





Lista - 6


 Codifique, compile e execute um programa que:


- (a) crie/abra um arquivo texto de nome "arq.txt",
- (b) permita que o usuário entre com diversos caracteres nesse arquivo, até que o usuário entre com o caractere '0',
- (c) reinicie o arquivo, fazendo o ponteiro apontar para seu início, e
- (d) lendo-o caractere por caractere, e escrevendo na tela todos os caracteres armazenados.


 Faça um programa em Linguagem C que receba do usuário um arquivo, e mostre na tela quantas linhas esse arquivo possui.


 Faça um programa que receba dois arquivos do usuário, e crie um terceiro arquivo com o conteúdo dos dois primeiros juntos (o conteúdo do primeiro seguido do conteúdo do segundo).


 Desenvolva um programa em C que lê o conteúdo de um arquivo e cria um arquivo com o mesmo conteúdo, mas com todas as letras minúsculas convertidas para maiúsculas. Os nomes dos arquivos serão fornecidos, via teclado, pelo usuário. A função que converte maiúscula para minúscula é o `toupper()`. Ela é aplicada em cada caractere da string.


 Faça um programa no qual o usuário informa o nome do arquivo, e uma palavra, e retorne o número de vezes que aquela palavra aparece no arquivo.


 Faça um programa em C que recebe um vetor de n números, converta cada um desses números para binário e grave a sequência de 0s e 1s em um arquivo texto. Cada número deve ser gravado em uma linha.

 Faça um programa que leia um arquivo com os preços de diversos produtos, e calcule o total da compra.

 Faça um programa que permita que o usuário entre com diversos nomes e telefone para cadastro, e crie um arquivo com essas informações, uma por linha. O usuário finaliza a entrada com 0 para o telefone.

 Dado um arquivo contendo um conjunto de nome e data de nascimento (dd mm aa, isto é 3 inteiros seguidos), construir outro arquivo contendo o nome e a idade. O programa deve ler além do nome do arquivo a ser lido, a data de hoje.

 Faça um programa em C que recebe como entrada o nome de arquivo de entrada e o nome de um arquivo de saída. O arquivo de entrada contém em cada linha o nome de uma cidade ocupando 40 caracteres e o seu número de habitantes. A rotina deverá ler o arquivo de entrada e gerar um arquivo de saída onde aparece o nome da cidade mais populosa seguida pelo seu número de habitantes.

 Faça um programa em C que recebe como entrada o nome de um arquivo de entrada e o nome de um arquivo de saída e o ano corrente. Cada linha do arquivo de entrada contém o nome de uma pessoa ocupando 40 caracteres e o seu ano de nascimento. A rotina deverá ler o arquivo de entrada e gerar um arquivo de saída onde aparece o nome da pessoa seguida por uma string que representa a sua idade.

Se a idade for menor do que 18 anos, escrever menor de idade

Se a idade for maior do que 18 anos, escrever maior de idade

Se a idade for igual a 18 anos, escrever entrando na maior idade

Codifique, compile e execute um programa em C que leia a profissão e o tempo de serviço (em anos), de cada um dos 50 funcionários de uma empresa e armazene-os no arquivo "emp.txt". Cada linha do arquivo corresponde aos dados de um funcionário. Utilize o comando `fprintf()`. Em seguida, leia o mesmo arquivo utilizando `fscanf()`. Apresente os dados na tela.

Crie um programa em C que receba como entrada o número de alunos de uma disciplina. Aloque dinamicamente dois vetores para armazenar as informações a respeito desses alunos. O primeiro vetor contém o nome dos alunos e o segundo contém suas notas finais. Crie um vetor que armazene a cada linha o nome do aluno e sua nota final. Use nomes com no máximo 40 caracteres. Se o nome não contém 40 caracteres, complete com espaço em branco

Registre tudo que ocorrer com o carro: manutenções, combustível, lavagens, etc., tudo com o valor gasto e a quilometragem. Além de um relatório geral, inclua uma consulta sobre custo por quilômetro e consumo médio. Armazene os dados em arquivo binário.

Implemente um controle simples de mercadorias em uma despensa doméstica. Sobre cada produto podem ser armazenado um código numérico, descrição e quantidade atual. O programa deve ter opções para entrada e retirada de produtos, bem como um relatório geral e um de produtos não disponíveis. Armazene os dados em arquivo binário.

Faça um programa gerenciar uma agenda de contatos. Para cada contato armazene o nome, o telefone e o aniversário (dia e mês). O programa deve permitir (1) inserir contato, (2) remover contato, (3) pesquisar um contato pelo nome, (4) listar todos os contatos, (5) listar os contatos cujo nome inicia com uma dada letra, (6) imprimir os aniversariantes do mês. Sempre que o programa for encerrado, os contatos devem ser armazenados em um arquivo binário. Quando o programa iniciar, os contatos devem ser inicializados com os dados contidos neste arquivo binário.

Faça um programa para atualizar contas bancárias. O programa deve abrir quatro arquivos binários: (a) contas dos clientes no dia anterior, (b) movimentações no dia (débitos e créditos), (c) contas criadas no dia (contendo o saldo inicial) e (d) contas removidas no dia. O saldo atual de cada conta deve ser atualizado com base nas movimentações diárias. Um novo arquivo binário de contas de clientes deve ser criado, contendo o saldo atualizado de cada cliente (com base nas movimentações diárias) e as novas contas. As contas removidas no dia não devem aparecer neste novo arquivo. Após criar este arquivo o programa é encerrado. Os vetores utilizados devem ser alocados dinamicamente. Os dados de cada cliente são: número da conta (produzido automaticamente pelo sistema), nome e saldo. Uma movimentação é composta de número da conta, tipo da operação (crédito ou débito), valor.

De acordo com o exercício anterior grave em um arquivo binário (a) todas as movimentações que produziram saldo negativo, (b) tentativas de remover uma conta que não existe.

Faça um programa que receba do usuário um arquivo que contenha o nome e a nota de diversos alunos (da seguinte forma: NOME: JOÃO NOTA: 8), um aluno por linha. Mostre na tela o nome e a nota do aluno que possui a maior nota.

Faça uma rotina em C que receba como entrada o nome de um arquivo de entrada e o nome de um arquivo de saída. O arquivo de entrada contém o nome de um aluno ocupando 40 caracteres e três inteiros que indicam suas notas. A rotina deverá ler o arquivo de entrada e gerar um arquivo de saída onde aparece o nome do aluno e as suas notas em ordem crescente.

Crie um programa em C que receba como entrada o número de alunos de uma disciplina. Aloque dinamicamente dois vetores para armazenar as informações a respeito desses alunos. O primeiro vetor contém o nome dos alunos e o segundo contém suas notas finais. Crie um vetor que armazene a cada linha o nome do aluno e sua nota final. Use nomes com no máximo 40 caracteres. Se o nome não contém 40 caracteres, complete com espaço em branco.

Crie um programa em C que receba como entrada o número de alunos de uma disciplina. Aloque dinamicamente em uma estrutura para armazenar as informações a respeito desses alunos (nome do aluno e sua nota final). Use nomes com no máximo 40 caracteres. Se o nome não contém 40 caracteres, complete com espaço em branco.

Crie um programa em C que declare uma estrutura (registro) para o cadastro de alunos.

- (a) Deverão ser armazenados, para cada aluno: matrícula, sobrenome (apenas um), e ano de nascimento.
- (b) Ao início do programa, o usuário deverá informar o número de alunos que serão armazenados
- (c) o programa deverá alocar dinamicamente a quantidade necessária de memória para armazenar os registros dos alunos.
- (d) O programa deverá pedir ao usuário que entre com as informações dos alunos.
- (e) Em seguida, essas informações deverão ser gravadas em um arquivo
- (f) Ao final, mostrar os dados armazenados e liberar a memória alocada.

Faça um programa para gerenciar as notas dos alunos de uma turma com um menu contendo as seguintes opções:

- 1. Definir informações da Turma;
- 2. Inserir aluno e notas;
- 3. Exibir alunos e médias;
- 4. Exibir alunos aprovados;
- 5. Exibir alunos reprovados;
- 6. Salvar dados em Disco;
- 7. Sair do programa (fim).

Faça a rotina que gerencia o menu dentro do main, e para cada uma das opções deste menu, crie uma sub-rotina específica.