

Exercício: Algoritmos e Estrutura de Dados I  
Universidade Federal do ABC

## Expressão pós-fixa

Normalmente, expressões matemáticas são apresentadas utilizando a notação *infixa*. Por exemplo:

$$(2 + 4) \times (7 - 2) = 30$$

Além desta, existem as notações *prefixa* e *pós-fixa*. Neste exercício, trataremos da *pós-fixa*. O exemplo anterior seria representado da seguinte forma em notação pós-fixa:

$$2\ 4\ +\ 7\ 2\ -\ \times = 30$$

Uma vantagem desta notação é o fato de dispensar o uso de parênteses para determinar a ordem das operações. A expressão é lida da esquerda para a direita. Quando um operador é encontrado, ele é imediatamente aplicado sobre elementos anteriores. A resolução do exemplo ocorre da seguinte maneira:

- O primeiro operador encontrado é a soma. Então, aplica-se a soma sobre os dois números anteriores:

$$6\ 7\ 2\ -\ \times = 30$$

- O segundo operador é a subtração, que é aplicado sobre os dois números anteriores:

$$6\ 5\ \times = 30$$

- Por fim, é aplicado o último operador sobre os dois números anteriores:

$$30 = 30$$

### Exercício:

Escreva um programa que receba uma expressão pós-fixa e retorne seu resultado.

### Código

Escrever o programa em linguagem C, salve o código no arquivo: **expressao.c**

O programa deverá utilizar uma pilha dinâmica (simplesmente ligada).

## Entrada

O sistema receberá apenas uma entrada no formato *string*, que conterá a expressão *pós-fixa*. A expressão pode ter até 100 caracteres.

Restrições:

- Cada elemento da expressão possui apenas um caractere. Portanto, os números de entrada são inteiros na faixa [0;9].

- Além dos números, a expressão admite cinco operadores, conforme descrito a seguir:

Operador	Uso	Operação realizada	Exemplo
<b>h</b>	[n1] <b>h</b>	$n1 \times 3$	$7h = 21$
<b>+</b>	[n1][n2] <b>+</b>	$n1 + n2$	$64+ = 10$
<b>-</b>	[n1][n2] <b>-</b>	$n1 - n2$	$64- = 2$
<b>f</b>	[n1][n2][n3] <b>f</b>	$(n1 \times n1) - ((n2) / 2) + n3$	$641f = 35$
<b>g</b>	[n1][n2][n3] <b>g</b>	$(n1 \times n2) + n3$	$641g = 25$

- **Importante:** apesar dos números da expressão original serem inteiros, o resultado das operações pode não ser. Devido a isso, considere o tipo **double** para os cálculos internos.

## Saída

Imprimir o resultado final da expressão pós-fixa, com duas casas decimais. Caso a expressão seja inválida, imprimir "ERRO".

Por exemplo, a expressão é "27++" é inválida. A equivalente dela em notação *infixa* seria "(2+7)+". Veja que o segundo operador "+" torna a expressão inválida.

## Exemplos:

	Entrada	Saída
<b>Exemplo 1</b>	5h	15.00
<b>Exemplo 2</b>	23+	5.00
<b>Exemplo 3</b>	93-	6.00
<b>Exemplo 4</b>	23+93-+	11.00
<b>Exemplo 5</b>	23+93--	-1.00
<b>Exemplo 6</b>	234f	6.50
<b>Exemplo 7</b>	234g	10.00
<b>Exemplo 8</b>	234f234g+	16.50
<b>Exemplo 9</b>	33+11f	36.50
<b>Exemplo 10</b>	33+11f1-	35.50
<b>Exemplo 11</b>	33+11f1-25g	76.00
<b>Exemplo 12</b>	33+11fh	109.50
<b>Exemplo 13</b>	33	ERRO
<b>Exemplo 14</b>	27++	ERRO