

# Algoritmos e Programação II

## Exercício sobre Registros

Abra os arquivos da pasta "exemplo5" e complete a solução conforme descrito a seguir.

1. O proprietário de uma mecânica deseja criar um cadastro dos carros dos clientes que atende. Você foi contratado para criar um cadastro simples usando **registros**. Portanto, você deve implementar um programa em C/C++ com as seguintes opções:

- 1 - Cadastrar uma marca
- 2 - Cadastrar um carro
- 3 - Listagem de carros
- 4 - Exclusão de uma marca
- 5 - Sair

Utilize os tipos definidos a seguir:

```
struct tMarca{
    int codigomarca;
    float nota;
};
```

```
struct tCarro
{
    char modelo[40];
    int ano;
    int codigomarca;
};
```

- Na opção 1 do menu devem ser cadastrados em um vetor os seguintes dados de **UMA** marca: código da marca (inteiro) e nota de avaliação da marca no mercado (real). Crie uma função com o protótipo a seguir para realizar a inserção de uma nova marca no vetor.

**void inserirM(tMarca \*M, int &n, int cod, float nota);**

- Crie uma função com o protótipo a seguir, para realizar a busca de uma marca, a partir de um código de marca informado. A função retorna a nota da marca se o código da marca for encontrado e 0.0, caso contrário.

**float buscaMarca(tMarca \*M, int n, int cod);**

- Na opção 2 do menu devem ser cadastrados em um vetor os seguintes dados de **UM** carro: modelo do carro, ano de fabricação e marca (código da marca). Um carro só pode ser cadastrado utilizando uma marca já cadastrada. Crie uma função com o protótipo a seguir para realizar a inserção de um novo carro neste vetor. Utilize a função **buscaMarca** para te auxiliar nesta opção.

**void inserirC(tCarro\* C, int &n, char \*mod, int ano, int codmarca);**

- Na opção 3 do menu devem ser listados todos os carros cadastrados (modelo, ano e nota da marca). Crie uma função com o protótipo a seguir para realizar esta tarefa.

**void listagem(tMarca\* M, int tamM, tCarro\*, int tamC);**

- Na opção 4 do menu deve ser excluída uma marca, após o fornecimento do código da marca pelo usuário. Como é necessário manter a integridade referencial, uma marca só pode ser excluída se não existir nenhum carro cadastrado para ela. Mostre uma mensagem de erro caso ela não possa ser excluída.

**void removerM(tMarca \*M, int &tamM, tCarro \*C, int tamC, int codmarca);**