

Na última prática (12) foi feito um programa com um menu para:

1. Criar (ao criar, ler o nome do arquivo, e caso o arquivo já exista ele deve ser resetado). Nenhum registro é inserido no arquivo através desta opção;
2. Listar sequencialmente o arquivo criado (avisar caso esteja vazio);
3. Inserir novos carros no final do arquivo;
4. Exibir a potência e a marca de um determinado carro, cujo modelo foi lido do teclado.

>> FIM DA ÚLTIMA PRÁTICA<<

→ Para a atividade de hoje, alterem a opção 3 para inserção ordenada que funciona assim: supondo que o arquivo está ordenado pelos carros de maior potencia, inserir novos carros mantendo o arquivo ordenado.

Observações:

- Há um exemplo no moodle para inserção ordenada de inteiros. Consultem o exemplo para ideias de como fazer este exercício.
- Usar a função `le_texto` abaixo no programa acima. Ela deverá ser implementada em um arquivo separado, por exemplo, `minha_biblioteca.h`. Incluir a linha `#include "minha_biblioteca.h"` na lista de bibliotecas.

/* A função `le_texto.h` recebe como parâmetros uma variável `string` a ser preenchida por leitura e o `sizeof` desta variável. O conteúdo digitado é lido em uma variável `dummy`, contendo 1 caractere a mais que a variável parâmetro. Assim, se o penúltimo caractere lido **não for** o `'\n'` isto significa que o número de caracteres digitados **ultrapassou** o tamanho da variável a ser lida */

```
#include <stdio.h>
#include <string.h>
```

```
void le_texto (char texto[ ], int size_texto) // string: ponteiro implícito
{
    char dummy[size_texto + 1]; // cabe um caractere a mais do que no texto:
    fflush(stdin);
    fgets(dummy, sizeof(dummy), stdin); // lê caracteres
    // O último caractere tem que ser '\n' para estar correto:
    while(strlen(dummy) - 1 != '\n')
    {
        printf("\nNumero de caracteres digitados excedeu tamanho do campo.\n");
        printf("Numero maximo de caracteres eh %d.\n", size_texto - 1);
        printf("Digite o conteudo novamente.\n");
        fflush(stdin);
        fgets(dummy, sizeof(dummy), stdin); // lê caracteres novamente
    }
    dummy[strlen(dummy)-1] = '\0'; // sempre precisa substituir o '\n'
    strcpy(texto, dummy); // transfere conteúdo digitado e sem o '\n' para texto
}
```