Projeto – Locadora de Veículos

Uma empresa de locação de veículos está contratando você para desenvolver seu sistema de locação de veículos. Segue abaixo as especificações que você deve seguir para desenvolver o sistema solicitado. Inicialmente, é necessário entender as entidades envolvidas no processo, as quais são descritas abaixo.

* Cliente: um cliente é uma pessoa que possui nome, endereço e telefone, sendo o endereço composto por rua, número, bairro e cidade. O telefone segue o formato: 19-9- XXXX-XXXX. Um cliente ou é uma pessoa física ou uma pessoa jurídica (fazer validação). Sendo pessoa física, o que o identifica é o seu CPF (formato: XXX.XXX.XXX-XX) (chave de busca). Sendo uma pessoa jurídica, o que o identifica é o seu CNPJ (formato: XX.XXX.XXX/XXXX-XX) (chave de busca). Para facilitar, supõe-se que é possível cadastrar no máximo 15 clientes.
* Carro: um carro é descrito pelas seguintes características: renavam (formato: XXXXXXXXXXX) (chave de busca), placa (formato: LLL-NNNN), marca, modelo, ano, cor e categoria. As cores possíveis são: preta (1), branca (2) e prata (3) (fazer validação). As categorias possíveis são: hatch (1), sedan (2), suv (3) e jipe (4) (fazer validação). Para facilitar, supõe-se que é possível cadastrar no máximo 10 carros.
* Cada categoria de carro está associada as seguintes informações: valor da diária e pontos fidelidade. O valor da diária é baseado, portanto, na categoria do carro. Cada categoria gera um pontuação ao cliente, que poderá ser utilizada para ganhar descontos quando for realizar futuras locações. Fazer validação para a categoria informada.
* Locação: uma locação é descrita pelas seguintes características: código da locação, tipo do cliente (física ou jurídica) (fazer validação), cliente (CPF ou CNPJ) (fazer validação), carro (Renavan) (fazer validação), data de retirada e data da devolução. O código deverá ser gerado automaticamente (auto incremento) e começará na numeração 1001. As datas são compotas por dia, mês e ano (exibir para o usuário no formato dd/mm/aa). Enquanto o carro não é devolvido, a data de devolução fica em 00/00/00.
* Cada cliente poderá locar no máximo 5 veículos. Haverá uma matriz que armazenará, para cada cliente, seus históricos de locação. Assim, existe uma matriz 15X5 (cliente X locação) que armazena o histórico de cada locação realizada por cada cliente. Cada célula da matriz armazena os dados de cada locação realizada por um dado cliente.
* Como não se sabe em qual posição da matriz um dado cliente se encontra, a fim de facilitar as buscas, existe um índice (como um índice de um livro) que armazena para cada cliente (identificado ou pelo CPF ou pelo CNPJ), a posição que o mesmo se encontra na matriz (linha). Assim, existe um vetor índice que armazena, em cada posição, o tipo do cliente (física ou jurídica), o identificador do cliente (CPF ou CNPJ) e a posição que o mesmo se encontra na matriz (linha).

O seu sistema disponibiliza as seguintes funcionalidades:

* Cadastro de Cliente: cadastra um cliente. Inicialmente é necessário informar se o cliente é uma pessoa física ou jurídica (fazer validação), para então continuar o cadastro. Não pode existir mais de um cliente com o mesmo CPF/CNPJ (chave de busca).
* Cadastro de Carro: cadastra um carro. Não pode existir mais de um carro com o mesmo Renavam (chave de busca). Não esqueça de validar as entradas para cor e categoria.
* Cadastro de Dados da Categoria: cadastra os dados referente a uma dada categoria de carro. É necessário verificar se a categoria informada existe.
* Cadastro de Locação: realiza uma locação. Como mencionado, no momento da locação a data de devolução fica em 00/00/00. Além disso, o código da locação (chave de busca) será auto incremento, como acima descrito. Note que a locação fica armazenada em um célula da matriz, acima descrita, que armazena o histórico de locação de cada cliente.

É necessário atualizar também o índice que guarda, para cada cliente, sua localização na matriz. É necessário checar se o cliente e o renavam informados na locação já foram previamente cadastrados, assim como se o tipo de cliente informado é válido. Além disso, é necessário checar se o renavam informado (i.e., o carro desejado) já não está sendo utilizado em uma locação em aberto (não é possível mais de um cliente locar o mesmo carro).

Caso contrário, a locação não é realizada. Para facilitar, supõe-se que um cliente só pode locar um veículo se ele não tiver nenhuma pendência, i.e., nenhuma locação em aberto (é necessário fazer a verificação).

* Devolve Carro: informa-se se pessoa física ou jurídica e o CPF/CNPJ do cliente e realiza- se a baixa da locação. Para realizar a baixa, atualiza-se a data de devolução.

Observe, em função das regras de locação, que ele sempre tem apenas uma locação em aberto (ou não tem nenhuma devolução a ser realizada). Note que a data de devolução deve ser posterior a data de retirada (fazer validação). Caso a devolução seja realizada no mesmo dia entende-se que há um cancelamento (neste caso, é necessário remover a locação do histórico do cliente (não aparecerá em nenhuma busca futura)). Assim que o cliente devolve o carro é informado ao mesmo o valor devido. Para tanto, computa-se o número de dias que o cliente ficou com o carro e multiplica-se pelo valor da diária de acordo com a categoria do carro locado. Esse é o valor devido.

Para simplificar, considera-se que todas as operações são realizadas dentro do mesmo ano. Contudo, se o cliente já realizou locações anteriores, consulta-se a última locação realizada (anterior a atual (por ordem de cadastro)) e a pontuação adquirida de acordo com a categoria do carro locado na época. Se a categoria gera uma pontuação de 1500 pontos, por exemplo, o cliente ganha 50,00 reais de desconto no total a ser pago. Supõe-se que a cada 1000 pontos ganha-se 50,00 reais de desconto.

O limite máximo de desconto não pode exceder a 30% do valor devido, i.e., o máximo de desconto deve ser de 30% sob o valor devido.

* Atualiza cadastro de um dado cliente: o usuário informa se deseja atualizar os dados de um cliente pessoa física ou pessoa jurídica, seu respectivo CPF/CNPJ, e, na sequência, realiza a atualização do referido campo de interesse. É possível alterar apenas telefone e endereço. Fazer validações necessárias.
* Lista os dados de um dado cliente: o usuário informa se deseja os dados de um cliente pessoa física ou pessoa jurídica, seu respectivo CPF/CNPJ, e, na sequência, exibe suas informações (nome, endereço e telefone). Fazer validações necessárias.
* Lista dados de uma determinada locação (código da locação, CPF/CNPJ, nome do cliente, renavam e datas): o usuário informa o código da locação e os dados da mesma são mostrados na tela. Contudo, em relação ao cliente, não basta mostrar apenas o CPF/CNPJ, mas também seu nome. Fazer validações necessárias.
* Lista locações de um dado cliente (usar função da opção anterior para exibir os dados): o usuário pode optar por mostrar todas (1), as encerradas (2) ou a em aberto (3). Caso o cliente não possua nenhuma locação em uma dada opção, deve-se informar ao usuário que neste caso não há nenhuma informação cadastrada. O usuário deve informar se o cliente é uma pessoa física ou uma pessoa jurídica e seu respectivo CPF/CNPJ para realizar a busca. Fazer validações necessárias.
* Lista locações em aberto: lista todas as locações em aberto, i.e., aquelas que o cliente ainda não realizou a devolução do carro (código da locação, CPF/CNPJ e nome do cliente). A listagem pode ser separada por tipo de cliente: pessoa física e pessoa jurídica (o usuário informa o tipo desejado). Fazer validações necessárias.
* Frequência de locação por categoria: considerando todas as locações (em aberto ou não), listar, das mais frequentes para as menos frequentes, a frequência com que cada categoria é locada. A listagem pode ser separada por tipo de cliente: pessoa física e pessoa jurídica (o usuário informa o tipo desejado). Fazer validações necessárias.
* Sair: encerra o programa.

Observações:

* Para especificar os tipos de dados é necessário utilizar struct, union (pelo menos 1) e enum (pelo menos 1).
* Para cada opção implementar uma ou mais funções.
* O main apenas gerencia o menu e as chamadas e os retornos das funções.
* Em relação as strings, sempre declarem as mesma com uma posição a mais, uma vez que internamente o C usa um caractere especial para delimitar fim de string.

Exemplo: telefone, que tem formato 19-9-XXXX-XXXX, deve ter tamanho 15 e não 14. Para padronizar, as strings que não tem formato definido devem ser declaradas com tamanho 50.

Observações sobre a correção:

* É necessário passar nos casos de teste. Contudo, a nota não será baseada apenas nos casos de teste. Os códigos serão comparados em termo de similaridade, será checado se o mesmo comtempla as especificações recomendadas e será avaliada também a qualidade de abstração da solução proposta.

Menu de Opções e ordem de entrada:

1 Cadastro de Cliente: tipo cliente > chave cliente > nome > endereço (rua, num, bairro, cidade) > telefone

2 Atualiza cadastro de um dado cliente: tipo cliente > chave cliente > opção (endereço ou telefone)

3 Lista os dados de um dado cliente: tipo cliente > chave cliente > mostrar nome, rua, num, bairro, cidade, telefone

4 Cadastro de Carro: renavam > categoria > cor > placa > marca > modelo > ano

5 Cadastro de Dados da Categoria: categoria > valor diaria > pontos fidelidade

6 Cadastro de Locação: renavam > tipo cliente > chave cliente > dia retirada > mês retirada > ano retirada

7 Devolve Carro: tipo cliente > chave cliente > dia entrega > mes entrega > ano entrega

8 Lista dados de uma determinada locação: código locação

9 Lista locações de um dado cliente: tipo cliente > chave cliente > opção (todas, encerradas ou abertas)

10 Lista locações em aberto: tipo cliente

11 Frequência de locação por categoria: tipo cliente

0 - Sair

Controle de fluxo:

Entradas

As entradas podem ser feitas utilizando:

scanf("%d", &x); // para inteiros

scanf("%u", &x); // para enumerações

scanf("%f", &x); // para pontos flutuantes

scanf(" %s", s); // para strings sem espaços

scanf(" %[^\n]%\*c", s); // para strings com espaços

IMPORTANTE:

**Qualquer discrepância entre as saídas abaixo e as saídas dos casos de testes, por favor, enviei um e-mail informando a discordância para os monitores:** [vinicius.s.bueno@unesp.br](mailto:vinicius.s.bueno@unesp.br) e [ana.klinke@unesp.br](mailto:ana.klinke@unesp.br)

1 - Cadastro de Cliente

Verifica Quantidades de Clientes cadastrados

printf("ERRO: sem espaco\n");

verifica tipo de cliente - CPF - CNPJ (caso inválido, retorne ao menu principal)

printf("ERRO: opcao invalida\n");

verifica se cliente ja foi cadastrado

printf("ERRO: ja cadastrado\n");

printf("Cadastrado com Sucesso\n");

2 - Atualiza cadastro de um dado cliente

SubMenu: 1 - adiciona endereço do cliente 2 - adiciona telefone cliente

LEITURA - pessoa fisica / pessoa juridica (caso inválido, retorne ao menu principal)

printf("ERRO: opcao invalida\n");

verifica se o cliente está cadastrado

printf("ERRO: nao cadastrado\n");

LEITURA - endereco / telefone

printf("ERRO: opcao invalida\n");

printf("Cadastrado com Sucesso\n");

3 - Lista os dados de um dado cliente

LEITURA - 1: pessoa fisica / 2: pessoa juridica (caso inválido, retorne ao menu principal)

printf("ERRO: opcao invalida\n");

verifica se o cliente está cadastrado

printf("ERRO: nao cadastrado\n");

Caso ache o cliente, mostrar os dados:

printf("nome: %s\n",x);

printf("rua: %s\n",x);

printf("num: %d\n",x);

printf("bairro: %s\n",x);

printf("cidade: %s\n",x);

printf("telefone: %s\n",x);

4 - Cadastro de Carro

Verifica Quantidades de Carros cadastrados

printf("ERRO: sem espaco\n");

verifica se o carro ja foi cadastrado

printf("ERRO: ja cadastrado\n");

Leitura de cores: preta (1), branca (2) e prata (3) (fazer validação). (caso inválido, retorne ao menu principal)

printf("ERRO: opcao invalida\n");

Leitura de categorias: hatch (1), sedan (2), suv (3) e jipe (4). (caso inválido, retorne ao menu principal)

printf("ERRO: opcao invalida\n");

printf("Cadastrado com Sucesso\n");

5 - Cadastro de Dados da Categoria

verifica se a categoria é válida (caso inválido, retorne ao menu principal)

printf("ERRO: opcao invalida\n");

printf("Cadastrado com Sucesso\n");

6 - Cadastro de Locação

LEITURA - 1: pessoa fisica / 2: pessoa juridica (caso inválido, retorne ao menu principal)

printf("ERRO: opcao invalida\n");

verifica se o cliente está cadastrado

printf("ERRO: nao cadastrado\n");

verifica o numero maximo de locação por cliente

printf("ERRO: sem espaco\n");

verifica se o carro foi cadastrado

printf("ERRO: nao cadastrado\n");

verifica se o carro já está locado

printf("ERRO: locacao em aberto\n");

verifica se o cliente possui locação em aberto

printf("ERRO: locacao em aberto\n");

printf("%d cadastrado com sucesso\n", codigo\_locacao);

7 - Devolve Carro

LEITURA - 1: pessoa fisica / 2: pessoa juridica (caso inválido, retorne ao menu principal)

printf("ERRO: opcao invalida\n");

verifica se o cliente está cadastrado

printf("ERRO: nao cadastrado\n");

verifica se a data é válida (caso inválido, retorne ao menu principal)

printf("ERRO: data invalida\n");

verifica se não existe locação em aberto

printf("ERRO: nenhuma locacao em aberto\n");

mostra o valor total devido

printf("Valor devido: %.2f\n", x);

mostra o desconto dado

printf("Desconto: %.2f\n", x);

caso a locação seja cancelada:

printf("Locacao cancelada\n");

8 - Lista dados de uma determinada locação

verifica se o codigo da locação está cadastrado

printf("ERRO: nao cadastrado\n");

verifica se a locação foi cancelada

printf("ERRO: locacao foi cancelada\n");

Caso ache, mostrar os dados:

printf("codigo da locacao: %d\n",x);

printf("cpf: %s\n",x); OU printf("cnpj: %s\n",x);

printf("nome: %s\n", x)

printf("renavam: %s\n",x);

printf("data retirada: %02d/%02d/%02d\n",x,y,z);

printf("data entrega: %02d/%02d/%02d\n",x,y,z);

9 - Lista locações de um dado cliente

LEITURA - 1: pessoa fisica / 2: pessoa juridica (caso inválido, retorne ao menu principal)

printf("ERRO: opcao invalida\n");

verifica se o cliente está cadastrado

printf("ERRO: nao cadastrado\n");

SubMenu: mostrar todas (1), as encerradas (2) ou a em aberto (3). (caso inválido, retorne ao menu principal)

printf("ERRO: opcao invalida\n");

Caso ache, mostrar os dados:

printf("codigo da locacao: %d\n",x);

printf("cpf: %s\n",x); OU printf("cnpj: %s\n",x);

printf("nome: %s\n", x)

printf("renavam: %s\n",x);

printf("data retirada: %02d/%02d/%02d\n",x,y,z);

printf("data entrega: %02d/%02d/%02d\n",x,y,z);

Nenhum dado de locação cadastrado:

printf("ERRO: nada cadastrado\n");

10 - Lista locações em aberto

LEITURA - 1: pessoa fisica / 2: pessoa juridica (caso inválido, retorne ao menu principal)

printf("ERRO: opcao invalida\n");

Caso ache, mostrar os dados:

printf("codigo da locacao: %d\n",x);

printf("cpf: %s\n",x); OU printf("cnpj: %s\n",x);

printf("nome: %s\n", x)

printf("renavam: %s\n",x);

printf("data retirada: %02d/%02d/%02d\n",x,y,z);

printf("data entrega: %02d/%02d/%02d\n",x,y,z);

Nenhum dado cadastrado:

printf("ERRO: nenhum dado cadastrado\n");

11 - Frequência de locação por categoria

LEITURA - 1: pessoa fisica / 2: pessoa juridica (caso inválido, retorne ao menu principal)

printf("ERRO: opcao invalida\n");

Caso ache, mostrar os dados das mais frequentes para as menos frequentes. Em caso de empate nas frequências utilize a ordem de prioridade a seguir:

printf("Categoria hatch: %d\n", x);

printf("Categoria sedan: %d\n", x);

printf("Categoria suv: %d\n", x);

printf("Categoria jipe: %d\n", x);

0 - Sair

printf("ERRO: opcao invalida\n");