

1. Na notação usual de expressões aritméticas, os operadores são escritos entre os operandos; por isso, a notação é chamada infixa. Na notação posfixa, ou polonesa, os operadores são escritos depois dos operandos. Eis alguns exemplos de expressões infixas e correspondentes expressões posfixas:

infixa	posfixa
$(A+B*C)$	$ABC*+$
$(A*(B+C)/D-E)$	$ABC+*D/E-$
$(A+B*(C-D*(E-F)-G*H)-I*3)$	$ABCDEF-* -GH* -*+I3*-$
$(A+B*C/D*E-F)$	$ABC*D/E*+F-$
$(A+B+C*D-E*F*G)$	$AB+CD*+EF*G*-$
$(A+(B-(C+(D-(E+F)))))$	$ABCDEF+-+--+$
$(A*(B+(C*(D+(E*(F+G))))))$	$ABCDEF*G+*+*+*$

Note que os operandos (A, B, C, etc.) aparecem na mesma ordem na expressão infixa e na correspondente expressão posfixa. Note também que a notação posfixa dispensa parênteses e regras de precedência entre operadores (como a precedência de \* sobre + por exemplo), que são indispensáveis na notação infixa.

Faça um programa que use pilha para calcular o valor de uma expressão passada em notação pós fixa.

Considere que há espaço em branco separando números de expressões

2. Parênteses e Colchetes

Considere o problema de decidir se uma dada sequência de parênteses e colchetes está bem-formada (ou seja, parênteses e colchetes são fechados na ordem inversa àquela em que foram abertos). Por exemplo, a sequência

$(([()]) )$

está bem-formada, enquanto  $([ ] )$  está malformada. Suponha que a sequência de parênteses e colchetes está armazenada em uma string.

Use uma pilha para resolver o problema. O algoritmo é simples: examine a string da esquerda para a direita e empilhe os parênteses e colchetes esquerdos à espera de que apareçam os correspondentes parênteses e colchetes direitos, que serão os responsáveis por removê-los da pilha.

3. Escreva um algoritmo que use uma pilha para inverter a ordem das letras de cada palavra de uma string, preservando a ordem das palavras. Por exemplo, para a string ESTE EXERCICIO E MUITO FACIL o resultado deve ser ETSE OIICREXE E OTIUM LICAF.

4. Digamos que nosso alfabeto contém apenas as letras a, b e c. Considere o seguinte conjunto de strings: c, aca, bcb, abcba, bacab, aacaa, bbcbb, ... Qualquer string desse conjunto tem a forma  $WcM$ , sendo  $W$  uma sequência de letras que só contém a e b e  $M$  é o inverso de  $W$  (ou seja,  $M$  é  $W$  lido de trás para frente). Escreva um programa que decida se uma string  $X$  pertence ou não ao nosso conjunto, ou seja, decida se  $X$  é da forma  $WcM$ .