



LISTA 5 – 09/10/2024

Conteúdos:

Estruturas de Dados Heterogêneas (registros): definição, declaração, armazenamento de dados e acesso aos elementos armazenados. Definição e utilização de vetores de registros.

Elabore os seguintes algoritmos:

1. Elabore um algoritmo preencha o cadastro de 10 alunos contendo matrícula, nome, endereço e média geral. Após o preenchimento, deve ser somado o valor 0,5 às médias de todos os alunos que tiveram médias maiores que 5. Em seguida escreva o vetor final.
2. Elabore um algoritmo que
 - Crie a estrutura tAnimal: **Registro tAnimal:**
nome: string
raca: string
idade: int
 - Declare um vetor Animais (50 posições de tAnimal) e cadastre 8 animais;
 - Verifique e escreva o nome do animal mais velho;
 - Pergunte ao usuário se deseja consultar os animais cadastrados solicitando uma raça para a busca. Se ele quiser, leia uma raça e, se existir animal da raça procurada, informe seus dados (nome e idade);
3. Escreva um programa para cadastrar dois clientes de uma loja. As informações necessárias para o cliente são: identidade, nome, endereço e telefone. OBS: Deve ser usada uma estrutura de registro para a construção deste cadastro.
4. Altere o exercício anterior para cadastrar no máximo 20 clientes. Ao final do cadastro de cada cliente deverá ser perguntado: "Novo Cliente (S/N)?". Caso o usuário escolha "N" ou chegar nos 20 clientes, o programa deve mostrar um relatório de todos os clientes cadastrados e finalizar. *OBS: Deve-se utilizar um vetor de registros na solução.*
5. Altere o exercício anterior para que, após o término de todos os cadastros, ou seja, quando o usuário digitar "N" na pergunta para novo cliente ou quando preencher o vetor com 20 clientes, a tela seja limpa e seja exibida uma nova tela perguntando se o usuário quer ver um relatório ou consultar um cliente individualmente. Se ele desejar ver o relatório, o sistema deverá exibi-lo conforme questão anterior. Se ele escolher consultar um cliente individualmente, o sistema deverá solicitar a identidade do cliente procurado, fazer uma busca no vetor e informar seus dados caso o cliente esteja cadastrado. Se não existir cliente cadastrado com aquela identidade, o sistema deverá informar "Cliente não cadastrado".

6. Uma empresa tem para cada um dos seus 200 funcionários uma ficha contendo o nome, número de horas trabalhadas e o número de dependentes. Considerando que:
 - A empresa paga 12 reais por hora e 40 reais por dependentes;
 - Sobre o salário são feitos descontos de 8,5% para o INSS e 5% para IR.Elabore um algoritmo que declare o registro do funcionário, e leia seus dados enquanto o usuário desejar cadastrar (não permitindo o cadastro de mais de 200 funcionários). Após a leitura, escreva o nome, salário bruto, os valores descontados para cada tipo de imposto e qual o salário líquido de cada um dos funcionários que foram cadastrados.
7. Elabore um algoritmo que defina um registro capaz de armazenar as seguintes informações sobre um determinado cliente de um banco: nome, CPF, RG, número da conta, data de abertura da conta e saldo. Em seguida manipule um vetor com 15 registros de clientes, onde cada registro é um elemento do tipo de dado definido. A manipulação do vetor é feita através das seguintes opções: cadastrar cliente, imprimir os dados de um determinado cliente com base no valor do campo CPF e imprimir todos os clientes com saldo negativo.
8. Um provedor de acesso à Internet mantém o seguinte cadastro de clientes: código do cliente, e-mail, número de horas de acesso, página ('S'-sim ou 'N'-não). Elaborar um algoritmo que insira um conjunto de registros (máximo 500), calcule e mostre um relatório contendo o valor a pagar por cada cliente, sabendo-se que as primeiras 20 horas de acesso custam R\$ 35,00 e as horas que excederam tem o custo de R\$ 2,50 por hora. Para os clientes que têm página, adicionar R\$ 40,00.
9. Crie um tipo registro chamado Endereco que contenha os campos Rua (string), Numero (int), Bairro (string), Cidade (string), Estado (string). Em seguida declare uma variável do tipo Endereco. Crie outro tipo registro chamado Pessoa que tenha os campos Nome (string), Ender (Endereco), Telefone (string) e Idade (int).
 - Declare um vetor pessoas com 10 posições do tipo Pessoa e leia seus dados;
 - Leia um endereço na variável do tipo endereço;
 - Faça uma busca e liste os nomes de todas as pessoas cadastradas no vetor cujo bairro é o mesmo bairro do endereço cadastrado na variável do tipo endereço.
10. Suponha que você esteja realizando uma pesquisa e precise obter os seguintes dados de um conjunto de n pessoas (n é informado pelo usuário): nome, sexo (1-M, 0-F), cor dos olhos, altura, peso e data de nascimento. Elabore um algoritmo que realize a leitura desses dados e imprima duas listagens. A primeira listagem deve conter todos os dados (exceto o sexo) das mulheres e a outra deve conter todos os dados (exceto o sexo) dos homens. Ou seja, vamos apenas criar uma listagem dos homens e outra das mulheres
11. Elabore um algoritmo que:
 - Crie o registro tAluno:

Registro tAluno:

nome: string

matriculaDoAluno: int

- Declare um vetor Alunos (3 posições de tAluno) e cadastre 3 alunos no vetor Alunos;
- Após o cadastro, escreva os dados de todos os alunos cadastrados;

- Crie o registro tDisciplina:

Registro tDisciplina:	<i>Obs: o usuário não deverá</i>
matriculaDoAluno: int	<i>digitar o valor do campo media.</i>
disciplina: string	<i>O sistema deverá calcular</i>
nota1: float	<i>automaticamente a média</i>
nota2: float	<i>aritmética das duas notas do</i>
media: float	<i>aluno e armazenar neste campo.</i>

- Crie o vetor Disciplinas (12 posições de tDisciplina) e cadastre, para cada um dos 3 alunos existentes no vetor Alunos, 4 disciplinas com as respectivas informações. Lembre-se que a matrícula do aluno tem que ser a mesma cadastrada no vetor Alunos.
- Informe o seguinte menu e proceda com a operação de acordo com a escolha do usuário:

MENU PRINCIPAL

-
- 1 – EXIBIR DADOS DE UM ALUNO**
 - 2 – EXIBIR MAIOR MEDIA**
 - 3 – EXIBIR MÉDIA DAS MÉDIAS POR ALUNO**
 - 4 – EXIBIR OS ALUNOS DE UMA DISCIPLINA**
 - 5 – SAIR DO SISTEMA**

- Se o usuário escolher a opção 1, pergunte qual a matrícula do aluno que ele deseja consultar. Depois que ele informar, procure a matrícula procurada no vetor Alunos. Se existir, mostre o nome do aluno. Em seguida, procure no vetor Disciplinas pelos dados daquele aluno e mostre os três resultados.
- Se o usuário escolher a opção 2, procure no vetor notas qual a matrícula do aluno que possui maior média e escreva A DISCIPLINA E O NOME DO ALUNO.
- Se o usuário escolher a opção 3, mostre a média das médias por ALUNO, informando o nome do aluno e a média aritmética de suas médias.
- Se o usuário escolher a opção 4, pergunte qual a disciplina que ele deseja consultar. Depois que ele informar, procure e mostre todos os nomes dos alunos daquela disciplina. - Se o usuário escolher 5, finalize o sistema.

12. Elabore um registro que represente corretamente uma conta bancária de uma pessoa contendo os campos “número do banco”, “número da agência”, “número da conta”, “nome do cliente”, “senha” e “saldo atual” e um vetor capaz de armazenar 100 contas. Leia os dados de N contas e, após a leitura, mostre uma tela com as opções “entrar” ou “sair”. Se o usuário escolher “entrar”, ele deverá digitar o número da conta e a senha. O algoritmo deverá localizar a conta no vetor e, se a senha estiver correta, mostrar um menu com as opções de “saque”, “depósito”, “saldo” e “voltar a tela anterior”. A opção que o usuário escolher deverá ser executada pelo sistema e, em seguida, esta mesmo menu de opções deverá ser exibido novamente. Se a senha estiver incorreta ou a conta não existir, o sistema deverá informar a mensagem “erro de busca”. Somente quando o usuário escolher “sair”, o sistema deverá finalizar.

13. Elabore um algoritmo que:

- Crie o registro tProduto: **Registro tProduto:**
descricao: string
quantidade: int
precoUnitario: float
subTotal: float

Obs: o campo subtotal deve ser calculado automaticamente, sendo ele resultante da quantidade x preço unitário do produto.

- Crie o registro tNotaFiscal **Registro tNotaFiscal:**
numero: int
data: string
itens: tProduto[5]
total: float

Obs: o campo total deve ser calculado automaticamente, sendo ele a soma de todos os SubTotais dos produtos cadastrados.

- Declare um vetor NotasFiscais (5 posições de tNotaFiscal) e leia seus dados.
- Informe o seguinte menu e proceda com a operação de acordo com a escolha do usuário:

MENU PRINCIPAL

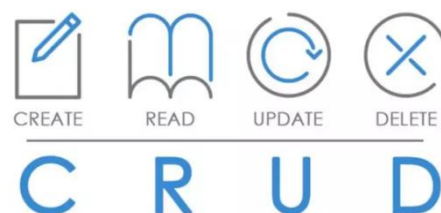
- 2 – RELATÓRIO DE NOTAS FISCAIS**
- 3 – BUSCAR NOTA POR NUMERO**
- 4 – EXIBIR NOTA COM MAIOR VALOR TOTAL**
- 5 – EXIBIR A QUANTIDADE TOTAL VENDIDA DE UM PRODUTO A PARTIR DA SUA DESCRIÇÃO**
- 6 – EXIBIR MÉDIA DO TOTAL DAS NOTAS**
- 7 – SAIR**

- Se o usuário escolher a opção 1, o algoritmo deverá mostrar todos os dados de todas as notas cadastradas.
- Se o usuário escolher a opção 2, solicite o número da nota e faça a busca. Se existir, mostre seus produtos. Se não existir, informe.
- Se o usuário escolher a opção 3, mostre os dados da nota que tem o maior valor total. - Se o usuário escolher a opção 4, solicite a descrição do produto, calcule e mostre o somatório de todas as quantidades vendidas daquele produto em todas as notas.
- Se o usuário escolher a opção 5, calcule e mostre a média de todos os totais das notas.
- Se o usuário escolher a opção 6, finalize o sistema.

14. Elabore um algoritmo que realize o controle dos veículos em um estacionamento a partir de um vetor de tVeículos (placa, modelo, data, hora de entrada, hora de saída e valor) e de um menu contendo as opções “Registrar entrada de veículo”, “Registrar saída de veículo”, “Listar veículos ativos no estacionamento”, “Informar quantidade de veículos por data”, “Informar total geral arrecadado” e “sair do sistema”. Considere que os campos “hora de entrada” e “hora de saída” armazenam somente a hora (sem minutos ou segundos). Considere que quando um veículo entra no estacionamento, a hora de saída é, obrigatoriamente, zerada. Considere ainda que o valor da hora no estacionamento custa 5 reais e que se o veículo entrar e sair na mesma hora não deverá efetuar pagamento. Finalmente, considere que o valor a ser pago deverá ser calculado pelo sistema a partir da hora de entrada e hora de saída.

15. **Projeto: implementação de um CRUD**

A partir de um registro à sua escolha, implemente um vetor de registros e as funcionalidades do menu abaixo:



MENU PRINCIPAL

-
- 1 – INCLUIR UM REGISTRO
 - 2 – BUSCAR UM REGISTRO
 - 3 – ALTERAR UM REGISTRO
 - 4 – EXCLUIR UM REGISTRO
 - 5 – LISTAR REGISTROS EXISTENTES
 - 6 – SAIR DO SISTEMA

Considere que:

- O registro escolhido deverá ter, obrigatoriamente, um campo chave primária do tipo inteiro (pesquisar o que significa “campo chave primária”);
- Na inclusão de um registro, o campo chave primária deverá ser único, ou seja, não deverá ser permitida a inclusão de um registro cujo campo chave primária já exista no vetor;
- Na busca por um registro no vetor, o usuário deverá escolher o campo a partir do qual será efetuada a busca. Por exemplo, no caso do registro tProduto da questão 12, a busca poderia ser feita pelo campo “descrição”, “quantidade”, etc. Se existir mais de um registro cadastrado com aquela informação, todos deverão ser listados como resultados da busca;
- Na alteração de um registro, o algoritmo deverá garantir que o campo chave primária, após alterado, não poderá estar duplicado no vetor.
- Na exclusão do registro, o vetor não deve ficar com “buracos”, ou seja, quando um registro for excluído, todos os registros posteriores a este deverão ser movidos uma posição para a frente;
- Na listagem dos registros existentes, o algoritmo não deverá mostrar as posições do vetor sem registros incluídos previamente (com lixos de memória).
- Não poderão ser executadas inclusões se o vetor estiver cheio e não poderão ser realizadas exclusões se o vetor estiver vazio. O algoritmo deverá informar se uma dessas duas situações ocorrerem.