

Trabalho 2

LEIA ATENTAMENTE AS REGRAS E O ENUNCIADO

R E G R A S

- O trabalho deverá ser realizado individualmente ou em dupla.
- Não serão aceitos trabalhos individuais ou com mais de 2 pessoas.
- O trabalho deverá ser enviado para o Google classroom até o dia 16/07/2023 (domingo).
- A data de entrega não será adiada.
- O nome do programa deverá ser, obrigatoriamente, locacao.cpp.
- O arquivo solicitado deverá ser ZIPADO (ZIP ou RAR) com o NOME E SOBRENOME DO ALUNO (para duplas colocar nome e sobrenome dos DOIS alunos).
- Caso o trabalho seja feito em dupla, apenas UM dos alunos deverá postar a solução no Classroom.
- Deverá ser compactado somente o arquivo .CPP (não enviar arquivos executáveis!)
- Não serão aceitos trabalhos enviados por email.
- Trabalhos com estruturas e/ou organizações semelhantes (plágio) serão penalizados com a nota zero.

ENUNCIADO

1) Programa: locacao.cpp

Crie um programa para administrar uma locadora de veículos, conforma especificação a seguir. O programa deve ter mesmos menus e mesmos layouts da interface com o usuário que foram implementados no protótipo **locacao.exe**, disponível para baixar no link:

<https://www.dropbox.com/scl/fi/b0ptajcff8jmv35sr5ayj/locacao.exe?dl=0&rlkey=mhaz87b0aqo2gid6dn8gf74kp>

Importante: esse executável foi compilado para Windows.

Inclusão de clientes: são necessários CPF, nome e data de nascimento.

- a) CPF deve ser válido.
- b) Nome do usuário deve ter pelo menos 4 caracteres.
- c) Data de nascimento deve ser fornecida no formato DDMMAAAA.
- d) Caso algum dado seja inválido, deve ser apresentada uma mensagem de erro e o dado deve ser solicitado novamente.
- e) Não podem existir dois clientes com o mesmo CPF.
- f) Cliente deve ter 18 anos ou mais no momento do cadastro (data atual do computador).

Inclusão de veículos: são necessários placa, tipo, modelo, ano de fabricação, valor da locação/dia e quilometragem.

- a) Placa: deve ter o formato XXX9999 onde XXX são letras e 9999 são dígitos.
- b) Não podem existir dois veículos com a mesma placa.
- c) Tipo do veículo: (C) Carro ou (M) Moto.
- d) Modelo deve de 4 caracteres a 30 caracteres.
- e) Ano de fabricação ≥ 2000 e ano de fabricação \leq ano atual.
- f) Valor da locação/dia deve ser um real > 0 .
- g) Quilometragem deve ser inteiro > 0 .
- h) Não podem existir dois veículos com a mesma placa.
- i) Caso algum dado seja inválido, deve ser apresentada uma mensagem de erro e o dado deve ser solicitado novamente.

Exclusão de clientes da locadora

- a) Para excluir um cliente, basta fornecer seu CPF.
- b) Um cliente que tem uma locação em curso não pode ser excluído.

Exclusão de veículos da locadora

- a) Para excluir um veículo basta fornecer sua placa.
- b) Um veículo que tem uma locação em curso não pode ser excluído.

Locação de veículo: são necessários o CPF do cliente, a placa do veículo e o número do cartão de crédito.

- a) CPF e placa devem existir no cadastro.
- b) Cada cliente só pode realizar uma locação por vez, ou seja, se o cliente já estiver com uma locação em curso ele não poderá fazer outra.
- c) Cada veículo só pode fazer parte de uma locação por vez, ou seja, se o veículo já está associado a uma locação em curso ele não pode ser locado em outra.
- d) Deve ser armazenada a data em que a locação teve início (data atual do computador).

Devolução de veículo locado: é necessário o CPF do cliente, a data da devolução e a quilometragem do veículo no ato da devolução.

- a) Data da devolução \leq hoje e data da devolução \geq data da locação.
- b) Cliente deve ter uma locação em curso.
- c) Quilometragem da devolução \geq quilometragem da locação.
- d) Valor a pagar = (número de dias locados x valor do veículo/dia) + (quilômetros rodados x 1,50).

Listagem de clientes

- a) A consulta de clientes deve ser apresentada conforme o layout definido no final desse documento e pode estar ordenada de forma crescente por CPF ou nome, à escolha do usuário.
- b) Se o usuário possuir uma locação em curso, os dados da locação devem ser apresentados abaixo dos dados do usuário.

Listagem de veículos

- a) A consulta de veículos deve ser apresentada conforme o layout definido no final desse documento e deve estar ordenada de forma crescente por placa.
- b) Se o veículo estiver associado a uma locação em curso, os dados do cliente que alugou devem ser apresentados abaixo dos dados do veículo.

Listagem de locações

- a) A consulta de locações deve ser apresentada conforme o layout definido no final desse documento e deve estar ordenada de forma crescente por CPF do cliente.

Informações gerais:

- a) A quantidade máxima de clientes e veículos é 50.
- b) Nome do cliente deverá ter, no máximo, 50 caracteres.
- c) Modelo do veículo deverá ter, no máximo, 30 caracteres.
- d) Não é necessário armazenar o histórico de locações por cliente ou veículo, basta armazenar a locação em curso.
- e) Todas as strings deverão ser armazenadas em caixa alta (nome do cliente e modelo do veículo).
- f) A implementação deve seguir o layout de interface definido no protótipo **locacao.exe**.
- g) Não poderão ser utilizadas variáveis globais. Todos os vetores deverão ser criados na função **main** e passados para as demais funções via parâmetros.

Dicas:

- Antes de iniciar a programação, crie as structs necessárias para resolver o problema e verifique se essas representações atendem às operações descritas no enunciado.
- Se desejar, utilize o tipo **long long** para CPF.
- Crie funções básicas para leitura dos dados e implemente nessas funções a validação dos dados. Exemplos:
 - Função para ler um número inteiro.
 - Função para ler um número inteiro no intervalo A..B (ano de fabricação)
 - Função para validar número do CC.
 - Função para ler a placa do veículo no formato XXX9999
 - Função para ler um data no formato DDMMAAAA.
 - Função para ler uma string com no mínimo n e no máximo k caracteres (nome do cliente, modelo do veículo, número do cartão de crédito).
 - Função para ler uma data no formato DDMMAAAA.
- Crie funções para cada uma das ordenações solicitadas.
- Crie funções para cada um dos menus (principal e secundários).
- Crie funções para cada uma das operações do programa descritas anteriormente (incluir cliente, incluir veículo, locar veículo, devolver veículo, consultar clientes, etc.).
- Nos exercícios resolvidos no material de sala de aula e nas listas de exercícios propostas encontram-se vários problemas semelhantes aos propostos nesse trabalho. Estude-os!
- Para recuperar a data/hora atual do computador use a função a seguir:

```
#include <time.h>

void data_hora_atual(int &dia, int &mes, int &ano,
                    int &hora, int &min, int &seg) {
    time_t t = time(NULL);
    struct tm lt = *localtime(&t);

    ano  = lt.tm_year + 1900;
    mes  = lt.tm_mon + 1;
    dia  = lt.tm_mday;
    hora = lt.tm_hour;
    min  = lt.tm_min;
    seg  = lt.tm_sec;
}
```