

# Algoritmos e Programação II

## VI Lista de Exercícios: Manipulação de Arquivos

Prof. Evandro C. R. Rosa  
UNIVALI

Nome Completo: \_\_\_\_\_ Matrícula: \_\_\_\_\_

1. Os códigos abaixo contêm um ou mais erros. Identifique-os e explique como corrigi-los:

- (a) 

```
fstream file(ios::in | ios::out);  
file.open("info.dat");  
if (!file) {  
    cout << "Não foi possível abrir o arquivo.\n";  
}
```
- (b) 

```
ofstream file;  
file.open("info.dat", ios::in);  
if (file) {  
    cout << "Não foi possível abrir o arquivo.\n";  
}
```
- (c) 

```
fstream data_file("info.dat", ios::in | ios::binary);  
int x = 5;  
data_file << x;
```
- (d) 

```
fstream data_file("info.dat", ios::in);  
char stuff[81];  
data_file.get(stuff);
```
- (e) 

```
fstream data_file("info.dat", ios::in);  
char stuff[81] = "abcdefghijklmnopqrstuvwxyz";  
data_file.put(stuff);
```
- (f) 

```
fstream data_file("info.dat", ios::out);  
struct Date { int month; int day; int year; };  
Date dt = { 4, 2, 98 };  
data_file.write(&dt, sizeof(int));
```

2. Escreva um programa que solicite ao usuário o nome de um arquivo. O programa deve exibir as 10 primeiras linhas do arquivo na tela. Se o arquivo tiver menos de 10 linhas, o conteúdo inteiro deve ser exibido, com uma mensagem indicando que todo o arquivo foi mostrado.
3. Escreva um programa que solicite ao usuário o nome de um arquivo. O programa deve exibir o conteúdo do arquivo na tela. Se o conteúdo do arquivo não couber em 24 linhas, o programa deve exibir 24 linhas por vez e depois pausar. A cada pausa, o programa deve esperar o usuário pressionar uma tecla para exibir as próximas 24 linhas.

4. Escreva um programa que solicite ao usuário o nome de um arquivo. O programa deve exibir as últimas 10 linhas do arquivo na tela. Se o arquivo tiver menos de 10 linhas, o conteúdo inteiro deve ser exibido, com uma mensagem indicando que todo o arquivo foi mostrado.
5. Escreva um programa que solicite ao usuário o nome de um arquivo. O programa deve exibir o conteúdo do arquivo na tela. Cada linha exibida deve ser precedida de um número de linha, seguido por dois pontos. A numeração das linhas deve começar em 1. Veja o exemplo:

```
1: Olá
2: Esse é um exemplo de arquivo
3: com apenas 3 linhas.
```

Se o conteúdo do arquivo não couber em uma única tela, o programa deve exibir 24 linhas por vez e depois pausar. A cada pausa, o programa deve esperar o usuário pressionar uma tecla para exibir as próximas 24 linhas.

6. Escreva um programa que solicite ao usuário o nome de um arquivo e uma string para busca. O programa deve procurar no arquivo todas as ocorrências da string especificada. Quando a string for encontrada, a linha que a contém deve ser exibida. Após todas as ocorrências serem localizadas, o programa deve informar quantas vezes a string apareceu no arquivo.
7. Escreva uma função chamada `array_para_arquivo`. A função deve aceitar três argumentos: o nome de um arquivo, um ponteiro para um array de inteiros e o tamanho do array. A função deve abrir o arquivo especificado no modo binário, escrever o conteúdo do array no arquivo e depois fechar o arquivo. Escreva outra função chamada `arquivo_para_array`. Esta função deve aceitar três argumentos: o nome de um arquivo, um ponteiro para um array de inteiros e o tamanho do array. A função deve abrir o arquivo no modo binário, ler seu conteúdo para o array e depois fechar o arquivo. Escreva um programa completo que demonstre essas funções, utilizando `array_para_arquivo` para gravar um array em um arquivo, e em seguida, usando `arquivo_para_array` para ler os dados do mesmo arquivo. Após ler os dados, exiba o conteúdo do array na tela.
8. Criptografia de arquivos é a ciência de escrever o conteúdo de um arquivo em código secreto. Seu programa de criptografia deve funcionar como um filtro, lendo o conteúdo de um arquivo, modificando os dados para um código e escrevendo o conteúdo codificado em um segundo arquivo. O segundo arquivo será uma versão do primeiro, mas escrito em código secreto. Embora existam técnicas de criptografia complexas, você deve criar uma simples. Por exemplo, você pode ler o primeiro arquivo um caractere por vez e adicionar 10 ao código ASCII de cada caractere antes de escrevê-lo no segundo arquivo.
9. Escreva um programa que descriptografe o arquivo produzido pelo programa do exercício anterior. O programa de descriptografia deve ler o conteúdo do arquivo codificado, restaurar os dados ao seu estado original e escrevê-los em outro arquivo.