

Problema: (BOCA: P2_2016_Q4) Após estudar muita programação, um certo aluno não conseguia mais escrever textos com números normais. Ao invés de usar números simples, ele estava escrevendo em forma de expressões com notação pós-fixada (i.e. operando da esquerda, operando da direita e operador) entre parênteses. Antes de passar o texto para outras pessoas lerem, ele precisa de sua ajuda para converter os valores das expressões entre parênteses para números comuns com precisão de duas casas decimais. Além disso ele precisa imprimir, no final, o maior valor que apareceu no texto.

Atenção: Desconsidere a precedência dos operadores, use a ordem da esquerda para direita como precedência. Portanto, o resultado da primeira operação (da esquerda para a direita) será sempre utilizado como operando da esquerda da próxima operação (ex. $2\ 3 + 6 / = 1$). Conforme o exercício da lista 3.

Por exemplo:

➤ $a\ b +$	representa	$(a + b)$
➤ $a\ b + c /$	representa	$((a + b) / c)$
➤ $a\ b * c / d - e *$	representa	$((((a * b) / c) - d) * e)$
➤ $1.5\ 1 + 2 *$	representa	$((1.5 + 1) * 2)$

- Entrada: um texto terminado com ponto '.' e com algumas expressões a serem avaliadas dadas entre parênteses. Não há outros parênteses no texto além dos que contêm as expressões.
- Saída: o texto com as expressões reduzidas a um valor numérico com precisão de duas casas decimais, e com o maior valor impressor novamente no final. Ver formatação de saída nos exemplos abaixo.

O programa deve ser entregue modularizado, para isso crie:

- a) Uma função que recebe dois números floats e um caractere *op* representando a operação a ser realizada com os dois números e retorna o resultado da operação entre o primeiro e o segundo números. As operações permitidas são as básicas de adição, subtração, multiplicação e divisão (representadas respectivamente por '+', '-', '*' e '/'). Cabeçalho: *float FazOperacao(float num1, float num2, char op);*
- b) Uma função que retorne o valor do resultado da expressão reduzida. Assuma que a função será chamada no início da leitura da expressão (após a leitura do parêntese de abertura) e retornará ao final da expressão (antes do parêntese de fechamento). Cabeçalho: *float ResolveExpressao();*

- Exemplo de Entrada:

Qual sera esse valor ($1.5 \cdot 1 +$), e esse ($502.0 - 1 + 99 + 6 /$), ou esse ($2.75 \cdot 1.1 - 2 * 3 /$). Isso ($2.5 \cdot 2.5 + 5 / 5 *$) eh igual a isso ($1.5 \cdot 3.5 + 5 / 5 *$), mas isso ($500.0 - 1 - 99 + 6 /$) nao eh igual a isso ($2 \cdot 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 *$). $(4 \cdot 2 - 2 * 4 /) != (10 \cdot 1 - 3 / 2 + 5 /)$, era mentira.

- Exemplo de Saída:

Qual sera esse valor 2.50, e esse 100.00, ou esse 1.10. MAIOR:100.00
Isso 5.00 eh igual a isso 5.00, mas isso 100.00 nao eh igual a isso 256.00. MAIOR:256.00
1.00 != 1.00, era mentira. MAIOR:1.00