

# Estoque de carros

Os carros têm a fama de ser uma paixão nacional dos brasileiros. Uma venda de carros é um mercado promissor, com uma quantidade muito grande variedade de ofertas, modelos e, claro, lojas! Além dos carros novos, o mercado de seminovos cresceu muito no ano passado, impulsionado pelo aumento dos preços dos carros novos e o crescimento baixo da economia, os consumidores estão preferindo bons carros, mas com menores preços.

Uma loja de seminovos, diferente de uma concessionária de uma marca específica, trabalha com um conjunto e variedade maior de carros. Buscando estudar as variações de mercado, o gerente de uma loja deseja um programa para contar o número de carros no estoque de uma determinada marca. E, para isso, vai precisar da sua ajuda!

Sua tarefa é implementar um código que contém a struct produto com os seguintes campos:

Campo	Tipo
marca	string de 80 caracteres
modelo	string de 80 caracteres
quantidade	um inteiro

Além disso, defina no seu código a função `pesquisarEstoque`:

```
int pesquisarEstoque(char marca[80], struct produto estoque[50], int n);
```

Essa função recebe uma string `marca`, um vetor de struct `produto` contendo todos os produtos no estoque (no máximo 50) e um número `n` contendo a quantidade de produtos que realmente existem no estoque. Nela, você precisa implementar a seguinte lógica: procurar em todos os elementos do vetor (até `n`), os produtos da **marca**. Se o produto for encontrado, sua função deve imprimir na tela o **modelo** do produto, um espaço, a quantidade disponível e deve terminar a linha.

Ao final, sua função deve retornar o somatório da quantidade de todos os produtos disponíveis.

## Restrições

A criação de outras funções auxiliares e o uso de outras bibliotecas são permitidos. Você é obrigado a declarar a struct e fazer os includes necessários no seu código.

Seu programa não deve ler dados da entrada padrão.

IMPORTANTE! Defina a função com o mesmo nome, incluindo letras maiúsculas e minúsculas, mesma ordem de argumentos e mesmos nomes nos argumentos. Caso contrário irá ocorrer erro de compilação. O nome da struct e de seus campos devem ser idênticos a tabela acima (incluindo letras maiúsculas e minúsculas). Além disso, os campos strings devem conter exatamente 80 caracteres, nenhum a mais ou a menos.

Ao enviar a sua solução pro MOJ, envie somente o arquivo com a extensão C com a função exigida do enunciado, as estruturas, a inclusão das bibliotecas utilizadas e as funções auxiliares (caso existam). Não inclua neste arquivo a função `main`. Crie uma função `main` para testar seu código, mas remova-a (ou comente) antes de enviar a solução para o MOJ.

## Exemplos

Suponha que o vetor chamado `estoque` esteja preenchido com 5 entradas:

Campo / Entrada	0	1	2	3	4
marca	Volkswagen	Nissan	Ford	Ford	Volkswagen
modelo	Gol	March	Ranger	Fiesta	Polo
quantidade	10	20	2	30	3

### Exemplo 1

Ao chamar a função:

```
pesquisarEstoque("Volkswagen", estoque, 5);
```

Sua função deve imprimir:

Gol 10

Polo 3

e retornar 13.

### **Exemplo 2**

Ao chamar a função:

```
pesquisarEstoque("Nissan", estoque, 5);
```

Sua função deve imprimir:

March 20

e retornar 20.

### **Exemplo 3**

Ao chamar a função:

```
pesquisarEstoque("Peugeot", estoque, 5);
```

Sua função não deve imprimir nada e retornar 0.

*Author: Daniel Sundfeld <daniel.sundfeld@unb.br>*