

## Lista 1 - Programação em C

Os itens da lista possuem ordem crescente de dificuldade, exceto pelos desafios. Os itens comuns da lista são para os iniciantes, e os desafios para quem quiser se aprofundar e quem já possui conhecimento prévio em C. Os códigos desenvolvidos para resolução dos exercícios devem ser colocados no GitHub pessoal de cada membro. Os arquivos necessários para resolução da lista podem ser encontrados em: [https://github.com/rafael-pastre/Treinamento\\_Programacao\\_C](https://github.com/rafael-pastre/Treinamento_Programacao_C)

1. Faça um programa que leia 3 números inteiros do terminal e apresente no terminal sua soma, média, e desvio padrão.
  - a. Desafio: Como fazer seu programa funcionar para N números, sendo que N é um valor definido em um macro “#define”?
  - b. Desafio: Como fazer seu programa funcionar para N números, sendo que N é um valor lido do terminal?
2. Modifique seu programa, de forma que ele utilize funções para retornar a soma, média e desvio padrão.
3. Leia os dados encontrado no arquivo “dados1.csv”, e para cada linha do arquivo, escreva os resultados da soma, média e desvio padrão em uma linha de outro arquivo chamado “resultados.csv”. Todos os números no arquivo de entrada são inteiros e estão separados por vírgulas. É sabido que o arquivo de entrada é composto somente por números inteiros e possui exatas 10 linhas.
  - a. Desafio: O arquivo “dados2.csv” possui um número desconhecido de linhas, faça um programa que execute o mesmo procedimento indicado para o arquivo “dados1.csv”, para todas as linhas existentes.
  - b. Desafio: O arquivo “dados3.bin” é um arquivo binário de 3 bytes onde cada byte é um número inteiro sem sinal. Encontre a soma, média e desvio padrão destes números.
4. Crie uma função que calcule as divisões do 1º e 2º número pelo 3º. Os resultados devem ser passados por referência, e a função deve retornar “-1” caso tenha ocorrido algum erro, e “0” em caso de sucesso.
5. Separe as funções criadas até aqui do arquivo principal fazendo uma biblioteca para elas. Para isso, coloque os cabeçalhos das funções em um arquivo “funcoes.h”, e a implementação das funções em um arquivo “funcoes.c”, e inclua sua biblioteca no arquivo principal utilizando “#include “funcoes.h””.
  - a. Desafio 1: O que são os macros para proteção de uma biblioteca? Implemente eles na “funcoes.h”.
  - b. Desafio 2: Se trocássemos a linguagem de programação para C++, como poderíamos fazer para a nossa biblioteca ter compatibilidade? Dica: pesquise pelo macro “\_\_cplusplus” e pelo comando ‘extern "C"’
6. Desafio Final: Sabendo que temos diversos tipos de dados na linguagem c, como fazer que as funções que implementou funcionem com qualquer tipo de dado? E se precisássemos chamar a função 2 vezes com tipos diferentes de dados em cada uma delas, como isso poderia ser feito? Obs: procure sobre diretivas de compilação.