

9ª LISTA DE EXERCÍCIOS

- 1) Escreva um programa que armazene os dados bancários dos clientes de um banco, além de imprimir relatórios com informações variadas. Para isso, você deverá implementar o seguinte menu de opções:

1 – Ler dados:

Os dados a serem lidos são o número da conta, o CPF (use uma string de 13 caracteres) e o saldo atual (pode ser positivo, negativo ou zero). A leitura se encerra com um número de conta negativo. A cada cliente lido, seus dados deverão ser armazenados em um arquivo texto CONTAS.DAT.

2 – Excluir cliente:

Deverá ser informado o número da conta a ser excluída e, no arquivo, esse valor deve ser trocado para 0.

3 – Relatório completo:

Lista os dados de todas as contas válidas (ao encontrar uma conta de número 0, essa linha deve ser pulada do relatório -> não pode aparecer uma linha em branco no relatório final).

4 – Relatório de devedores:

Listar todos os dados dos clientes com saldo negativo.

5 – Otimizar BD:

Criar uma nova versão do arquivo CONTAS.DAT, a fim de eliminar os clientes excluídos. Para isso, crie um arquivo temporário TEMP.DAT para salvar apenas os dados dos clientes válidos de CONTAS.DAT. Ao final, exclua o arquivo CONTAS.DAT e renomeie TEMP.DAT para CONTAS.DAT, usando as funções `remove(nomeArq)` e `rename(nomeOld, nomeNew)`, ambas da biblioteca `stdio.h`

Observações:

- a) Cada item do menu deverá ser implementado em uma função separada.*
- b) Coloque todas as funções na biblioteca banco.h e inclua ela no seu programa principal.*

- 2) Desenvolva um sistema para controlar o inventário de uma rede de lojas de ferragens. O seu sistema deverá informar qual o estoque atual da rede de lojas, o estoque de uma loja específica, em que loja existe estoque disponível de um item qualquer, além de um relatório contendo os preços de cada item disponível na rede. O programa deverá utilizar apenas arquivos binários, e um menu de opções que permita a inclusão e exclusão dos itens comercializados, a atualização da quantidade em estoque e o valor de cada item. Para implementar o sistema, use as estruturas descritas abaixo (se precisar criar outras, faça isso):

Item:

struct contendo o número de registro (inteiro), o nome do item (string) e o preço.

Estoque:

struct contendo o número da loja (inteiro), um vetor de 200 elementos do tipo item, um vetor de 200 elementos (inteiros) representando a quantidade disponível do item correspondente na loja.

O menu principal do sistema deverá ter, pelo menos, as seguintes opções:

1 – Cadastrar produtos:

Inserir os dados de cada produto no arquivo binário ITENS.DAT.

2 – Inserir no estoque:

Inserir um produto no arquivo binário ESTOQUE.DAT

3 – Realizar venda:

Procurar o registro no arquivo ESTOQUE.DAT e dar baixa do produto na loja onde foi feita a venda.

4 – Atualizar produto:

Procura pelo código do produto no arquivo ITENS.DAT e atualiza seu preço.

5 – Comprar produto:

Incrementa a quantidade disponível do produto no arquivo ESTOQUE.DAT

6 – Otimizar BD:

Realiza a limpeza dos arquivos ITENS.DAT e ESTOQUE.DAT, de forma similar à explicada no exercício anterior.

7 – Relatório de estoque:

Imprime todos os itens em estoque, loja por loja.

8 – Relatório de produto:

Imprime todos os dados dos produtos.

9 – Relatório consolidado:

Imprime o total em estoque de cada produto, somando os valores disponíveis em cada loja. Se não conseguir, não precisa fazer esse.

Observações:

- a) O arquivo ESTOQUE.DAT você pode conter produtos repetidos, desde que em lojas diferentes.*
- b) Você deverá definir todas as funções do sistema, tentando manter todas as funções (inclusive a main) o menos complexas possível.*
- c) Coloque todas as funções na biblioteca lojas.h e inclua ela no seu programa principal.*
- d) Você pode desenvolver seu sistema de forma livre, adicionando mais funcionalidades e arquivos, tanto para o BD quanto para as suas bibliotecas.*