

Exercício 2 - TAD

Sobre a entrega:

- exercício **INDIVIDUAL**
- submissão no **run.codes** – veja instruções no arquivo **run_codes.pdf**
- até **07/09/2020**, 23:59h

Objetivos: revisar conceitos básicos de TAD enfatizando a independência de implementação e o ocultamento de informações.

Implementação

Problema: implementar, em linguagem C, a versão 2 do TAD Sacola apresentado em aula.

Simplificações:

- considere que o elemento armazenado é do tipo *char*

Projete e implemente a versão 2 do TAD Sacola, que armazena um par contendo o elemento **Base** e um inteiro para o número de ocorrências (conforme apresentado em aula). O TAD deve conter as funções declaradas na definição do TAD (**sacola.h**) que são:

- **Criar**
- **Limpar**
- **Contar**
- **Inserir**

Requisito:

- você deverá implementar o arquivo **sacola2.c**, utilizando os arquivos **main.c**, **sacola.h** e **Makefile** (disponibilizados no Tidia), **sem alterá-los**.

No Tidia, também está disponível o arquivo **sacola.c**, que implementa a versão mais simples (versão 1) do TAD apresentada em aula. A objetivo é que o mesmo programa em **main.c**, usando o mesmo *header* **sacola.h**, funcione tanto com a implementação em **sacola.c** quanto com versão 2, que você irá implementar em **sacola2.c**.

Teste na sua máquina as duas versões, trocando apenas o arquivo **.c** no **Makefile**, o que ilustrará a independência de implementação (aplicação/programa principal não precisa ser alterado se a implementação da estrutura muda).

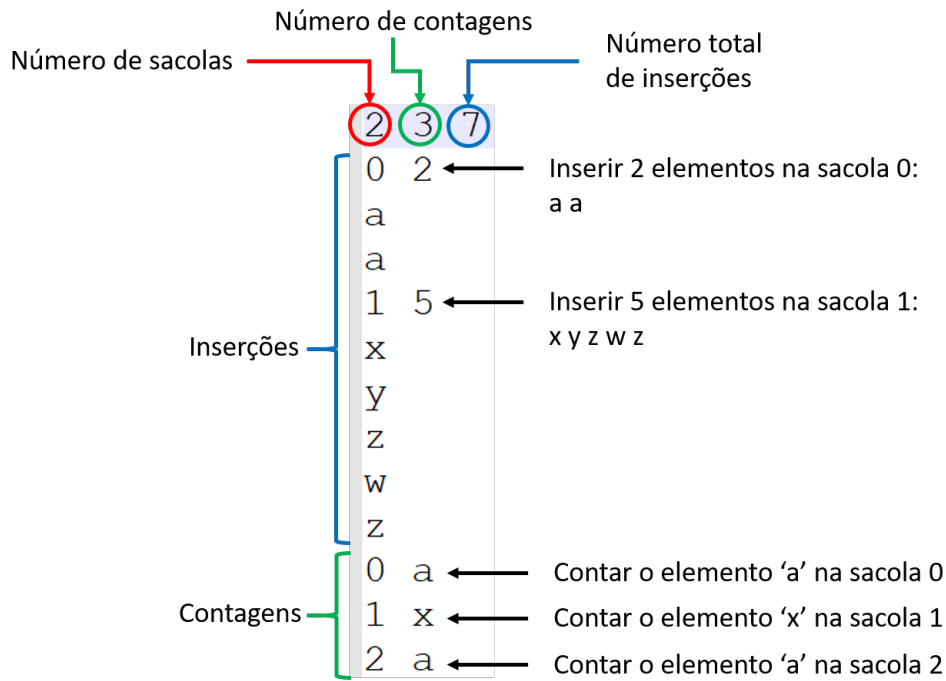
Entregar:

- código (em zip) com: **main.c**, **sacola.h**, **Makefile** e **sacola2.c** (lembrando que os 3 primeiros devem ser exatamente os arquivos disponibilizados no Tidia).

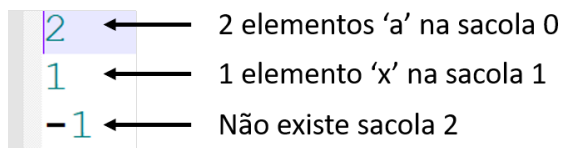
PS: não é necessário fazer relatório para este exercício.

Exemplo de Entrada e Saída Esperada

Entrada



Saída



IMPORTANTE:

- ✓ Lembre-se e aplique as “boas práticas de programação”, tais como: organizar e documentar bem o código, definir adequadamente os nomes de variáveis e funções, modularizar o código, identificar autor do código em todos os arquivos, etc...
- ✓ Nomeie os arquivos de modo a facilitar correção do código.
- ✓ No relatório, inclua nome, número USP, identificação do exercício, e apresente as informações requisitadas de modo claro, objetivo e completo.