

LEXSIM - Avaliador Lexico e Sintático

Por Neilor Tonin, URI  Brasil**Timelimit: 1**

Uma das formas mais interessantes do uso de pilhas é a na avaliação de uma expressão matemática. Pode-se, através da pilha, fazer a análise léxica de uma expressão (indicar se uma expressão possui um operando inválido, como por exemplo um símbolo qualquer que não está presente nem na tabela de operadores, nem na tabela de operandos) e também a análise sintática. A análise sintática pode indicar que está faltando um ou mais parênteses, sobrando um ou mais parênteses, sobrando operador, 2 operandos sucessivos, etc. A tarefa aqui é determinar se uma expressão está correta ou não.

Entrada

Como entrada, são válidos:

- a) Operandos: todas as letras maiúsculas ou minúsculas ('a'..'z', 'A'..'Z') e números (0...9).
- b) Parênteses.
- c) Operadores: deverão ser aceitos os seguintes operadores segundo a tabela de prioridades apresentada abaixo:

Operador	Prioridade
\wedge	6
$*$, $/$	5
$+$, $-$	4
$>$, $<$, $=$, $\#$, $\%$	3
AND (\cdot)	2
OR ($ $)	1

Para facilitar a implementação, será utilizado um ponto para representar o AND (.) e o Pipe (|) para representar o OR.

Obs.: Como restrição, não será permitida a entrada de expressões com operadores unários, como por exemplo o '-' de: $4 * -2$

A finalização da entrada será determinada pelo final do arquivo de entrada EOF().

Saída

Como saída, para cada expressão de entrada deverá ser gerado uma linha indicando o resultado do processamento. Se a expressão estiver correta, esta deverá ser transformada para a forma posfixa. Se não for possível, deverá ser impressa a mensagem "Lexical Error!" indicando erro léxico ou "Syntax Error!" indicando o erro de sintaxe, nesta ordem.

Exemplo de Entrada	Exemplo de Saída
((A+ (A+B) * C	Syntax Error! Syntax Error! AB+C*

(A+B) *% Exemplo de Entrada	Lexical Error Exemplo de Saída
(a+b*c)/2*e+a	abc*+2/e*a+
(a+b*c)/2*(e+a)	abc*+2/ea+*
(a+b*c)/2*(e+a	Syntax Error!
(ab+*c)/2*(e+a)	Syntax Error!
(a+b*cc)/2*(e+a	Syntax Error!
("a+b*cc)/2*(e+a	Lexical Error!
a+b-c	ab+c-
a-b*c/d+e	abc*d/-e+