

Algoritmos e Programação II



Abra o arquivo “main-carro.cpp” 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 listas lineares dinâmicas. 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 marca{
    int codigomarca;
    float nota;
    struct marca *prox;
};
```

```
struct carro{
    char modelo[40];
    int ano;
    int codigomarca;
    struct carro *prox;
};
```

- Na opção 1 do menu devem ser cadastrados em uma **lista dinâmica encadeada (SEM CABEÇA)** os seguintes dados: 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** nesta lista.

```
void inserirM(marca*&, int, float);
```

- 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(marca *, int);
```

- Na opção 2 do menu devem ser cadastrados em uma **lista dinâmica simplesmente encadeada** do tipo PILHA os seguintes dados: modelo do carro, ano de fabricação e marca (código). 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** nesta lista. Utilize a função **buscaMarca** para te auxiliar nesta opção.

```
void inserirC(carro*& char *, int, int);
```

- 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(marca* , carro*);
```

- Na opção 4 do menu deve ser excluída uma marca, após 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(marca*&, carro *, int);
```

- Apague todas as listas encadeadas ao sair do programa.