

Atividade de Programação 4

Expressões Numéricas em RNP

Algoritmos e Estruturas de Dados I

Tempo limite: 1s

Algumas calculadoras que realizam cálculos financeiros, como a HP 12C, utilizam o método Notação Polonesa Reversa, (RPN, em inglês, *Reverse Polish Notation* ou muitas vezes conhecida por método pós-fixado - *Postfix*), para encontrar o valor das expressões numéricas digitadas. Por exemplo, para avaliarmos uma expressão na notação usual (*Infix*) como: $(1+2) * (8-3)$, em notação *Postfix* temos: $12+83-*$, cujo resultado é 15.

O objetivo desta tarefa consiste em: dado uma expressão na forma de RPN (*Postfix*), calcular o valor final da mesma.

Cada elemento de entrada, na forma tradicional, pode ser classificado como um operador: $+$, $-$, $*$, $/$ ou dígito: 0 , ..., 9 .

Uma forma para encontrar o resultado de uma expressão em RPN, é através do armazenamento gradual do RPN em uma **pilha**. Cada dígito (valor numérico) encontrado deve ser empilhado. Para cada operador encontrado na expressão deve-se desempilhar dois dígitos e efetuar a operação numérica correspondente, o resultado da operação deve ser armazenado na mesma pilha novamente, para uso futuro.

Por exemplo, dado a *string* que representa o RPN: $12+83-*$, os dígitos **1** e **2** são empilhados assim que forem lidos (considerando que os valores numéricos nessa atividade podem ter no máximo um único dígito). Ao se ler o operador $+$ deve-se desempilhar os valores **1** e **2** e realizar a soma dos mesmos, armazenando o seu resultado (valor 3) na pilha. Depois os dígitos **8** e **3** são empilhados como dois diferentes valores numéricos. Quando o operador $-$ é lido, desempilha-se os dígitos 8 e 3, realiza-se a subtração e empilha-se o resultado (valor 5). Por último, ao se ler o operador $*$, desempilham-se os valores **3** e **5** para realização da multiplicação, que resulta em 15, que deve ser empilhado. O método termina quando não há mais elementos a serem lidos a partir da *string* que representa a RPN. Quando isso ocorre, exibe-se o resultado final, que deve ser retirado da pilha (a qual deve ficar vazia).

O programa que soluciona esta tarefa **deve** utilizar uma ou mais **pilhas**, que serão representadas na forma de uma **lista simplesmente encadeada**, para armazenar e manipular os dados da RPN.

As precedências dos operadores utilizados na expressão são: $+$, $-$, $*$, $/$, onde os operadores $+$ e $-$ tem a mesma precedência, assim como o $*$ e $/$.

Os operadores devem ser binários, ou seja, **exigem** a existência de dois dígitos para realização de uma operação matemática. Não devem ser considerados operadores unários, que exigem apenas

um dígito, tais como: -3, +5, +1+2.

Entrada

A entrada para cada teste inicia-se informado a *string* da expressão numérica na forma *Postfix (RPN)*. O tamanho máximo da *string* correspondente a entrada é **N**, onde $(1 \leq N \leq 1024)$. A *string* NÃO DEVE possuir espaços em branco. Por exemplo, 49-2*5+ **✗** CORRETO, 4 9- 2 * 5 + **✗** ERRADO.

Os valores numéricos possuem APENAS um dígito, ou seja, TODOS os dígitos para as operações só conterão valores de 0 a 9, assim, a expressão 56+ significa, 5+6.

Saída

A saída do programa deve ser mostrada em uma linha a qual mostra o resultado da expressão numérica. Caso a expressão de entrada seja invalida deve-se exibir: *Invalid Expression*, como nos casos: 5*, 789+, +4, 78*/+.

Alguns exemplos de saída podem ser vistos a seguir:

- **Expressão RPN:** 53+2*3/ (significa: (5+3)*2/3)
Resultado: 32
- **Expressão RPN:** 54-3-2+ (significa: 5-4-3+2)
Resultado: 0

Dicas:

- Quando for calcular a expressão, deve-se observar que a mesma é lida como um *string*, mas as operações com RPN são realizadas com números inteiros. O programa deve tratar essa conversão, portanto.
- Não é possível ter entrada de números negativos, mas o resultado da expressão pode ser negativo. Por exemplo:
 - **Expressão RPN:** 45-
Resultado: -1

Restrições:

- O programa deve ser escrito em C;
- Deve-se ter no código as funções de inicialização, remoção (*pop*), inserção (*push*) e pilha vazia;
- A Pilha deve ser representada como uma lista simplesmente encadeada;
- É prevista a ocorrência de uma expressão de entrada como um RPN inválido onde

a mensagem de saída deve ser: `Invalid Expression`.

- Todos os casos de teste válidos irão gerar resultados que são números inteiros. Mesmo na ocorrência de uma divisão.
- Todo valor numérico de entrada tem apenas um único dígito, porém o valor de saída pode ter quantos dígitos forem necessários.

Exemplo de Entrada	Exemplo de Saída
235+3*+	26

Exemplo de Entrada	Exemplo de Saída
32*1-	5

Exemplo de Entrada	Exemplo de Saída
4*	<code>Invalid Expression</code>