

(BOCA: P3_2019_Q3_a) Problema: Imagine um cenário em que você tenha que controlar os carros de uma concessionária de aluguel de carros. Sua primeira tarefa, será ler uma lista de carros (com informação de identificação, número de passageiros, tipo e quilometragem para cada carro) representando os veículos de uma concessionária (no máximo 100 veículos) e exibir uma lista de carros que sigam os critérios de buscas feitas por clientes. Uma busca será baseada nos seguintes critérios: número de passageiros e quilometragem. Ela deverá exibir os veículos com capacidade de transportar passageiros igual a informada e com quilometragem menor ou igual a informada. Quando o cliente informar -1 para algum critério de busca (ex. para quantidade de passageiros ou quilometragem), aquele(s) critério(s) deverão ser desconsiderados.

- Entrada: na primeira linha, um inteiro n representando o número de carros da concessionária; na linhas seguintes, serão dados os n carros, sendo cada um deles em uma linha e contendo as seguintes informações separadas por espaço: identificador único (inteiro), quantidade de passageiros que o carro suporta (inteiro), tipo do carro (string com no máximo 100 caracteres) e uma quilometragem (inteiro). Após os carros, serão dadas as buscas, sendo uma em cada linha. Cada busca será composta por uma quantidade de passageiros que o carro suporta (inteiro) e uma quilometragem máxima (inteiro), separados por uma vírgula. Veja exemplos abaixo.
- Saída: A saída será uma lista de carros para cada uma das buscas realizadas. Cada busca será considerada um caso. A lista deverá exibir cada carro que atende os critérios da busca de um caso em uma linha. A descrição de um carro será dada pela mensagem “CARRO (#1): #2 de #3 passageiros e com #4 km”, em que #1 é o identificador do carro, #2 é o tipo, #3 é o número de passageiros e #4 é a quilometragem. Ver exemplos abaixo!

O aluno deverá implementar o tipo *tCarro* (0.2 pts). Funções deverão ser criadas para tratar um carro, *tCarro*, individualmente. Todo acesso ao tipo *tCarro* deve ser feito por funções específicas com os seguintes cabeçalhos:

- *tCarro LeCarro()*; (0.4 pts) Essa função deverá ler as informações de um carro da entrada padrão e retornar o carro lido.
- *void ImprimeCarro(tCarro carro)*; (0.1 pts) Essa função receberá um carro como argumento, e imprimirá o seu conteúdo na saída padrão seguindo a formatação pedida.
- *int ObtemNumPassageirosCarro(tCarro carro)*; (0.1 pts) Essa função receberá um carro como argumento, e retornará a capacidade de passageiros daquele carro.
- *int ObtemKmCarro(tCarro carro)*; (0.1 pts) Essa função receberá um carro como argumento, e retornará a quilometragem daquele carro.

O aluno deverá implementar o tipo *tConcessionaria* (0.2 pts). Funções deverão ser criadas para tratar uma concessionária, *tConcessionaria*, individualmente. Todo acesso ao tipo *tConcessionaria* deve ser feito por funções específicas com os seguintes cabeçalhos:

- *tConcessionaria LeCarrosConcessionaria();* (0.4 pto) Essa função deverá ler a quantidade de carros de uma concessionária e seus respectivos carros da entrada padrão. Por fim, ela deve retornar a concessionária com seus respectivos carros.
- *void ListaCarrosConcessionaria(tConcessionaria concessionaria, int pass, int km);* (1.0 pto) Essa função receberá uma concessionária, um número de passageiros e uma quilometragem como argumento, e imprimirá a lista de carros que atendem os critérios de busca informados como argumento (i.e., número de passageiros e quilometragem).

O aluno deverá utilizar a função *main* dada abaixo, ou seja, ela e seu conteúdo não poderão ser alterados:

```
int main(){
    tConcessionaria concessionaria;

    concessionaria = LeCarrosConcessionaria();

    int qtdPassageiros, km, i = 0;
    while(scanf("%d,%d", &qtdPassageiros, &km) == 2){
        printf("Caso %d:\n", ++i);
        ListaCarrosConcessionaria(concessionaria, qtdPassageiros, km);
    }

    return 0;
}
```

- Exemplos de Entradas:

9
10 5 Gol 15000
11 5 Gol 12000
14 5 Gol 65000
20 5 Corola 11000
21 5 Corola 35000
31 2 Saveiro 45000
33 2 Saveiro 55000
43 7 Doblo 35000
44 7 Spin 15000
5,15000
5,30000
-1,30000
2,-1
2,50000
7,30000
2,15000
-1,-1

1
10 5 Palio 15000
-1,-1
5,15000
5,30000
-1,30000
2,-1
5,-1

- Exemplos de Saídas:

Caso 1:
CARRO (10): Gol de 5 passageiros e com 15000 km
CARRO (11): Gol de 5 passageiros e com 12000 km
CARRO (20): Corola de 5 passageiros e com 11000 km
Caso 2:
CARRO (10): Gol de 5 passageiros e com 15000 km
CARRO (11): Gol de 5 passageiros e com 12000 km
CARRO (20): Corola de 5 passageiros e com 11000 km
Caso 3:
CARRO (10): Gol de 5 passageiros e com 15000 km
CARRO (11): Gol de 5 passageiros e com 12000 km
CARRO (20): Corola de 5 passageiros e com 11000 km
CARRO (44): Spin de 7 passageiros e com 15000 km
Caso 4:
CARRO (31): Saveiro de 2 passageiros e com 45000 km
CARRO (33): Saveiro de 2 passageiros e com 55000 km
Caso 5:
CARRO (31): Saveiro de 2 passageiros e com 45000 km
Caso 6:
CARRO (44): Spin de 7 passageiros e com 15000 km
Caso 7:

Caso 8:

CARRO (10): Gol de 5 passageiros e com 15000 km
CARRO (11): Gol de 5 passageiros e com 12000 km
CARRO (14): Gol de 5 passageiros e com 65000 km
CARRO (20): Corola de 5 passageiros e com 11000 km
CARRO (21): Corola de 5 passageiros e com 35000 km
CARRO (31): Saveiro de 2 passageiros e com 45000 km
CARRO (33): Saveiro de 2 passageiros e com 55000 km
CARRO (43): Doblo de 7 passageiros e com 35000 km
CARRO (44): Spin de 7 passageiros e com 15000 km

Caso 1:

CARRO (10): Palio de 5 passageiros e com 15000 km

Caso 2:

CARRO (10): Palio de 5 passageiros e com 15000 km

Caso 3:

CARRO (10): Palio de 5 passageiros e com 15000 km

Caso 4:

CARRO (10): Palio de 5 passageiros e com 15000 km

Caso 5:

Caso 6:

CARRO (10): Palio de 5 passageiros e com 15000 km