

---

Roberto Sadao Yokoyama e Amaury Kruehl Budri

## 1 Descrição do Projeto

O presente trabalho tem como objetivo a implementação da **Calc** (escolha um nome para sua Calc), uma calculadora simples capaz de realizar operações com números inteiros de tamanho arbitrário (limitados apenas pela memória principal do computador). Cada grupo deverá personalizar a sua própria versão da calculadora.

## 2 Requisitos da Calculadora

### 2.1 Entrada

A **Calc** deve permitir a leitura dos operandos tanto via teclado quanto por meio de arquivos texto. Os operandos serão números inteiros na base decimal, com  $N > 0$  dígitos.

### 2.2 Operações

O programa deve apresentar um menu de operações ao usuário. As operações obrigatórias são:

- Soma (+)
- Subtração (-)
- Multiplicação (\*)
- Divisão Inteira (/)
- Módulo (%)
- Operação Personalizada

### 2.3 Saída

O resultado da operação deve ser exibido conforme o modo de entrada:

- Entrada via teclado → saída exibida na tela;
- Entrada via arquivo → saída registrada em um novo arquivo texto.

### 2.4 Encerramento

A execução da **Calc** deve ser encerrada quando o usuário selecionar a opção “Sair”. Enquanto essa opção não for escolhida, será possível realizar múltiplas operações sobre os operandos carregados ou novos operandos.

## 3 Definições e Exemplos

### 3.1 Exemplo de uso

Operando 1	Operação	Operando 2	Saída
1001	+	19	1020
2025	-	-25	2050
-22	*	10	-220
1980	/	45	44
45	%	8	5

### 3.2 Definição do módulo

Sejam  $a$  (dividendo) e  $n$  (divisor). Existem inteiros únicos  $q$  (quociente) e  $r$  (resto) tais que:

$$a = q \cdot n + r, \quad 0 \leq r < n.$$

## 4 Operação Personalizada

Cada grupo deve implementar uma operação extra de sua escolha. Exemplos:

- Fatorial ( $n!$ )
- $\text{Xor}(a, b)$
- Máximo Divisor Comum (MDC)
- Fatoração (ex.:  $793 = 13 \cdot 61$ )
- Ordenação de  $n$  números ( $n \geq 10$ )
- Multiplicação de matrizes

A operação personalizada deve possuir maior complexidade que as operações básicas. Por exemplo, a operação de potência ( $\wedge$ ) não será aceita.

## 5 Regras de Implementação

- A implementação deve estar o mais próximo possível da Linguagem C padrão (ANSI C).
- Não é permitido o uso de bibliotecas que implementem números grandes (`BigInt`).
- A estrutura de dados e os algoritmos para representar e manipular os números são de livre escolha, desde que:
  - Utilizem memória e processamento de forma eficiente;
  - Evitem alocações desnecessárias ou redundantes;
  - Evitem cópias de grandes estruturas de dados;
  - Reduzam laços de repetição desnecessários.
- A operação personalizada deve utilizar a mesma estrutura de dados escolhida, embora o resultado possa ser armazenado em outro tipo de estrutura.

## 6 Entrega e Apresentação

### 6.1 Entrega via Moodle

O grupo deverá enviar:

- Os arquivos fonte necessários para a compilação;
- Um arquivo `Makefile` e/ou `leia-me.txt` com instruções de compilação (parâmetros, bibliotecas, compilador etc.);
- Um relatório em PDF contendo:
  1. Requisitos implementados e funcionando;
  2. Casos de teste descritos e ilustrados com capturas de tela da execução da `Calc`;
  3. Detalhamento de correções ou modificações feitas após a apresentação (incluindo trechos de código e explicações).

### 6.2 Apresentação em Laboratório

- Reservar previamente a data e o horário no cronograma da disciplina;
- Preparar ao menos 3 casos de teste;
- Compilar e executar a `Calc`, demonstrando os testes preparados;
- Responder às perguntas da arguição, demonstrando domínio sobre o trabalho.

## 7 Critérios de Avaliação

A avaliação seguirá critérios que variam do conceito F até A, considerando:

- Capacidade de compilação e execução;
- Implementação das operações requeridas;
- Uso adequado de memória e processamento;
- Implementação correta e funcional da operação personalizada;
- Clareza nas respostas durante a arguição.