Noções básicas para Programação HP

Daniel Rossato

Henrique Camargo

Introdução

- Modo RPN
  - Mais fácil de implementar
  - Mais rápido para inserir expressões matemáticas
    - Não precisa de parênteses
    - Não precisa “lembrar” de resultados
  - Exemplo:

$$\frac{3+5}{7+6}$$

Introdução

- Algébrico
  - 7 + 6 =
  - Anotar o resultado
  - 3 + 5 =
  - Dividir pelo resultado da primeira conta
    - / 13 =
  - Total: 12 teclas, mais esforço de anotar/lembrar o resultado da primeira operação

Introdução

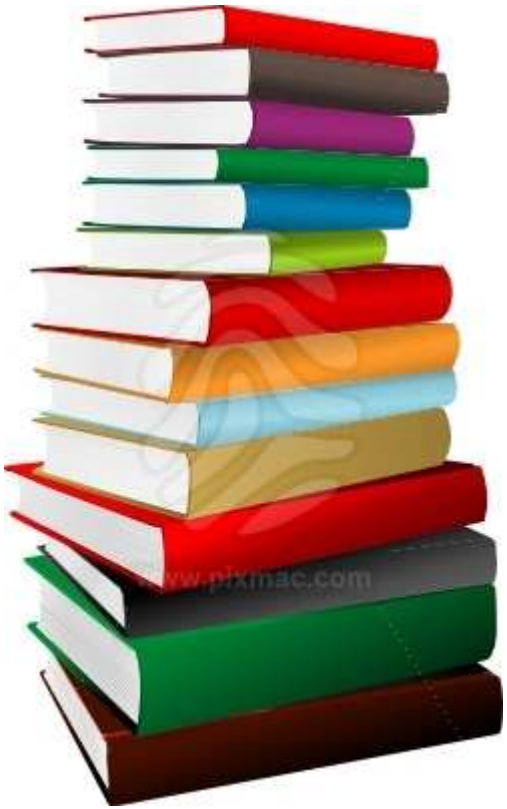
- RPN
  - 3 ENT 5 + 7 ENT 6 + /
  - Total: 9 teclas, sem esforço para lembrar

RPN

- Como funciona?
  - Operandos primeiro, operadores depois
    - 3 + 5 ⇔ 3 5 +
    - 4/3 ⇔ 4 3 /
    - Sin(30) ⇔ 30 sin
  - Pilha
    - Resultados anteriores armazenados numa memória LIFO (Last In, First Out)

RPN

- Pilha
  - Só coloca no topo (push)
  - Só tira do topo (pop)
  - Pra tirar um livro do meio, tem que tirar todo o resto
  - A pilha da HP é visualizada de ponta-cabeça



RPN

- O que faz um operador
  - "pop" no número de argumentos que ele usa
    - Adição; 2 argumentos
    - Raiz quadrada: 1 argumento
    - Função trigonométrica (sin, cos): 1 argumento
  - Calcula o resultado
  - "push" no resultado

RPN

- Pilha
  - Últimos valores são utilizados primeiro (LIFO)
  - Exemplo: 3 4 5 6 + => 3 4 11

RPN

- Exemplos
  - 3 2 +
  - 5 4 *
  - /
  - 2 +
- Alternativa
  - 2 3 2 + => 2 5 na pilha
  - 5 4 * => 2 5 20 na pilha
  - / => 2 5/20 na pilha (ou ¼)
  - + => 2+1/4

$$2 + \frac{3+2}{5 \cdot 4}$$

RPN

- Exemplo:

$$1 - \frac{\sqrt{3+2} \cdot \left(4 - \left(3 + 2 \cdot 5\right)\right)}{2 / \left(7 - 3\right)}$$

- Várias maneiras de resolver!
  - Don’t panic!!!
  - Começar com o mais externo, e ir aprofundando, resolvendo cada operando

RPN

$$1 - \frac{\sqrt{3+2} \cdot \left(4 - \left(3 + 2 \cdot 5\right)\right)}{2 / \left(7 - 3\right)}$$

- Operação mais externa: -
  - Primeiro operando: 1
  - Segundo operando: fração
    - Processar fração
    - Primeiro operando: multiplicação – processar
      - Primeiro operando: raiz
        - » etc., etc., etc...

RPN

$$1 - \frac{\sqrt{3+2} \cdot \left(4 - \left(3 + 2 \cdot 5\right)\right)}{2 / \left(7 - 3\right)}$$

- **1 3 2 + RAIZ 4 3 2 5 * + - * 2 7 3 - / / -**

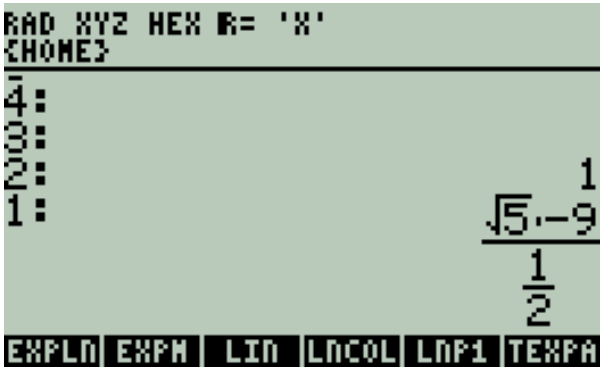
RPN

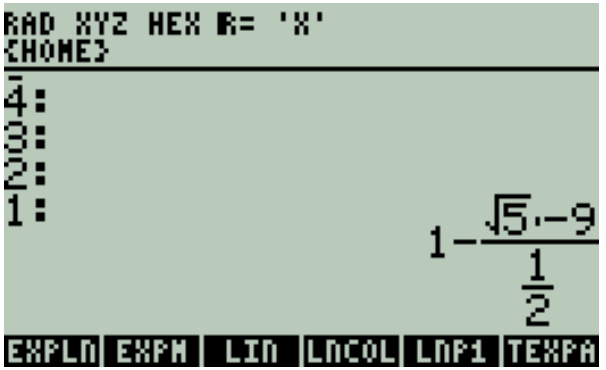
$$1-\frac{\sqrt{3+2}\cdot\left(4-\left(3+2\cdot 5\right)\right)}{2/\left(7-3\right)}$$

- 1 3 2 + RAIZ 4 3 2 5 * + - * 2 7 3 - / / -

RPN

$$1-\frac{\sqrt{3+2}\cdot\left(4-\left(3+2\cdot 5\right)\right)}{2/\left(7-3\right)}$$



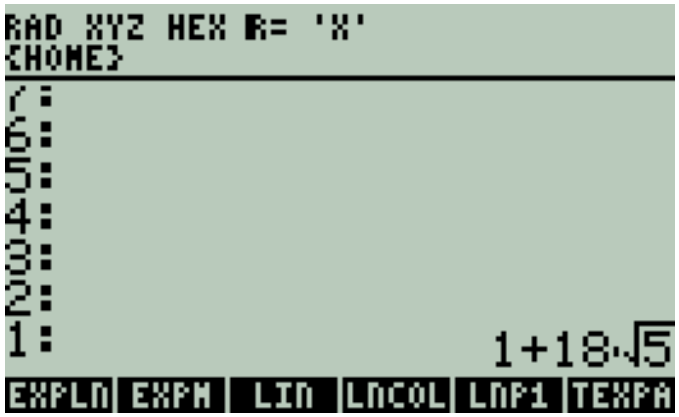


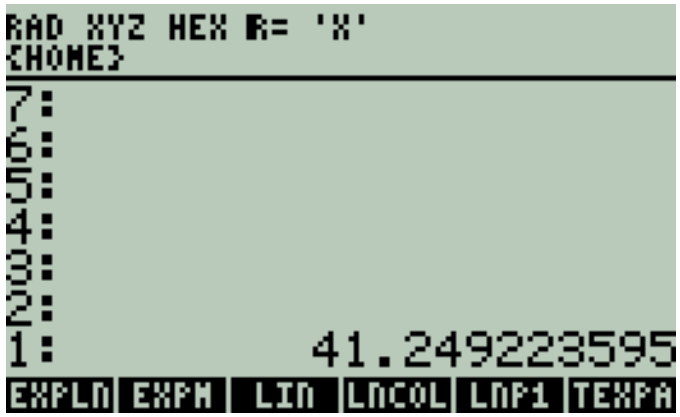
- 1 3 2 + RAIZ 4 3 2 5 * + - * 2 7 3 - / / -

RPN

- Simplificar resultado:

- Eval - ->NUM



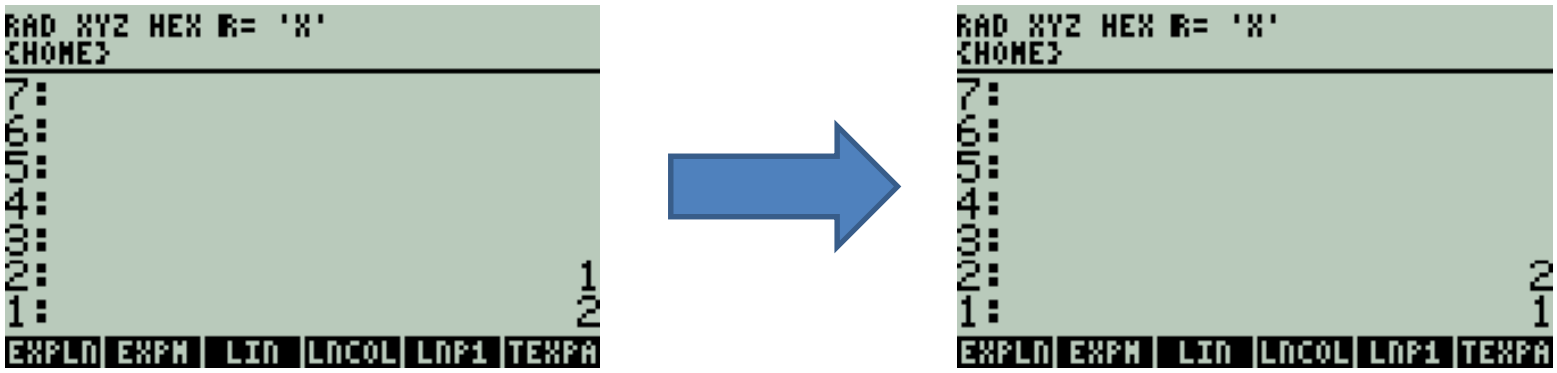


RPN

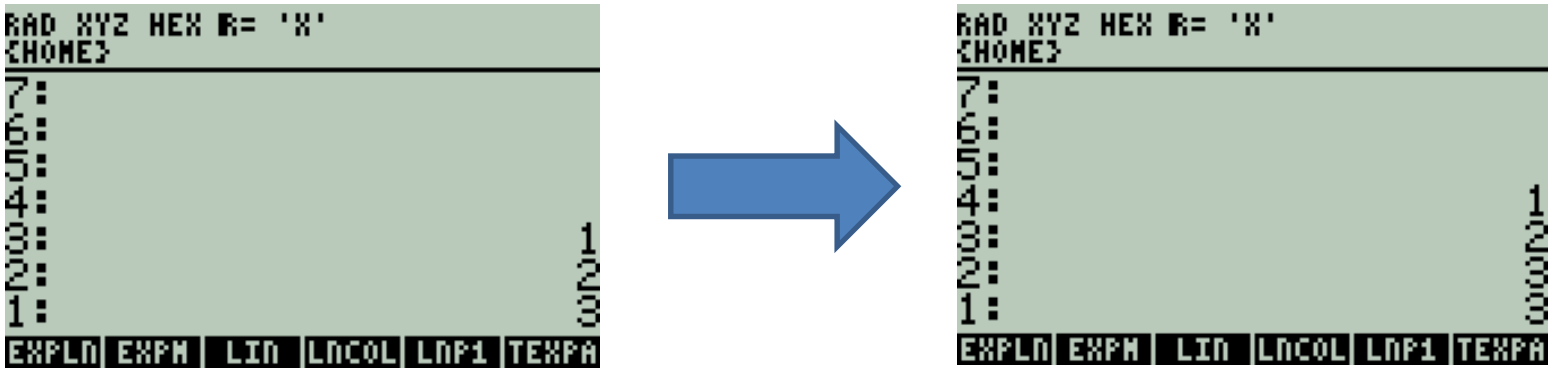
- Não vá embora!! Não desista!
- Funções úteis para operações com pilha
  - SWAP (seta para direita)
  - DUP (enter)
  - PICK
- SWAP é muito útil até se acostumar a "pensar" em pilha.

RPN

- SWAP

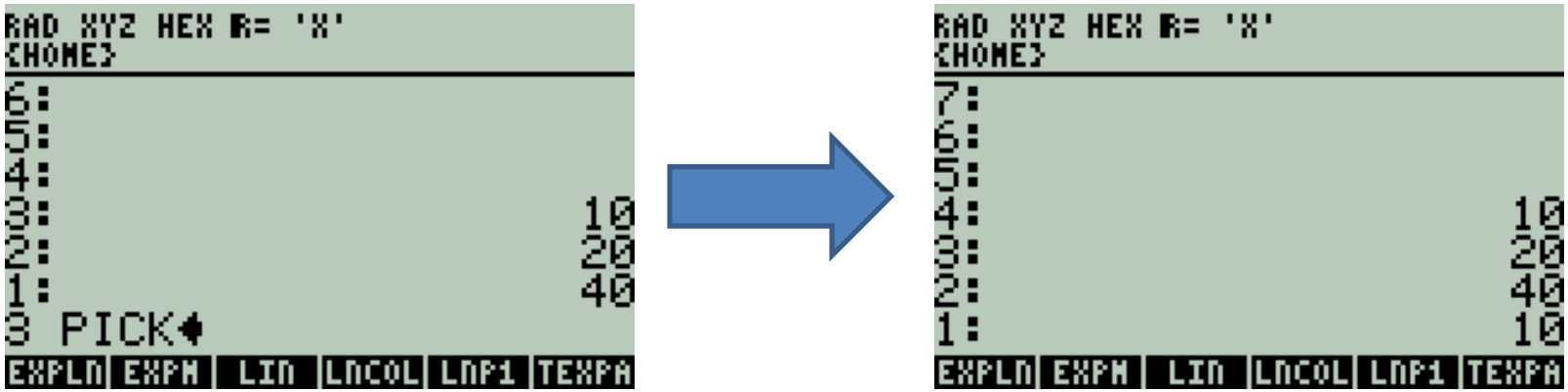

RAD XYZ HEX R= 'X'
{HOME}
7:
6:
5:
4:
3:
2: 1
1: 2
EXPLN EXPM LIN LNCOL LNP1 TEXPA
RAD XYZ HEX R= 'X'
{HOME}
7:
6:
5:
4:
3:
2: 2
1: 1
EXPLN EXPM LIN LNCOL LNP1 TEXPA

- DUP


RAD XYZ HEX R= 'X'
{HOME}
7:
6:
5:
4:
3:
2: 1
1: 3
EXPLN EXPM LIN LNCOL LNP1 TEXPA
RAD XYZ HEX R= 'X'
{HOME}
7:
6:
5:
4:
3:
2: 1
1: 3
EXPLN EXPM LIN LNCOL LNP1 TEXPA

RPN

- PICK
  - Argumento: no. da linha



RPN

- Menu pilha
  - Ativado apertando a seta pra cima
    - VIEW
    - EDIT
    - PICK
    - ROLL
    - ROLLD
    - DUPN
    - DROPN
    - KEEP

RPN

- VIEW
  - Visualizar elemento
- EDIT
  - Editar elemento
- PICK
  - Duplica o elemento selecionado na base da pilha
- ROLL
  - Gira a pilha pra cima

RPN

- ROLLD
  - Gira pilha pra baixo
- DUPN
  - Duplica todos os elementos até a linha selecionada
- DROPN
  - Deleta todos os elementos abaixo da linha selecionada (incluindo a linha)
- KEEP
  - Deleta todos os elementos acima da linha selecionada

RPN

- Divirta-se com os comandos aprendidos!

- Sim, divirta-se.

Variáveis

- Armazenam valores, equações, strings, programas, etc.
- Para acessar o valor de uma variável, digite o nome
- Para referenciar a variável, digite entre aspas simples: ‘X1’
- O nome pode conter números, mas deve sempre começar com uma letra

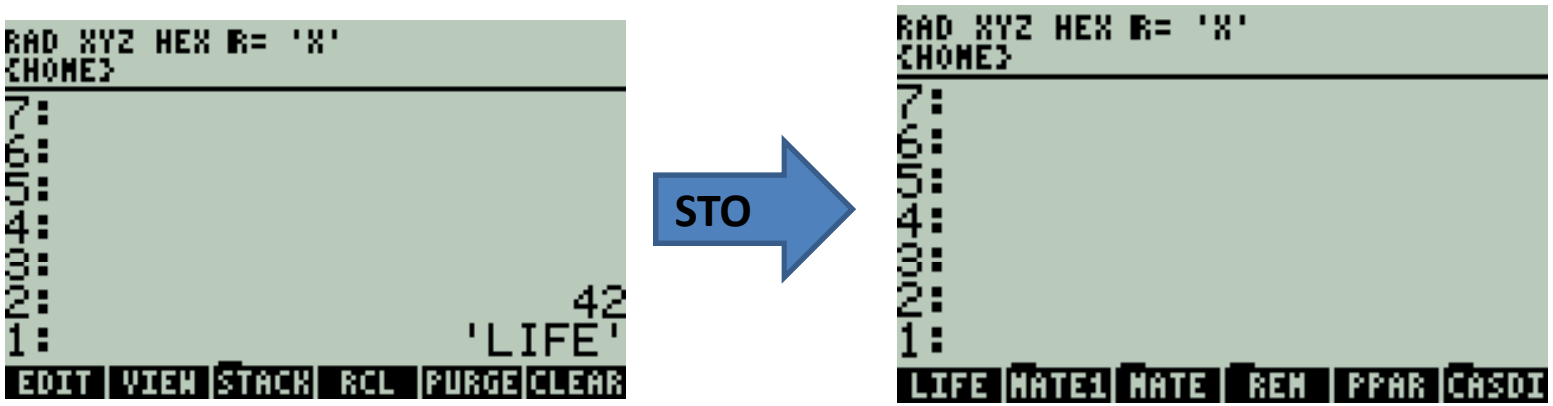
Variáveis

- Para guardar um valor numa variável:
  - Coloque o valor na pilha
  - Nome da variável entre aspas simples
  - STO

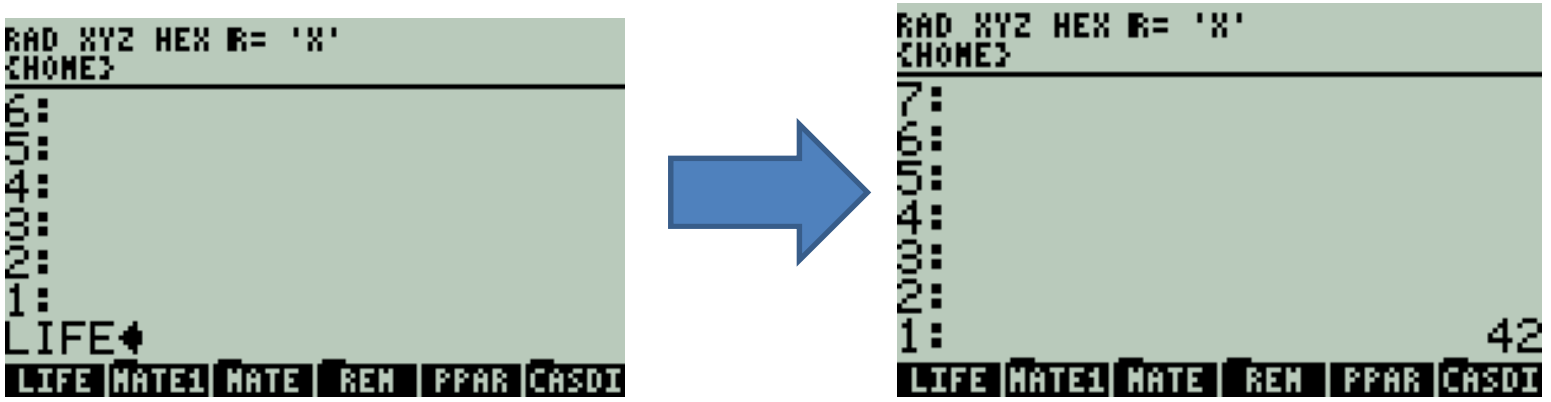
- As variáveis estarão disponíveis no menu VAR. Pode-se invocar o valor da variável pressionando o botão correspondente, ou com o nome da variável.

Variáveis

- Exemplo

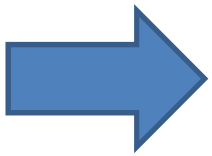


- Acessando



Variáveis

- Apagando – PURGE



Variáveis

- RELEMBRANDO!!!
- Entre aspas -> Para armazenar valores e apagar a variável
- Sem aspas -> Para acessar o valor

UserRPL

- Linguagem de programação das calculadoras HP
- São scripts de funções RPN, utilizam a pilha da mesma maneira
- Representados pelo símbolo << >>
- Armazenados dentro de variáveis

UserRPL

- Exemplo – Paralelo de Resistores

$$\frac{1}{R_{eq}} = \frac{1}{R_1} + \frac{1}{R_2} \qquad R_{eq} = \frac{1}{\frac{1}{R_1} + \frac{1}{R_2}}$$

- R1 INV R2 INV + INV
- Supondo que R1 e R2 estejam na pilha:

<< INV SWAP INV + INV >>

UserRPL

- Nem sempre manipulação de pilha é trivial
- Outro modo de resolver:

- << `R2` STO `R1` STO R1 INV R2 INV + INV >>

- Problema: variáveis “lixo” no final do programa
- Solução: variáveis locais

UserRPL

- << -> R1 R2
  << R1 INV R2 INV + INV >>
  >>

- Sem variáveis “lixo”
- Se houver variáveis com o mesmo nome, ignora
- Jeito recomendado de fazer as coisas.

UserRPL

- Funções
  - RTFM : LEIA O MANUAL!!
  - No manual completo, todas as funções estão descritas em detalhe.
  - Se você consegue fazer uma operação, você consegue fazer o programa pra isso

UserRPL

- Vários tipos de argumentos são possíveis:
  - Valores
  - Expressões
  - Variáveis
  - Vetores
  - Matrizes
  - Outros programas
  - Etc.

UserRPL

- Exemplo:
  - Decompor fator de potência
  - Recebe Potência aparente e fator de potência
  - Devolve Potência Ativa e Reativa

  - P = S*FP
  - Q = S*sin(acos(FP))

UserRPL

- Comando ->TAG – formata a saída
  - << -> S FATOR
    << S FATOR * “Ativa” ->TAG S FATOR ACOS
    SIN * “Reativa” ->TAG >>
    >>

UserRPL

- Separar real e imaginário
  - Sem variáveis locais:
    - << DUP RE SWAP IM >>
  - Com variáveis locais:
    - << -> NUMERO << NUMERO RE NUMERO IM >> >>

- Se o programa tem um argumento só, em geral não é necessário usar variáveis locais

UserRPL

- Dada uma impedância e uma corrente, dar a tensão e a potência:

- $U = Z \cdot I$
- $P = U \cdot I^*$

UserRPL

- << -> I Z
  << Z I * “Tensão” ->TAG
  DUP I CONJ * “Potencia” ->TAG
  Z ARG COS “FP” ->TAG
  >>
  >>

UserRPL

- Lançamento parabólico
  - Recebe velocidade de lançamento e ângulo
  - Devolve alcance e altura máxima
    - t = 2*V*SIN(alfa)/9.8
    - h = V SIN(alfa) t/2 - (9,8 (t/2)^2)/2
    - x = V*COS(alfa)*t
    - << -> V ALFA
      << 2 V ALFA SIN * * 9.8 / `T` STO
      V ALFA COS T * * “Alcance” ->TAG
      V ALFA SIN T 2 / ** 9.8 T 2 / SQ * 2 / -
      “Altura” ->TAG T “Tempo” ->TAG
      >>
    - >>

UserRPL

- Bhaskara
  - Ax^2 + bx + c = 0

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

  - << -> A B C
    << B NEG B SQ 4 A C * * - RAIZ+ 2 A * /
    B NEG B SQ 4 A C ** - RAIZ – 2 A * /
    >>
    >>

  - RAIZ é o símbolo de raiz quadrada

Agora é com vocês!

- Dêem exemplos de programas que vocês querem
- Vamos fazer todos juntos!

Perguntas

- ~~Encher linguiça~~