Exercícios: Arquivos

1. Escreva um programa que:

   (a) Crie/abra um arquivo texto de nome “arq.txt”

   (b) Permita que o usuário grave diversos caracteres nesse arquivo, até que o usuário entre com o caractere ‘0’

   (c) Feche o arquivo

   Agora, abra e leia o arquivo, caractere por caractere, e escreva na tela todos os caracteres armazenados.

2. Faça um programa que receba do usuário um arquivo texto e mostre na tela quantas linhas esse arquivo possui.

3. Faça um programa que receba do usuário um arquivo texto e mostre na tela quantas letras são vogais.

4. Faça um programa que receba do usuário um arquivo texto e mostre na tela quantas letras são vogais e quantas são consoantes.

5. Faça um programa que receba do usuário um arquivo texto e um caracter. Mostre na tela quantas vezes aquele caractere ocorre dentro do arquivo.

6. Faça um programa que receba do usuário um arquivo texto e mostre na tela quantas vezes cada letra do alfabeto aparece dentro do arquivo.

7. Faça um programa que receba do usuário um arquivo texto. Crie outro arquivo texto contendo o texto do arquivo de entrada, mas com as vogais substituídas por ‘*’.

8. Faça um programa que leia o conteúdo de um arquivo e crie 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.

9. 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).

10. Faça um programa que receba o nome de um arquivo de entrada e outro de saída. O arquivo de entrada contém em cada linha o nome de uma cidade (ocupando 40 caracteres) e o seu número de habitantes. O programa 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.

11. 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.

12. Abra um arquivo texto, calcule e escreva o número de caracteres, o número de linhas e o número de palavras neste arquivo. Escreva também quantas vezes cada letra ocorre no arquivo (ignorando letras com acento). Obs.: palavras são separadas por um ou mais caracteres espaço, tabulação ou nova linha.

13. 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.

14. Dado um arquivo contendo um conjunto de nome e data de nascimento (DD MM AAAA, isto é, 3 inteiros em sequência), faça um programa que leia o nome do arquivo e a data de hoje e construa outro arquivo contendo o nome e a idade de cada pessoa do primeiro arquivo.

15. Faça um programa que receba como entrada o ano corrente e o nome de dois arquivos: um de entrada e outro de saída. Cada linha do arquivo de entrada contém o nome de uma pessoa (ocupando 40 caracteres) e o seu ano de nascimento. O programa 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, escreva "menor de idade"
    - Se a idade for maior do que 18 anos, escreva "maior de idade"
    - Se a idade for igual a 18 anos, escreva "entrando na maior idade"

16. Faça um programa que recebe um vetor de 10 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.

17. Faça um programa que leia um arquivo que contenha as dimensões de uma matriz (linha e coluna), a quantidade de posições que serão anuladas, e as posições a serem anuladas (linha e coluna). O programa lê esse arquivo e, em seguida, produz um novo arquivo com a matriz com as dimensões dadas no arquivo lido, e todas as posições especificadas no arquivo ZERADAS e o restante recebendo o valor 1.
    Ex: arquivo "matriz.txt"

    ```
    3 3 2 /*3 e 3 dimensões da matriz e 2 posições que serão anuladas*/
    1 0
    1 2  /*Posições da matriz que serão anuladas.
    ```

    arquivo "matriz_saida.txt"
    saída:

    ```
    1 1 1
    0 1 0
    1 1 1
    ```

18. Faça um programa que leia um arquivo contendo o nome e o preço de diversos produtos (separados por linha), e calcule o total da compra.

19. 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.

20. Crie um programa 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 arquivo 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.

21. Crie um programa 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. Em seguida, salve os dados dos alunos em um arquivo binário. Por fim, leia o arquivo e mostre o nome do aluno com a maior nota.

22. Faça um programa que recebe 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. O programa 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.

23. Escreva um programa que leia a profissão e o tempo de serviço (em anos) de cada um dos 5 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.

24. Implemente um controle simples de mercadorias em uma despensa doméstica. Para cada produto armazene 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.

25. 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

    (a) inserir contato
    (b) remover contato
    (c) pesquisar um contato pelo nome
    (d) listar todos os contatos
    (e) listar os contatos cujo nome inicia com uma dada letra
    (f) 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.

26. Crie um programa que declare uma estrutura 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.

Ao iniciar o programa, forneça ao usuário uma opção para carregar os registros do arquivo para a memória do computador alocando dinamicamente a quantidade de memória necessária.
Dica: para que o usuário possa entrar com novos dados, além dos que foram obtidos a partir do arquivo, use a função realloc() para realocar a quantidade de memória usada.

27. Faça um programa para gerenciar as notas dos alunos de uma turma salva em um arquivo. O programa deverá ter um menu contendo as seguintes opções:

    (a) Definir informações da turma;
    (b) Inserir aluno e notas;
    (c) Exibir alunos e médias;
    (d) Exibir alunos aprovados;
    (e) Exibir alunos reprovados;
    (f) Salvar dados em Disco;
    (g) Sair do programa (fim).

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

28. Faça um programa que recebe como entrada o nome de um arquivo de entrada e o nome de um arquivo de saída. Cada linha do arquivo de entrada possui colunas de tamanho de 30 caracteres. No arquivo de saída deverá ser escrito o arquivo de entrada de forma inversa. Veja um exemplo:

    Arquivo de entrada:
    Hoje é dia de prova de AP
    A prova está muito fácil
    Vou tirar uma boa nota

    Arquivo de saída: Aton aob amu rarit uov
    Licáf otium átse avorp A
    PA ed avorp ed aid é ejoH

29. Codifique um programa que manipule um arquivo contendo registros descritos pelos seguintes campos: codigo_vendedor, nome_vendedor, valor_da_venda e mes.
    A manipulação do arquivo em questão é feita através da execução das operações disponibilizadas pelo seguinte menu:

    - Criar o arquivo de dados;
    - Incluir um determinado registro no arquivo;
    - Excluir um determinado vendedor no arquivo;
    - Alterar o valor de uma venda no arquivo;
    - Imprimir os registros na saída padrão;
    - Excluir o arquivo de dados;
    - Finalizar o programa.

    Os registros devem estar ordenados no arquivo, de forma crescente, de acordo com as informações contidas nos campos codigo_vendedor e mes. Não deve existir mais de um registro no arquivo com mesmos valores nos campos codigo_vendedor e mês.