Lista 10 de Exercícios

Exercícios sobre Arquivos

1) Considere o tipo Funcionario, que representa um funcionário de uma empresa, definido pela estrutura a seguir:

```
struct funcionario {
  char nome[81]; /* nome do funcionário */
  float valor_hora; /* valor da hora de trabalho em Reais */
  int horas_mes; /* horas trabalhadas em um mês */
};
typedef struct funcionario Funcionario;
```

Escreva uma função em C que preencha um vetor de ponteiros para Funcionario com os dados lidos de um arquivo texto. Essa função deve receber como parâmetros o vetor de ponteiros para Funcionario (representado pelo seu comprimento e pelo endereço de seu primeiro elemento) previamente alocado e o nome do arquivo de entrada. Nesse arquivo de entrada, os dados de cada funcionário são armazenados em duas linhas: uma com o seu nome (cadeia com até 80 caracteres), e outra com o valor de sua hora de trabalho e com o número de horas trabalhadas em um mês (nessa ordem). Um exemplo desse formato é mostrado abaixo.

```
João da Silva
15.0 160
Manuel Santos
15.0 80
Fulana de Tal
23.5 40
```

Considere que: não existem linhas em branco no arquivo; o parâmetro vet já vem com todas as suas posições inicializadas com NULL; o comprimento do vetor (parâmetro n) é no mínimo igual à quantidade de registros de funcionários no arquivo de entrada; se não for possível abrir o arquivo, a função deve imprimir a mensagem “ERRO” e terminar a execução do programa. A função implementada deve ter o seguinte protótipo:

```
void carrega (int n, Funcionario** vet, char* arquivo);
```

2) Considere um arquivo texto onde cada linha contém a matrícula de um aluno e seu respectivo CR, como ilustra o exemplo a seguir:

```
9010087-2  7.3
8820324-3  8.7
9210478-5  9.2
```

9020256-8 6.7
Implemente uma função que receba como parâmetros o nome de um arquivo no formato descrito acima e a matrícula de um aluno, e retorne o CR do aluno. Se não for encontrado no arquivo a matrícula procurada, a função deve retornar –1.0. Se não for possível abrir o arquivo, a função deve imprimir a mensagem "ERRO" e terminar a execução do programa. O protótipo dessa função deve ser:

float busca (char* arquivo, char* matricula);

3) Considere um arquivo texto com as notas dos alunos de uma disciplina. Cada linha do arquivo contém a matrícula de um aluno (cadeia de nove caracteres), seguida pelos valores de suas três notas (P1, P2 e P3). Considere ainda que não existem linhas em branco no arquivo. Um exemplo desse formato é mostrado abaixo.

9010087-2 2.0 4.3 6.5
8820324-3 7.0 8.2 8.5
9210478-5 6.0 7.5 7.8
9020256-8 3.0 0.5 4.2

Escreva uma função que receba como parâmetros o número de matrícula de um aluno e o nome de um arquivo com as notas de uma disciplina no formato descrito acima, e retorne a média do aluno na disciplina. A média de um aluno é calculada pela fórmula (P1+P2+P3)/3. Caso o número de matrícula passado como parâmetro não seja encontrado no arquivo, a função deve retornar -1.0. Por exemplo, considerando um arquivo com o conteúdo apresentado acima, sua função deve retornar 7.9 caso o número de matrícula passado como parâmetro seja “8820324-3”. O protótipo da função deve ser:

float media (char* mat, char* nome_arquivo);

Obs.: Se não for possível abrir o arquivo de entrada, a função deve imprimir a mensagem “ERRO” e terminar a execução do programa.

4) Faça um programa que leia um nome de “arquivo .c” digitado pelo usuário e gere um novo que remove as linhas com comentários “//”

5) Escreva um programa que leia de um arquivo, cujo nome será fornecido pelo usuário, um conjunto de números reais e armazene em um vetor. O programa ao final calcula a media dos números lidos.

6) Faça um programa que gere 10 arquivos com o nome “testeXX”, onde XX é ”01”, ..., ”10” (exemplo: teste01.txt”). Cada arquivo deve conter dentro dele o texto ”Texto do arquivo [NÚMERO DO ARQUIVO]”.

7) Um arquivo do tipo texto, chamado ‘‘numeros.txt’’ contém uma quantidade desconhecida de números reais. Escreva um programa que leia estes números, os coloque em ordem crescente e depois os grave em um arquivo binário chamado ‘‘numeros.bin’’.

8) Escreva um programa que grave os dados de alunos conforme a estrutura abaixo em um arquivo do tipo binário de acesso aleatório.

```
typedef struct _ALUNO {
   char nome [81];
   float nota1 , nota2;
} ALUNO;
```

a. Faça uma função para mostrar os alunos que tem média maior do que 6,0.
b. Faça um menu de opções para incluir, alterar, consultar e excluir um aluno.