

# EP1 - MAC0316 Conceitos de Linguagens de Programação

---

Washington Luiz Meireles de Lima

nUSP: 10737157

Ygor Tavela Alves

nUSP: 10687642

## Instruções

- Primeiramente deve-se compilar este programa utilizando o comando `make`.
- Após o passo da compilação, caso seja necessário apenas traduzir a gramática para uma sintaxe reconhecida pelo parser do racket, execute:

```
./mcalc < arquivoteste
```

- Caso, queira também executar o código traduzido utilizando o parser do racket, execute:

```
./mcalc < arquivoteste | ./direto
```

## Testes

- Todos os arquivos de testes, estão localizados na pasta `tests` e estão nomeados com o seguinte padrão `test-<tipo>-<numero>`, onde `tipo := div | if | func` e `numero := 1 | 2 | 3`, ou seja, um possível teste para a operação de divisão está nomeado como `test-div-2`.
- Há um shell script básico na pasta raiz, `tests.sh`, que realiza a execução de todos os testes, imprimindo as suas saídas no terminal. Para executá-lo basta executar:

```
./tests.sh
```

- Lembrando que o `tests.sh` deve ter permissão para execução, para isto, basta utilizar o comando `chmod u+x tests.sh`

## Documentação

Nos códigos estão presentes alguns comentários sobre o código em si, além disso, as decisões tomadas referente a escolha da nossa gramática é descrita abaixo para cada caso implementado:

1. Operação de divisão: é utilizado notação infixa padrão para realizar tal operação, assim como as outras operações matemáticas simples.
2. Condicionais if: Como pedido no enunciado do EP, a expressão falsa é interpretada quando a condição for igual a 0, caso contrário, é interpretado a expressão verdadeira. Em relação à sintaxe, foi optado pela seguinte gramática a ser traduzida:

**if** condição **then** expressão verdadeira **else** expressão falsa

3. Funções: Inicialmente as funções foram pré-definidas, estando a disposição apenas as funções twice(dobra o argumento), squared(quadrado do argumento) e factorial(fatorial do argumento), em todos os casos para executar uma função deve-se utilizar o nome da função, seguido de uma abertura de '()' com o argumento dentro do parênteses, exemplos:

**factorial(7)**

**twice(4)**