



Exercícios para auxiliar nos estudos – *strings, structs* e arquivos

Os exercícios não devem ser entregues. Eles servem apenas como auxílio no estudo para terceira avaliação.

1. Faça um programa que receba do usuário um arquivo texto e mostre na tela quantas linhas esse arquivo possui.
2. Faça um programa que receba do usuário um arquivo texto e um caractere. Mostre na tela quantas vezes aquele caractere ocorre dentro do arquivo.
3. 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 '*'.
4. 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). Os nomes dos três arquivos devem ser fornecidos pelo usuário.
5. 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.
6. Melhore o programa anterior, fazendo estatísticas sobre o arquivo: informe quantas (e o percentual) de palavras com 2, 3, 4, 5 e mais de 5 letras que existem no arquivo.
7. Crie um programa que receba como entrada o número de alunos de uma disciplina. Construa uma estrutura para armazenar informações a respeito desses alunos: nome do aluno, data de nascimento, 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.
8. Faça um programa que copie um arquivo binário de inteiros (denominado "números.bin") para um outro arquivo binário de inteiros (denominado "pares.bin") de tal forma que apenas os números pares sejam copiados.
9. Faça um programa que gere e salve em um arquivo os 10 mil primeiros números primos.
10. Escreva um programa que lê e modifica um arquivo texto lido trocando cada letra pela sua correspondente maiúscula. Exemplo: o arquivo abaixo deve ser modificado como a seguir:

Antes

```
a certificacao de metodologias que nos
auxiliam a lidar com o julgamento imparcial
das eventualidades aponta para a melhoria dos
niveis de motivacao departamental.
```

Depois

A CERTIFICACAO DE METODOLOGIAS QUE NOS
AUXILIAM A LIDAR COM O JULGAMENTO IMPARCIAL
DAS EVENTUALIDADES APONTA PARA A MELHORIA DOS
NIVEIS DE MOTIVACAO DEPARTAMENTAL.

11. Faça um programa de criptografia de dados, ou seja, um programa capaz de ler um arquivo texto, codificar este arquivo através de alguma técnica de alteração do código ASCII (exemplo: somar 1 ao valor ASCII de cada caracter), e escrever em disco o arquivo codificado. Crie um outro programa que descriptografe um arquivo criado pelo programa de criptografia, realizando a operação inversa: ler o arquivo do disco, decodificar e escrever o novo arquivo em disco descriptografado. Lembre-se que para que seja possível criptografar/descriptografar um arquivo a função de codificação deve possuir uma função inversa. Exemplo de código de criptografia:

```
tamanho = strlen(linha);  
  
for (i=0; i < tamanho; i++)  
    linha_cripto[i] = linha[i] + 1; // Soma 1 ao caracter da string  
linha_cripto[tamanho]='\0'; //Adiciona a marca de fim de string
```

Desafios: Você consegue desenvolver uma função de criptografia/descriptografia que seja mais sofisticada e menos óbvia do que essa? Você conseguiria adaptar este programa para criptografar qualquer tipo de arquivo, seja ele um arquivo texto ou binário?

Data a declaração da *string*, conforme:
`char str[100];`

Explique o que faz cada um dos comandos abaixo

```
scanf("%[^\\n]", str);  
scanf(" 99%[^\\n]", str);
```

12. Faça um programa que apresente um menu com as opções: adicionar um cliente, alterar informações do cliente, imprimir informações do cliente e remover um cliente. As informações de clientes que devem ser armazenadas são: nome, endereço, data de cadastro. Essas informações do cliente devem ser armazenadas em um arquivo. Lembre-se que a alteração das informações do cliente e remoção também devem ser feitas no arquivo.
13. Faça um programa que leia uma matriz 3x3 do arquivo e calcule o determinante.
14. Faça um programa que leia uma matriz quadrada de qualquer dimensão ($n \times n$) e calcule o determinante.
15. Dentro da pasta “listas” no Google Drive existem dois arquivos usados neste exercícios. O primeiro é o “car.data.txt”, em que cada linha do arquivo representa informações de um veículo. O arquivo “car.names.txt” contém o nome de cada um dos atributos (colunas) do arquivo “car.data.txt”. Analisando os arquivos, mostre:
- A quantidade de carros com 2, 3, 4 e mais de 5 portas;
 - Quantidade de carros com capacidade para mais de 4 pessoas;
 - Indique a porcentagem de carros dessa lista que tem “med” como valor do atributo “safety” (6ª coluna).

Os dados foram adquiridos diretamente do *Machine Learning Repository* (UCI) e podem ser acessados pelo link:
<https://archive.ics.uci.edu/ml/datasets/Car+Evaluation>