

## Problema F

### Posfixa

Arquivo fonte: posfixa.{ c | cpp | java | py }  
Autor: Prof. Me. Sérgio Luiz Banin (Fatec São Paulo)

Em geral quando alguém precisa escrever uma expressão aritmética utiliza uma forma de notação conhecida como notação Infixa. Você deve estar pensando: “Caramba! Começou mal esse texto, o que é essa tal de Infixa?”

Bem, não se espante, o nome pode não ser familiar, mas você conhece essa notação. Ao escrever as expressões aritméticas como essas:

$$A + B * 2$$

$$(A + B) * 2$$

$$(A - B) / (C - D)$$

you está usando a notação Infixa. Essa forma de escrita tem esse nome porque os operadores são posicionados no meio da expressão, entre os operandos. Agora, acho que vem outra questão no seu pensamento: “Como assim, essa forma de notação? Por acaso, existe outra?”

Pois é: a resposta é SIM!! Existem as notações Prefixas e Posfixas. E aqui vamos nos concentrar nesta segunda, Posfixa, também conhecida como Notação Polonesa Reversa em homenagem ao matemático polonês Jan Łukasiewicz que a desenvolveu no ano de 1924.

Essa proposta é muito interessante pois elimina a necessidade de uso dos parênteses na expressão. Por exemplo, as expressões acima podem ser reescritas em notação Posfixa como mostrado a seguir

Infixa	Posfixa	Interpretação da Posfixa
$A + B * 2$	AB2*+	Tome A e guarde, tome B e guarde, tome 2 e guarde, multiplique os dois anteriores ( $B*2$ ) e guarde, some os dois anteriores ( $A+B*2$ ). Esse é o resultado.
$(A + B) * 2$	AB+2*	Tome A e guarde, tome B e guarde, some os dois anteriores ( $A+B$ ) e guarde, tome o 2 e guarde, multiplique os dois anteriores. Esse é o resultado.
$(A-B)/(C-D)$	AB-CD-/	Tome A e guarde, tome B e guarde, subtraia os dois anteriores e guarde, tome C e guarde, tome D e guarde, subtraia os dois anteriores e guarde, divida os dois anteriores (que são os resultados $A-B$ e $C-D$ )

Na interpretação acima, A, B, C, D e algarismos são operandos da expressão. Quando usamos o termo “guarde” pense em uma pilha de itens guardados. Cada operando (variável ou algarismo) que você guarda é colocado nessa pilha. Quando você atinge um operador a operação deve ser feita com os dois itens que estiverem no topo da pilha, que são removidos dela, e o resultado obtido fica no topo dessa pilha. Quando a expressão chega ao final, na pilha só existirá um valor, que é o resultado final.

As operações disponíveis são cinco e as prioridades matemáticas de sua execução devem ser respeitadas, ou seja, as operações de maior prioridade são feitas antes.

Saiba que esse sistema Posfixo tem sido muito utilizado em calculadoras eletrônicas e na computação. Os compiladores quando analisam uma expressão aritmética Infixa eles a transformam em Posfixa para efeitos

Figura F.1: Símbolos usados e ordem de prioridade

Operação	Símbolo	Maior prioridade	Potenciação
Adição	+	Média prioridade	Multiplicação e Divisão
Subtração	-	Baixa prioridade	Adição e Subtração
Multiplicação	*		
Divisão	/		
Potenciação	^		

de realização dos cálculos. Como curiosidade saiba que a empresa Hewlett-Packard desenvolveu toda uma geração de calculadoras eletrônicas que utilizavam essa forma de expressão aritmética.

Muito bem, agora que você já conhece o que são as notações Infixa e Posfixa e tem uma noção de que a notação Posfixa é muito importante, chegou a hora de trabalhar.

Escreva um programa capaz de converter expressões aritméticas válidas em Notação Infixa e as converta para Notação Posfixa.

## Entrada

A entrada tem vários casos de teste. Cada caso de teste é um string contendo variáveis, valores literais, operadores aritméticos e parênteses. Nomes de variáveis contém uma única letra maiúscula (de A a Z); valores literais contém um único algarismo (de 1 a 9); estes símbolos ( +, -, \*, /, ^ ) são os operadores aritméticos. Esse string não tem qualquer outro caractere além desses, nem mesmo espaços em branco.

Garante-se que todas as expressões contidas na entrada são válidas em formato Infixo e aceitam a devida conversão para Posfixo.

O arquivo de entrada termina com uma linha contendo um único caractere que é o ponto final "." (sem as aspas).

## Saída

Para cada string da entrada o programa deve imprimir a correspondente expressão Posfixa seguida de um pulo de linha.

### Exemplo de Entrada 1

```
A+B
A/2
A+B*C
(A+B)*C
(A+B)/(C-D)
X^Y/Z
X^(Y/Z)
2*A+B
X+Y^2
(X+Y)^2
.
```

### Exemplo de Saída 1

```
AB+
A2/
ABC*+
AB+C*
AB+CD-/
XY^Z/
XYZ/^
2A*B+
XY2^+
XY+2^
```