

Lista 09 - Ponteiros e Structs

Objetivo: Neste laboratório, iremos utilizar o conceito de **structs** para representar entidades e objetos do mundo real em nossos programas. Iremos também utilizar o conceito de ponteiros nas funções. Esta lista é a união das Listas 8 e 9. Portanto, o valor é **10 pontos**.

Entrega: Ao final desta prática, você deverá enviar no **SIGAA** da disciplina um arquivo **.zip** contendo os arquivos de extensão **.c**, com extensão **.exe** (resultados das compilações) e arquivos de extensão **.h**. A prática é individual e cada aluno(a) é responsável por editar, executar e submeter seu próprio trabalho.

Formato da Entrega:

Defina o nome do arquivo **.zip** como **Lista_09_PrimeiroNome_UltimoNome.zip**.

Seu código deve ser desenvolvido em C, estar bem organizado e indentado, possuir funções ou procedimentos (se necessário) e conter comentários relevantes. Cópias de trabalho são inaceitáveis, podem ser detectadas automaticamente e receberão nota zero.

ATIVIDADE 0 – Configuração Inicial: Construa um programa que tenha um menu principal para acessar funções ou procedimentos que resolvam os exercícios a seguir. Após realizar uma das operações indicadas no menu, o menu deverá voltar a ser exibido para que o usuário possa escolher uma nova opção. Lembre-se de tratar opções inválidas e criar uma opção para sair do menu. Salve e envie o seu arquivo com o nome *programas_lista09.c*.

Atividade 1

Divisores – Escreva uma função chamada **divisores** que:

- recebe como parâmetro um número inteiro *n* por valor e dois números inteiros *max* e *min* por referência;
- retorna 0 se *n* é primo e 1 caso contrário;
- Se *n* não for primo, as variáveis *max* e *min* devem assumir os valores do maior e do menor divisores inteiros de *n*, respectivamente. Desconsidere o número 1 e o próprio número *n*.

Atividade 2

Revendedora de Carros (v1.0) – Seja uma estrutura para descrever os carros de uma determinada revendedora, contendo os seguintes campos:

- **marca:** string de tamanho 15
- **ano:** inteiro
- **cor:** string de tamanho 10

- **preco:** real

Implemente os seguintes trechos de programa:

1. Escrever a definição da estrutura carro.
2. Declarar um vetor do tipo da estrutura definida acima, de tamanho lido pelo teclado.
3. Definir uma função para preencher o vetor.
4. Definir uma função que receba um preço e imprima os carros (marca, cor e ano) que tenham preço igual ou menor ao preço recebido.

Atividade 3

Revendedora de Carros (v1.1) – Usando a estrutura e código do exercício anterior, defina uma nova função que leia a marca de um carro e imprima as informações de todos os carros dessa marca (preço, ano e cor).

Atividade 4

Revendedora de Carros (v1.2) – Usando a estrutura e código do exercício anterior, defina uma função que leia uma marca, ano e cor e informe se existe ou não um carro com essas características. Se existir, informar o preço.

Atividade 5

Distância entre pontos (v2.0) – Escreva um programa que define um tipo de dados Ponto através de uma struct. Em seguida, o programa deverá ler uma sequência de pontos e os guarda em um vetor de pontos. O tamanho do vetor deverá ser lido do teclado. Após a leitura dos pontos a partir do teclado, as seguintes funções deverão ser chamadas para calcular os dois pontos mais próximos e os dois pontos mais distantes:

1. A função **mostraPontosMaisProximos** que determina os dois pontos mais próximos.
2. A função **mostraPontosMaisDistantes** que determina os dois pontos mais distantes.