

## Lista 8 (EXERCÍCIOS)

1) Escreva um programa que leia um arquivo contendo sequências de tuplas (nome, número), separadas por linhas. Considere um máximo de 256 entradas, cada nome com um máximo de 255 caracteres.

2) Modifique o programa anterior. Desta vez, inclua uma rotina que ordene os registros de acordo com os valores numéricos, em ordem decrescente. Os registros ordenados deverão ser salvos substituindo o arquivo original.

Exemplo:

João 14

Pedro 3

José 43

Saída:

Pedro 3

João 14

José 43

Dica: você pode escrever sua própria rotina de ordenação, entretanto, irá agregar mais conhecimento se aprender a utilizar a rotina `qsort`.

3) Modifique o programa anterior para incluir um registro. Remova as entradas excedentes caso o limite de 256 entradas seja extrapolado. Como critério de escolha, remova os registros cujos valores numéricos sejam os maiores.

4) Crie um programa que receba por linha de comando um argumento, sendo este o nome de um arquivo. O programa deverá então receber uma string como entrada do usuário e procurar quantas vezes esta string ocorre no arquivo.

5) Crie um programa que receba dois arquivos como parâmetros por linha de comando. O programa deverá gerar um relatório em arquivo texto com a comparação byte a byte dos dois arquivos, apontando em qual endereço (offset) eles diferem e por quais valores (em hexadecimal). Sugestão: chame seu programa de `bdiff`.

6) Escreva um programa que remova todas as quebras de linha de um arquivo texto (considere tanto as quebras de linha do Unix, `\n`, quanto do Windows, `\r\n`).

7) Escreva um programa que conta o tamanho de um arquivo.

Dica: abra o arquivo em modo binário! Caso contrário, o tamanho observado em modo texto é um tamanho virtual.

8 ) Escreva um programa que conta a quantidade de letras (ignorando espaços, números e outros caracteres) em um arquivo.

9) Escreva um programa que conta a quantidade de palavras em um arquivo.

10) Crie um arquivo “test.txt” e preencha-o com “abcdefghijklmnopqrstuvxz”. Qual a saída esperada do seguinte programa?

```
FILE *fp = fopen("test.txt", "rb+");  
fseek(fp, 0, SEEK_SET);  
fwrite("A", 1, 1, fp);  
fwrite("B", 1, 1, fp);  
fseek(fp, 5, SEEK_SET);  
fwrite("C", 1, 1, fp);  
fseek(fp, 0, SEEK_END);  
fwrite("D", 1, 1, fp);  
fseek(fp, 1, SEEK_END);  
fwrite("E", 1, 1, fp);  
fclose(fp);
```

OBS: utilize um editor hexadecimal para interpretar os resultados no arquivo texto.

11) Altere o código anterior para que “z” seja sobrescrito por “D” e logo em seguida “E” seja escrito.

12) Crie uma variável do tipo inteiro e atribua algum valor. Salve essa variável para um arquivo vazio utilizando fwrite. Abra o arquivo escrito com um editor hexadecimal e avalie qual a byte-order do inteiro (little endian ou big endian).

13) Escreva um programa que ao ser compilado e executado escreva seu próprio código fonte na tela.

14) Repita o programa anterior, dessa vez escreva o código fonte ao contrário, indo do final do arquivo ao começo.

15) Escreva um programa que conta o número de linhas em um arquivo.

16) Formatos de arquivo são utilizados para padronizar arquivos, criando uma estrutura ordenada de dados. Procure pela descrição do formato de arquivo BMP (bitmap) e escreva seu analisador BMP. Seu programa deve receber como parâmetro por linha de comando o nome de um arquivo BMP e informar a altura, largura, número de bits por pixel e tamanho da imagem.

17) Crie um programa que leia um arquivo “mapa.txt”. Nesse arquivo, um mapa 2D estará presente, por exemplo:

```
*****A*****
*****          ***** *
*                ***** ** *
*                *          *
*****          *          *****
*****          *****      ***
B                *****      ***
*****
```

Nesta tarefa, considere o mapa como de tamanho fixo. Os asteriscos são áreas que proíbem movimento enquanto espaços em branco são áreas que permitem movimento.

Escreva um programa que encontra um caminho, numa sequência de movimentos C, B, E, D (cima,baixo, esquerda e direita) entre os pontos A e B. Salve essa sequência em um arquivo texto.

18) Escreva um programa que receba como parâmetros um arquivo de mapa e um arquivo de sequências, do exercício anterior, e imprima na tela um replay passo a passo do movimento de uma bolinha caminhando de A até B.