

Programação Imperativa – Trabalho 2 – Recuperação do RA2

Prof. Alcides Calsavara

Escreva um programa na linguagem **C** que leia uma sequência de registros (implementados por meio de um `struct`) sobre carros armazenados em um arquivo texto denominado **dados.txt**, armazene esses registros em memória por meio de uma lista encadeada e, finalmente, permita o usuário realizar diversas operações sobre essa lista, conforme detalhado a seguir.

Cada registro de um carro deve ter os seguintes campos:

1. Marca do carro (string de, no máximo, 20 caracteres)
2. Modelo do carro (string de, no máximo, 20 caracteres)
3. Ano de fabricação do carro (valor inteiro)
4. Kilometragem do carro (valor inteiro)
5. Preço do carro (valor real)

Os registros de carros no arquivo de entrada devem estar em ordem arbitrária. No entanto, a lista em memória deve ser construída de modo que os registros fiquem em ordem crescente com respeito aos preços dos carros. Ainda, o arquivo de entrada só pode ser lido um única vez. Uma forma simples de atender a esse requisito é fazer a inserção de cada registro na posição correta da lista. Isto é, o registro de um novo carro deve ser inserido na lista em uma posição tal que todos os registros anteriores à essa posição da lista refiram-se a carros com preço inferior ao preço do novo carro, ao mesmo que todos os registros posteriores refiram-se a carros com preço superior ao preço do novo carro. Dessa forma, quando o último registro for inserido, a lista já estará correta!

As operações disponíveis na interface do usuário devem ser as seguintes:

1. Exibir a lista completa de registros.
2. Exibir todos os registros (completos) de carros de uma marca específica, fornecida pelo usuário, via teclado.
3. Exibir todos os registros (completos) de carros cujo preço esteja entre um valor mínimo e um valor máximo, fornecidos pelo usuário, via teclado.
4. Inserir um novo registro de carro na lista; todos os dados do carro devem ser fornecidos pelo usuário, via teclado. A inserção do novo registro deve preservar a ordem crescente da lista com respeito aos preços dos carros.
5. Remover todos os registros de carros na lista cuja kilometragem seja superior a um valor fornecido pelo usuário, via teclado.
6. Sair do programa.

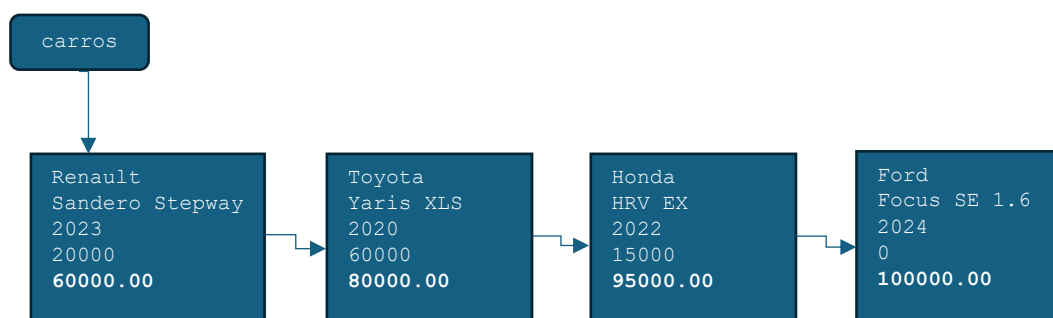
A interação do programa com o usuário deve ficar em *loop*, possibilitando todas as operações acima, até que o usuário opte por sair do programa.

A lista encadeada deve ser implementada em memória dinamicamente alocada (memória *heap*), com uso das funções `malloc` ou `calloc` para criar cada registro, bem como com uso da função `free` para liberar a memória usada por cada registro que seja removido da lista.

O arquivo **dados.txt** deve conter, ao menos, 10 registros de carros. Exemplo de formato do arquivo contendo apenas 4 registros e com os dados na ordem definida acima (marca, modelo, ano, kilometragem e preço):

```
Honda
HRV EX
2022
15000
95000.00
Renault
Sandero Stepway
2023
20000
60000.00
Toyota
Yaris XLS
2020
60000
80000.00
Ford
Focus SE 1.6
2024
0
100000.00
```

Para esse arquivo de dados, a correspondente lista encadeada gerada, supondo que o primeiro elemento seja apontado pela variável “carros”, deverá ter a seguinte forma:



Deve-se observar que os elementos da lista não estão na mesma ordem que os registros no arquivo de entrada: eles estão em ordem crescente com respeito aos preços dos carros.