

ALGORITMOS II

3ª LISTA DE EXERCÍCIOS – ARQUIVO (TEXTO)

Desenvolva todos os exercícios listados em Linguagem C.

- 1 Escreva um programa que mostre um arquivo texto no vídeo. O nome do arquivo deve ser fornecido pelo teclado.
- 2 Faça um programa que conte o número de vogais e consoantes de um arquivo texto.
- 3 Faça um programa que processe um texto convertendo as letras maiúsculas em minúsculas. A inicial de cada frase deve ficar maiúscula.
- 4 Faça um programa que conte o número de linhas, o número de caracteres e o número de palavras de um arquivo texto.
- 5 Faça um programa no qual o usuário informa o nome do arquivo texto e uma palavra, e retorne o número de vezes que aquela palavra aparece no arquivo.
- 6 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 '*’.
- 7 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).
- 8 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.
- 9 Escreva um programa que leia uma matriz de números inteiros armazenada em um arquivo texto com o seguinte formato: a primeira linha do arquivo contém o número de linhas (m) e número de colunas (n) da matriz e, as m linhas seguintes contém as informações das linhas da matriz.

Exemplo:

3	3	
1	2	3
4	5	6
7	8	9

- 10 Faça um programa que copie um arquivo texto em outro arquivo mantendo a estrutura de linhas.
- 11 Dada uma matriz de 1000 linhas e 80 colunas, contendo um texto, faça um programa para compactar e descompactar o texto. Isto é, cada palavra deve ser armazenada no vetor uma única vez:
 - ° leia a matriz;
 - ° construa um vetor onde cada elemento é um registro com:
 - um vetor com 20 elementos para armazenar a palavra;
 - um vetor com 10 elementos para armazenar as posições onde a palavra ocorre na matriz.

Está sendo suposto que cada palavra terá no máximo 20 letras e que o número máximo de vezes que ela ocorre no texto (matriz) é 10.

- 12 Faça um programa que leia o arquivo texto "texto.txt" e grave todas as palavras iniciadas com vogais no arquivo texto "vogais.txt". O programa deve também mostrar na tela o arquivo "texto.txt", durante a procura das palavras, e o número de palavras gravadas em "vogais.txt".
- 13 Faça um programa em C 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

- 14 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.

Exemplo: 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
```

- 15 Dado um arquivo texto, faça um programa para substituir as palavras (traduzir o texto). Suponha que exista um arquivo dicionário. Leia cada palavra do texto, verifique se ela existe no dicionário e então faça a substituição, caso contrário, não substitua a palavra.
- 16 Faça um programa que formate um texto qualquer, imprimindo 60 caracteres por linha com margens direita e esquerda alinhadas, páginas numeradas e com no máximo 60 linhas por página. O arquivo é composto por linhas de texto e linhas de controle, onde as linhas de controle podem ser do tipo: .pp (a próxima linha começa com parágrafo) e, .ds e .de (o texto entre essas duas linhas não deve ser formatado).
- 17 Dados dois vetores inteiros A e B, de tamanho máximo 30 e 40 respectivamente, faça um programa que leia os vetores A e B, ordene-os e gere o vetor C ordenado a partir de A e B. Os vetores A e B devem ser lidos de dois arquivos texto e o vetor C deve ser gravado em um arquivo texto.

Observação: Os nomes dos arquivos devem ser fornecidos pelo teclado.

Exemplo:

Ler:	arq1.dat	arq2.dat	arq3.dat
	10	24	3
	3	11	8
			10
			11
			24

- 18** Considere um arquivo de dados do tipo texto com o seguinte conteúdo:

```
3
Cida Souza
5.5
4.0
José Silva e Silva
7.5
8.5
Maria das Dores
5.0
6.0
```

O arquivo acima é apenas um exemplo. Neste arquivo de alunos a primeira linha contém o número de alunos no arquivo. As linhas seguintes contém os seguintes dados:

```
nome do aluno com no máximo 50 caracteres;
nota da primeira prova;
nota da segunda prova.
```

Escreva um programa que imprima os nomes de todos os alunos que têm a média das duas notas menor que 5.0

- 19** Escreva um programa que leia um arquivo texto contendo linhas de dados. Em cada linha do arquivo há o nome de um aluno e duas notas. Estes dados estão separados por ponto e vírgula. Existe um ponto e vírgula ao final de cada linha. O formato dos dados é o seguinte:

```
Cida Souza; 5.5; 4.0;
Jose Silva e Silva: 7.5; 8.5;
```

O programa deve ler estes dados e imprimir os valores lidos, a média das duas notas e se o aluno foi aprovado ou não (media \geq 5). O formato de saída é:

```
Cida Souza 5.5 4.0 4.25 reprovado
Jose Silva e Silva 7.5 8.5 8.0 aprovado
```

- 20** Escreva um programa que compare dois arquivos especificados pelo usuário e imprima sempre que os caracteres dos dois arquivos coincidirem. Por exemplo:

```
arquivo1.txt
Olá, pessoal!

arquivo2.txt
Oi, como vai?
```

Neste caso, os caracteres na primeira e décima primeira posição são iguais nos dois arquivos. A saída do seu programa deve ser algo como:

```
1 - O (79)
11 - a (97)
```

Os valores entre parênteses são os respectivos códigos ASCII dos caracteres.

- 21** Escreva um programa que leia um arquivo texto contendo linhas de dados. Em cada linha do arquivo há o nome de um aluno e duas notas. Estes dados estão separados por ponto e vírgula. Existe um ponto e vírgula ao final de cada linha. O formato dos dados é o seguinte:

```
ze sa; 10.0; 9.0;
antonio silva: 9.0; 7.0;
```

O programa deve ler estes dados e imprimir os valores lidos, a média das duas notas e se o aluno foi aprovado ou não (média ≥ 5). O formato de saída é:

```
ze sa 10.0 9.0 8.0 aprovado
antonio silva 9.0 7.0 8.0 aprovado
```

- 22** Escreva um programa que armazene em um arquivo binário os dados de pessoas, de acordo com a estrutura a seguir.

```
struct pessoa{
    char nome[50];
    int idade;
    float altura;
    char telefone[10];
} amigos[20];
```

Em seguida, faça um programa que leia os dados do arquivo gerado anteriormente e salve-os num novo arquivo (formato texto) utilizando uma saída formatada.

FORMATO:

[nome] tem [idade] anos e [altura] de altura.
Tel.: [telefone].

- 23** Considere um arquivo texto (denominado "dados.txt") que contém, em cada linha, as seguintes informações sobre os modelos de uma determinada agência:

- nome (cadeia de 30 caracteres);
- sexo (um caractere, podendo ser 'M' ou 'F');
- cor dos olhos (um caractere, podendo ser 'A', 'B', 'C', ou 'D');
- altura (valor real em metros);
- peso (valor real em quilos).

Faça um programa que separe este arquivo texto em dois arquivos binários de registros cujos campos são as informações descritas acima: o 1º arquivo deve conter apenas os modelos do sexo masculino ('M') e o 2º arquivo deve conter apenas os modelos do sexo feminino ('F').

- 24** 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.