



Universidade Federal de Santa Catarina Campus Araranguá - ARA
Centro de Ciências, Tecnologias e Saúde
Departamento de Computação
Linguagem de Programação I -- "DEC0012"

DESCRIÇÃO DO SEGUNDO TRABALHO PRÁTICO

Modalidade: trabalho em dupla ou individual. As duplas não precisam ser as mesmas do primeiro trabalho.

1. Enunciado (ementa semi-aberta):

Desenvolver um sistema para gerenciamento de uma empresa rodoviária de transporte. Definir um tipo de atuação para sua empresa (frota de ônibus, cargas, veículos de locação, etc). O sistema deve permitir:

- Cadastro e remoção de cidades. Cada registro deve ter: (código da cidade, cidade, estado);
- Cadastro e remoção de veículos rodoviários. Os registros irão variar de acordo com o tipo de atuação da sua empresa. Os registros deverão ter pelo menos: (código do veículo, placa, <outras *min.2* informações que irão variar de acordo com a atuação>);
- Cadastro de rotas. Deverá ser possível cadastrar rotas. Cada rota deverá ter um código, nome da cidade, distâncias entre essas cidades, e outras informações que dependerão da atuação da sua empresa, ex: valor de pedágio depende da quantidade de eixos do veículo;
- Cadastro de itinerário. Um itinerário é composto entre uma rota, e um veículo que fará esta rota, assim como informações sobre o que se está transportando (depende da atuação). A rota deverá ter um nome, geralmente formado entre a **cidade1-cidade2**;
- Relatório dos itinerários contendo as cidades e distâncias;
- 02 Relatórios financeiros, a escolha do aluno, dependendo da atuação da empresa.**

Atenção:

- quando utilizada uma cidade (a) em alguma rota (c), a mesma não mais poderá ser removida em (c). Se deseja remover uma cidade, remover primeiro a dependência (uso desta na rota);
- mesma situação para veículo. Só pode ser removido se não estiver em uso no itinerário, em (d).

2. Requisitos:

- Desenvolver em linguagem C;
- Utilizar structs e modularização;
- Todas as informações, após cadastradas, devem ser armazenadas em arquivos de **texto** ou **binários**. O sistema deve portanto, ser persistente, e toda vez que for executado, recuperar as informações de execuções anteriores e carregar no programa nas *structs*. Cada tipo de cadastro deve ter seu próprio arquivo para armazenamento;

(DICA: não precisa "garimpar" no arquivo em disco as informações para cadastrar ou remover. Simplesmente, apagar o arquivo anterior e salvar tudo novamente. O importante é não perder as informações anteriores);

- Utilizar alocação dinâmica de memória. Toda a vez que o arquivo necessitar ser lido ou um novo cadastro realizado, alocar com **malloc** ou **calloc**;
- Liberar memória a cada remoção, ou ao sair do sistema, com **free()**, evitando *memory leaks* (vazamento de memória).

3. Entregáveis:

- a) Apresentação (somente do código e execução do mesmo) a ser realizada presencialmente para o professor em sala no dia 16/12/2022, horário da aula;
- b) A dupla (se for o caso) deve apresentar para validar a nota;
- c) Código a ser postado no Moodle até a data de apresentação. Deve compilar (será utilizado `gcc *.cpp -o exe` pelo professor);
- d) Trabalhos similares = ZERO.