

Lista de Exercícios 9 – Registros e Arquivos

Registros

Programa 1)

Crie um programa que armazene dados de N livros em um vetor, informados por um usuário. Os dados dos livros deverão ser estruturados em registros/struct, que possuirão os seguintes campos: id, título, ano, autor. As seguintes opções deverão ser dadas ao usuário:

- a) Um procedimento para cadastrar livros
- b) Um procedimento que lista livros por ano (usuário informa um ano)
- c) Um procedimento que permite atualizar um nome do livro. Usuário digita o nome antigo para pesquisa.

Arquivo Texto

Programa 2) Crie um programa que ofereça ao usuário as seguintes opções:

- a) Crie um procedimento que escreva em um arquivo texto números digitados pelo usuário.
- b) Crie um procedimento que leia números escritos em um arquivo (item anterior) e depois mostre a soma desses números. Sugere-se utilizar a função *atoi*
- c) Crie um procedimento que abra um arquivo HTML e mostre na tela o seu conteúdo. Você poderá baixar uma página HTML usando o seu navegador e salvá-lo em na mesma pasta do seu programa.
- d) Crie um procedimento que abre um arquivo HTML faça uma cópia desse arquivo (backup). O nome do arquivo de backup deverá ser informado pelo usuário.

Registros e Arquivo Binários

Programa 3) Crie um programa em linguagem C para registro de veículos em um estacionamento. Os registros deverão implementados como *struct* e deverá ter os seguintes campos/atributos:

- Código (int) – Auto numerável
- Placa (char(7))
- Modelo (char(30))
- Entrada (>> Struct Data (dia, mes, ano, hora, minuto)
- Saída (>> Struct Data (dia, mes, ano, hora, minuto)
- Valor pago (float) - Preenchido automaticamente (R\$1 por hora estacionada)

O programa deverá ter um menu com as opções:

- a) Cadastrar Veículos
- b) Listar Veículos
- c) Alterar Veículos (a partir da placa informada)
- d) Pesquisar por Placa. Deve-se exibir todos os dados do veículo após encontrar o registro ou caso não seja encontrado, deve-se informar que o registro não existe.
- e) Pesquisar por Modelo. Deve-se exibir todos os dados do veículo após encontrar o registro ou caso não seja encontrado, deve-se informar que o registro não existe.
- f) Lista por Data. Usuário informa uma data inicial e final e o sistema exibe os dados dos veículos com data de saída entre a data inicial e final informada.

As opções apresentadas acima devem utilizar *struct* e salvar os dados em memória primária (um vetor de veículos com tamanho pré-definido).

As opções seguintes deverão manipular arquivos binários.

- g) Backup dos dados em Arquivo Binário - Usuário informa o nome de um arquivo e programa salva os dados dos estudantes em arquivo binário.
- h) Restauração de dados de Arquivo Binário – Usuário informa o nome de um arquivo binário e programa carregar dados salvo no arquivo na memória primária (um vetor de Alunos).

Sugestão de Implementação - Passo-a-Passo

Sugere-se os seguintes passos para o desenvolvimento do Trabalho:

1º Crie um *struct* Veículo. Lembre-se de colocar ponto e virgula (;) ao final da definição da *struct*.

2º Crie o procedimento *main* e os demais procedimentos para as opções do *menu*.

3º Crie um vetor do tipo Veículos para armazenar os dados de veículos. O vetor poderá ser criado de forma global (fora do *main*) ou local (dentro do *main*), sendo que neste último caso será necessário passar o vetor como parâmetro para os procedimentos que irão utilizá-lo.

```
struct Veículo list[MAX];
```

4º Defina o limite máximo de Veículos a serem cadastrados. Exemplo: #define MAX 10

Nesse caso, você terá duas opções de implementação:

- a) Mais simples: na opção de cadastro sempre exigir que o usuário cadastre todos os veículos possíveis (limite máximo).
- b) Mais sofisticado: coloque no vetor de Veículos o número de veículos que o usuário desejar. Nesse caso você deverá controlar os procedimentos de imprimir e busca para não realizar consultas em parte do vetor de Veículo que não estejam preenchidas.

Informações sobre cópias

As questões são individuais. Em caso de cópias de trabalho a pontuação será zero para os autores originais e copiadores. Não serão aceitas justificativas como: “Fizemos o trabalho juntos, por isso estão idênticos”.

