

LISTA DE EXERCICIOS 01 - UNIDADE 2

PROGRAMACAO IMPERATIVA E FUNCIONAL - TURMA A E B - 2021.2

- **Deadline:** Segunda-feira, 25/10/2021, as 23:59.
 - **Conteúdo:** Pre-processadores C, bibliotecas, leituras e escrita de arquivos, structs.
 - **Linguagem:** C
 - O arquivo correspondente de cada questão deverá iniciar com a letra q em minúsculo, seguida pelo número da questão, *underline*, seu login (ou seja, as iniciais do seu e-mail do CESAR School) e com a extensão .c (exemplo, q1_ptlb.c). **Qualquer arquivo enviado fora desse padrão será automaticamente descartado;**
 - Os arquivos deverão ser salvos dentro de um diretório cujo nome será o seu **login do CESAR School**. Este diretório deverá ser compactado em um arquivo **.zip** e submetido no Google Classroom de PIF de sua respectiva turma (atente a qual turma você pertence). **Qualquer arquivo enviado fora deste padrão será automaticamente descartado;**
 - Não é permitida a importação de bibliotecas exceto as criadas por você, stdio.h, stdlib.h, e string.h.
 - O nome das funções implementadas **deve ser igual ao indicado pela questão**, caso contrário o script de correção não conseguirá identificar sua questão e ela será automaticamente zerada.
-

1- Faça um programa em C que recebe um array de inteiros do usuário e aplique uma destas funções:

- a. newReverse: retorna o inverso do array;
- b. elemNum: retorna o número de vezes que o elemento x aparece no array;
- c. unique: retorna um array apenas com os elementos que não se repetem;
- d. deleteN: remove o enésimo elemento de um array;

Seu programa deve ter:

- i. Instruções que pedem o tamanho do array e seus elementos ao usuário;
- ii. Instruções que perguntem ao usuário que operações ele deseja fazer com o seu array (menu de opções), sendo R = inverter a lista (*newReverse*), E = contar o número de vezes que um elemento aparecer no array (*elemNum*), U = retorna apenas os elementos que não se repetem (*unique*), D = deleta o enésimo elemento (*deleteN*). Os resultados devem ser apresentados na tela.
- iii. Instruções que perguntem ao usuário se ele deseja salvar os resultados obtidos em um arquivo chamado “resultados.txt”;
- iv. Todas as suas funções devem ser implementadas em uma biblioteca chamada “**operaArray.h**”. Esta biblioteca deve ser chamada em seu programa através da diretiva #include.

2- Desenvolva um programa em C que implemente um sistema bancário. Seu programa deve ter:

- i. Instruções que leiam dados bancários de vários clientes em um arquivo chamado “CLIENTES.txt” (exemplo no classroom). Estes dados incluem (1) número da conta (int), (2) nome (string) e sobrenome (string) do cliente, e (3) saldo (float).
- ii. Instruções que leiam diversas transações bancárias presentes em um arquivo chamado “TRANSACTIONS.txt”. Os dados deste arquivo incluem (1) número da conta (int), (2) tipo da transação (char – positiva = entrada de dinheiro, negativa = saída de dinheiro), e (3) valor da transação (float);
- iii. Instruções que calculem o novo saldo dos clientes após realizarem as transações contidas em “TRANSACTIONS.txt”.
- iv. Instruções para salvar em um arquivo “DEVEDORES.txt” apenas os clientes com saldo negativo. As informações contidas neste arquivo devem incluir (1) número da conta (int), (2) nome e sobrenome do cliente (String) e (3) saldo atual;
- v. Defina uma struct **dadosClientes** para guardar informações sobre clientes;