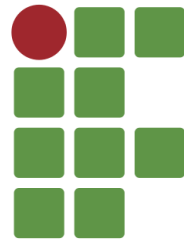


*Acertou Miseravi*

Programação de Computadores I

Ciência da Computação

Prof. Daniel Saad Nogueira Nunes



**INSTITUTO  
FEDERAL**  
Brasília

# 1 Introdução

A aritmética é uma parte importante da matemática que lida com operação sobre números. Neste projeto deverá ser implementado três operações básicas sobre inteiros:

- Adição.
- Subtração.
- Multiplicação.

Contudo, uma observação precisa ser feita: os inteiros podem ser arbitrariamente grandes!

## 2 Especificação

O projeto deve ser elaborado em um único arquivo fonte, o qual pode ser modularizado através de várias funções.

### 2.1 Entrada

A primeira linha da entrada possui um inteiro  $n$ , o qual indica a quantidade de casos de teste.

As próximas  $n$  linhas, contém cada, uma pergunta, que é descrita por três inteiros:  $a$ ,  $b$  e  $o$ .

- Se  $o = 1$ , então os números  $a$  e  $b$  devem ser somados ( $a + b$ ).
- Se  $o = 2$ ,  $b$  deve ser subtraído de  $a$  ( $a - b$ ).
- Se  $o = 3$ ,  $a$  deve ser multiplicado por  $b$  ( $a \cdot b$ ).

**Restrições:**

- $1 \leq n \leq 10^3$
- $0 \leq a, b \leq 10^{500}$ .
- $1 \leq o \leq 3$ .
- Nenhum número será informado com zeros à esquerda.

### 2.2 Saída

Para cada pergunta, o resultado da operação deve ser impresso em uma linha. Não deverá haver zeros mais à esquerda.

### 2.3 Exemplo

Abaixo temos um exemplo com a entrada e a saída esperada.

### Entrada:

```
7
1 1 1
1 2 2
10 10 3
127349847218943757471829 9942174127482147421798427421 1
127349847218943757471829 9942174127482147421798427421 3
1 1000 2
1234 1234567890123456789
```

### Saída:

```
2
-1
100
9942301477329366365555899250
1266134356158986930307618580257140119125290908623009
-999
-1234567890123455555
```

## 2.4 Documentação

O código deve ser bem documentado, com presença de comentários explicando os trechos mais complexos do código. Além disso, um arquivo README deve ser providenciado com a devida identificação do autor descrevendo o projeto e instruindo como o código deve ser compilado.

## 2.5 Modularização

É importante que o programador divida o programa em várias funções, para facilitar o entendimento, o desenvolvimento e a utilização dos módulos do programa por terceiros.

## 3 Critérios de correção

Deve ser utilizada a linguagem de programação C para a implementação do caça-palavras.

Para validação da correção do algoritmo, testes automatizados serão realizados, então é **crucial** que a saída esteja conforme o especificado.

Serão descontados pontos dos códigos que não possuírem indentação.

### 3.1 Ambiente de Correção

Para a correção dos projetos, será utilizada uma máquina de 64-bits com sistema operacional GNU/LINUX e compilador GCC 10.2.0, logo é imprescindível que o sistema seja capaz de ser compilado e executado nesta configuração.

## 4 Considerações

- GDB, Valgrind e ferramentas gráficas associadas podem ajudar na depuração do código.
- Este trabalho deve ser feito **individualmente**.
- Não serão avaliados trabalhos que não compilem.
- Como parte da correção é automatizada, deverá ser impresso apenas o que a especificação pede. Atentem-se para a formatação da saída.
- A incidência de plágio será avaliada automaticamente com nota 0 para os envolvidos. Medidas disciplinares também serão tomadas.
- O trabalho deve ser entregue dentro de uma pasta zipada com a devida identificação do aluno no prazo combinado pelo ambiente virtual de aprendizagem da disciplina.