

Notação Polonesa — Portal

 200.17.137.109/novobsi/Members/rodrigo/laboratorio-de-programacao/exercicios/exercicio-9/

Na notação usual de expressões aritméticas, chamada notação "infixa", os operadores são escritos entre os operandos. Por exemplo:

$7 - 4 + 2$

O problema é que podemos ter expressões ambíguas, por exemplo, a última expressão acima poderia ser interpretada como:

$((7 - 4) + 2) = 5$

ou como

$(7 - (4 + 2)) = 1.$

Alternativamente, na Notação Polonesa, ou notação prefixa, os operadores são escritos antes dos operandos e, nesse caso, os parênteses não são necessários pois não há ambiguidade. As duas últimas expressões acima, por exemplo, são escritas respectivamente como

$+ - 7 4 2$

e

$- 7 + 4 2$

Em qualquer que seja a notação, a expressão continua sendo avaliada de "dentro para fora", sendo que, na notação prefixa, quanto mais "interna" a operação, mais à direita ela aparece. Assim, o primeiro operador é o mais externo e o último é o mais interno.

A avaliação de uma expressão aritmética em notação prefixa com operadores estritamente binários é feita através do seguinte algoritmo:

Leia a expressão da direita para a esquerda

Para cada token lido

```
{
  se for operando então
    empilha-o numa pilha
  se for operador então
    {
      operando1 = desempilha um elemento
      operando2 = desempilha um elemento
      calcula operando1 operador operando2
      empilha o resultado na pilha
    }
}
```

retorna o topo da pilha como resultado

Nosso problema consiste em implementar esse método de avaliação de expressões em notação prefixa.

Formato de Entrada

A entrada consiste de várias linhas, cada uma correspondendo a uma expressão aritmética escrita em notação prefixa. Por simplicidade, temos que:

1) Os operandos são apenas números naturais

- 2) A expressão envolve apenas os operadores binários +, -, *, e /.
- 3) Os tokens são separados por espaços.

DICA:

A função `split()` do python divide uma string em um array de tokens (strings).

Exemplo:

```
"* + 10 20 30" ----> {"*", "+", "10", "20", "30"}
```

Formato de Saída

Para cada expressão do arquivo de entrada, o programa deve imprimir, no arquivo de saída, uma linha com o resultado da avaliação.

Importante:

- 1) O operador "/" deve ser interpretado na aritmética inteira. Assim, por exemplo
 $/ 10\ 4 = 2$.
- 2) Nenhuma expressão conterá uma divisão por zero.

Exemplos

Entrada:

```
* 199 - + 725 148 + 902 885
+ - + 879 608 842 - - 251 43 906
- 484 390
+ 635 + + 114 927 557
- 224 - + 18 309 - 620 683
+ + + 403 408 - 917 853 + + 568 791 + 692 322
/ + 162 661 + 2 + 932 6
/ + - 837 35 - 332 124 - - 260 605 - 751 463
/ + 376 - 466 399 + * 310 707 378
- 782 197
```

Saída:

```
-181886
-53
94
2233
-166
3248
0
-1
0
585
//cursor aqui
```