
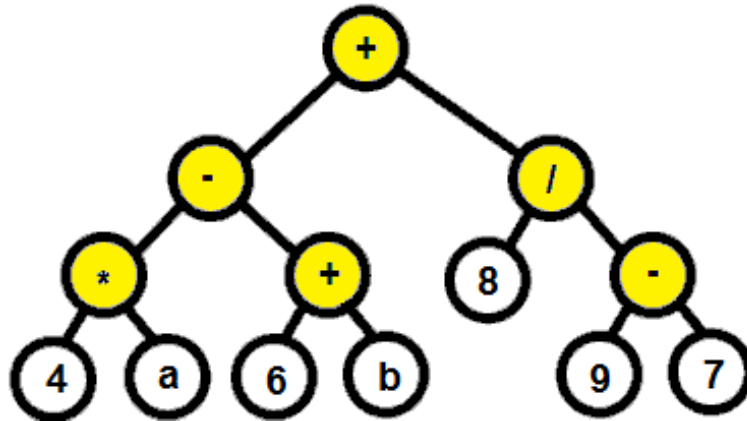


Árvore Binária de Expressão

Por Neilor Tonin, URI  Brasil

Timelimit: 2

A árvore binária de expressão aritmética é uma aplicação específica de uma árvore binária para avaliar expressões. Ela pode ser usado para representar uma expressão algébrica ou booleana, como por exemplo, a expressão $4 * a - (6 + b) + 8 / (9 - 7)$ que é apresentada na figura abaixo.



Essas árvores podem representar expressões que contêm operadores unários e binários. As árvores de expressão são implementadas como árvores binárias, principalmente porque permitem ao usuário encontrar rapidamente o que está procurando.

O limite superior de passos necessários para encontrar a informação requerida em árvores binárias igual a $\log_2 N$, em que N indica o número de todos os nós de uma árvore.

A fim de fazer um exercício diferente, o professor pediu para listar uma expressão armazenada em uma árvore binária, nível a nível, iniciando no primeiro nível (zero) e terminando no nível n .

Entrada

A entrada contém vários casos de teste. Cada caso de teste consiste de uma expressão aritmética contendo no mínimo dois operandos e uma operação simples e no máximo até 100 elementos. Esta expressão poderá conter letras maiúsculas, letras minúsculas, números, parênteses e operações aritméticas básicas (+, -, *, /) conforme o exemplo abaixo. Cada operando pode ter apenas um dígito ('0'-'9') ou letra ('a', 'B', etc). O final da entrada é indicado pelo fim de arquivo (EOF). O final da entrada é indicado por final de arquivo (EOF).

Saída

Para cada caso de teste, seu programa deverá imprimir várias linhas de saída correspondentes aos níveis da árvore de expressão e contendo todos os elementos presentes em cada um destes níveis, da esquerda para a direita. Estas linhas devem iniciar sempre com a mensagem "Nível n: ", conforme o exemplo fornecido abaixo. Imprima uma linha em branco entre dois casos de teste.

Exemplo de Entrada	Exemplo de Saída
4 * a - (6 + b) + 8 / (9 - 7)	Nível 0: +
a + b	Nível 1: -/
(a + b * c) * a - 4 * 5 - 6 + 1 +	Nível 2: *+8-
c * 3	Nível 3: 4a6b97
	Nível 0: +

Nivel 1: ab

Nivel 0: +

Nivel 1: +*

Nivel 2: -1c3

Nivel 3: -6

Nivel 4: **

Nivel 5: +a45

Nivel 6: a*

Nivel 7: bc
