de "loop" termina.

Note que WHILE ... REPEAT ... END termina quando o "flag" é falso, mas DO ... UNTIL ... END termina quando o "flag" é verdadeiro. Caso você queira mudar o valor de uma condição de teste quando verdadeira, adicione NOT como o último comando: WHILE ... NOT REPEAT ou UNTIL ... NOT END.

Para outro exemplo de WHILE ... REPEAT ... END, ver "PAD (Preencha com Espaços no Início)", à página 257.

Estruturas de Programas Superpostas

Uma vez que uma condição em uma estrutura de programa é como um sub-programa, a condição em si pode conter uma estrutura de programa. A estrutura dentro da condição é chamada estrutura *interna*, e a estrutura que contém a condição é chamada a estrutura *externa*. O programa "SORT (Classificar uma lista)", à página 270, demonstra "loops" definidos superpostos.

Não existe limite para o número de níveis que você pode superpor, exceto talvez sua habilidade para entender a lógica. Em alguns casos, é mais fácil armazenar a estrutura interna em programas e utilizar seus nomes como condições nas estruturas externas.

Programas Interativos

Alguns programas requerem instruções do usuário, isto é, instruções suas quando você está executando o programa. Quando o usuário precisa suprir valores para as variáveis, um programa pode solicitar entradas. Quando o usuário precisa escolher entre diversas alternativas, um programa pode solicitar uma seleção.

Este capítulo demonstra como um programa pode solicitar uma entrada ou uma seleção, utilizando os comandos a seguir do menu PROGRAM CONTROL.

| Comando | Descrição |
|-----------------|--|
| HALT | Suspende execução de programa. |
| <i>s</i> WAIT | Suspende execução do programa por <i>s</i> segundos. |
| KEY | Devolve uma cadeia de caracteres se uma tecla foi pressionada. |
| <i>f s</i> BEEP | Emite um som de frequência <i>f</i> por <i>s</i> segundos. |
| CLLCD | Apaga o visor. |
| <i>n</i> DISP | Apresenta um objeto na linha <i>n</i> do visor. |
| CLMF | Restaura a apresentação normal do visor, quando o programa termina a execução. |

Solicitando uma Entrada

A seqüência a seguir cria um menu de entrada personalizado para as variáveis A, B e C, emite um som para alertar o usuário e interrompe a execução para receber a entrada.

```
... ( STO A B C ) MENU 440 .1 BEEP HALT ...
```

O menu apresentado mostra os rótulos `A`, `B` e `C`, os quais lembram os rótulos do menu Solver. Após introduzir um valor na pilha, o usuário pode, simplesmente, pressionar uma destas teclas para armazenar o valor na variável correspondente. Após entrar os valores, o usuário precisa pressionar `CONT`, para continuar a execução do programa.

Solicitando uma Escolha

Para tarefas complexas é melhor escrever uma série de programas pequenos, cada um executando uma pequena tarefa. Em alguns casos, o usuário tem diversas opções para executar uma das tarefas. Um método é escrever programas alternativos para executar, aquela tarefa.

Admita que uma tarefa é completada e o usuário precisa escolher entre os programas HOP, SKIP e JUMP, para a próxima tarefa. A seqüência a seguir cria um menu do usuário personalizado para os programas HOP, SKIP e JUMP, emite um som para alertar o usuário e termina a execução do programa

```
... { HOP SKIP JUMP } MENU 440 .1 BEEP »
```

O menu apresentado mostra os rótulos `HOP`, `SKIP` e `JUMP`, os quais lembram os rótulos do menu USER. Quando o usuário pressiona uma das teclas do menu, a próxima tarefa é executada. Aquela tarefa pode terminar com uma seqüência similar, oferecendo ao usuário um conjunto diferente de opções e assim por diante, através de toda a tarefa complexa.

Um Exemplo Mais Complicado

O exemplo abaixo apresenta uma mensagem, espera até que o usuário pressione uma tecla e verifica que a tecla está definida (isto é, representa uma das escolhas). Se a tecla está definida, a ação correspondente é executada; se a tecla não está definida, uma mensagem de erro é apresentada e o processo se reinicia.

Este exemplo utiliza outras estruturas de programa, descritas no capítulo anterior. Existe uma estrutura "externa" `DO ... UNTIL ... END` que se repete até que o usuário pressione uma tecla definida. A condição `DO` externa contém uma estrutura "interna" `DO ... UNTIL ... END`, que se repete até que o usuário pressione uma tecla. A condição `UNTIL` externa contém uma expressão condicional, que apresenta uma mensagem de erro se a tecla está indefinida.

Na listagem abaixo, o tamanho das margens marca a estrutura exterior, as condições, a estrutura interior e suas condições.

Seqüência

```
( ( 'APPLE' 'BANANA'  
  'CHERRY' )
```

```
DO
```

```
  CLLCD
```

```
  'PRESS'
```

```
  1 DISP
```

```
  ' [A] FOR APPLE'
```

```
  2 DISP
```

```
  ' [B] FOR BANANA'
```

```
  3 DISP
```

```
  ' [C] FOR CHERRY'
```

```
  4 DISP
```

```
DO UNTIL KEY END
```

```
UNTIL
```

Comentários

Esta lista contém as possíveis respostas. Ela permanece na pilha até que a estrutura DO ... UNTIL ... END a seguir devolva 1, 2, ou 3, indicando a escolha do usuário.

Inicia a condição de "loop" externo. Esta condição apresenta mensagens de opções, as quais dizem ao usuário quais as escolhas e obtém uma resposta do usuário.

Apaga o visor.

Mensagem de opção para a linha 1.

Apresenta mensagem.

A mensagem de opção para a linha 2.

Apresenta mensagem.

A mensagem de opção para a linha 3.

Apresenta mensagem.

A mensagem de opção para a linha 4.

Apresenta mensagem.

Este "loop" indefinido interno se repete, até que o usuário pressione uma tecla. O comando KEY devolve 0 se nenhuma tecla foi pressionada, ou uma cadeia (representando a tecla) e 1 se uma tecla foi pressionada. Quando o "loop" termina, a cadeia permanece na pilha.

Inicia a condição externa de teste. Esta condição verifica se a tecla pressionada era uma tecla definida.

| | |
|---|---|
| <code>{ 'A' 'B' 'C' }</code> | Esta lista contém as teclas definidas. Existe uma correspondência bi-unívoca entre as teclas definidas e os possíveis resultados. |
| <code>SWAP POS</code> | Compara a cadeia da tecla com a lista de teclas definidas. POS (<i>posição</i>) devolve 1 se a cadeia da tecla é 'A', 2 se a cadeia da tecla é 'B', 3 se a cadeia da tecla é 'C' ou 0 se não é igual a nenhuma das três |
| <code>IF DUP</code> | Faz uma cópia da posição para utilizar como um "flag". Se a posição é 1, 2, ou 3, executa a condição THEN. Se a posição é 0, executa a condição ELSE. |
| <code>THEN 1</code> | A tecla estava definida, assim coloque um "flag" verdadeiro na pilha. |
| <code>ELSE</code> | A tecla estava indefinida, assim apresente no visor uma mensagem de erro e emita um sinal audível. |
| <code>CLLCD 1 'BAD KEY' 1 DISP</code> | Apresenta uma mensagem de erro. |
| <code>440 .1 BEEP</code> | Emite um som. |
| <code>1 WAIT</code> | Espera 1 segundo. |
| <code>END</code> | Termina a estrutura IF ... THEN ... ELSE ... END. Se a tecla estava definida, a posição e um "flag" verdadeiro estão na pilha. Se a tecla estava indefinida, somente a posição (a qual é também um "flag" falso) está na pilha. |
| <code>END</code> | Termina o "loop" indefinido externo. Se a tecla estava definida, o "loop" termina com a posição na pilha. Se a tecla estava indefinida, a condição de "loop" é repetida. |
| <code>GET</code> | Dada a lista de possíveis resultados e uma posição, obtém-se o resultado correspondente. |

EVAL

Processe o resultado. Neste exemplo, EVAL não tem efeito porque o resultado é uma cadeia. Num exemplo mais realístico, o resultado poderia ser um programa (possivelmente armazenado em uma variável), assim EVAL seria necessário.

CLMF

Ative a apresentação normal da pilha no visor.

Quando esta seqüência é executada, o usuário vê as mensagens de opção.

```
Press
[A] for Apple
[B] for Banana
[C] for Cherry
```

Se o usuário pressionar uma tecla que não [A], [B] ou [C], um som é emitido e a mensagem de erro aparece por 1 segundo.

```
Bad key
```

Então a mensagem de opção reaparece. Quando o usuário pressiona [A], [B], ou [C] a cadeia "APPLE", "BANANA" ou "CHERRY", é devolvida ao nível 1.

Modificando a lista de possíveis resultados, as mensagens de opção e a lista de teclas definidas, você pode tornar esta seqüência mais significativa do que simplesmente colocar uma cadeia na pilha. De modo mais geral, através do uso de variáveis locais e colocando esta seqüência dentro de uma estrutura de variável local você pode fazer o programa que segue.

```

« → KEYS M1 M2 M3 M4
« DO CLLCD
  M1 1 DISP
  M2 2 DISP
  M3 3 DISP
  M4 4 DISP
  DO UNTIL KEY END
  UNTIL KEYS SWAP POS
  IF DUP
  THEN 1
  ELSE CLLCD ' 'BAD KEY' ' 1 DISP
    440 .1 BEEP 1 WAIT
  END
  END
»
GET EVAL CLMF
»

```

Se você armazenar este programa em uma variável denominada KEY?, você poderá executar o exemplo acima através dos seguintes passos

```

( 'APPLE' 'BANANA' 'CHERRY' )
( 'A' 'B' 'C' )
'PRESS'
'[A] FOR APPLE'
'[B] BANANA'
'[C] CHERRY'
KEY?

```

Exemplos de Programação

Este capítulo contém 20 programas para sua HP-285. Estes programas são úteis e, o mais importante, demonstram uma variedade de técnicas de programação. Para cada programa você encontrará a informação a seguir.

- **Diagrama da Pilha.** Um diagrama da pilha é uma tabela com duas colunas mostrando "Argumentos" e "Resultados". "Argumentos" mostra o que precisa estar na pilha antes do programa ser executado; "Resultados" mostra o que o programa deixa na pilha.

O diagrama da pilha não mostra tudo; um programa que muda a memória do usuário ou apresenta objetos no visor, pode não ter efeito na pilha.

- **Técnicas.** Esta é a parte mais interessante. Quando você entender como uma técnica é utilizada neste capítulo, você pode utilizá-la em seus próprios programas.
- **Programas Necessários.** Alguns programas chamam outros como sub-rotinas. Você pode entrar os programas necessários e o programa que os chama em qualquer ordem, mas você precisa entrá-los todos antes de executar o programa principal.
- **Programa e Comentários.** Este capítulo formata a listagem do programa para mostrar sua estrutura e processo. Você não necessita seguir o formato da listagem quando você entra um programa. Entretanto, assegure-se de digitar espaços onde eles aparecem na listagem ou entre objetos aparecendo em linhas separadas.

Você pode digitar um programa caractere a caractere, ou você pode utilizar os menus para digitá-lo comando a comando. Não faz diferença desde que o resultado seja igual à listagem.

Quando você digitar o programa, você pode omitir todos os fechamentos de parênteses e delimitadores *que aparecem bem no final do programa*; quando você pressionar `ENTER` o fechamento de parênteses e delimitadores é adicionado automaticamente.

- Exemplo. Os exemplos pressupõem formato de apresentação STD (padrão) no visor. Para selecionar formato de apresentação STD, pressione STD `ENTER` ou use o menu MODE.

A técnica mais importante demonstrada neste capítulo é *programação estruturada*: pequenos programas utilizados para construir outros programas. Os programas abaixo são utilizados em outros programas.

- BOXS é utilizado por BOXR.
- MULTI é utilizado em EXCO.
- PAD e PRESERVE são utilizados em BDISP.
- Σ GET é utilizado em $\Sigma X2$, $\Sigma Y2$ e ΣXY .
- SORT e LMED são utilizados em MEDIAN.

Funções Caixa (Box Functions)

Esta seção contém dois programas:

- BOXS calcula a superfície total de uma caixa.
- BOXR utiliza BOXS para calcular a relação entre a superfície e o volume para uma caixa.

BOXS (Superfície de uma Caixa)

Dada a altura, largura e comprimento de uma caixa, calcule a área total de seus seis lados.

| Argumentos | Resultados |
|----------------|------------|
| 3: altura | 3: |
| 2: largura | 2: |
| 1: comprimento | 1: área |

Técnicas

- Estrutura de variável local. Variáveis locais permitem que você atribua nomes a argumentos, sem conflitar com variáveis globais. Tal como variáveis globais, variáveis locais são convenientes porque você pode utilizar argumentos qualquer número de vezes, sem manter controle sobre suas posições na pilha; diversamente de variáveis globais, variáveis locais desaparecem quando a estrutura de programa que as criou termina.

Uma estrutura de variável local tem três partes.

1. Um comando denominado “ \rightarrow ”. Quando você digita este comando, lembre-se de colocar espaços antes e depois dele. (Como qualquer comando, \rightarrow é escrito utilizando caracteres normais e reconhecidos somente quando é separado por espaços. Não confunda este comando de um caractere com delimitadores como # ou \llcorner .)
2. Um ou mais nomes.
3. Um procedimento (expressão, equação ou programa) que inclui os nomes. Este procedimento é denominado o procedimento de *definição*.

Quando uma estrutura de variável local é processada, uma variável local é criada para cada nome. Os valores para as variáveis locais são tomados da pilha. O procedimento de definição é então processado, substituindo os valores das variáveis locais.

Para apreciar-se o poder de variáveis locais, compare a versão de BOXS dada abaixo com a versão que aparece à página 244.

- Função do Usuário. Este tipo de programa funciona tanto em sintaxe RPN como em algébrica. Uma função do usuário é um programa que consiste unicamente de uma estrutura de variável local e devolve exatamente um resultado.

Programa

```
 $\llcorner$   
 $\rightarrow$  HU 1
```

Comentários

Inicia o programa.

Cria variáveis locais para a altura, largura e comprimento. Por convenção letras minúsculas são utilizadas. Os valores são tomados da pilha (em RPN) ou dos argumentos da função do usuário (em sintaxe algébrica).

Programa

$1/2 * (H * W + H * L + W * L)$

»

ENTER

BOXS STO

Comentários

A expressão de definição para a área da superfície. O processamento da função do usuário causa o processamento desta expressão, devolvendo a área para a pilha.

Termina o programa.

Coloca o programa na pilha.

Armazena o programa em BOXS.

Exemplo. Uma das vantagens de funções do usuário é que elas funcionam tanto em RPN como em sintaxe algébrica. Calcule a superfície de uma caixa de 12 polegadas de altura, 16 polegadas de largura e 24 polegadas de comprimento; faça o cálculo primeiro em RPN e, então, em sintaxe algébrica.

Para a versão RPN primeiro entre a altura e a largura.

USER

12 ENTER

16 ENTER

| | |
|------|----|
| 3: | |
| 2: | 12 |
| 1: | 16 |
| BOXS | |

Então digite o comprimento e execute BOXS.

24 BOXS

| | |
|------|------|
| 3: | |
| 2: | |
| 1: | 1728 |
| BOXS | |

A superfície da área é 1728 polegadas quadradas.

Agora tente a versão algébrica.

BOXS (12, 16, 24 EVAL

| | |
|------|------|
| 3: | |
| 2: | 1728 |
| 1: | 1728 |
| BOXS | |

Novamente, a área da superfície é 1728.

BOXS sem Variáveis Locais

O programa seguinte utiliza apenas operações da pilha para calcular a superfície de uma caixa. Compare este programa com BOXS.

| Argumentos | Resultados |
|----------------|------------|
| 3: altura | 3: |
| 2: largura | 2: |
| 1: comprimento | 1: área |

Programa

```
« .  
DUP2 *  
ROT .  
4 PICK  
*  
+  
ROT ROT  
*  
+  
2 *  
»
```

Comentários

Começa o programa
Calcula wl .
Move w para o nível 1.
Copia h para o nível 1.
Calcula wh .
Calcula $wl + wh$.
Move h e l para os níveis 2 e 1.
Calcula hl .
Calcula $wl + wh + hl$.
Calcula $2(wl + wh + hl)$.
Fim do programa.

Como esta versão de BOXS não é uma função do usuário, ela não pode ser utilizada em sintaxe algébrica.

BOXR (Relação entre a Superfície e o Volume de uma Caixa)

Dada a altura, largura e comprimento de uma caixa, calcule a relação de sua superfície para seu volume.

| Argumentos | Resultados |
|----------------|----------------|
| 3: altura | 3: |
| 2: largura | 2: |
| 1: comprimento | 1: área/volume |

Técnicas

- Funções do usuário superpostas. BOXR é uma função do usuário cuja expressão de definição utiliza BOXS em seu cálculo. Por sua vez, BOXS pode ser utilizada para definir outras funções do usuário.

Lembre-se de que BOXS foi definida utilizando h , w e l como variáveis locais, e observe abaixo que BOXS toma x , y e z como argumentos na definição de BOXR. Não faz diferença se as variáveis locais nas duas definições casam ou não, porque cada conjunto de variáveis locais é independente da outra. Entretanto, é essencial que as variáveis locais sejam consistentes dentro de uma única definição.

Programa

```
«
+ XYZ
+ BOXS (X,Y,Z)
/ ( * * 2 )
»
```

ENTER

BOXR STO

Comentários

Começa o programa.

Cria variáveis locais para altura, largura e comprimento. Este programa utiliza x , y e z , ao invés de h , w e l .

Começa a expressão de definição com a função do usuário BOXS.

Divide pelo volume da caixa.

Fim do programa.

Coloca o programa na pilha.

Armazena o programa como BOXR.

Exemplo. Calcule a relação entre a superfície e o volume para uma caixa de 9 polegadas de altura, 18 polegadas de largura e 21 polegadas de comprimento; faça o cálculo primeiro em RPN e, então, em sintaxe algébrica.

Para a versão RPN, primeiro entre a altura e a largura.

USER
9 ENTER
18 ENTER

```
3:
2:
1:          9
          18
BOXR BOXS
```

Em seguida, digite o comprimento e execute BOXR.

21 BOXR

```
3:
2:
1:          .428571428571
BOXR BOXS
```

A relação é .428571428571.

Agora tente a versão algébrica.

BOXR (9,18,21 EVAL

```
3:
2:          .428571428571
1:          .428571428571
BOXR BOXS
```

Novamente, a relação é .428571428571.

Números de Fibonacci

Dado um inteiro n , calcule o n -ésimo número de Fibonacci F_n , onde

$$F_0 = 0, \quad F_1 = 1, \quad F_n = F_{n-1} + F_{n-2}$$

Esta seção inclui dois programas, cada um demonstrando uma solução para este problema.

- FIB1 é uma função do usuário que é definida *recursivamente* — sua expressão de definição contém seu próprio nome. FIB1 é curta, fácil de entender e utilizável em objetos algébricos.
- FIB2 é um programa com “loop” definido. Não é utilizável em objetos algébricos, é mais longo e mais complicado do que FIB1, mas é mais rápido.

FIB1 (Números de Fibonacci, Versão Recursiva)

| Argumentos | Resultados |
|------------|------------|
| 1 : n | 1 : F_n |

Técnicas

- IFTE (Função If-Then-Else). A expressão de definição para FIB1 contém a função condicional IFTE, a qual pode ser utilizada tanto em RPN como em sintaxe algébrica. (FIB2 utiliza a estrutura de programa IF ... THEN ... ELSE ... END.)
- Recursividade. A expressão de definição para FIB1 é escrita em termos de FIB1, da mesma forma que F_n é definida em termos de F_{n-1} e F_{n-2} .

Programa

```

«
  → N
  .
  IFTE (N ≤ 1,
    N,
    FIB1 (N - 1) + FIB1 (N - 2))
  .
»

```

ENTER

FIB1 STO

Comentários

Inicia o programa.
 Define a variável local.
 Inicia a expressão de definição.
 Se $n \leq 1$,
 Então $F_n = n$;
 Caso contrário $F_n = F_{n-1} + F_{n-2}$.
 Fim da expressão de definição.
 Fim do programa.
 Coloca o programa na pilha.
 Armazena o programa como FIB1.

Exemplo. Calcule F_6 utilizando sintaxe RPN e F_{10} utilizando sintaxe algébrica.

Primeiro calcule F_6 utilizando RPN.

USER
 6 FIB1

```

3:
2:
1: 8
FIB1 BOND BONS

```

A seguir, calcule F_{10} utilizando sintaxe algébrica.

FIB1 (10 EVAL

```

3:
2:
1: 55
FIB1 BOND BONS

```

FIB2 (Números de Fibonacci, Versão "Loop")

| Argumentos | Resultados |
|------------|------------|
| 1: n | 1: F_n |

Técnicas

- IF ... THEN ... ELSE ... END. FIB2 utiliza a forma condicional de estrutura de programa. (FIB1 utiliza IFTE.)
- START ... NEXT ("loop" definido). Para calcular F_n , FIB2 começa com F_0 e F_1 e repete o "loop" para calcular sucessivos F_i 's.

Programa

```

«
  + N
  «
    IF N 1 ≤
    THEN N
    ELSE
      0 1
      2 N
      START
        DUP
      ROT
      +
    NEXT
    SWAP DROP
  END
»
»
[ENTER]
['] FIB2 [STO]

```

Comentários

Inicia o programa.
 Cria a variável local.
 Inicia o programa de definição.
 If $n \leq 1$,
 Então $F_n = n$;
 Inicia condição ELSE.
 Coloca F_0 e F_1 na pilha.
 De 2 até n ,
 Executa o seguinte "loop":
 Faz uma cópia do F mais recente (inicialmente F_1).
 Move o F anterior (inicialmente F_0) para o nível 1.
 Calcula o próximo F (inicialmente F_2).
 Repete o "loop".
 Elimina F_{n-1} .
 Fim da condição ELSE.
 Fim do programa de definição.
 Fim do programa.
 Coloca o programa na pilha.
 Armazena o programa como FIB2.

Exemplo. Calcule F_6 e F_{10} . Note que FIB2 é mais rápido do que FIB1.

Calcule F_6 .

```
USER
6 FIB2
```

```
3:
2:
1: 8
FIB2 FIB1 BOMR BONS
```

Calcule F_{10} .

```
10 FIB2
```

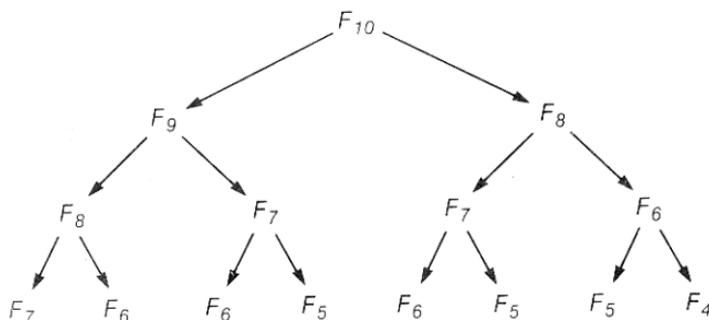
```
3:
2: 8
1: 55
FIB2 FIB1 BOMR BONS
```

Comparação de FIB1 e FIB2

FIB1 calcula valores intermediários F_i mais do que uma vez, enquanto FIB2 calcula cada F_i intermediário somente uma vez. Conseqüentemente, FIB2 é mais rápido.

A diferença em velocidade aumenta com o tamanho de n porque o tempo requerido por FIB1 cresce exponencialmente com n , enquanto que o tempo requerido por FIB2 cresce somente linearmente com n .

O diagrama abaixo mostra os passos iniciais de FIB1 calculando F_{10} . Note o número de cálculos intermediários: 1 na primeira linha, 2 na segunda linha, 4 na terceira linha e 8 na quarta linha.



Execução Passo-a-Passo

É mais fácil entender como um programa funciona se você executá-lo passo-a-passo, vendo o efeito de cada passo na pilha. Fazer isso, pode auxiliá-lo a depurar seus próprios programas ou ajudá-lo a entender programas escritos por terceiros.

Esta seção mostra como executar FIB2 passo-a-passo, mas você pode aplicar estas regras para qualquer programa. As regras gerais são:

1. utilize VISIT para inserir o comando HALT no programa. Coloque HALT onde você deseja iniciar execução passo-a-passo. (Você verá como a posição de HALT dentro de FIB2 afeta a execução.)
2. execute o programa. Quando o comando HALT é executado, o programa pára (indicado pelo anúncio de parada).
3. selecione o menu PROGRAM CONTROL.
4. pressione `Ⓢ` uma vez para ver o próximo passo de programa apresentado no visor e, então, executado.

Você pode agora:

- continuar pressionando `Ⓢ` para apresentar no visor e executar passos seqüenciais.
 - pressionar `■` `CONT` para continuar a execução normal.
 - pressionar `KILL` para abandonar a execução do programa.
5. quando você desejar que o programa volte a executar normalmente, utilize VISIT para remover HALT do programa.

Para o primeiro exemplo, insira HALT como primeiro comando em FIB2.

Apague a pilha e selecione o menu USER.

CLEAR
 USER

```
3:  
2:  
1:  
FIB2 FIB1 BOWR BOWS
```

Utilize VISIT para devolver FIB2 para a linha de comando.

FIB2 VISIT

```
→ n  
◀  
IF n 1 ≤  
THEN n
```

Insira o comando HALT.

INS CONTRL HALT

```
◀ HALT ↔ n  
◀  
IF n 1 ≤  
SST HALT ABORT KILL WAIT KEY
```

Armazene a versão editada de FIB2.

ENTER

```
3:  
2:  
1:  
SST HALT ABORT KILL WAIT KEY
```

Calcule F_1 . Inicialmente, nada ocorre exceto que o anúncio  aparece.

USER
1 FIB2

```
3:  
2:  
1:  
FIB2 FIB1 BOWR BOWS 1
```

Selecione o menu PROGRAM CONTROL e execute SST (*single-step= passo-a-passo*). Olhe para a linha superior do visor para ver o primeiro passo apresentado antes de ser executado.

CONTRL
SST

```
→ n  
2:  
1:  
SST HALT ABORT KILL WAIT KEY 1
```

Note que $\rightarrow n$ constitui um passo; “passo” é uma unidade lógica e não apenas o próximo objeto do programa.

Examine as regras gerais no início desta seção. Você executou os primeiros quatro passos e agora você pode escolher uma das três alternativas para o passo 5.

Para este exemplo, pressione `EST` repetidamente até que o anúncio  desapareça, indicando que FIB2 foi completada. (Estes passos individuais não são mostrados aqui.)

O cálculo para F_1 executa somente a condição THEN em FIB2. Para o segundo exemplo, execute `3 FIB2` e prossiga passo-a-passo através do cálculo de F_3 . Isto executa a condição ELSE, incluindo o "loop" START ... NEXT. Você verá que para $n = 3$, o "loop" START ... NEXT é executado duas vezes.

Para o terceiro exemplo, suponha que você queira seguir passo-a-passo o "loop" START ... NEXT por completo — vendo a pilha antes de cada iteração do "loop", mas não prosseguindo passo-a-passo para todos os passos FIB2 ou no "loop" em si. Para fazer isso, mova o comando HALT para dentro do "loop". Neste caso FIB2 não será interrompido até que atinja o "loop". Você pode utilizar  `CONT` (*continue*) para executar o "loop" uma iteração por vez.

Utilize VISIT para devolver FIB2 para a linha de comando.

`USER`

`FIB2`  `VISIT`

```

 HALT → n
  *
    IF n 1 ≤
    THEN n
  
```

Utilize as teclas do menu do cursor para eliminar HALT. Então insira HALT como mostrado (em seguida ao comando START).

```

    IF n 1 ≤
    THEN n
    ELSE 0 1 2 n
        START HALT♦DUP R...
  
```

Armazene a versão editada de FIB2.

`ENTER`

```

3:
2:
1:
FIB2 FIB1 BOWR BOWS
  
```

Inicie o cálculo para F_3 . FIB2 parará antes de executar o "loop".

`3 FIB2`

```

3:
2:
1:
FIB2 FIB1 BOWR BOWS
  
```

Continue a execução do "loop". FIB2 será interrompido antes de executar o "loop" uma segunda vez.

■ CONT

```
3:
2:
1:
FIB2 FIB1 BOWR BOWS
```

Continue a execução do "loop". Como esta é a última iteração do "loop", FIB2 executará até o final.

■ CONT

```
3:
2:
1:
FIB2 FIB1 BOWR BOWS
```

Quando você terminar de experimentar com FIB2, não se esqueça de utilizar VISIT para remover o comando HALT.

Desenvolvendo e Agrupando Completamente

Esta seção contém dois programas:

- MULTI repete um programa até que este não tenha efeito.
- EXCO utiliza MULTI, para desenvolver e agrupar completamente.

MULTI (Execução Múltipla)

Dado um objeto e um programa que atua nesse objeto, aplique o programa ao objeto repetidamente até que o objeto fique inalterado.

| Argumentos | Resultados |
|------------------------|-----------------------------|
| 2: <i>objeto</i> | 2: |
| 1: « <i>programa</i> » | 1: <i>objeto resultante</i> |

Técnicas

- DO ... UNTIL ... END (“loop” indefinido). A condição DO contém os passos a serem repetidos; a condição UNTIL contém o teste que determina se deve repetir ambas as condições novamente (se falso) ou sair do “loop” (se verdadeiro).
- Programas como argumentos. Embora, normalmente, atribua-se nomes a programas (e execute-os chamando-os por seus nomes), programas podem ser colocados na pilha e utilizados como argumentos para outros programas.
- Processamento de variáveis locais. O argumento programa para ser executado repetidamente é armazenado em uma variável local. É prático armazenar um objeto em uma variável local quando você não sabe antecipadamente de quantas cópias você necessitará.

MULTI demonstra uma das diferenças entre as variáveis globais e locais: se uma variável global contém um nome ou programa, o conteúdo da variável é processado quando o nome é processado; mas o conteúdo da variável local é sempre simplesmente recuperado. Conseqüentemente, MULTI utiliza o nome local para colocar o argumento programa na pilha e, então, executa um comando EVAL explícito para processar o programa.

Programa

```
«
→ P
»
DO
  DUP
  P EVAL
.
UNTIL
  DUP
.
  ROT
  SAME
```

Comentários

Inicia o programa.
Cria a variável local *p* que contém o argumento programa.
Inicia o programa de definição.
Inicia a condição DO.
Efetua uma cópia do objeto.
Aplica o programa ao objeto, devolvendo uma nova versão. (O comando EVAL é necessário para executar o programa, porque variáveis locais sempre devolvem seus conteúdos à pilha sem processamento.)
Inicia a condição UNTIL.
Efetua uma cópia da nova versão do objeto.
Move a versão antiga para o nível 1.
Testa se a versão antiga e a nova versão são a mesma.

Programa

END

»
»

ENTER

· MULTI STO

Comentários

Fim da condição UNTIL.

Fim do programa de definição.

Fim do programa.

Coloca o programa na pilha.

Armazena o programa como MULTI.

Exemplo. MULTI é demonstrado no próximo programa.

EXCO (Desenvolvendo e Agrupando Completamente - Expand and Collect Completely)

Dado um objeto algébrico, execute EXPAN repetidamente até que o algébrico não mude, então execute COLCT repetidamente até que o algébrico não mude. Em alguns casos o resultado será um número.

| Argumentos | Resultados |
|----------------|----------------|
| 1: 'algébrico' | 1: 'algébrico' |
| 1: 'algébrico' | 1: z |

Técnicas

- Programação estruturada. EXCO chama o programa MULTI duas vezes. Mesmo que você não utilize MULTI em nenhum outro lugar, a eficiência de repetir-se todos os comandos em MULTI, simplesmente incluindo seu nome uma segunda vez, justifica escrever MULTI como um programa separado.

Programas Necessários

- MULTI (página 253) executa repetidamente os programas que EXCO fornece como argumentos.

Programa

```
«  
« EXPAN »  
MULTI  
  
« COLCT »  
MULTI  
  
»
```

ENTER
EXCO STO

Exemplo. Desenvolva e agrupe completamente a expressão.

$$3x(4y + z) + (8x - 5z)^2.$$

Entre a expressão.

USER
3 X X
(4 X Y + Z) +
(8 X X - 5 X Z) ^ 2
ENTER

```
2:  
1: '3*X*(4*Y+Z)+(8*X-5*  
Z)^2'  
EXCO MULT FIB2 FIB1 BOMR BONS
```

Desenvolva e agrupe completamente.

EXCO

```
2:  
1: '12*X*Y-77*X*Z+64*X^  
2+25*Z^2'  
EXCO MULT FIB2 FIB1 BOMR BONS
```

Expressões com muitos produtos de somas ou com potências podem tomar muitas iterações de EXPAN, para serem desenvolvidas completamente, resultando num longo tempo de execução para EXCO.

Comentários

Inicia o programa.
Coloca EXPAN na pilha.
Executa EXPAN até que o objeto algébrico não se modifique mais.
Coloca COLCT na pilha.
Executa COLCT até que o objeto algébrico não mude mais.
Fim do programa.

Coloca o programa na pilha.
Armazena o programa como EXCO.

Apresentando no Visor um Inteiro Binário

Esta seção contém três programas:

- PAD é um programa utilitário que converte um objeto em uma cadeia, para apresentação no visor, alinhado à direita.
- PRESERVE é um programa utilitário para utilização em programas que mudam o estado da calculadora (modo angular, base binária, etc).
- BDISP apresenta no visor um inteiro binário nas bases HEX, DEC, OCT e BIN. Ele chama PAD para apresentar os números no visor, alinhados à direita e chama PRESERVE para preservar a base binária.

PAD (Preenche com Espaços no Início)

Converte um objeto em uma cadeia e se a cadeia contém menos do que 23 caracteres, adiciona espaços no início.

Quando uma cadeia curta é apresentada no visor através de DISP, ela aparece *alinhada à esquerda*: seu primeiro caractere aparece ao lado esquerdo do visor. A posição do último caractere é determinada pelo comprimento da cadeia.

Ao adicionar espaços no início de uma cadeia curta, PAD move a posição do último caractere para a direita. Quando a cadeia tem 23 caracteres de comprimento, ela aparece *alinhada à direita*: seu último caractere aparece ao lado direito do visor.

PAD não tem efeito em cadeias com mais do que 22 caracteres.

| Argumentos | Resultados |
|------------|-------------|
| 1: object | 1: "object" |

Técnicas

- WHILE ... REPEAT ... END ("loop" indefinido). A condição WHILE contém um teste que determina se deve executar a condição REPEAT e testar novamente (se verdadeira), ou se deve saltar a condição REPEAT e sair do "loop" (se falsa).

- Operações com cadeias. PAD demonstra como converter um objeto para a forma de cadeia, contar o número de caracteres e concatenar duas cadeias.

Programa

```

«
+STR

WHILE
  DUP SIZE 23<

REPEAT
  " " SWAP +
END
»

```

Comentários

Inicia o programa.
 Assegura que o objeto está em forma de cadeia. (Cadeias não são afetadas por este comando.)
 Inicia a condição WHILE.
 A cadeia contém menos do que 23 caracteres?
 Inicia condição REPEAT.
 Adiciona um espaço no início.
 Fim da condição REPEAT.
 Fim do programa.

ENTER

· PAD STO

Coloca o programa na pilha.
 Armazena o programa como PAD.

Exemplo. PAD é demonstrado no programa BDISP.

PRESERVE (Salvar e Restabelecer o Estado Anterior)

Dado um programa na pilha, armazene o estado corrente, execute o programa e, então, restaure o estado anterior.

| Argumentos | Resultados |
|-----------------|----------------------------|
| 1: « programa » | 1: (resultado do programa) |

Técnicas

- RCLF e STOF. PRESERVE utiliza RCLF (*recall flags = chama "flags"*) para registrar o estado corrente da calculadora em um inteiro binário e STOF (*store flags = armazena "flags"*) para restaurar o estado a partir daquele inteiro binário.

- Estrutura de variável local. PRESERVE cria uma variável local somente para remover o objeto da pilha brevemente; seu programa de definição faz muito pouco, exceto processar o argumento programa na pilha.

Programa

```

⌘
RCLF

⇨F
⌘
EVAL
F STOF

⌘
⌘

```

```

[ENTER]
[ ] PRESERVE [STO]

```

Comentários

Inicia o programa.
 Recupera um inteiro binário de 64 bits representando o estado de todos os 64 "flags" do usuário.
 Armazena o inteiro binário em uma variável local *f*.
 Inicia o programa de definição.
 Executa o argumento programa.
 Restaura o estado de todos os 64 "flags" do usuário.
 Fim do programa de definição.
 Fim do programa.

Coloca o programa na pilha.
 Armazena o programa como PRESERVE.

Exemplo: PRESERVE é demonstrado no programa BDISP.

BDISP (Apresentação de Binário no Visor = Binary Display)

Apresenta no visor um número nas bases HEX, DEC, OCT e BIN.

| Argumentos | Resultados |
|---------------|---------------|
| 1: # <i>n</i> | 1: # <i>n</i> |
| 1: <i>n</i> | 1: <i>n</i> |

Técnicas

- IFERR ... THEN ... END (armadilha de erro). Para acomodar números reais, BDISP inclui o comando $R \rightarrow B$ (*real-para-binário*). Entretanto, este comando causa um erro se o argumento *já* é um inteiro binário.

Para manter a execução se um erro ocorrer, o comando $\mathbb{R} \rightarrow \mathbb{B}$ é colocado dentro de uma condição IFERR. Como nenhuma ação é requerida quando um erro ocorre, a condição THEN não contém comandos.

- Ativando LAST. Caso um erro ocorra, LAST precisa estar ativado para devolver o argumento à pilha. BDISP ativa o “flag” 31 para, sob controle de programa, ativar o recurso de recuperação LAST.
- “Loop” FOR ... NEXT. (“loop” definido com contador). BDISP executa um “loop” de 1 a 4, cada vez apresentando no visor n em uma base diferente, em uma linha diferente.

O contador do “loop” (denominado j neste programa) é uma variável local. É criado pela estrutura de programa FOR ... NEXT (ao invés de por um comando \rightarrow) e é automaticamente incrementado por NEXT.

- Subprogramas. BDISP demonstra três usos para subprogramas.
 1. BDISP contém um subprograma principal e uma chamada para PRESERVE. O subprograma principal vai para a pilha e é processado por PRESERVE.
 2. Quando BDISP cria uma variável local para n , o programa de definição é um subprograma.
 3. Existem quatro subprogramas que “personalizam” a ação do “loop”. Cada subprograma contém um comando para modificar a base binária, e cada iteração do “loop” executa um destes subprogramas.

Programas Necessários

- PAD (página 257) expande uma cadeia até 23 caracteres de modo que DISP a mostre alinhada à direita.
- PRESERVE (página 258) armazena o estado corrente, executa os subprogramas principais e restaura o estado.

Programa

```
«
«
DUP
31 SF
IFERR
  R+B
THEN
END

→ N
«
  CLLCD
  « BIN »
  « OCT »
  « DEC »
  « HEX »
  1 4
  FOR J
    EVAL

    N+STR

    PAD
    JDISP
  NEXT
  »
  »
PRESERVE
```

»

ENTER

BDISP STO

Comentários

Inicia o programa.
Inicia o subprograma principal.
Efetua uma cópia de n .
Ativa "flag" 31 para ativar LAST.
Inicia armadilha de erro.
Converte n para um inteiro binário.
Se um erro ocorrer,
Não faz nada (não há comandos na condição THEN).
Cria a variável local n .
Inicia o programa de definição.
Apaga o visor.
Subprograma para BIN.
Subprograma para OCT.
Subprograma para DEC.
Subprograma para HEX.
Primeiro e último valores do contador.
Inicia "loop" com contador j .
Processa um dos subprogramas da base (inicialmente aquele para HEX).
Forma uma cadeia mostrando n na base corrente.
Preenche a cadeia até 23 caracteres.
Apresenta a cadeia na j -ésima linha.
Incrementa j e repete o "loop".
Fim do programa de definição.
Fim do subprograma principal.
Armazena o estado corrente, executa o subprograma principal e restaura o estado.
Fim do programa.
Coloca o programa na pilha.
Armazena o programa como BDISP.

Exemplo. Mude para a base DEC, apresente # 100 em todas as bases e verifique que BDISP restaurou a base DEC.

Apague a pilha e selecione o menu BINARY.

```
3:
2:
1:
DEC= HEX OCT BIN STWS RCWS
```

Assegure-se de que a base corrente é DEC e digite # 100.

```
3:
2:
1: # 100d
DEC= HEX OCT BIN STWS RCWS
```

Execute BDISP.(Não mude de menu, uma vez que você querará ver o menu BINARY no próximo passo.)

BDISP

```
# 64h
# 100d
# 144o
# 1100100b
```

Volte para a apresentação normal do visor e verifique a base corrente.

```
3:
2:
1: # 100d
DEC= HEX OCT BIN STWS RCWS
```

Embora o subprograma principal tenha deixado a calculadora na base BIN, PRESERVE restaurou a base DEC.

Para verificar que BDISP também funciona para números reais, tente 144.

144 BDISP

```
# 90h
# 144d
# 220o
# 10010000b
```

Estatísticas com Somatórios

Para executar cálculos estatísticos com dados emparelhados muitas vezes é útil calcular a soma dos quadrados (Σx^2 e Σy^2) e a soma dos produtos (Σxy) das duas variáveis. Esta seção contém cinco programas:

- SUMS cria uma variável ΣCOV que contém a matriz de covariância para a matriz estatística corrente ΣDAT .
- ΣGET extrai um número da posição especificada em ΣCOV .
- ΣX^2 utiliza ΣGET para extrair Σx^2 de ΣCOV .
- ΣY^2 utiliza ΣGET para extrair Σy^2 de ΣCOV .
- ΣXY utiliza ΣGET para extrair Σxy de ΣCOV .

Se ΣDAT contém n colunas, ΣCOV é uma matriz $n \times n$. Os programas ΣX^2 , ΣY^2 e ΣXY se referem a ΣPAR (*parâmetros estatísticos*) para determinar quais colunas contém os dados x (denominados C_1) e os dados y (denominados C_2).

Técnicas

- Operações matriciais. Estes programas demonstram como transpor uma matriz, como multiplicar duas matrizes e como extrair um elemento de uma matriz.
- Programas utilizáveis em objetos algébricos. Uma vez que ΣX^2 , ΣY^2 e ΣXY são compatíveis com a sintaxe algébrica (não há argumentos da pilha, um resultado é colocado na pilha), você pode utilizar seus nomes como variáveis comuns em uma expressão ou equação.
- Convenção ΣPAR . Diversos comandos para estatística com dados emparelhados utilizam uma variável denominada ΣPAR para especificar um par de colunas em ΣDAT . ΣPAR contém uma lista com quatro números, os dois primeiros especificando colunas. (Os outros dois números são os coeficientes angular e linear da regressão linear).

SUMS assegura que ΣPAR existe executando 0 PREDV DROP; o comando PREDV (*predicted value = valor projetado*) cria ΣPAR com valores padrão se ΣPAR não existe ainda, e DROP retira o valor projetado computado para 0.

ΣX^2 , ΣY^2 e ΣXY utilizam os valores armazenados em ΣPAR para determinar qual elemento extrair de ΣCOV .

SUMS (Matriz de Somatórios Estatísticos)

Cria uma variável ΣCOV que contém a matriz de covariância da matriz estatística ΣDAT .

Como um exemplo, se ΣDAT é a matriz $n \times 2$.

$$\begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_n & y_n \end{bmatrix}$$

então ΣCOV conterà a matriz de covariância.

$$\begin{bmatrix} \Sigma x^2 & \Sigma xy \\ \Sigma xy & \Sigma y^2 \end{bmatrix}$$

| Argumentos | Resultados |
|------------|------------|
| 1 : | 1 : |

Programa

```

❖
RCLΣ
DUP
TRN
SNAP #
'ΣCOV' STO
⊖ PREDV DROP
»

```

Comentários

Inicia o programa.
 Recupera o conteúdo da matriz estatística $n \times m$ ΣDAT .
 Efetua uma cópia.
 Transpõe a matriz. O resultado é uma matriz $m \times n$.
 Multiplica as matrizes para produzir a matriz de covariância $m \times m$. (Sem trocar a posição das matrizes o produto seria uma matriz $n \times n$).
 Armazena a matriz de covariância na variável ΣCOV .
 Assegura que ΣPAR existe.
 Fim do programa.

Coloca o programa na pilha.
 Armazena o programa como SUMS.

Σ GET (Obtém um elemento de Σ COV)

Dados p e q , cada um indicando ou a primeira ou a segunda *posição* em Σ PAR, extraia o elemento rs de Σ COV, onde r e s são o primeiro e segundo *elementos* correspondentes em Σ PAR.

Σ GET é chamada por $\Sigma X2$, $\Sigma Y2$ e ΣXY com os seguintes argumentos.

- Para $\Sigma X2$, $p = 1$ e $q = 1$.
- Para $\Sigma Y2$, $p = 2$ e $q = 2$.
- Para ΣXY , $p = 1$ e $q = 2$.

| Argumentos | Resultados |
|------------|--------------------------|
| 2 : 1 ou 2 | 2 : |
| 1 : 1 ou 2 | 1 : elemento rs de COV |

Programa

```
⌘  
ΣCOV  
ΣPAR  
  
DUP  
5 ROLL  
GET  
SWAP  
4 ROLL  
GET  
2 →LIST  
GET  
⌘
```

ENTER

⌘ ΣGET STO

Comentários

Inicia o programa.
Coloca a matriz de covariância na pilha.
Coloca a lista de parâmetros estatísticos na pilha.
Efetua uma cópia.
Move p para o nível 1.
Obtém r , o p -ésimo elemento em Σ PAR.
Move Σ PAR para o nível 1.
Move q para o nível 1.
Obtém s , o q -ésimo elemento em Σ PAR.
Coloca $\{r, s\}$ na pilha.
Obtém o elemento r,s de Σ COV.
Fim do programa.

Coloca o programa na pilha.
Armazena o programa como Σ GET.

ΣX^2 (Soma dos Quadrados de x)

Calcule Σx^2 , onde os x 's são os elementos de C_1 (a coluna especificada pelo primeiro parâmetro em ΣPAR).

| Argumentos | Resultados |
|------------|------------------|
| 1 : | 1 : Σx^2 |

Programa

```
«  
 1 1  
  ΣGET  
»  
[ENTER]  
[ ] ΣX2 [STO]
```

Comentários

Inicia o programa.
Especifica C_1 duas vezes.
Extrai Σx^2 .
Fim do programa.
Coloca o programa na pilha.
Armazena o programa como ΣX^2 .

ΣY^2 (Soma dos Quadrados de y)

Calcula Σy^2 , onde os y 's são os elementos de C_2 (a coluna especificada pelo segundo parâmetro em ΣPAR).

| Argumentos | Resultados |
|------------|------------------|
| 1 : | 1 : Σy^2 |

Programa

```
«  
 2 2  
  ΣGET  
»  
[ENTER]  
[ ] ΣY2 [STO]
```

Comentários

Inicia o programa.
Especifica C_2 duas vezes.
Extrai Σy^2 .
Fim do programa.
Coloca o programa na pilha.
Armazena o programa como ΣY^2 .

ΣXY (Soma dos Produtos de x e y)

Calcule Σxy , onde os x 's e y 's são os elementos correspondentes de C_1 e C_2 (as colunas especificadas pelos primeiro e segundo parâmetros em ΣPAR).

| Argumentos | Resultados |
|------------|-----------------|
| 1 1 | 1 1 Σxy |

Programa

```
«  
1 2  
ΣGET  
»
```

ENTER

ΣXY STO

Comentários

Inicia o programa.

Especifica C_1 e C_2 .

Extrai Σxy .

Fim do programa.

Coloca o programa na pilha.

Armazena o programa como ΣXY .

Exemplo. Calcule ΣX^2 , ΣY^2 e ΣXY para os seguintes dados estatísticos:

| | |
|----|----|
| 18 | 12 |
| 4 | 7 |
| 3 | 2 |
| 11 | 1 |
| 31 | 48 |
| 20 | 17 |

Os passos gerais são os seguintes:

1. entre os dados estatísticos.
2. execute SUMS para criar a matriz de covariância ΣCOV .
3. execute $\Sigma X2$, $\Sigma Y2$ e ΣXY .
4. se ΣDAT contém duas ou mais colunas (isto é, se cada ponto de dados contém mais do que duas variáveis):
 - a. execute $COL\Sigma$ para especificar novos valores para C_1 e C_2 . Os valores são armazenados em ΣPAR .
 - b. execute $\Sigma X2$, $\Sigma Y2$ e ΣXY .

Agora, tente o exemplo dado acima.

Apague a pilha, selecione o menu STAT e apague ΣDAT .

CLEAR
 STAT
CLΣ

```
3:  
2:  
1:  
Σ+ Σ- NΣ CLΣ STOΣ RCLΣ
```

Entre dados e, então, verifique se você entrou todos os seis pontos de dados.

18,12 Σ+
 4,7 Σ+
 3,2 Σ+
 11,1 Σ+
 31,48 Σ+
 20,17 Σ+
NΣ

```
3:  
2:  
1: 6  
Σ+ Σ- NΣ CLΣ STOΣ RCLΣ
```

Elimine o número de pontos de dados.

DROP

```
3:  
2:  
1:  
Σ+ Σ- NΣ CLΣ STOΣ RCLΣ
```

Crie a matriz de covariância ΣCOV .

USER
SUMS

```
0:
2:
1:
ΣPAR ΣCOV ΣDAT ΣXY ΣY2 ΣX2
```

Calcule Σx^2 .

ΣX2

```
0:
2:
1: 1831
ΣPAR ΣCOV ΣDAT ΣXY ΣY2 ΣX2
```

Calcule Σy^2 .

ΣY2

```
0:
2: 1831
1: 2791
ΣPAR ΣCOV ΣDAT ΣXY ΣY2 ΣX2
```

Calcule Σxy .

ΣXY

```
0: 1831
2: 2791
1: 2089
ΣPAR ΣCOV ΣDAT ΣXY ΣY2 ΣX2
```

Se a matriz estatística tem mais do que duas colunas, você poderia especificar novos valores para C_1 e C_2 . Para praticar, especifique $C_1 = 1$ e $C_2 = 2$ (os valores correntes).

O comando $\text{COL}\Sigma$ está disponível no menu STAT, mas aqui é mais fácil evitar o comando e permanecer no menu USER.

1 ENTER
2 COLΣ ENTER

```
0: 1831
2: 2791
1: 2089
ΣPAR ΣCOV ΣDAT ΣXY ΣY2 ΣX2
```

Você pode agora executar $\Sigma X2$, $\Sigma Y2$ e ΣXY para o novo par de colunas C_1 e C_2 .

Não se esqueça de executar SUMS novamente sempre que adicionar ou eliminar dados da matriz estatística ΣDAT .

Mediana de Dados Estatísticos

Esta seção contém três programas:

- SORT ordena os elementos de uma lista.
- LMED calcula a mediana de uma lista classificada.
- MEDIAN utiliza SORT e MED para calcular a mediana dos dados estatísticos correntes.

SORT (Classifica uma lista)

Classifique uma lista em ordem ascendente.

| Argumentos | Resultados |
|---------------|----------------------------|
| 1 : { lista } | 1 : { lista classificada } |

Técnicas

- Classificação tipo “bolha”. Iniciando com o primeiro e segundo números na lista, SORT compara números adjacentes e move os números maiores para o fim da lista. Este processo é executado uma vez para mover o número maior para a última posição na lista, novamente para mover o segundo maior para a penúltima posição da lista e assim por diante.
- “Loops” definidos superpostos. O “loop” externo controla a posição de parada cada vez que o processo é executado; o “loop” interno executa de 1 até a posição de parada cada vez que o processo é executado.
- Estruturas de variável local superpostas. SORT contém duas estruturas de variável local, a segunda dentro do programa de definição da primeira. Este processo de superposição é feito por conveniência; é mais fácil criar a primeira variável local tão logo seu valor seja computado, portanto removendo seu valor da pilha, ao invés de computar ambos os valores e criar as duas variáveis locais simultaneamente.

- FOR ... STEP e FOR ... NEXT (“loops” definidos). SORT utiliza dois contadores: - 1 STEP decrementa o contador para o “loop” externo a cada iteração; NEXT incrementa o contador para o “loop” interno de 1 a cada iteração.

Programa

```

«
DUP SIZE 1 - 1

FOR J
  1 J
  FOR K

    K GETI → N1

    «
    GETI → N2

    «
    DROP
    IF N1 N2 >

    THEN
      KN2PUTI

      N1 PUT

    END

    »
  »
NEXT
- 1STEP
»
[ENTER]
['] SORT [STO]

```

Comentários

Inicia o programa.
Da penúltima posição para a primeira posição,
Inicia o “loop” externo com o contador j .
Da primeira para a j -ésima posição.
Inicia o “loop” interno com o contador k .
Obtenha o k -ésimo número na lista e armazene-o na variável local n_1 .

Inicia o programa de definição externo.
Obtém o próximo número na lista e armazene-o na variável local n_2 .

Inicia o programa de definição interno.
Elimina o contador.
Se os dois números estão na ordem errada,
Então faça o seguinte:
Coloque o segundo de volta na k -ésima posição.
Coloque o k -ésimo elemento de volta na próxima posição.
Fim da condição THEN.
Fim do programa de definição interno.
Fim do programa de definição externo.
Incremente k e repita o “loop” interno.
Decrementa j e repita o “loop” externo.
Fim do programa.
Coloque o programa na pilha.
Armazene o programa como SORT.

Exemplo.

Classifique a lista {8, 3, 1, 2, 5}.

```
USER  
1 8,3,1,2,5 SORT
```

```
0:  
2:  
1: { 1 2 3 5 8 }  
SORT  SWAP  SCOV  SDAT  SWY  SWZ
```

LMED (Mediana de uma Lista)

Dada uma lista classificada, calcule a mediana. Se a lista contém um número ímpar de elementos, a mediana é o valor do elemento central. Se a lista contém um número par de elementos, a mediana é o valor médio dos elementos imediatamente acima e abaixo do centro.

| Argumentos | Resultados |
|--------------------------|-------------------------------------|
| 1 { lista classificada } | 1 mediana de uma lista classificada |

Técnicas

- FLOOR e CEIL. Para um inteiro, tanto FLOOR como CEIL devolvem este inteiro; para um número não inteiro, FLOOR e CEIL devolvem inteiros sucessivos que delimitam o não inteiro.

Programa

```
«  
DUP SIZE  
1 + 2 /  
  
+ P  
  
«  
DUP  
P FLOOR GET  
  
SWAP  
P CEIL GET
```

Comentários

Inicia o programa.
O tamanho da lista.
A posição central na lista (fracionária para lista com número par de elementos).
Armazena a posição central da variável local *p*.
Inicia o programa de definição.
Faz uma cópia da lista.
Obtém o número na posição central ou abaixo.
Move a lista para o nível 1.
Obtém o número acima ou na posição central.

Programa

+ 2 /

»

»

ENTER

LMED STO

Comentários

A média dos dois números próximos da posição central ou na própria.

Fim do programa de definição.

Fim do programa.

Coloca o programa na pilha.

Armazena o programa como LMED.

Exemplo.

Calcule a média da lista que você classificou utilizando SORT.

USER

LMED

```
0:
1:
2:
3:
LMED SORT SPAR ZCOW EDAT ZNY 3
```

LMED é chamado por MEDIAN.

MEDIAN (Mediana de Dados Estatísticos)

Devolve um vetor representando as medianas das colunas dos dados estatísticos.

| Argumentos | Resultados |
|------------|-----------------------------|
| 1 : | 1 : [$x_1 x_2 \dots x_m$] |

Técnicas

- Arranjos (arrays), listas e elementos da pilha. MEDIAN extrai uma coluna de dados de ΣDAT em forma de vetor. Para transformar o vetor em uma lista, MEDIAN coloca os elementos do vetor na pilha e, então, combina-os em uma lista. Desta lista a mediana é calculada utilizando SORT e LMED.

A mediana para a m -ésima coluna é calculada primeiro, e a mediana para a primeira coluna é calculada por último, de modo que cada mediana que é calculada é movida para o nível da pilha acima das medianas previamente calculadas.

Após todas as medianas terem sido calculadas e posicionadas corretamente na pilha, elas são combinadas em um vetor.

- FOR ... NEXT (“loop” definido com contador). MEDIAN utiliza um “loop” para calcular a mediana de cada coluna. Como as medianas são calculadas em ordem inversa (última coluna primeiro), o contador é utilizado para inverter a ordem das medianas.

Programas Necessários:

- SORT (página 220) arranja uma lista em ordem ascendente.
- LMED (página 272) calcula a mediana de uma lista classificada.

Programa

```
«  
RCLΣ  
  
DUP SIZE  
  
LIST→ DROP  
  
→ NM  
«  
' ΣDAT' TRN  
  
1 M  
FOR M  
Σ -  
  
ARRY→ DROP  
  
N→LIST  
SORT  
LMED
```

Comentários

Inicia o programa.
Coloca uma cópia da matriz estatística corrente ΣDAT na pilha para cópia de segurança.
Coloca a lista {n m} na pilha, onde *n* é o número de linhas em ΣDAT e *m* é o número de colunas.
Coloca *n* e *m* na pilha. Elimina o tamanho da lista.
Cria variáveis locais para *n* e *m*.
Inicia o programa de definição.
Transpõe ΣDAT. Agora *n* é o número de colunas em ΣDAT e *m* é o número de linhas.
A primeira e última linhas.
Para cada linha, faz o seguinte:
Extrai a última linha em ΣDAT. Inicialmente esta é a *m*-ésima linha, a qual corresponde a *m*-ésima coluna na ΣDAT original.
Coloca os elementos da linha na pilha. Elimina o índice da lista {*n*}, uma vez que *n* já está armazenado em uma variável local.
Faz uma lista de *n* elementos.
Classifica a lista.
Calcula a mediana da lista.

Programa

```
JROLLD  
  
NEXT  
M 1 →LIST  
→ARRY  
  
»  
SWAP  
STOZ  
»
```

```
ENTER  
· MEDIAN STO
```

Exemplo. Calcule a mediana dos dados à página 268. (Este exemplo pressupõe que você digitou os dados.) Existem duas colunas de dados, assim MEDIAN devolverá um vetor com dois elementos.

Calcule a mediana.

```
USER  
· MEDI
```

```
3:  
2:  
1: [ 14.5 9.5 ]  
ΣDAT MEDIAN LMEQ SORT ΣPAR ΣCOV
```

As medianas são 14.5 para a primeira coluna e 9.5 para a segunda coluna.

Mudando de Diretórios

Esta seção contém dois programas:

- UP fornece um menu de diretórios-pai.
- DOWN fornece um menu de subdiretórios.

Estes programas não têm utilidade para quem sempre se lembra da estrutura completa do diretório e sabe exatamente onde está em todos os momentos. Para aqueles que ocasionalmente se confundem, estes programas são úteis.

Comentários

Move a mediana para o nível correto da pilha.

Incrementa j e repete o "loop".

Faz a lista $\{m\}$.

Combina todas as medianas em um vetor de m elementos.

Fim do programa de definição.

Move ΣDAT original para o nível 1.

Restaura ΣDAT a seu valor anterior.

Fim do programa.

Coloca o programa na pilha.

Armazena o programa como MEDIAN.

UP (Muda para um Diretório-Pai)

Cria um menu que contém os nomes do diretório-pai, seu diretório-pai, etc de volta até o diretório HOME.

| Argumentos | Resultados |
|------------|------------|
| 1 : | 1 : |

Técnicas

- Lista dos diretórios-pai. UP utiliza PATH para devolver os nomes do diretório corrente e todos os diretórios-pai.
- Sub-conjunto de uma lista. UP utiliza SUB para remover o nome do diretório corrente da lista PATH.
- Menu personalizado. UP utiliza MENU para criar um menu personalizado de diretórios-pai da lista PATH modificada.

Programa

```
«  
PATH  
1  
OVER SIZE 1 -  
SUB  
  
MENU  
»
```

```
ENTER  
UP STO
```

Comentários

Inicia o programa.
Coloca a lista de trilha na pilha.
Coloca 1 na pilha.
Coloca *size* - 1 na pilha.
Cria um sub-conjunto da lista PATH que inclui todos os nomes menos o último nome (o diretório corrente).
Cria um menu de diretórios-pai.
Fim do programa.

Coloca o programa na pilha.
Armazena o programa como UP.

Exemplo. Do diretório HOME, crie uma hierarquia de subdiretórios D1, D2 e D3; então utilize UP para mudar de D3 para D1.

Apague a pilha e mude para o diretório HOME.

HOME

```
3:
2:
1:
MEM MENU ORDER PATH HOME CRDIR
```

Crie um subdiretório D1 e mude para ele.

```
3:
2:
1:
MEM MENU ORDER PATH HOME CRDIR
```

Repita o processo para os subdiretórios D2 e D3.

```
3:
2:
1:
MEM MENU ORDER PATH HOME CRDIR
```

Apresente o menu de diretórios-pai.

```
3:
2:
1:
HOME D1 D2
```

Mude para o diretório D1.

```
3:
2:
1:
HOME D1 D2
```

DOWN (Muda para um Subdiretório)

Crie um menu que contenha o nome de todos os subdiretórios do diretório corrente.

| Argumentos | Resultados |
|----------------------------------|----------------------------------|
| <input type="button" value="1"/> | <input type="button" value="1"/> |

Técnicas

- Lista de variáveis. DOWN utiliza VARS para devolver a lista de variáveis e subdiretórios ao diretório corrente.
- Armadilha de erro. Para verificar se um nome na lista VARS é um diretório, DOWN utiliza um nome como argumento para RCL; uma vez que diretórios não podem ser recuperados para a pilha, se o nome é nome de um diretório, um erro ocorrerá, e o nome é adicionado à lista de nomes de diretórios.

Programas

```
«  
VARS  
  
+ V  
  
«  
 ( )  
  
1 0 SIZE  
FOR J  
  V J GET  
  IFERR RCL DROP  
  
THEN +  
  
END  
  
NEXT  
MENU  
  
»  
»
```

DOWN

Comentários

Inicia o programa.

Coloca na pilha uma lista dos nomes de todas as variáveis e subdiretórios.

Armazena a lista VARS em uma variável local *v*.

Inicia o programa de definição.

Coloca a lista de nomes de diretórios na pilha (inicialmente vazia).

Coloca 1 e *tamanho de v* na pilha.

Para cada nome em *v*, faz o seguinte: Obtém o nome.

Tenta recuperar o conteúdo de uma variável com aquele nome, se bem sucedido, elimina o conteúdo.

Se RCL causou um erro, o nome é necessariamente um nome de diretório, assim adiciona o nome à lista de nomes de diretórios.

Fim da condição THEN e da estrutura de programa.

Repete para o próximo nome em *v*.

Cria um menu personalizado para os nomes de diretório.

Fim do programa de definição.

Fim do programa.

Coloca o programa na pilha.

Armazena o programa como DOWN.

Exemplo. No exemplo anterior (página 277) você criou uma hierarquia de subdiretórios D1, D2 e D3 e completou o exemplo fazendo D1 o diretório corrente. Para este exemplo, mude para D2 e então D3.

Apresente o menu de subdiretórios.

DOWN

```
3:
2:
1:
02
```

Mude para baixo para D2.

D2

```
3:
2:
1:
02
```

Apresente o menu de subdiretórios.

DOWN

```
3:
2:
1:
03
```

Mude para baixo para D3.

D3

```
3:
2:
1:
03
```


Apêndices e Índices

| | |
|-------------------|---|
| Página 282 | A: Atendimento ao Usuário, Baterias e Assistência Técnica. |
| 296 | B: Notas para Usuários de Calculadoras HP RPN. |
| 302 | C: Notas para Usuários e Calculadoras Algébricas. |
| 306 | D: Diagramas dos Menus. |
| 327 | Índices das Teclas. |
| 332 | Índice por Assunto. |

A

Atendimento ao Usuário, Baterias e Assistência Técnica

Este apêndice contém informações para auxiliá-lo quando você tiver problemas com sua calculadora. Se houver dúvidas sobre como utilizar a calculadora e não encontrar um tópico apropriado no Índice (página 5) ou no Índice por Assunto (página 332), veja “Respostas a Perguntas Frequentes”, abaixo. Se não encontrar uma resposta a sua pergunta, você pode consultar nosso Serviço de Atendimento ao Usuário, utilizando o endereço ou número de telefone indicado na parte interna da contra-capas.

Se você necessitar substituir as baterias, veja página 286. Se sua calculadora parece não estar funcionando adequadamente, veja “Determinando se a Calculadora Necessita de Reparos” à página 289. Se a calculadora necessitar de assistência técnica, veja “Garantia Limitada de Um Ano” à página 291 e “Se a Calculadora Necessitar Reparos”, à página 293.

Respostas a Perguntas Frequentes

Pergunta: *A calculadora não liga quando eu pressiono . O que está errado?*

Resposta: Pode ser um problema simples que você pode resolver imediatamente, ou a calculadora pode necessitar de assistência técnica. Veja a seção “Verificando se a Calculadora Necessita de Reparos” à página 289.

Pergunta: *Como posso verificar se a calculadora está funcionando adequadamente?*

Resposta: Realize o teste contínuo, como descrito à página 290.

Pergunta: *Como posso apagar tudo da memória da calculadora?*

Resposta: Pressione e mantenha pressionadas as teclas e solte-as a seguir, como descrito em “Inicializando Toda a Memória” (“Memory Reset”) à página 20.

Pergunta: *O que significam três pontos (...) à direita da linha do visor?*

Resposta: Os três pontos, denominados *reticências*, indicam que o objeto apresentado no visor é muito longo para ser apresentado em uma linha.

Pergunta: *Como posso apresentar no visor o objeto por inteiro?*

Resposta: Utilize  **EDIT** ou  **VISIT** para devolver o objeto para a linha de comando, como descrito em "Editando Objetos Existentes", à página 173. Você pode então utilizar as teclas do cursor para apresentar no visor qualquer parte do objeto. Para cancelar a edição, pressione .

Pergunta: *O que significa "objeto"?*

Resposta: "Objeto" é um termo geral para designar quase tudo com o que você trabalha. Números, expressões, arranjos, programas e assim por diante são todos tipos de objetos. Veja "Principais Recursos e Conceitos", à página 25, para uma breve descrição de tipos de objetos ou veja o capítulo 16, "Objetos", para uma discussão detalhada dos tipos de objetos.

Pergunta: *A calculadora emite um sinal audível e apresenta no visor BAD ARGUMENT TYPE. O que está errado?*

Resposta: Os objetos na pilha não são do tipo correto para o comando que você tentou executar. Por exemplo, executar  sem o nome no nível 1 causa este erro. Utilize CATALOG para verificar o argumento correto para o comando, como descrito em "O Catálogo de Comandos", à página 31.

Pergunta: *A calculadora emite um sinal audível (bip) e apresenta no visor TOO FEW ARGUMENTS. O que está errado?*

Resposta: Existem menos objetos na pilha do que exigidos pelo comando que você tentou executar. Por exemplo, executar  com somente um número na pilha causa este erro. Utilize CATALOG para verificar os argumentos corretos para o comando, como descrito em "O Catálogo de Comandos", à página 31.

Pergunta: *A calculadora emite um sinal audível (bip) e apresenta no visor uma mensagem diferente das duas relacionadas acima. Como descobrir o que está errado?*

Resposta: Veja Apêndice A, "Mensagens" no Manual de Referência.

Pergunta: *Como desligar o sinal audível (bip)?*

Resposta: Digite 51 SF . Isto ativa o "flag" 51, o qual desliga o sinal (bip).

Pergunta: *Como posso imprimir uma cópia do visor?*

Resposta: Pressione e mantenha pressionado , pressione  e solte .

Pergunta: As teclas de **A** a **R** não funcionam. O que está errado?

Resposta: Você acidentalmente selecionou a Chave do Menu, de modo que as teclas de **A** a **R** selecionam menus, a menos que você pressione **■** antes. Para desligar a Chave do Menu pressione **■** **MENUS**.

Pergunta: Não consigo encontrar algumas variáveis que utilizei anteriormente. Para onde elas foram?

Resposta: Você pode ter utilizado as variáveis num diretório diferente. Se você não se lembra qual diretório você estava utilizando, você necessitará verificar todos os diretórios.

Pergunta: Como posso determinar qual é a memória disponível?

Resposta: Execute **MEM** **ENTER** para devolver o número de bytes disponíveis na memória.

Pergunta: Por que o cursor mudou de aparência?

Resposta: O cursor indica o modo de entrada corrente. Os modos de entrada são: Imediato (cursor vazio), Algébrico, (cursor parcialmente preenchido) ou Alfanumérico (cursor sólido). A forma do cursor indica modo de substituição (cursor quadrado) ou modo de inserção (cursor em seta). Ver "Como o Cursor Indica Modos", à página 172.

Pergunta: Eu digitei um nome (ou pressionei um tecla do menu USER), mas o nome não foi para a pilha. Por que não?

Resposta: Você entrou um nome sem aspas, o qual se refere ao conteúdo de uma variável. Para colocar um nome na pilha, pressione **'** primeiro. (Ver "Nomes Com e Sem Aspas", à página 57.)

Pergunta: Quando eu calculo a raiz cúbica de -27 , por que o resultado não é -3 ?

Resposta: Todo número tem três raízes cúbicas, duas das quais são números complexos. A HP-28S devolve uma das três raízes, denominada o valor principal. Para argumentos reais positivos o valor principal é a raiz real; para argumentos reais negativos o valor principal é uma das raízes complexas. Para calcular a b -ésima raiz real de um número real a , digite o seguinte programa.

« \rightarrow A B 'SIGN(A)*ABS(A)^INV(B)' »

Pressione **'** **RROOT** **STO** para armazenar o programa em uma variável **RROOT** (raiz real). Você pode então encontrar a raiz cúbica de -27 digitando **27** **CHS** **ENTER** **3** **ENTER** **RROOT** **ENTER**.

Pergunta: *A calculadora está mais lenta do que o usual e o anúncio  está piscando. O que está acontecendo?*

Resposta: A calculadora está em modo de monitoração de impressão. Pressione  **PRINT** **TRAC** para desligar o modo de monitoração da impressão.

Pergunta: *A impressora imprime diversas linhas rapidamente e, então, diminui a velocidade. Por que?*

Resposta: A calculadora transmite rapidamente uma certa quantidade de dados para a impressora, então diminui a velocidade de transmissão, para assegurar que a impressora não perca dados.

Pergunta: *Como posso aumentar a velocidade de impressão?*

Resposta: Se a sua impressora está conectada à rede por um adaptador, a calculadora pode enviar dados com mais velocidade sem risco de perdas. Para selecionar impressão mais rápida, digite 52 SF **ENTER**. Este comando ativa o "flag" 52, o qual controla a velocidade de impressão. Quando a impressora não estiver conectada à rede por um adaptador, digite 52 CF **ENTER** para desativar o "flag" 52 e voltar à velocidade normal de impressão.

Pergunta: *A impressora perde caracteres ou imprime caracteres ■ . O que está errado?*

Resposta: A distância ou ângulo entre a impressora e a calculadora pode estar muito grande, ou pode haver uma obstrução, bloqueando a transmissão. Veja o manual da impressora, para maiores detalhes sobre a posição da impressora e calculadora.

Pergunta: *Qual é a diferença entre STO e STORE?*

Resposta: O comando STO atribui um valor especificado a uma variável. O menu STORE contém comandos que executam *aritmética de armazenamento* utilizando o valor de uma variável como um argumento e atribuindo o valor resultante à variável.

Pergunta: *Eu esperava um resultado simbólico, mas obtive um resultado numérico. Por que?*

Resposta: Existem valores atribuídos a uma ou mais variáveis. Elimine o conteúdo das variáveis (ver "Eliminando uma Variável", à página 52) e tente novamente.

Pergunta: *Quando eu pressiono **DRAW** , o visor se apaga, o anúncio (•) pisca e então pára, mas eu não vejo pontos do gráfico no visor. Por que?*

Resposta: Os valores calculados estão fora do intervalo de traçado de gráfico corrente. Ver "Mudando a Escala de Traçado de Gráficos", à página 91.

Pergunta: *Eu processei uma variável ou uma expressão e agora a calculadora não responde. Pressionar  não tem efeito. O que aconteceu?*

Resposta: Você definiu uma variável em termos dela mesmo, criando uma definição circular e agora a calculadora está executando um "loop infinito". Para terminar o "loop", execute uma Parada de Sistema, como segue:

1. pressione e mantenha pressionada .
2. pressione .
3. solte .

Então redefina a variável para remover a definição circular.

Se você não encontrar uma resposta a sua pergunta, consulte nosso Serviço de Atendimento ao Usuário, utilizando o endereço ou o telefone indicado na contra-capa interna.

Baterias

A HP-28S é alimentada por três baterias alcalinas. Tipicamente, um conjunto novo de baterias será suficiente para aproximadamente seis meses a um ano de uso. Entretanto, a vida esperada da bateria depende de como a calculadora é utilizada.

Utilize somente baterias alcalinas tipo N. Não utilize baterias recarregáveis.

Indicador de Bateria Fraca

Quando o anúncio de bateria fraca () aparece no visor, a HP-28S pode continuar operando pelo menos 10 horas. Se a calculadora for desligada quando o anúncio aparece pela primeira vez, a Memória Contínua será preservada por aproximadamente um mês.

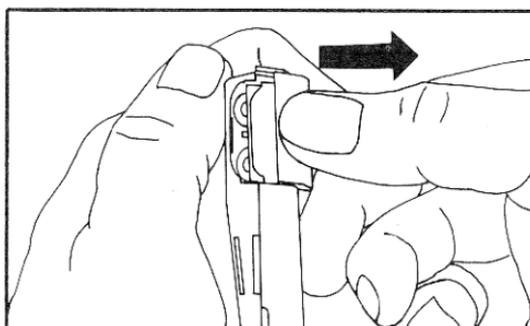
Instalação das Baterias

Se você acabou de adquirir a HP-28S e está instalando as baterias pela primeira vez, tome tanto tempo quanto desejar para completar este procedimento.

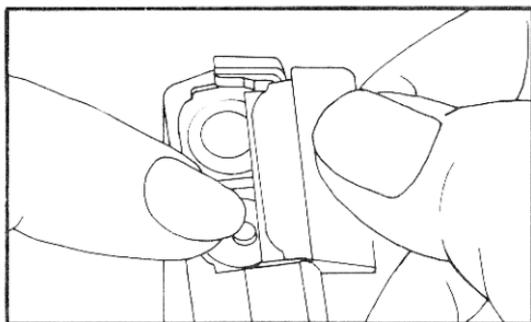
Entretanto, se você está substituindo as baterias, deve ter em mente que existe um limite de tempo para completar estes procedimentos, se deseja preservar a informação armazenada dentro da calculadora (Memória Contínua). Uma vez que o compartimento de baterias tenha sido aberto, você precisa substituir as baterias e fechá-lo dentro de um minuto, para evitar perda da Memória Contínua. Portanto, você deve ter as novas baterias à mão, antes de abrir o compartimento de baterias. Também precisa assegurar-se que a calculadora está desligada, durante todo o processo de trocas de baterias.

Para instalar as baterias:

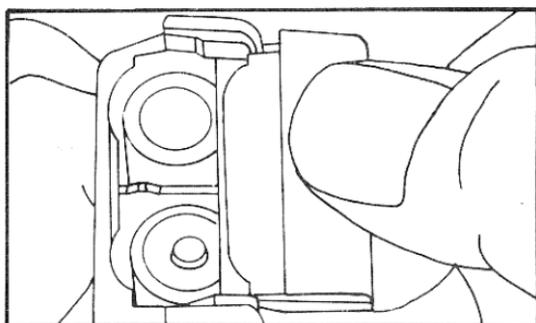
1. tenha três baterias novas do tipo N ao alcance da sua mão.
2. abra a calculadora para expor o teclado e o visor. Se você estiver substituindo as baterias, certifique-se de que a calculadora esteja desligada. *Não pressione **ON** até o final do processo de substituição das baterias. A Memória Contínua poderá ser apagada se a calculadora estiver ligada, durante a substituição das baterias.*
3. segure a calculadora com o compartimento da bateria voltado para você e deslize a porta do compartimento na direção da parte de trás da calculadora, como indicado.



4. bata com suavidade a calculadora contra a palma da sua mão, para retirar as baterias.
5. coloque as três baterias novas. Respeite a polaridade (+ e -) apresentada no diagrama da calculadora.
6. pressione as novas baterias no compartimento, utilizando a parte da porta da bateria que se estende além da placa metálica de contato. Pressione até que a placa de contato se alinhe com o sulco na caixa da calculadora.



7. deslize a placa de contato da porta para dentro do encaixe. Se necessário, use seu dedo para empurrar as baterias para dentro do compartimento, de modo que a porta deslize sobre elas até que se trave em seu lugar.



Manutenção da Calculadora

Para limpar o visor, utilize um pano levemente umedecido com água. Evite molhar a calculadora.

Não lubrifique a dobradiça.

Limites Ambientais

A fim de manter a confiabilidade do produto, você deve observar os seguintes limites de temperatura e umidade da HP-28S:

- temperatura de operação: 0° a 45°C (32° a 113°F).
- temperatura de armazenamento: -20° a 65°C (-4° a 149°F).
- umidade de operação e armazenamento: máximo de 90% de umidade-relativa a 40°C (104°F).

Determinando se a Calculadora Necessita de Reparos

Utilize estas orientações para determinar se a calculadora está funcionando adequadamente. Se a calculadora necessitar de reparos, veja “Garantia Limitada de Um Ano”, à página 291 e “Se a Calculadora Necessitar de Reparos”, à página 293.

Se nada aparece no visor quando você pressiona :

1. verifique o contraste do visor.
 - a. Pressione e mantenha pressionada .
 - b. Pressione diversas vezes.
 - c. Solte .
 - d. Se o visor permanece em branco, pressione e repita os passos a, b, c.

2. troque as baterias conforme explicado à página 286.
3. se os passos 1 e 2 não restaurarem a operação da calculadora, ela requer assistência técnica. Veja “Garantia Limitada de Um Ano”, à página 291 e “Se a Calculadora Necessitar de Reparos”, à página 293.

Se o visor é visível, mas nada acontece quando você pressiona teclas:

1. execute uma Parada de Sistema.
 - a. Pressione e mantenha pressionada **[ON]**.
 - b. Pressione **[▲]**.
 - c. Solte **[ON]**.
2. se a calculadora ainda não responde, execute uma inicialização de memória.
 - a. Pressione e mantenha pressionada **[ON]**.
 - b. Pressione e mantenha pressionada **[INS]** e **[▶]**.
 - c. Solte **[INS]** e **[▶]**.
 - d. Solte **[ON]**.
3. se os passos 1 e 2 falharem em retornar a calculadora à operação normal, ela requer assistência técnica. Ver “Garantia Limitada de Um Ano”, à página 291 e “Se a Calculadora Necessitar de Reparos”, à página 293.

O Teste Contínuo

Se a calculadora funciona, mas você acha que não está funcionando corretamente:

1. Se você tem uma impressora, ligue-a. Durante o teste a calculadora imprime números que são úteis se a calculadora necessitar reparos.
2. Inicie o teste contínuo.
 - a. Pressione e mantenha pressionada **[ON]**.
 - b. Pressione **[◀]**.
 - c. Solte **[ON]**.

O teste contínuo prossegue automaticamente. (Se o teste não prosseguir, você provavelmente pressionou por engano. Isto inicia um teste diferente, utilizado na fábrica, que requer entradas do teclado. Termine este auto-teste, executando uma Parada de Sistema, descrita no passo 4 abaixo e, então, inicie o teste contínuo correto.)

3. observe a mensagem de teste. O teste mostra linhas verticais e horizontais, o visor em branco, um padrão aleatório e então apresenta o resultado do teste.

- A mensagem `OK28S` indica que a calculadora passou no teste.
- Uma mensagem tal como `1 FAIL` indica que a calculadora não passou no teste. O número indica a natureza da falha. Quando você enviar a calculadora para reparo, inclua o número da falha e, se disponível, a fita com a impressão do teste.

Se você interromper o teste contínuo pressionando uma tecla, o teste devolve uma mensagem de falha, uma vez que ele não espera nenhuma tecla digitada. *Tal mensagem de falha não indica problema com a calculadora.*

4. interrompa o teste executando uma Parada de Sistema.

- a. Pressione e mantenha pressionada .

- b. Pressione .

- c. Solte .

5. se o teste devolve uma mensagem de falha e *you não causou a falha interrompendo o teste*, a calculadora necessita de reparos. Ver "Garantia Limitada de Um Ano" abaixo e "Se a Calculadora Necessitar de Reparos", à página 293.

Garantia Limitada de Um Ano

A HP-28S é garantida pela Tesis (com exceção das baterias e do eventual dano por elas causado), contra defeitos de material e montagem por um ano, a partir da data da compra original. Se você a vender ou presenteá-la, a garantia será automaticamente transferida ao novo proprietário e permanecerá válida pelo período original de um ano.

Durante o período de garantia repararemos, ou a nosso critério, substituiremos, sem qualquer ônus, o produto comprovadamente defeituoso, quando for enviado, com porte pago, a um dos Postos de Assistência Técnica da Tesis. (A substituição poderá ser feita com modelo mais novo ou de funcionalidade equivalente ou melhor.)

O que Não Está Coberto

As baterias e o dano por elas causado não estão cobertos por esta garantia. Consulte o fabricante das baterias sobre as garantias contra vazamento das mesmas.

Esta garantia não se aplica se o produto foi danificado por acidente ou mau uso, ou como resultado de modificação executada por terceiros que não a Tesis ou Centros de Serviços Autorizados.

Nenhum outro tipo de garantia expressa será dada. A decisão de reparo ou substituição do produto é exclusivamente da Tesis.

Os produtos são vendidos tendo por base as especificações aplicáveis por ocasião da fabricação. A Tesis não se obriga a modificar ou atualizar seus produtos, depois que estes são vendidos.

Se a Calculadora Necessitar de Reparos

Você poderá obter assistência técnica para sua calculadora sempre que ela necessitar de reparos, estando ou não no período de garantia; se fora do período de garantia haverá um custo de reparo. A Tesis Informática S/A é a única empresa autorizada a prestar assistência técnica às calculadoras Hewlett-Packard no Brasil. Há vários postos de recebimento das calculadoras distribuídos convenientemente pelas grandes capitais do país. Você pode enviar sua calculadora para reparos através deles ou diretamente à Tesis, conforme instruções a seguir.

Instruções para Remessa de Calculadoras

Se sua calculadora necessitar de reparos, envie-a acompanhada do seguinte:

- breve descrição do problema observado.
- nota de compra (ou cópia dela) se ainda estiver no período de garantia.
- nota de remessa (para pessoas jurídicas).

A calculadora deve ser acondicionada na embalagem original ou em embalagem fornecida pela ECT. Recomendamos o uso do sistema SEDEX. As despesas de remessa correm por conta do cliente e as de retorno, por conta da Tesis.

Endereços da Assistência Técnica:

São Paulo:

Tesis Informática S/A
Al. Rio Negro, 750 - Alphaville
06400 Barueri - S.Paulo
tel.: (011) 421-1444

Rio de Janeiro:

Tesis Informática S/A
Praia de Botafogo, 228 - 6 andar - salas 611/614
22230 Rio de Janeiro - RJ
Tel.: (021) 552-6422

Use o telefone (011) 421-1444 para saber o endereço do posto de recebimento mais próximo, caso não queira usar o sistema SEDEX.

Custo dos Reparos

Para agilizar os reparos feitos fora do período de garantia, a Tesis adota o sistema de custo fixo, eliminando a necessidade de orçamento e a demora por ele provocada. Entretanto, este sistema não se aplica às calculadoras danificadas por acidente ou negligência, quando os custos serão determinados pelas peças necessárias e mão-de-obra envolvida.

Depois de Expirar a Garantia

A Tesis oferece um contrato de manutenção que tem início ao findar o prazo de garantia e que cobre todos os custos de reparos efetuados durante a sua vigência, garantindo ao cliente um suporte efetivo por um custo inferior ao de um reparo isolado. Solicite uma proposta de contrato de manutenção, remetendo os dados abaixo, à Tesis Informática S/A.

Modelo da calculadora: _____

No. de série: _____

Data de compra: _____

NOME: _____

EMPRESA: _____

ENDEREÇO: _____

CIDADE: _____

ESTADO: _____ TELEFONE: _____

CEP: _____

Nota Importante:

Não se esqueça de incluir uma cópia da nota de compra caso sua calculadora esteja no período de garantia.

Garantia de Reparos

O material e a mão-de-obra utilizados no reparo, realizado fora do período de garantia, são garantidos por 90 dias, contados da data do mesmo.

Informações Sobre a Assistência Técnica Internacional

Não são todos os Postos de Assistência Técnica de calculadoras Hewlett-Packard que oferecem seus serviços a todos os modelos de calculadoras HP. Contudo, se você adquiriu sua calculadora de um revendedor autorizado, pode ter certeza de que o serviço estará disponível no país onde se deu a aquisição.

Se acontecer de você estar fora do país onde se deu a aquisição, entre em contato com o Posto de Assistência Técnica HP local para verificar se a sua calculadora pode ser nele reparada. Se não for possível, faça a gentileza de remetê-la para o "United States Calculator Service Center", cujo endereço está no verso da contra-capa. Escrevendo para esse endereço, você poderá obter a relação dos Postos de Assistência Técnica de outros países.

Todas as despesas para envio e reimportação são de sua responsabilidade.

Notas Para Usuários de Calculadoras HP RPN

Começando com a HP-35 em 1972, a Hewlett-Packard desenvolveu uma série de calculadoras científicas e para negócios, baseadas na interface da pilha RPN. Embora existam muitas diferenças na capacidade e aplicações destas diversas calculadoras, todas elas partilham uma implementação comum da interface básica de pilha, que torna muito fácil para um usuário acostumado a utilizar uma calculadora, aprender o uso de outras.

A HP-28S também utiliza uma pilha e lógica RPN, como temas centrais de sua interface de usuário. Entretanto, a pilha de quatro níveis e a estrutura de registradores fixos das calculadoras anteriores são inadequadas para suportar os múltiplos tipos de objetos e capacidades de matemática simbólica da HP-28S. Assim, enquanto a HP-28S é uma evolução natural da interface RPN "original", existem suficientes diferenças entre a HP-28S e suas predecessoras para requerer um pouco de "familiarização", se você já está acostumado com outras calculadoras RPN. Neste apêndice, salientaremos as principais diferenças.

A Pilha Dinâmica

A diferença mais acentuada na interface básica da HP-28S, comparada com calculadoras HP anteriores que usavam RPN, é o tamanho da pilha. As outras calculadoras apresentavam uma pilha fixa de quatro níveis, consistindo dos registradores X, Y, Z e T, aumentados por um único LAST X ou registrador L. Esta pilha está sempre "cheia" – mesmo quando você a "apaga" tudo o que você faz é preencher a pilha com zeros.

A HP-285 não tem tamanho fixo para sua pilha. Quando você entra novos objetos na pilha, novos níveis são dinamicamente criados à medida que são necessários. Quando você remove objetos da pilha, ela encolhe, até o ponto que esteja vazia. Assim a HP-285 pode gerar um erro `‘‘TOO FEW ARGUMENTS’’` que as calculadoras RPN anteriores da HP não podiam.

A implementação de uma pilha dinâmica, em oposição a uma fixa, gera as seguintes diferenças específicas entre a HP-285 e as calculadoras de pilha fixa:

níveis numerados. O tamanho indefinido da pilha da HP-285 torna os nomes dos níveis X Y Z T inadequados— ao invés disso, os níveis são numerados; assim o nível 1 é o análogo ao registrador X, o 2 ao Y, 3 ao Z e o 4 ao T. Os rótulos de tecla $1/x$ e x^2 foram preservados na HP-285 em razão da familiaridade — eles tornam as teclas mais visíveis do que o nome real dos seus comandos— INV e SQ respectivamente. Entretanto, o comando RPN $X < > Y$ teve seu nome alterado para SWAP (trocar), na HP-285.

manipulação da pilha. A HP-285 requer um conjunto mais genérico de comandos de manipulação da pilha do que as calculadoras de pilha fixa. $R \blacklozenge$ e $R \blacktriangledown$, por exemplo, foram substituídos por ROLL e ROLLD, respectivamente, cada um dos quais requer um argumento adicional para especificar quantos níveis da pilha devem rolar. O menu STACK (pilha) contém diversos comandos de manipulação da pilha que não existem nas calculadoras de pilha fixa.

sem cópia automática do registrador T. Em calculadoras de pilha fixa, o conteúdo do registrador T é duplicado no registrador Z, sempre que a pilha desce, isto é, quando um número é removido da pilha. Isto oferece um meio conveniente para multiplicação por constantes — você pode preencher a pilha com cópias de uma constante então, multiplicá-la por uma série de números entrando cada número, pressionando \boxed{x} então, \boxed{CLx} após você ter registrado cada resultado. Você não pode fazer isso na HP-285 — mas é fácil criar um programa desta forma.

```
«12345 * » 'MULTI' STO
```

onde 12345 representa uma constante típica. Tudo o que você tem a fazer é pressionar \boxed{USER} , entrar um número e pressionar **MULT**, entrar um novo número e pressionar **MULT** novamente e, assim por diante, para executar multiplicação por uma constante. Você pode deixar resultados sucessivos na pilha.

memória da pilha. Uma pilha dinâmica tem a vantagem de você poder utilizar tantos níveis quantos necessitar para qualquer cálculo, sem preocupar-se em perder objetos “empurrados para fora do topo da pilha”, à medida que você entra novos objetos. Também tem a desvantagem de você comprometer uma quantidade significativa de memória com objetos antigos, se você os deixar na pilha, após terminar um cálculo. Com a HP-28S, você deveria habituar-se a descartar objetos desnecessários da pilha.

DROP versus CLX. Em calculadoras de pilha fixa, CLX significa “substitua o conteúdo do registrador X com 0 e desative a elevação da pilha” (ver abaixo). Sua finalidade primordial é desfazer-se de um número antigo, antes de substituí-lo por um novo— mas você também pode usá-lo como uma maneira de entrar 0. Na HP-28S, CLX é substituído por DROP (deixa cair) que faz o que seu nome implica— deixa cair os objetos no nível 1 da pilha e o resto da pilha também desce um espaço para preenchê-lo. Não é entrado nenhum 0. Similarmente, CLEAR deixa cair todos os objetos da pilha, ao invés de substituí-los por zeros, como faz o seu análogo CLST (CLEAR STACK) das calculadoras de pilha fixa.

Desativação da Elevação da Pilha e ENTER

Certos comandos em calculadoras de pilha fixa (ENTER \uparrow , CLX, $\Sigma+$, $\Sigma-$) exibem uma característica peculiar chamada *desativação da elevação da pilha*. Isto é, após a execução de quaisquer destes comandos, o próximo número entrado na pilha substitui o conteúdo corrente do registrador X, ao invés de empurrá-lo ao registrador Y. Esta característica é inteiramente ausente na HP-28S. Novos objetos entrados na pilha *sempre* empurram os objetos anteriores da pilha para cima, para níveis mais altos.

O registrador X e ENTER, em calculadoras de pilha fixa, têm papéis duplos oriundos mais da única linha do visor das calculadoras do que da estrutura de pilha. O registrador X atua como um registrador de entrada, bem como um registrador ordinário da pilha — quando você digita um número, os dígitos são criados no registrador X, até que uma tecla não numérica termine a entrada. A tecla ENTER \uparrow é fornecida para separar-se duas entradas de números consecutivas. Mas em adição a terminar a entrada de dígitos, a tecla ENTER \uparrow também copia o conteúdo do registrador X no Y e desativa a elevação da pilha.

Na HP-285, cada um destes papéis duplos é separado — não existe desativação da elevação da pilha. Uma linha de comando completamente distinta do nível 1 (o “registrador X”) é utilizada para entrada de comandos. ENTER é utilizado *somente* para processar o conteúdo da linha de comando — ele não duplica o conteúdo do nível 1. Note, entretanto, que a tecla ENTER executará DUP (o qual copia nível 1 no nível 2), se não houver linha de comando presente. Esta característica de ENTER é oferecida parcialmente em nome da similaridade com calculadoras anteriores.

Prefixada versus Pós-fixada

Os comandos da HP-285 usam uma sintaxe estritamente pós-fixada. Todos os comandos que utilizam argumentos requerem que esses argumentos estejam presentes na pilha antes que o comando seja executado. É um desvio da convenção utilizada por calculadoras RPN anteriores, nas quais os argumentos especificando um número de registrador, o número de um “flag” e assim por diante não são entrados na pilha, mas são entrados *após* o comando em si — por exemplo, STO 25, TONE 1, CF 03, etc. Este último método tem a vantagem de economizar um nível da pilha, mas a desvantagem de requerer um formato inflexível — STO na HP-41, por exemplo, precisa sempre ser seguido por um número de registrador de dois dígitos.

Operações similares da HP-285 são mais próximas em estilo a operações *indiretas* nas calculadoras de pilha fixa, onde você pode utilizar um *registrador i* (ou qualquer registrador, no caso da HP-41) para especificar o registrador, número do “flag”, etc. endereçados por um comando. Você pode visualizar STO, RCL, etc. na HP-285, como se o nível 1 fosse utilizado como um *registrador i*. RCL, por exemplo, significa “recupere o conteúdo da variável (“registrador”) denominada no nível 1” — equivalente a RCL IND X na HP-41.

Você deve também estar ciente de que a maioria dos comandos da HP-285 remove seus argumentos da pilha. Se você executar, por exemplo, 123 ‘X’ STO, o 123 e o ‘X’ desaparecem da pilha. Sem este comportamento, a pilha seria sobrecarregada com argumentos “velhos”. Se você deseja manter o 123 na pilha, deve executar 123 DUP ‘X’ STO.

Registrador versus Variáveis

Calculadoras de pilha fixa podem trabalhar eficientemente, apenas com números reais em ponto flutuante, para os quais a estrutura de registradores da pilha e registradores numerados de memória, de sete bytes fixos, é adequada (a HP-41 introduziu um objeto primitivo de dados Alfanuméricos, limitado pelo formato de sete bytes). A HP-28S substitui registradores de dados numerados por variáveis com nomes atribuídos. Variáveis, em adição a possuírem uma estrutura flexível de forma que podem acomodar diferentes tipos de objetos, têm nomes que podem auxiliá-lo a lembrar-se de seu conteúdo mais prontamente do que números de registrador.

Se você deseja duplicar registradores numerados na HP-28S, pode usar um vetor:

```
( 50 ) 0 CON 'REG' STO
```

cria um vetor com 50 elementos inicializados com 0;

```
« 'REG' SWAP GET » 'NRCL' STO
```

cria um programa NRCL que recupera o n -ésimo elemento do vetor, onde n é um número no nível 1;

```
« 'REG' SWAP ROT PUT » 'NSTO' STO
```

cria o programa análogo para armazenar NSTO.

P

LASTX versus LAST

O comando LASTX, em calculadoras de pilha fixa, devolve o conteúdo do registrador LASTX (ou L), o qual contém o último valor utilizado do registrador X. Este conceito é generalizado na HP-28S com o comando LAST, que devolve os últimos um, dois, ou três argumentos tomados da pilha por um comando (nenhum comando utiliza mais do que três argumentos). Assim $1\ 2 + \text{LASTX}$ devolve 3 e 2 à pilha em uma calculadora de pilha fixa, mas $1\ 2 + \text{LAST}$ devolve 3, 1 e 2 à pilha na HP-28S.

Embora o LAST da HP-28S seja mais flexível do que seu predecessor LASTX, você deve ter em mente que mais comandos da HP-28S utilizam argumentos da pilha do que seus correspondentes nas calculadoras de pilha fixa anteriores. Isto significa que argumentos de LAST são atualizados mais freqüentemente, e até mesmo comandos, tais como DROP ou ROLL, irão substituir os argumentos de LAST.

Lembre-se também de que UNDO pode substituir a pilha por completo, o que para simples recuperação de erro pode ser preferível a LAST.

Notas para Usuários de Calculadoras Algébricas

Muitas calculadoras, incluindo a grande maioria das calculadoras “quatro operações” simples, utilizam variações da interface de calculadora *algébrica*. O nome é derivado da característica que a seqüência de teclas, utilizadas para operações simples, é estreitamente próxima da maneira pela qual os cálculos são especificados em expressões algébricas, “no papel”. Isto é, para processar $1 + 2 - 3$, você pressiona $\boxed{1} \boxed{+} \boxed{2} \boxed{-} \boxed{3} \boxed{=}$.

Esta interface funciona muito bem para expressões contendo números e *operadores* — funções tais como $+$, $-$, \times e $/$ que são escritas em notação infixa entre seus argumentos. Calculadoras mais sofisticadas permitem que você entre parênteses para especificar a precedência (a ordem das operações). Entretanto, a introdução de funções de prefixo, tais como SIN, LOG, etc, leva a duas variações diferentes:

- calculadoras algébricas comuns utilizam uma combinação de estilos — operadores infixos permanecem infixos, mas funções prefixadas são entradas em um estilo pós-fixado (como em calculadoras RPN). Por exemplo, $1 + \text{SIN}(23)$ é entrado como $\boxed{1} \boxed{+} \boxed{2} \boxed{3} \boxed{\text{SIN}} \boxed{=}$. Este método tem a vantagem de poder mostrar resultados intermediários e preservar processamentos tecla-a-tecla de funções de prefixo (isto é, sem parênteses), mas tem a desvantagem de perder a correspondência com notação matemática comum que é a vantagem principal da interface algébrica.

- calculadoras de “entrada direta de fórmulas” e computadores com linguagem BASIC, que têm um modo de execução imediata, permitem que você digite uma expressão completa em sua forma algébrica comum e, então, compute o resultado, quando você pressiona uma tecla de terminação (com rótulos diversos como `ENTER`, `ENDLINE`, `RETURN` etc). Este método tem a vantagem de preservar a correspondência entre expressões escritas e seqüência de teclas, mas usualmente a desvantagem de não fornecer resultados intermediários. (O modo CALC da HP-71B é uma exceção.) Você necessariamente tem que conhecer a forma completa de uma expressão antes que comece a entrá-la — é difícil “ir trabalhando de acordo com os resultados intermediários”, variando os cálculos de acordo com os mesmos.

Familiarizando-se com a HP-28S

A lógica operacional da HP-28S é baseada na lógica matemática conhecida como “Notação Polonesa”, desenvolvida pelo matemático polonês Jan Lukasiweicz (1878-1956). A notação convencional algébrica coloca operadores aritméticos *entre* os números ou variáveis relevantes, quando processa expressões algébricas. A notação de Lukasiweicz especifica os operadores *antes* das variáveis. Uma variação desta lógica especifica os operadores *após* as variáveis — isto é denominado “Notação Polonesa Reversa” (Reverse Polish Notation) ou “RPN” abreviadamente.

A idéia básica de RPN é que você entra números ou outros objetos na calculadora em primeiro lugar, e, então, executa um comando que atua nestes elementos (denominados “argumentos”). A “pilha” é somente a seqüência de objetos esperando para ser utilizada. A maior parte dos comandos devolve seus resultados à pilha, onde podem então ser utilizados como argumentos para operações subseqüentes.

A HP-28S utiliza uma interface de pilha RPN porque ela fornece a flexibilidade necessária para suportar a ampla variedade de capacidade matemáticas da HP-28S de uma maneira uniforme. Todas as operações da calculadora, incluindo aquelas que não podem ser expressas como expressões algébricas, são executadas da mesma maneira — argumentos da pilha, resultados à pilha.

Não obstante, a utilização da pilha RPN para simples aritmética é, provavelmente, o maior obstáculo para usuários de calculadoras algébricas, tentando aprender a utilizar calculadoras RPN. RPN é muito eficiente, mas requer que você mentalmente rearranje uma expressão, antes que possa calcular resultados. A capacidade da HP-28S de interpretar expressões algébricas sem tradução, deve tornar a transição do uso de calculadoras algébricas mais direta e simples do que foi possível em calculadoras RPN anteriores. O visor de quatro linhas também pode ajudar a tirar uma parte do mistério da pilha, mostrando-lhe os conteúdos de até quatro níveis de cada vez.

Para o propósito de processar-se expressões algébricas, a HP-28S é essencialmente uma calculadora de "entrada direta de fórmulas". Isto é, para processar uma expressão algébrica, tudo que você tem a fazer é precedê-la com um \square , digitar a expressão na sua forma algébrica, incluindo operadores infixos, funções prefixadas e parênteses e, então, pressionar \square EVAL para ver o resultado. Você pode utilizar esse método até mesmo para simples cálculos aritméticos:

\square 1 \square + \square 2 \square - \square 3 \square EVAL devolve 0.

Exceto pelo \square precedente, estas são as mesmas teclas que você utilizaria em uma calculadora algébrica simples, onde você substituiria \square EVAL por \square =.



Nota

Não confunda a tecla \square = da HP-28S com aquela encontrada em calculadoras algébrica — na HP-28S, \square = é utilizada para o único propósito de criar-se equações algébricas (descritas em "ÁLGEBRA").

Quando você utilizar a HP-28S como uma “calculadora de entrada direta de fórmulas” cada resultado que você computar é mantido na pilha, a qual assume o papel de uma “pilha histórica”. Isto permite que você salve resultados antigos indefinidamente, para reutilização mais tarde. Também permite que você quebre cálculos extensos em menores, mantendo cada resultado parcial na pilha e então combinando os resultados quando eles estiverem todos disponíveis. (Quando levado ao extremo, esta é a essência de aritmética em RPN.) A pilha histórica é muito mais fácil de ser utilizada e muito mais poderosa do que a única função “resultado” disponível em calculadoras algébricas ou em BASIC.

Uma característica chave da HP-28S é que você não necessita preocupar-se com a lógica RPN, se é melhor ou pior do que lógica algébrica. Você pode escolher a lógica que é mais adequada ao problema em questão e combinar expressões algébricas com manipulações RPN.

Diagramas dos Menus

Este apêndice mostra os comandos em cada menu da HP-28S. Os menus estão relacionados em ordem alfabética, de **ÁLGEBRA** até **TRIG**. Para informação detalhada sobre um menu, veja o Dicionário no Manual de Referência. O Dicionário descreve todos os menus, relacionados em ordem alfabética. Para informações detalhadas sobre um comando em particular, veja o Índice de Operações à página 381 do Manual de Referência. O Índice de Operações relaciona todos os comandos em ordem alfabética e dá uma referência de página para a descrição do comando no Dicionário.

Este apêndice não inclui os menus das operações interativas oferecidas por **CATALOG**, **FORM**, o **Solver** e **UNITS**.

- **CATALOG** é descrito no capítulo 22 e demonstrado à página 31.
- **FORM** é descrito em “Utilizando **FORM**” à página 112. Para detalhes ver “**ALGEBRA (FORM)**” no Manual de Referência.
- O **Solver** é descrito no capítulo 8, “O **Solver**”. Para detalhes, ver “**SOLVE**” no Manual de Referência.
- **UNITS** é descrito em “O Catálogo **UNITS**” à página 141. Para detalhes, ver “**UNITS**” no Manual de Referência.

Para cada menu neste apêndice, os comandos são agrupados pelas linhas que aparecem no visor a cada vez. Pressionando **[NEXT]**, muda para a próxima linha e, pressionando **[PREV]**, muda para a linha anterior.

A coluna denominada “Comando” é o nome que aparece no visor. A coluna denominada “Descrição” é uma breve descrição do comando ou seu nome completo. A coluna denominada “Página” refere-se a um exemplo, descrição ou menção do comando neste manual. Para comandos sem referência de página, ver o Índice de Operações no Manual de Referência.

ALGEBRA

| | Comando | Descrição | Página |
|-------------------------------|----------------|---------------------------|---------------|
| Linha 1 | COLCT | Agrupa termos | 111 |
| | EXPAN | Desenvolve produtos | 111 |
| | SIZE | Tamanho | |
| | FORM | Forma expressão algébrica | 112 |
| | OBSUB | Substitui objeto | |
| | EXSUB | Substitui expressão | |
| <input type="checkbox"/> NEXT | | | |
| Linha 2 | TAYLR | Série de Taylor | |
| | ISOL | Isola | 112 |
| | QUAD | Forma quadrática | |
| | SHOW | Mostra variável | |
| | OBGET | Obtém objeto | |
| | EXGET | Obtém expressão | |

ARRAY

| | Comando | Descrição | Página |
|-------------------------------|---------|----------------------------|--------|
| Linha 1 | →ARRY | Pilha para array | 275 |
| | ARRY→ | Array para pilha | 274 |
| | PUT | Coloca elemento | |
| | GET | Obtém elemento | |
| | PUTI | Coloca e incrementa índice | |
| | GETI | Obtém e incrementa índice | |
| <input type="checkbox"/> NEXT | | | |
| Linha 2 | SIZE | Tamanho | 274 |
| | RDM | Redimensiona | |
| | TRN | Transpõe | 264 |
| | CON | Array constante | |
| | IDN | Matriz identidade | |
| | RSD | Residual | |
| <input type="checkbox"/> NEXT | | | |
| Linha 3 | CROSS | Produto vetorial | 126 |
| | DOT | Produto escalar | 126 |
| | DET | Determinante | 128 |
| | ABS | Valor absoluto | |
| | RHRM | Norma de linha | |
| | CHRM | Norma de coluna | |
| <input type="checkbox"/> NEXT | | | |
| Linha 4 | R→C | Real para complexo | |
| | C→R | Complexo para real | |
| | RE | Parte real | |
| | IM | Parte imaginária | |
| | CONJ | Conjugado | |
| | NEG | Negativo | |

BINARY

| | Comando | Descrição | Página |
|-------------------------------------|---------|-----------------------------------|--------|
| Linha 1 | DEC | Modo decimal | 140 |
| | HEX | Modo hexadecimal | 139 |
| | OCT | Modo octal | 140 |
| | BIN | Modo binário | 140 |
| | STWS | Armazena tamanho de palavra | 139 |
| | RCWS | Recupera tamanho de palavra | |
| <input type="button" value="NEXT"/> | | | |
| Linha 2 | RL | Girar à esquerda | |
| | RR | Girar à direita | |
| | RLB | Girar byte à esquerda | |
| | RRB | Girar byte à direita | |
| | R→B | Real para binário | 261 |
| | B→R | Binário para real | |
| <input type="button" value="NEXT"/> | | | |
| Linha 3 | SL | Deslocar à esquerda | |
| | SR | Deslocar à direita | |
| | SLB | Deslocar byte à esquerda | |
| | SRB | Deslocar byte à direita | |
| | ASR | Deslocamento aritmético à direita | |
| <input type="button" value="NEXT"/> | | | |
| Linha 4 | AND | And | |
| | OR | Or | |
| | XOR | Or exclusivo | |
| | NOT | Not | |

COMPLEX

| | Comando | Descrição | Página |
|-------------------------------------|---------|-----------------------|--------|
| Linha 1 | R→C | Real para complexo | 83 |
| | C→R | Complexo para real | 83 |
| | RE | Parte real | 83 |
| | IM | Parte imaginária | 84 |
| | CONJ | Conjugado | 84 |
| | SIGN | Sinal | 84 |
| <input type="button" value="NEXT"/> | | | |
| Linha 2 | R→P | Retangular para polar | 86 |
| | P→R | Polar para retangular | 85 |
| | ABS | Valor absoluto | 85 |
| | NEG | Negativo | 85 |
| | ARG | Argumento | 85 |

LIST

| | Comando | Descrição | Página |
|-------------------------------|---------|----------------------------|--------|
| Linha 1 | →LIST | Pilha para lista | 181 |
| | LIST→ | Lista para pilha | 181 |
| | PUT | Coloca elemento | 271 |
| | GET | Obtém elemento | 237 |
| | PUTI | Coloca e incrementa índice | 271 |
| | GETI | Obtém e incrementa índice | 271 |
| <input type="checkbox"/> NEXT | | | |
| Linha 2 | POS | Posição | 237 |
| | SUB | Subconjunto | 276 |
| | SIZE | Tamanho | 271 |

LOGS

| | Comando | Descrição | Página |
|-------------------------------|----------------|------------------------------|---------------|
| Linha 1 | LOG | Logaritmo comum | 78 |
| | ALOG | Antilogaritmo comum | 78 |
| | LN | Logaritmo natural | 78 |
| | EXP | Exponencial | 78 |
| | LNP1 | Logaritmo natural de $1 + x$ | 78 |
| | EXPM | Exponencial menos 1 | 78 |
| <input type="checkbox"/> NEXT | | | |
| Linha 2 | SINH | Seno hiperbólico | 78 |
| | ASINH | Arco seno hiperbólico | 78 |
| | COSH | Cosseno hiperbólico | 78 |
| | ACOSH | Arco cosseno hiperbólico | 78 |
| | TANH | Tangente hiperbólica | 78 |
| | ATANH | Arco tangente hiperbólica | 78 |

MEMORY

| | Comando | Descrição | Página |
|-------------------------------|---------|---------------------------------|--------|
| Linha 1 | MEM | Memória disponível | 188 |
| | MENU | Cria menu personalizado | 195 |
| | ORDER | Ordena variáveis | 184 |
| | PATH | Trilha corrente | 67 |
| | HOME | Seleciona diretório HOME | 71 |
| | CRDIR | Cria diretório | 66 |
| <input type="checkbox"/> NEXT | | | |
| Linha 2 | VARs | Variáveis no diretório corrente | 184 |
| | CLUSR | Apaga diretório corrente | 184 |

MODE

| | Comando | Descrição | Página |
|----------------|--------------|---|--------|
| Linha 1 | STD | Formato padrão de apresentação de números | 38 |
| | FIX | Formato fixo de apresentação de números | 38 |
| | SCI | Formato científico de apresentação de números | 38 |
| | ENG | Formato de engenharia de apresentação de números | 38 |
| | DEG | Modo angular graus | 74 |
| | RAD | Modo angular radianos | 74 |
| NEXT | | | |
| Linha 2 | CMD | Ativa ou desativa COMMAND | 210 |
| | UNDO | Ativa ou desativa UNDO | 211 |
| | LAST | Ativa ou desativa LAST | 211 |
| | ML | Ativa ou desativa múltiplas linhas | 208 |
| | RDX, PRMD | Ativa ou desativa RDX, Imprime e apresenta modos | 37 |

PLOT

| | Comando | Descrição | Página |
|-------------------------------------|---------------|---|--------|
| Linha 1 | STEQ | Armazena equação | 90 |
| | RCEQ | Recupera equação | |
| | PMIN | Traça gráficos dos mínimos | 95 |
| | PMAX | Traça gráficos dos máximos | 95 |
| | INDEP | Independente | |
| | DRAW | Traça o gráfico | 90 |
| <input type="button" value="NEXT"/> | | | |
| Linha 2 | PPAR | Recupera parâmetros de traçado de gráfico | 90 |
| | RES | Resolução | |
| | AXES | Eixos | |
| | CENTR | Centro | 94 |
| | *W | Multiplica largura | |
| <input type="button" value="NEXT"/> | | | |
| Linha 3 | *H | Multiplica altura | 93 |
| | STOΣ | Armazena sigma | |
| | RCLΣ | Recupera sigma | |
| | COLΣ | Colunas sigma | |
| | SCLΣ | Escala para sigma | |
| <input type="button" value="NEXT"/> | | | |
| Linha 4 | DRWΣ | Traça gráfico estatístico | |
| | CLLCD | Apaga visor | |
| | DGTIZ | Digitaliza | |
| | PIXEL | Pixel | |
| | DRAX | Traça eixos | |
| | CLMF | Apaga mensagem de flag | |
| PRLCD | Imprime visor | | |

PRINT

| | Comando | Descrição | Página |
|-------------------------------------|---------|------------------------------------|--------|
| Linha 1 | PR1 | Imprime nível 1 | 151 |
| | PRST | Imprime pilha | 152 |
| | PRVAR | Imprime variável | 152 |
| | PRLCD | Imprime visor | 149 |
| | CR | Carro à direita | |
| | TRAC | Ativa ou desativa modo monitoração | 150 |
| <input type="button" value="NEXT"/> | | | |
| Linha 2 | PRSTC | Imprime pilha (compacto) | |
| | PRUSR | Imprime variáveis do usuário | |
| | PRMD | Imprime modos | |

PROGRAM BRANCH

| | Comando | Descrição | Página |
|----------------|----------------|-------------------------------|---------------|
| Linha 1 | IF | Inicia condição | 226 |
| | IFERR | Inicia condição IF ERROR | 227 |
| | THEN | Inicia condição THEN | 226 |
| | ELSE | Inicia condição ELSE | 226 |
| | END | Termina estrutura de programa | 226 |
| | NEXT | | |
| Linha 2 | START | Inicia loop definido | 228 |
| | FOR | Inicia loop definido | 229 |
| | NEXT | Termina loop definido | 228 |
| | STEP | Termina loop definido | 230 |
| | IFT | Comando If-Then | 227 |
| | IFTE | Função If-Then-Else | 226 |
| | NEXT | | |
| Linha 3 | DO | Define loop definido | 231 |
| | UNTI | Define loop indefinido | 231 |
| | END | Termina estrutura de programa | 231 |
| | WHIL | Define loop definido | 232 |
| | REPER | Define loop indefinido | 232 |
| | END | Termina estrutura de programa | 232 |

PROGRAM CONTROL

| | Comando | Descrição | Página |
|-------------------------------|---------|------------------------------------|--------|
| Linha 1 | SST | Passo único | 250 |
| | HALT | Suspende programa | 234 |
| | ABORT | Aborta programa | |
| | KILL | Aborta programas suspensos | 250 |
| | WAIT | Pausa no programa | 234 |
| | KEY | Devolve cadeia representando tecla | 234 |
| <input type="checkbox"/> NEXT | | | |
| Linha 2 | BEEP | Emite sinal audível (bip) | 234 |
| | CLLCD | Apaga visor | 234 |
| | DISP | Apresenta no visor | 234 |
| | CLMF | Apaga mensagem de flag | 234 |
| | ERRN | Número de erro | |
| | ERRM | Mensagem de erro | |

PROGRAM TEST

| | Comando | Descrição | Página |
|----------------|---------|---------------------------|--------|
| Linha 1 | SF | Ativa flag | 205 |
| | CF | Desativa flag | 205 |
| | FS? | Flag ativado? | 225 |
| | FC? | Flag desativado? | |
| | FS?C | Flag ativado? Desative | |
| | FC?C | Flag desativado? Desative | |
| NEXT | | | |
| Linha 2 | AND | And | |
| | OR | Or | |
| | XOR | Or exclusivo | |
| | NOT | Not | 232 |
| | SAME | Mesmo | 231 |
| | == | Igual | 222 |
| NEXT | | | |
| Linha 3 | STOF | Armazena flags | 156 |
| | RCLF | Recupera flags | 156 |
| | TYPE | Tipo | 232 |
| | | | |

REAL

| | Comando | Descrição | Página |
|----------------|---------|------------------------------------|--------|
| Linha 1 | NEG | Negativo | 78 |
| | FACT | Fatorial (gama) | 78 |
| | RAND | Número aleatório | 78 |
| | RDZ | Gera semente para número aleatório | 78 |
| | MAXR | Máximo real | 79 |
| | MINR | Mínimo real | 79 |
| NEXT | | | |
| Linha 2 | ABS | Valor absoluto | |
| | SIGN | Sinal | |
| | MANT | Mantissa | |
| | XPON | Expoente | |
| NEXT | | | |
| Linha 3 | IP | Parte inteira | |
| | FP | Parte fracionária | |
| | FLOOR | Piso | 272 |
| | CEIL | Teto | 272 |
| | RND | Arredonda | |
| NEXT | | | |
| Linha 4 | MAX | Máximo | |
| | MIN | Mínimo | |
| | MOD | Módulo | |
| | %T | Porcentagem do total | |

SOLVE

| | Comando | Descrição | Página |
|-------------------------------------|----------------|------------------------------|---------------|
| Linha 1 | STEQ | Armazena equação | 64 |
| | RCEQ | Recupera equação | |
| | SOLVR | Menu de variáveis do Solver | 102 |
| | ISOL | Isola | 110 |
| | QUAD | Forma quadrática | 108 |
| | SHOW | Mostra variável | |
| <input type="button" value="NEXT"/> | | | |
| Linha 2 | ROOT | Encontra raízes (Rootfinder) | |

STACK

| | Comando | Descrição | Página |
|-------------------------------|---------|--------------------------------------|--------|
| Linha 1 | DUP | Duplica | 178 |
| | OVER | Para cima | 178 |
| | DUP2 | Duplica dois objetos | 178 |
| | DROP2 | Elimina dois objetos | 179 |
| | ROT | Rola | 178 |
| | LIST→ | Lista para pilha | 181 |
| <input type="checkbox"/> NEXT | | | |
| Linha 2 | ROLLD | Rola para baixo | 178 |
| | PICK | Copia elemento do nível n | 178 |
| | DUPN | Duplica n objetos | 178 |
| | DROPN | Elimina n objetos | 179 |
| | DEPTH | Determina número de objetos na pilha | 181 |
| | →LIST | Pilha para lista | 181 |

STAT

| | Comando | Descrição | Página |
|----------------|--------------|-------------------------------------|--------|
| Linha 1 | $\Sigma+$ | Sigma + | 132 |
| | $\Sigma-$ | Sigma - | 133 |
| | $N\Sigma$ | Sigma N | 134 |
| | $CL\Sigma$ | Apaga sigma | 132 |
| | $STO\Sigma$ | Armazena sigma | 275 |
| | $RCL\Sigma$ | Recupera sigma | 264 |
| NEXT | | | |
| Linha 2 | TOT | Total | |
| | MEAN | Média | 134 |
| | SDEV | Desvio padrão | 135 |
| | VAR | Variância | 135 |
| | MAX Σ | Máximo de sigma | |
| | MIN Σ | Mínimo de sigma | |
| NEXT | | | |
| Linha 3 | COL Σ | Colunas de sigma | 136 |
| | CORR | Correlação | 136 |
| | COV | Covariância | 136 |
| | LR | Regressão linear | 137 |
| | PREDV | Valor projetado | 137 |
| | | | |
| NEXT | | | |
| Linha 4 | UTPC | Distribuição chi-quadrado superior | |
| | UTPF | Distribuição f de Snedecor superior | |
| | UTPN | Distribuição normal superior | |
| | UTPT | Distribuição t de Student superior | |
| | COMB | Combinações | |
| | PERM | Permutações | |

STORE

| | Comando | Descrição | Página |
|-------------------------------------|---------|----------------------------|--------|
| Linha 1 | STO+ | Armazena somando | |
| | STO- | Armazena subtraindo | |
| | STO* | Armazena multiplicando | |
| | STO/ | Armazena dividindo | |
| | SNEG | Armazena tornando negativo | |
| | SINV | Armazena invertendo | |
| <input type="button" value="NEXT"/> | | | |
| Linha 2 | SCONJ | Armazena conjugado | |

STRING

| | Comando | Descrição | Página |
|-------------------------------|---------|--------------------|--------|
| Linha 1 | →STR | Objeto para cadeia | 258 |
| | STR→ | Cadeia para objeto | 175 |
| | CHR | Caractere | 156 |
| | NUM | Caractere numérico | 156 |
| | →LCD | Cadeia para visor | 157 |
| | LCD→ | Visor para cadeia | 157 |
| <input type="checkbox"/> NEXT | | | |
| Linha 2 | POS | Posição | |
| | SUB | Subconjunto | |
| | SIZE | Tamanho | 258 |
| | DISP | Apresenta no visor | 156 |

TRIG

| | Comando | Descrição | Página |
|----------------|---------|-------------------------------------|--------|
| Linha 1 | SIN | Seno | 74 |
| | ASIN | Arco Seno | 74 |
| | COS | Cosseno | 74 |
| | ACOS | Arco cosseno | 74 |
| | TAN | Tangente | 74 |
| | ATAN | Arco tangente | 74 |
| NEXT | | | |
| Linha 2 | P→R | Polar para retangular | 76 |
| | R→P | Retangular para polar | 76 |
| | R→C | Real para complexo | 76 |
| | C→R | Complexo para real | 76 |
| | ARG | Argumento | 76 |
| NEXT | | | |
| Linha 3 | →HMS | Decimal para horas-minutos-segundos | 76 |
| | HMS→ | Horas-minutos-segundos para decimal | 76 |
| | HMS+ | Horas-minutos-segundos + | 76 |
| | HMS- | Horas-minutos-segundos - | 76 |
| | D→R | Graus para radianos | 77 |
| | R→D | Radianos para graus | 77 |

Índice das Teclas

Este índice descreve as ações das teclas no teclado da calculadora. Em primeiro lugar está o índice alfabético das teclas no teclado da esquerda, seguido por um índice alfabético das teclas no teclado da direita. Por último, está um índice das teclas no menu do cursor (os rótulos brancos acima da coluna superior do teclado da direita).

Este índice inclui teclas prefixadas tais como   e  . Não estão incluídas teclas de caracteres tais como  até  e  até , as quais sempre escrevem um caractere na linha de comando. (Outras teclas de caracteres incluem delimitadores tal como , operadores tal como , e constantes simbólicas tal como  . Esses caracteres têm significados especiais para calculadora, mas suas teclas são simplesmente teclas de caractere.) Se você não encontrar uma tecla relacionada neste índice, ela é uma tecla de caractere.

Para cada tecla, há uma breve descrição de sua ação e uma referência de página. Se a tecla não estiver mencionada neste manual ou se desejar informações adicionais sobre qualquer tecla, veja o Índice de Operações à página 381 no Manual de Referência.

Teclado da Esquerda

| Tecla | Descrição | Página |
|---|--|--------|
|  ALGEBRA | Seleciona o menu ALGEBRA. | 110 |
|  ARRAY | Seleciona o menu ARRAY. | 124 |
|  BINARY | Seleciona o menu BINARY. | 138 |
|  BRANCH | Seleciona o menu PROGRAM BRANCH. | 222 |
|  CATALOG | Inicia o comando catálogo. | 196 |
|  COMPLX | Seleciona menu COMPLEX. | 83 |
|  CONTRL | Seleciona o menu PROGRAM CONTROL. | 234 |
| LC | Ativa e desativa modo letras minúsculas. | 168 |
|  LIST | Seleciona o menu LIST. | 102 |
|  MENUS | Liga e desliga chave do menu. | 192 |
|  MEMORY | Seleciona o menu MEMORY. | 182 |
|  PRINT | Seleciona o menu PRINT. | 149 |
|  REAL | Seleciona o menu REAL. | 78 |
|  STACK | Seleciona o menu STACK. | 176 |
|  STAT | Seleciona o menu STAT. | 131 |
|  STORE | Seleciona o menu STORE. | 191 |
|  STRING | Seleciona o menu STRING. | 156 |
|  TEST | Seleciona o menu PROGRAM TEST. | 225 |
|  UNITS | Seleciona o catálogo UNITS. | 141 |
|  α | Muda modo de entrada. | 171 |

Teclado da Direita

| Tecla | Descrição | Página |
|---|--|--------|
| <input type="button" value="ATTN"/> (<input type="button" value="ON"/>) | Aborta execução do programa; apaga a linha de comando; sai dos catálogos, FORM, apresentação de gráficos. | 216 |
| <input type="button" value="CHS"/> | Muda o sinal do número na linha de comando ou executa NEG. | 168 |
| <input type="button" value="CLEAR"/> | Apaga a pilha. | 179 |
| <input type="button" value="COMMAND"/> | Move um elemento da pilha de comando para a linha de comando. | 174 |
| <input type="button" value="CONT"/> | Continua um programa interrompido. | 235 |
| <input type="button" value="CONVERT"/> | Executa uma conversão de unidade. | 143 |
| <input type="button" value="CUSTOM"/> | Seleciona o último menu personalizado apresentado no visor. | 192 |
| <input type="button" value="d/dx"/> | Derivada. | 117 |
| <input type="button" value="DROP"/> | Elimina um objeto da pilha. | 179 |
| <input type="button" value="EDIT"/> | Copia o objeto no nível 1 para a linha de comando para edição. | 173 |
| <input type="button" value="EEX"/> | Entra expoenta na linha de comando. | 168 |
| <input type="button" value="ENTER"/> | Analisa sintaxe e processa linha de comando. | 173 |
| <input type="button" value="EVAL"/> | Processa um objeto. | 118 |
| <input type="button" value="LAST"/> | Devolve os últimos argumentos. | 179 |
| <input type="button" value="LOGS"/> | Seleciona o menu LOGS. | 77 |
| <input type="button" value="MODE"/> | Seleciona o menu MODE. | 36 |
| <input type="button" value="NEXT"/> | Apresenta próxima linha dos rótulos de menu. | 192 |
| <input type="button" value="ON"/> (<input type="button" value="ATTN"/>) | Liga a calculadora; aborta a execução de programas; apaga linha de comando; sai de catálogos, FORM, traçado de gráficos. | 216 |
| <input type="button" value="OFF"/> | Desliga a calculadora. | 20 |
| <input type="button" value="PLOT"/> | Seleciona o menu PLOT. | 89 |
| <input type="button" value="PREV"/> | Apresenta no visor a linha anterior de rótulos de menu. | 192 |

| Tecla | Descrição | Página |
|---|---|--------|
|  PURGE | Elimina uma ou mais variáveis. | 183 |
|  RCL | Recupera o conteúdo de uma variável, sem processá-la. | 183 |
|  ROLL | Move o objeto no nível $n+1$ para o nível 1. | 178 |
| SOLV | Seleciona o menu SOLVE. | 99 |
| STO | Armazena um objeto em uma variável. | 183 |
|  SWAP | Troca os objetos nos níveis 1 e 2. | 178 |
| TRIG | Seleciona o menu TRIG. | 74 |
|  UNDO | Substitui o conteúdo da pilha. | 180 |
| USER | Seleciona o menu USER. | 49 |
|  VIEW ▲ | Move a janela do visor uma linha para cima. | 177 |
|  VIEW ▼ | Move a janela do visor uma linha para baixo. | 177 |
|  VISIT | Copia um objeto para a linha de comando para edição. | 173 |
|  x^2 | Eleva um número ou matriz ao quadrado. | 40 |
|  $1/x$ | Calcula o inverso. | 40 |
| + | Adiciona dois objetos. | 41 |
| - | Subtrai dois objetos. | 41 |
| x | Multiplica dois objetos. | 41 |
| ÷ | Divide dois objetos. | 42 |
|  % | Porcentagem. | 43 |
|  %CH | Variação percentual. | 43 |
|  ^ | Eleva um número a uma potência. | 42 |
|  \sqrt{x} | Calcula a raiz quadrada. | 40 |
|  | Integral definida ou indefinida. | 120 |
|  | Tecla de prefixo. | 29 |
|  | Seleciona o menu do cursor ou restaura o último menu. | 168 |
|  | Retrocesso. | 168 |
|  NUM | Força um resultado numérico. | 75 |

Menu do Cursor

O menu do cursor é rotulado em branco sobre as teclas do menu (a fileira superior do teclado da direita). O menu do cursor está ativado quando a linha de comando está presente e não há rótulos do menu no visor. Para selecionar o menu do cursor, quando os rótulos de menu estão apresentados, pressione . Para restaurar o menu anterior, pressione  uma segunda vez.

| Tecla | Descrição | Página |
|--|---|--------|
|  | Muda entre modos de substituição e inserção. | 167 |
|   | Elimina todos os caracteres à esquerda do cursor. | 168 |
|  | Elimina caractere na posição do cursor. | 167 |
|   | Elimina caractere na posição do cursor e todos os caracteres à direita. | 168 |
|  | Move o cursor para cima. | 167 |
|   | Move o cursor totalmente para a cima. | 168 |
|  | Move o cursor para baixo. | 167 |
|   | Move o cursor totalmente para baixo. | 168 |
|  | Move o cursor à esquerda. | 167 |
|   | Move o cursor totalmente à esquerda. | 168 |
|  | Move o cursor à direita. | 167 |
|   | Move o cursor totalmente à direita. | 168 |

Índice por Assunto

Números de página em **negrito** indicam referências principais; números de página em caracteres normais indicam referências secundárias.

A

- Abrindo o estojo, 18
- Agrupando um algébrico, 11,256
- Anúncios, 27, 29
- Apagando
 - a pilha, 44
 - dados estatísticos, 132
 - toda memória, 20
- Argumentos
 - definidos, 25
 - elementos de arranjos (array), 272
 - ordem dos, 41,43
 - utilização, 197
- Armazenando parâmetros de traçado
 - de gráfico, 96
- Arranjos (arrays)
 - definidos, 124
 - em sintaxe algébrica, 157
 - utilização mínima de memória, 191
- Aspas, nomes com, 57
- Assistência técnica, 293-295
- Associando termos, 114-115
- Atenção, 216
- Auto-testes, 218-219
- Avaliação, 198-199
- Avaliando uma expressão, utilizando o Solver, 65
- Avaliando uma variável, 50, 56

B

- Base para inteiros binários, 139
- Baterias, 286-288
- BDISP, programa, 259-262
- Beep, modo, 206
- Bolha, classificação, 270
- BOXR, programa, 245-246
- BOXS, programa, 241-244

C

- Cadeias, 156-157, 258
- Calculadoras algébricas, 302,305
- Cálculos em cadeias, 45
- Cancela operação do sistema, 215
- Caractere nova linha, 169
- Catálogo
 - de comandos, 196-197
 - de unidades, 141-143
- Clareando o visor, 21
- Classe-dados, objetos, 199
- Classe-nome, objetos, 199-201
- Classe-procedimento, objetos, 199, 201-204
- Classes de objetos, 199
- Comandos, 164-165
 - catálogo de, 26, 29, 31-33
- Compartimento da bateria, localização, 19
- Comutando termos, 113-114
- Condição, 225-226

Constantes, modo, 206-207
Contraste, visor, 21, 216
Contador, **228-230**, 260, 271,274
Contínuo, teste, 218
Copiando objetos da pilha, 178
Correlação, 136
Corrigindo erros, 47
COT, programa, 80-81
Cotangente, 80-81
Covariância, matriz, 263
Criando
 um diretório, 183
 uma variável, 49,54
Cursor, modos indicativos, 172

D

Dados dependentes, 136
Delimitadores, 26, 28, 169
Depurando programas, 250
Desempenho, maximizando, 190-191
Desenvolvendo um algébrico, 111, 256
Desfazer, 47
Desligar automaticamente, 20
Desvio padrão, 135
Determinante, 128
Diagnósticos, 218-219
Diferenciação, 117-120
Digitalizar, 93, 99
Diretório corrente, 60,184
Diretórios, **183-187**
 benefícios, 62, 66, 71, 183
 criando, 60
 mudando de, 275-279
Dois números, funções, 40

E

Editando, 69, 173
 dados estatísticos, 133
Eliminando, objetos da pilha, 179
Eliminando
 um diretório, 187
 uma variável, 52
Empréstimos, cálculos, 103-106
ENTER, 24, 173
Entrada personalizada, menu, 234, 235

Entrada, menu personalizado de,
 234-235
Entrada, modos, 51, **169-172**, 207
Entre expoente, 39
Equação corrente, 90
Equações, **162-163**
 processando, 203
 quadráticas, 107
 raízes de, 107
 traçando gráficos, 97
Erro, armadilha, 227-228, 259, 278
Escala de um gráfico, 91
Escalar, 126
Escurecendo o visor, 21
Espaçamento de saída impressa, 214
Estado corrente, 258
Estado, preservando, 258
Estimativas para o Solver, 99, 102
Estojo, abrindo e fechando, 18
Estruturas condicionais, 223-228
Exceções, matemáticas, 211-212
EXCO, programa, 255-256
Expoente, 38
Exponencial, função, 77-78
Expressões, 34, 161-162
 numérica, integração, 122-123
 processando, 202-203
 processando utilizando o Solver, 65
 de cálculos da pilha, 59-60
 zero de, 92, **98-100**, 107
Extremos de um gráfico, 94
Extremos de um gráfico, 96

F

Fechando o estojo, 18
FIB1, programa, 247
FIB2, programa, 248-253
Fibonacci, números, 246-249
Financeiros, cálculos, 103-106
Flags, 205, 225, 258
Força, unidade, 146
Formal, variável, 200
Função analítica, 164-165
Função, 164-165
 de um número, 41
 de dois números, 41
 processando, 203-204

G

GΣO, programa, 148
Galão, conversão, 146
Gama, função, 78
Garantia, 291-293
Globais, nomes, 159
 processando, 200-201
Globais, variáveis, 80, 182-183
Gráficas, cadeias, 157
Gram, conversão, 147-148
Graus, modo angular, 73
Graus-minutos-segundos, 76

H

Hiperbólicas, funções, 77-78
HOME, diretório, 60, 71
Horas-minutos-segundos, 76
HP RPN, calculadoras, 296-301
HP-Solver. Ver Solver

I

Igualdade, teste, 224
Imprimindo um gráfico, 91
Incluindo termos, 115
Incremento para o contador, 230
Indefinido, loops, **231-232**, 254, 257
Independentes, dados, 136
Infinito, resultado, 211-212
Inicializando a memória, 20
Inserção, modo, 70
Integração, 120-123
Inteiro binário, tamanho de palavra,
 210
Inteiro, base, 209-210
Inteiros binários, 156, 206
Inverso, de um número, 40
Invertendo uma matriz, 128
Isolando uma variável, 109-116

J

KEY?, programa, 239

L

Last, argumentos, 179-180
Letras minúsculas, modo, 26, 28
Linear, regressão, 137
Lineares, sistemas de equações, 130
Linha de comando, 22, 166
 recuperando, 174
Listas, 158, 276
 elementos de, 272
LMED, programa, 272-273
Locais, nomes, 159
 processando, 200
Locais, variáveis, 80, 86, 147, 179,
 222-223, 242, 259
 processamento de, 254
 superpostas, 270
Logarítmicas, funções, 77-78
Loop, estruturas de, 228
Loops definidos, **228-230**, 248, 260,
 274
 superpostos, 270-271

M

Mantissa, 38
Manutenção, 289
Marcador de base, 139
Matriz estatística corrente, 132
Matriz, definida, 124
Matriz, operações, 263
Máximo, de uma expressão, 100
Média, 134
MEDIAN, programa, 272-275
Mediana, definição, 272
Memória contínua, 20
Memória disponível, 188
Memória, inicialização, 20, 217
Memória, pouca, 188-190
Mensagem, imprimindo, 151
Menu do cursor, 30, 69, 166-168
Menu, chave, 169
Menu, rótulos, 27, 31
Menu, teclas, 27, 31
Menus personalizados, 192, **195**, 276,
 277
Milhas por hora, conversão, 145

Milímetro, conversão, 144
Mínimo, de uma expressão, 100
Modo angular, 73,205-206
Modo auto CR, 213
Modo de entrada alfanumérico, 55,
170-172
Modo de entrada algébrico, 34,51,
170-172
Modo de entrada imediato, **170-172**
Modos padrão, 205
Modos, 205-214
 indicados pelo cursor, 172
Movendo objetos da pilha, 178
Mudando o nome, de uma variável, 52
Mudando
 diretórios, 275-279
 uma variável, 51
MULTI, programa, 253-255
Multi-linhas, modo, 208

N

Negação, 40, 79
Negativo, número, 39
Níveis, da pilha, 176
Nível 1, imprimindo, 151
Nomes, **159-160**
 com aspas e sem aspas, 57
Numérica, variável, 49
Numérico, modo resultado, 203-204
Número, modo de apresentação, 37,
209
Números aleatórios, 78
Números complexos, 82, 155
Números reais, 155

O

OSR, programa, 147
Objetos algébricos, 161-163
 processamento, 202-203
Objetos, 154
Objetos, classes de, 199
Objetos, tipos de, 26-29
Off, automático, 20
Onça, conversão, 147-148
Operação, 164-165

Ordem dos argumentos, 41, 43
Overflow, 212

P

PAD, programa, 257-258
Pai, diretório, 60, 183, 275
Parâmetros de traçado de gráficos, 89
Parâmetros estatísticos, 136, 263
Passo único, execução, 250-253
Porcentagens, 43
Pés cúbicos, conversão, 146
Pés por segundo, conversão, 145
Pés, unidades, 146
Pi, 74-75
Pilha, 22, 272
 apagando, 44
 imprimindo, 152
Pilha, **176-181**
Pilha, diagrama, definição, 240
Pilha, flags de, 225
Pilha, lógica de, 25
Pilha, níveis da, 27, 31
Polares, coordenadas, 84-88
Polegada, conversão, 144
Ponto, 169
Ponto de dados, 132
Ponto decimal, 36, 209
Pós-fixada, notação, 25
Posições decimais, 37
Potências, 42
Pouca memória, 188-190
Pré-fixadas, unidades, 144
PRESERVE, programa, 258-259
Principal, valor, 206
Programação estruturada, 202, 241,
255
Programas, 160-161
 como argumentos, 254
 em algébricos, 263
 processando, 201-201
Programas, estruturas, 161
 processando, 201
Projetados, valores, 137
Proposição, 162
PSUM, programa, 86-88

Q

- Quadrado, 40
- Quadrática, expressões e equações, 107

R

- Radianos, modo angular, 73
- Raiz de uma equação, 107
- Raiz quadrada, 40
- Raiz, marca de, definição, 36
- Raízes, 42
- Recuperação, modos, 210-211
- Recuperando uma variável, 50, 56
- Recursividade, 246-247, 249
- Registro corrente, imprimindo, 150
- Regra da cadeia, 118-119
- RENAME, programa, 54-55
- Reservados, nomes, 159-160
- Restaurando a pilha, 180
- Resultados, modo, 207
- Retardando o processamento, 198
- Retrocesso, 27, 30
- RPN, 25

S

- Σ GET, programas, 265
- Σ X2, programa, 266
- Σ XY, programa, 267
- Σ Y2, programa, 266
- Saída de impressão, localização, 19
- Sem aspas, nomes, 57
- Separadores, 169
- Simbólicas, constantes, 163
- Simbólica, integração, 121
- Simbólico, modo de resultado, 203
- Sinal audível, modo, 206
- Sistema de equações lineares, 130
- Sistema, interrupção, 217
- Solver, 63-64, **98-109**
- SORT, programa, 270-272
- String, 156-157, 258
- Subdiretório, 60, 183, 275
- Subprogramas, 260
- SUMS, programa, 263-264

- Superpostas, estruturas e programas, 233
- estruturas e variáveis locais, 270
- funções do usuário, 245
- "loops" definidos, 270-271

T

- Tamanho de palavra, 138-139
- Tarefas internas, 190-191
- Taylor, série, 120
- Tecla de prefixo, 27, 29
- Teclado, 26-27, 328-330
- Teclado, teste, 219
- Temperatura, conversão, 143-144
- Teste, funções e comandos, 224
- Tipos de objetos, 26, 28
- Traçando gráficos, **89-97**
- Trace, imprimindo, 150, 213
- Traduzindo, um gráfico, 93
- Transpondo, 264
- Trigonométricas, funções, 73-77
- Trilha corrente, 60, 184-185
- Troca sinal, 39

U

- Um número, funções, 40
- Underflow, 212
- Unidade, catálogo, 26, 29
- Unidades, cadeias, 144-145
- Usuário, flags, 225
- Usuário, funções do, 79-81, 166, 202, 242, superpostas, 245
- Usuário, memória, 48
- Usuário, menu personalizado do, 235
- Usuário, menu personalizado, 235
- Utilização de comandos, 157

V

- Valor do dinheiro no tempo, 103-106
- Variáveis, 48
 - criando, 49, 54
 - eliminando, 52
 - imprimindo, 152
 - isolando, 109-116

variância, 135
vetores, definição, 124
Velocidade de impressão, 213
Vírgula, 169
Visor, contraste, 21
Visor, imprimindo, 149, 216

W,X,Y,Z

Zero de uma expressão, 92, **98-100**,
107, 162

O apêndice A descreve como verificar se sua calculadora necessita de reparos e como proceder para obter o serviço necessário.
Em caso de dúvidas quanto à assistência técnica entre em contato com

- São Paulo
Tesis Informática S/A
Al. Rio Negro, 750 — Alphaville
06400 — Barueri — S. Paulo
tel.: (011) 421-1444

- Rio de Janeiro
Tesis Informática S/A
Praia do Botafogo, 228 — 6º andar — salas 611/614
22250 — Rio de Janeiro — R.J.
tel.: (021) 552-6422

Se estiver fora do Brasil, consulte o escritório local da Hewlett-Packard ou envie-a para o

- Hewlett-Packard
Calculator Service Center
1030 N.E. Cirele Blvd.
Corvallis, OR 97330, U.S.A.
tel.: (503) 757-2002

Índice

Página 15 Como Utilizar Este Manual

17 Parte 1: Princípios Básicos

Para Iniciar • Efetuando Cálculos Aritméticos • Utilizando Variáveis • Repetindo Cálculos • Funções de Número Real • Funções de Número Complexo • Traçando Gráficos • O Solver • Soluções Simbólicas • Cálculo • Vetores e Matrizes • Estatísticas • Aritmética Binária • Conversão de Unidades • Imprimindo

153 Parte 2: Sumário dos Recursos da Calculadora

Objetos • Operações, Comandos e Funções • A Linha de Comando • A Pilha • Memória • Menus • Catálogo de Comandos • Processamento • Modos • Operações de Sistema

221 Parte 3: Programação

Estruturas de Programa • Programas Interativos • Exemplos de Programação

281 Apêndices e Índices

Atendimento ao Usuário, Baterias e Assistência Técnica • Notas para Usuários de Calculadoras HP RPN • Notas para Usuários de Calculadoras Algébricas • Diagramas dos Menus • Índice de Teclas • Índice por Assunto



HEWLETT
PACKARD

**Número do Manual para Pedidos Avulsos:
00028-90095**

00028-90026

Impresso no Brasil 8/88