

Trabalho 1: Calculadora de Expressões Aritméticas

Professores: Dr. Marcelo G. Manzato (mmanzato@icmc.usp.br)
Dr. Gustavo E. A. P. A. Batista (gbatista@icmc.usp.br)
Dr. Adinovam H. M. Pimenta (adinovam@gmail.com)
Estagiário PAE: Diego Silva (diego.fsilva@gmail.com)
Monitores: Felipe A. R. Dias (felipe.alegria.dias@usp.br)
Guilherme C. de Oliveira (guilherme.caixeta.oliveira@usp.br)

1 Descrição

O objetivo deste projeto é desenvolver uma calculadora de expressões aritméticas. As expressões aritméticas a serem avaliadas podem conter operações de soma, subtração, multiplicação e divisão, representadas pelos símbolos $+$, $-$, $*$ e $/$, respectivamente. O seu programa deve aceitar o uso de parênteses nas expressões e resolver as operações binárias de acordo com a precedência dos operadores/ocorrência de parênteses.

Considere as seguintes informações no desenvolvimento do projeto:

- Os operandos são valores reais e não negativos;
- Espaços em branco e tabulações presentes nas expressões devem ser desconsiderados;
- A calculadora lê uma expressão por linha, terminada pelo caractere ';' (ponto e vírgula);
- A cada expressão lida, a calculadora devolve a resposta em seguida;
- O final da execução é indicado pelo caractere ',' (vírgula) no final de uma linha contendo uma expressão;
- As respostas devem exibir duas casas de precisão decimal com aproximação por truncamento;
- Considere que não haverá erros de *overflow*;
- Considere que cada expressão terá, no máximo, 1.000 caracteres, incluindo dígitos, espaços, operadores e etc.
- A calculadora deve ler da entrada padrão e escrever na saída padrão.

Caso uma expressão mal formada seja informada ou aconteça alguma divisão por zero, sua calculadora deve exibir a seguinte mensagem de erro no lugar da resposta: **Expressao incorreta**.

2 Exemplos de Entrada e Saída

A seguir são apresentados exemplos de entrada e saída para que você teste seu código enquanto desenvolve o exercício. Este são apenas exemplos ilustrativos, somente uma pequena parte das operações está representada. Enquanto estiver desenvolvendo, elabore novos testes para validar seu código.

Entrada

```
(3 + 5 * 8) + 1;  
1 + 2 * 3 + 4;  
2.5 * 2;  
(1 + 2) - 1.5;  
((1 + 3) * (3 + 5));  
1 * 2 / 8,
```

Saída

```
44  
11  
5  
1.5  
Expressao incorreta  
0.25
```

3 Instruções Complementares

- O trabalho deve ser desenvolvido individualmente por cada aluno, sendo este responsável por decidir quais estruturas de dados implementar, assim como as melhores opções de implementação;
- A implementação do trabalho será em linguagem C (padrão C90). Pode-se utilizar qualquer IDE/compilador para o desenvolvimento contanto que seja compilado adequadamente no compilador `gcc` usando as *flags* `-Wall`, `-ansi` e `-pedantic`.
- Compacte seu projeto e submeta o arquivo `.zip` no <http://run.codes>. O código da disciplina é: **BL9E**.

4 Critérios de Avaliação

O projeto será avaliado quanto à corretude (faz tudo que deveria fazer?), organização do código (lembre-se de usar o conceito de TAD), legibilidade e documentação interna (comentários no código, indentação, etc.).