**Exercícios de fixação de conteúdo:**

**STRUCT's e ARQUIVO BINÁRIO**

1. Utilizando estrutura de dados, fazer um programa parametrizado em C /C++, que permita o armazenamento em Arquivo Binário, de nome, endereço e telefone de várias pessoas e, implemente funções de consultas por campos diferentes e relatório em ordem alfabética pelo nome.
2. Fazer um programa para simular uma agenda de telefones. Portanto, é preciso armazenar os dados em Arquivo Binário, onde para cada pessoa, devem-se ter:

* Nome
* E-mail
* Endereço (contendo campos para Rua, número, complemento, bairro, CEP, cidade, estado, país).
* Telefone (contendo campo para DDD e número).
* Data de aniversário (contendo campo para dia, mês, ano).
* Observações: Uma linha (string) para alguma observação especial.

(a) Definir a estrutura acima.

(b) Utilizar um arquivo binário para agendar vários nomes.

(c) Definir uma função de consulta por primeiro nome: mostrar os dados da pessoa com esse nome (pode haver nomes iguais).

(d) Implementar uma função para varrer por mês de aniversário: Mostrar os dados de todas as pessoas que fazem aniversário nesse mês.

(e) Implementar uma função de busca por dia e mês de aniversário: Mostrar os dados de todas as pessoas que fazem aniversário nesse dia e mês.

(f) Implementar uma função que insere pessoa: insere por ordem alfabética de nome. Utilizar a Inserção Direta.

(g) Implementar uma função para retirar pessoas: retira todos os dados dessa pessoa.

(h) Possibilitar o uso da agenda para com as opções:

* + Exibir: nome, telefone e e-mail.
  + Exibir: todos os dados.

O programa deve ter um menu principal oferecendo as opções acima.

1. Construa uma estrutura aluno com nome, número de matrícula (não pode se repetir – Chave Primária) e curso. Implemente o cadastro em arquivo binário do aluno, e possibilite realizar a busca dos dados para uma consulta ou alteração dos mesmos. O programa deve ser modularizado e parametrizado adequadamente.
2. Crie uma estrutura representando os alunos do curso de Computação e Informática. A estrutura deve conter a matrícula do aluno (chave primária), nome, nota da primeira prova, nota da segunda prova e nota da terceira prova e deve estar armazenada em arquivo binário. Implemente funções que:

(a) Permita ao usuário entrar com os dados de vários alunos.

(b) Encontre o aluno com maior nota da primeira prova.

(c) Encontre o aluno com maior média geral.

(d) Encontre o aluno com menor média geral.

(e) Um relatório que mostre, para cada aluno, se ele foi aprovado ou reprovado, considerando o valor 6 para aprovação.

1. Faça um programa que armazene em registro de dados (estrutura composta - structs) e arquivo binário, os dados de um funcionário de uma empresa, compostos de: Nome, Idade, Sexo Biológico (M/F), CPF, Data de Nascimento, Código do Setor onde trabalha (0-99), Cargo que ocupa (string de até 30 caracteres) e Salário. Os dados devem ser digitados pelo usuário, armazenados na estrutura e exibidos na tela. Além de possibilitar a implementar de funções Alteração, Exclusão e Relatórios para consultas específicas, crie as suas.
2. Utilizando structs e arquivos binários, escolha um jogo de cartas, baseado em um "baralho " do tipo "super trunfo" (cada carta possui um conjunto de atributos). Implemente a parte de distribuição (sorteio) de cartas para “n” jogadores, considerando que cada jogador irá receber 5 cartas. Exiba na tela as cartas que cada um dos jogadores recebeu. E implemente funções para Jogar.
3. Utilizando funções, faça um programa que leia um inteiro “n” e:
   * Crie e leia um arquivo binário com os dados de “n” carros: marca (máximo 15 letras), ano e preço.
   * Leia um valor “p” e mostre as informações de todos os carros com preço menor que “p”. Repita este processo até que seja lido um valor p=0.
4. Utilizando funções, faça um programa que leia um inteiro “n” e:
   * Crie e leia um arquivo binário com dados de “n” livros: título (máximo 30 letras), autor (máximo 15 letras) e ano.
   * Procure um livro por título, perguntando ao usuário qual título deseja buscar. Mostre os dados de todos os livros encontrados.
5. Faça um programa que, armazene em arquivo binário e controla o consumo de energia dos eletrodomésticos de uma casa e:
   * Crie e leia vários eletrodomésticos que contém nome (máximo 15 letras), potencia (real, em kWh) e tempo ativo por dia (real, em horas).
   * Leia um tempo t (em dias), calcule e mostre o consumo total na casa e o consumo relativo de cada eletrodoméstico (consumo/consumo total) nesse período de tempo. Apresente este último dado em porcentagem. Implementar com funções parametrizadas.
6. Implemente um programa em C/C++, utilizando funções, structs e arquivo binário, que gerencie o estoque de um mercado e para isso é preciso:
   * Criar e ler um produtos, com os dados: código (inteiro), nome (máximo 15 letras), preço e quantidade.
   * Leia um pedido, composto por um código de produto e a quantidade. Localize este código no arquivo e, se houver quantidade suficiente para atender ao pedido integralmente, atualize o estoque e informe o usuário. Repita este processo até ler um código igual a zero.
7. Faça um programa, modularizado e parametrizado, que controle o fluxo de voos nos aeroportos de um país. Com v=5 (voos) e a=5 (aeroportos) e:
   * Crie e leia um arquivo de voos, sendo que cada voo contém um código de aeroporto de origem e um de destino.
   * Crie um arquivo de aeroportos, sendo que cada aeroporto contém seu código, quantidade de voos que saem e quantidade de voos que chegam.
8. Faça um programa, modularizado e parametrizado, para armazenar um livro de receitas e:
   * Crie um arquivo de receitas, que deve ter nome (máximo 25 letras), quantidade de ingredientes e ingredientes.
   * Para cada receita, leia seu nome e a quantidade de ingredientes. Então crie e leia um vetor de ingredientes, sendo que cada ingrediente contém nome e quantidade.
   * Procure receita por nome, mostrando seus ingredientes se encontrar. Se não encontrar, informe ao usuário. Repita o processo até digitar uma string vazia.
9. Faça um programa que armazena filmes produzidos por vários diretores e:
   * Crie e leia um arquivo binário de diretores, cada um contendo nome (máximo 20 letras), quantidade de filmes e filmes. O membro filmes é um vetor, que deve ser criado após ter lido quantidade de filmes. Cada filme é composto por nome, ano e duração.
   * Procure um diretor por nome, mostrando todos os filmes que ele já produziu. Repita o processo até digitar uma string vazia.