

Notação Polonesa Reversa

Seres humanos escrevem suas sentenças matemáticas com o uso de chaves, colchetes, parênteses e ainda por cima tem uma hierarquia de operações. Se isto não for usado, temos problemas. Por exemplo:

$5 + 4 * 3$ é igual a 17, ou é igual a 27 ???

Todo compilador precisa entender expressões aritméticas escritas por humanos, logo, todo compilador precisa entender tais sentenças. Será que não haveria uma maneira de se escrever estas sentenças, sem NENHUM parênteses, e SEM nenhuma hierarquia ?

Esta maneira foi inventada por **Jan Lukiziewicz**, em 1951 e passou a ser denominada NOTAÇÃO POLONESA. Agora, mediante a introdução de um caracteres separador de operandos, as sentenças podem ser escritas sem parênteses e sem ambigüidade.

Na notação polonesa, escrevem-se os operandos e (antes = préfixada ou) depois (pósfixada) a operação que eles sofrerão.

Por exemplo, a expressão, $5 * (6 + 2) - 12/4$, ficaria 5, 6, 2, +, *, 12, 4, /, -

Para que qualquer compilador consiga achar o valor de uma expressão com parênteses, ele sempre:

1. Precisa transformar a expressão em RPN
2. Calcular a RPN

Algoritmo de Normal para RPN

Este algoritmo recebe uma cadeia de caracteres (denominada Q) representando a expressão aritmética escrita de maneira convencional, eventualmente com o uso de parênteses. A saída é outra cadeia (denominada R) e que é equivalente a cadeia original (a Q) mas EXPRESSA EM RPN.

Para este algoritmo funcionar, consideraremos a seguinte escala de prioridades: 3=E, 2=*, 1=+-/. Quando duas operações de mesma prioridades existirem faremos da esquerda para a direita, a menos que haja parenteses.

```
1: cadeia função NORMAL2RPN (cadeia Q)
2: pilha PILHA
3: empilhe (PILHA, "(")
4: insira ")" ao final de Q
5: enquanto houver elementos em Q faça
6:   se é operando então
7:     coloque-o na resposta R
8:   senão
9:     se é "(" então
10:       empilhe-o
11:     senão
12:       se é o operador (*) então
13:         retire da pilha todos os operadores com prioridade igual ou maior a (*) colocando-os na resposta R
14:         empilhe (*)
15:       senão
16:         se é "+" então
17:           desempilhe da PILHA e coloque na resp R cada operador ate achar um "("
18:           remova o "("
19:         fim se
20:       fim se
21:     fim se
22:   fim se
23: fim enquanto
```

Mini chinês Fazer chinês com a expressão $5 * (6 + 2) - 12/4$

Deve dar: 5, 6, 2 + * 12, 4 / -

A propósito compare o resultado encontrado ao analisar uma expressão parecida com a acima: $5 * (6 + 2) E 12/4$, que dá como resposta 5, 6, 2 + 12 E * 4 / -

Algoritmo de resolução de RPN

A entrada deste algoritmo é a cadeia já convertida pelo algoritmo anterior e a saída é um número real representando o resultado final numérico da expressão.

```
1: real função ResolveRPN (cadeia R)
2: pilha PILHA
3: real A,B
4: enquanto houver elementos em R faça
5:   se é um operando então
6:     empilhe-o
7:   senão
8:     se é um operador (*) então
9:       desempilhe B
10:      desempilhe A
11:      Calcule A (*) B
12:      empilhe o resultado
13:    fim se
14:  fim se
15: fim enquanto
```

O valor final na pilha é o resultado desejado

Mini Chinês

Faça um chinês com: $5 * (6 + 2) - 12/4$

Transformado em 5, 6, 2, + * 12, 4 / -

E dará:

5,6,2; depois 5,8; depois 40,12,4; depois 40,3 e 37

Exemplos

1. $((4 - 8) + (((8 * 8) - 3) * 2)) / (3 + 5)$
2. $((6 * 1) + ((5 * ((9 - 8) + (2/1))) * (2 + 2))) + 5$
3. $((((6 * ((8 + 7) - (5 * (2E1)))) - (4/2)) - (6 - 1)) / 8$
4. $(1 + 1) + (((((6 - 5) + ((3 - 3) + ((4 - 6) * (6/3)))) * (3 + 7)) + (5 + 1)) - 9)$
5. $(((((8 - 1) * (6 + 9)) - (9 * 7)) - 1) / (5 - 3)$
6. $((9 - ((3 * 3) + (3 / (4 * 2)))) + 1) + (8 - 3)$
7. $((0E2) + ((7 + (((6/2) + 9) - (6/3))) + 8)) - (4 + 8)$
8. $((((4 + 1) + (((((9 + 7) / ((9 * 5) * (7 + 6))) + (1 - 1)) + 8)) - (5 - 4)) + (1 * 4)$

Respostas

1. $48 - 88 * 3 - 2 * + 35 + /$, com resposta=14.75
2. $61 * 598 - 21 / + * 22 + * + 5 +$, com resposta=71.00
3. $687 + 521E * - * 42 / - 61 - - 8 /$, com resposta=2.88
4. $11 + 65 - 33 - 46 - 63 / * + + 37 + * 51 + + 9 - +$, com resposta=-31.00
5. $81 - 69 + * 97 * - 1 - 53 - /$, com resposta=20.50
6. $933 * 342 * / + - 1 + 83 - +$, com resposta=5.63
7. $02E762/9 + 63 / - + 8 + + 48 + -$, com resposta=13.00
8. $41 + 97 + 95 * 76 + * / 11 - + 8 + + 54 - - 14 * +$, com resposta=16.03

🔑 - Para você fazer

Nos exercícios a seguir, converta a expressão para RPN e depois calcule o resultado correto. A letra *E* significa *elevado a* e indica a operação potência.

1. $((((2E2) - (8 - 1)) - 9) + (1 + 4)) + (2 * 3)$
2. $(3 * 1) * ((3 * 3) - (3 + ((5 - 6) + ((8 - 1) - 3))))$
3. $((3 + 2) * (((0E1) * (2 / (4 + (8 * 2)))) + 7)) + (3 - 3)$
4. $(((((2/2) - (3 + ((2 * 2) + (3 + 8)))) * (7 * 3)) / (6 * 7)) - (2/2)) * (8/1)$

Para responder este exercício, você deve fazer duas coisas. Primeiro, deve localizar - na expressão RPN - qual o caracter pedido (usando o algoritmo 1). Depois, deve apresentar o resultado numérico dessa expressão RPN. (usando o algoritmo 2).

1. Caracter [12] _____. Resultado=_____.
2. Caracter [8] _____. Resultado=_____.
3. Caracter [11] _____. Resultado=_____.
4. Caracter [24] _____. Resultado=_____.

