

## Lista de exercícios – Arquivos

1. 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' [zero]; (c) reinicie o arquivo, fazendo o *stream* apontar para seu início; e (d) lendo-o caractere por caractere, e escrevendo na tela todos os caracteres armazenados.
2. Faça um programa que receba do usuário um arquivo, e mostre na tela quantas linhas tem esse arquivo. Teste o programa com arquivos de quantidade de linhas variáveis.
3. 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).
4. Desenvolver um programa 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. (Use a função da linguagem que converte maiúscula para minúscula). Aplique-a a cada caractere da *string*.
5. 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.
6. 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:

<b>Arquivo de entrada:</b> Hoje é dia de prova de PRG1 A prova está muito fácil Vou tirar uma boa nota	<b>Arquivo de saída:</b> Aton aob amu rarit uov Licáf otium átse avorp A 1GRP ed avorp ed aid é ejoH
---	---

7. Faça um programa 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.
8. Faça um programa que leia um arquivo com os preços de diversos produtos, e calcule o total da compra.
9. Faça um programa que permita que o usuário entre com diversos nomes e telefone para cadastro, e crie um arquivo com essas informações, com cada par de dados (nome e telefone) em cada linha. O usuário finaliza a entrada com string vazia para nome e 0 para o telefone.
10. Dado um arquivo contendo um conjunto de nome de pessoas e data de nascimento (DD MM AAAA, isto é 3 inteiros seguidos), construir outro arquivo contendo os nomes e a idade de cada pessoas. O programa deve ler o nome do arquivo a ser lido e carregar a data atual do sistema.
11. 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 em cada linha o nome de uma cidade 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.
12. Faça um programa 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 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 e 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.

13. Codifique, compile e execute um programa 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 "empresa.txt". Cada linha do arquivo corresponde aos dados de um funcionários. Em seguida, leia o mesmo arquivo utilizando e apresente os dados na tela.
14. Crie um programa que receba como entrada o número de alunos de uma disciplina. Crie estrutura(s) para armazenar as informações a respeito desses alunos. Após a leitura, armazene os dados (nome do aluno e sua nota final) em um arquivo. Cada linha do arquivo contém dados de um aluno. Use nomes com no máximo 40 caracteres. Se o nome não contém 40 caracteres, complete com espaços em branco.
15. Codifique um programa, que manipule um arquivo contendo registros descritos pelos seguintes campos: código do vendedor, nome do vendedor, valor da venda e mês de referencia. A manipulação do arquivo em questão é feita através da execução das operações disponibilizadas pelo seguinte *menu*:
  - 1 - Para criar o arquivo de dados;
  - 2 - Para incluir um determinado registro no arquivo;
  - 3 - Para excluir um determinado vendedor no arquivo;
  - 4 - Para alterar o valor de uma venda no arquivo;
  - 5 - Para imprimir os registros na saída padrão;
  - 6 - Para excluir o arquivo de dados;
  - 0 - Para finalizar o programa.Os registros devem estar ordenados no arquivo, de forma crescente, de acordo com as informações contidas nos campos código do vendedor e mês. Não deve existir mais de um registro no arquivo com os mesmos valores nos campos código do vendedor e mês.
16. Abra um arquivo cujo nome é informado como argumento na linha de comando. 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.
17. 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 de produtos e suas quantidades em estoque. Seu programa também deve gerar um relatório com os produtos não disponíveis (que estão com estoque vazio). Armazene os dados em arquivo.
18. Faça um programa que gerencia uma agenda de contatos. Para cada contato armazene o nome da pessoa, o telefone e o aniversário (DD/MM/AAAA). 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. Quando o programa iniciar, os contatos armazenados no arquivo devem ser transferidos para alguma estrutura para manipulação.
19. Faça um programa para atualizar contas bancárias. O programa deve abrir quatro arquivos: (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 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. Utilize algumas estruturas para alocar os dados. 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. Também grave em um arquivo (a) todas as movimentações que produziram saldo negativo, e (b) tentativas de remover uma conta que não existe.

20. Faça um programa que lê um arquivo que contém na primeira linha três informações (N, M, Q) referentes às dimensões de uma matriz (N e M refere-se à quantidade de linhas e colunas, respectivamente, e Q representa a quantidade de posições que serão anuladas). Em seguida, no mesmo arquivo, deve ser lido Q linhas com as posições da matriz que devem ser anuladas (zeradas). Faça um programa que abra esse arquivo em modo leitura e imprima a matriz com as dimensões dadas no arquivo lido, e todas as posições especificadas no arquivo ZERADAS e o restante recebendo o valor 1.

Exemplo: arquivo "matriz.txt"

```
3 3 2          # 3 e 3 dimensões da matriz; 2 posições que serão anuladas*/
1 0
1 2          # [1,0] e [1,2] são as posições da matriz que serão anuladas.
```

Saída:

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

PS: Observem que matrizes em linguagens type C indexam os elementos a partir de 0.

21. Faça um programa para gerenciar os dados e as notas dos alunos de uma turma com um *menu* contendo as seguintes opções:

- 1 - Inserir alunos e notas;
- 2 - Inserir as notas dos alunos;
- 3 - Exibir alunos e notas;
- 4 - Exibir alunos, notas e médias;
- 5 - Exibir alunos aprovados;
- 6 - Exibir alunos reprovados;
- 7 - Mostrar o aluno com a maior média;
- 7 - Salvar dados em Disco;
- 0. Sair do programa (fim).

- a) Faça a rotina que gerencia o *menu*, e para cada uma das opções deste menu, crie uma função específica.
- b) O programa deve manipular dois arquivos ("aluno.txt" e "notas.txt"): O primeiro arquivo deverá armazenar, para cada aluno, por linha: matrícula, nome e ano de nascimento. O segundo arquivo deverá armazenar por linha a matrícula e três notas.
- c) O programa deverá utilizar estruturas (tupla, lista e/ou dicionário) para armazenar os dados dos alunos.
- d) A cada leitura de dados, as mesmas devem ser gravadas nos respectivos arquivos.