

Lista de Exercícios 09: Vetores de Tipos Estruturados

Questão 1) Considerando as declarações do tipo Aluno abaixo, implemente uma função que tenha como valor de retorno a média final obtida pelos alunos de determinada turma. A nota final de cada aluno é dada pela média das três provas.

```
typedef struct aluno Aluno;
struct aluno {
    char nome[81];
    char matricula[8];
    char turma;
    float p1;
    float p2;
    float p3;
};

float media_turma (int n, Aluno alunos[], char turma);
```

Para o teste, suponha, por exemplo, que a turma tem 05 alunos, e que a informação de cada aluno vai ser fornecida via teclado.

Exemplo de execução:

```
Cadastrando os alunos:
Aluno 1:
    Nome: Monica
    Matricula: 1234
    Turma: A
    Nota da P1: 10.0
    Nota da P2: 8.0
    Nota da P3: 5.0
Aluno 2:
    Nome: Magali
    Matricula: 2345
    Turma: A
    Nota da P1: 7.0
    Nota da P2: 5.0
    Nota da P3: 6.0
Aluno 3:
    Nome: Rosinha
    Matricula: 3456
    Turma: A
    Nota da P1: 7.5
    Nota da P2: 5.3
    Nota da P3: 10.0
Media da turma A: 7.09
```

Questão 2) Considere que os dados dos alunos de uma turma estão sendo fornecidos através do arquivo de entrada chamado alunos.txt. Escreva e estruture um programa para que os dados deste arquivo sejam lidos e para cada aluno deve ser escrito em um único arquivo de saída a seguinte seguinte informação:

“Matricula: XXXXX Media aluno: 99.99”;

A média de cada aluno é dada pela média aritmética das três provas.

Seu arquivo de saída também deve conter a média da turma, da seguinte forma: “Media da turma: YYYY”.

Exemplo de execução:

Para o arquivo de entrada:

Monica 1234 A 10 8 5
Magali 2345 A 7 5 6
Rosinha 3456 A 7.5 5.3 10.0

A saída esperada é:

Matricula: 1234 Media aluno: 7.67
Matricula: 2345 Media aluno: 6.00
Matricula: 3456 Media aluno: 7.60
Media da turma A: 7.09

Questão 3) Bibtex é um software de gerenciamento de referências comumente usado pelo LaTeX. Abaixo, ilustra-se a base BibTex de um artigo

```
@article{turing_paper
  author = {Turing, A. M.},
  title = {Computing Machinery and Intelligence},
  journal = {Mind},
  volume = {LIX},
  number = {236},
  pages = {433-460},
  year = {1950},
}
```

Faça um programa que leia um arquivo no formato BibTex e exiba na tela o título do artigo entre aspas duplas, seguido na linha seguinte pelo autor, o número de páginas e o ano, como ilustrado abaixo.

"Computing Machinery and Intelligence"

Turing, A. M., 28 pages, 1950.

Questão 4) 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 podem existir linhas em branco no arquivo. Um exemplo desse formato é:

9010087 -2	2.0	4.3	6.5
8820324 -3	7.0	8.2	8.6
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, e retorne a média do aluno na disciplina. A média de um aluno é calculada pela fórmula $(P1 + P2 + P3)/3$. O protótipo da função deve ser:

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

Caso o número de matrícula passado como parâmetro não seja encontrado no arquivo, a função deve retornar -1.0. Se não for possível abrir o arquivo de entrada, a função deve imprimir a mensagem "Erro" e terminar a execução do programa.

Questão 5) Considere a existência de um arquivo texto, denominado “turma.txt”, com um cadastro de uma turma. Para cada aluno da turma, existem duas linhas no arquivo: na primeira linha, encontra-se o nome do aluno; na linha seguinte, encontram-se as duas notas obtidas pelo aluno. Considere que podem existir linhas em branco no arquivo. Um exemplo deste arquivo é mostrado a seguir:

```
Fulano Pereira
9.0 8.0
Beltrano Silva
4.0 5.0

Sicrano Santos
3.0 7.0
Fulana Souza
4.0 4.0

Maria Paula
6.0 7.0
```

Escreva um programa completo que leia o conteúdo do arquivo (“turma.txt”) com o formato anterior e crie outro arquivo, com o nome “aprovados.txt”, com a lista dos alunos que obtiveram médias maiores ou iguais a 5.0, seguindo a mesma ordem do arquivo de entrada. Cada linha do arquivo de saída deve conter o nome e a média obtida pelo aluno. Se esse arquivo anterior fosse usado num exemplo, a saída obtida seria:

Fulano Pereira	8.5
Sicrano Santos	5.0
Maria Paula	6.5

Se o arquivo de entrada não puder ser aberto, deve-se imprimir a mensagem “Erro” na tela e abortar o programa. Pode-se considerar que sempre será possível abrir o arquivo de saída.

Questão 6) Escreva um programa que crie um pequeno cadastro de veículos para aluguel em uma locadora de veículos em um vetor de estruturas com os seguinte membros:

- **marca:** string com tamanho máximo de 15 caracteres
- **modelo:** string com tamanho máximo de 20 caracteres
- **valor da diária:** float
- **número de portas:** int

Após a criação do cadastro, o programa deve, em processo repetitivo, ler do teclado o número de portas e o valor máximo da diária do veículo desejado e, em seguida, apresentar na tela uma relação contendo a marca e o modelo de todos os veículos cadastrados e que satisfaçam às condições solicitadas. O processo repetitivo termina quando for informado o valor 0 (zero) para o número de portas.

Observações:

1. O número de veículos a serem cadastrados deve ser lido no início do programa no intervalo de 1 a 20 e enquanto não for digitado um número no intervalo definido o programa não deve prosseguir.
2. O programa deve conter necessariamente:
 - a. Uma função que receba o número de veículos a serem cadastrados e retorne por parâmetro o vetor de estrutura com os veículos cadastrados;

- b. Uma função para imprimir uma relação contendo a marca e modelo de todos os veículos que tenham as características indicadas a ser chamada no programa principal.

Questão 7) Crie um programa em C que permita gerenciar informações de livros. O programa deve:

1. Definir uma estrutura chamada **Livro** com os seguintes campos:
 - **titulo:** string com tamanho máximo de 50 caracteres
 - **autor:** string com tamanho máximo de 50 caracteres
 - **ano:** inteiro
 - **preco:** float
2. Implementar as seguintes funcionalidades:
 - **Cadastrar Livro:** Permite ao usuário cadastrar informações de um novo livro e adicioná-lo a um vetor de structs **Livro**. O vetor deve ter um tamanho máximo predefinido (por exemplo, 100 livros).
 - **Listar Livros:** Exibe na tela as informações de todos os livros cadastrados no vetor.
 - **Salvar Livros em Arquivo:** Salva os dados de todos os livros do vetor em um arquivo texto chamado "livros.txt". Cada livro deve ser armazenado em uma linha do arquivo, com os campos separados por um caractere especial (por exemplo, ';').
 - **Carregar Livros do Arquivo:** Lê os dados do arquivo "livros.txt" e carrega-os para o vetor de structs **Livro**. Esta funcionalidade deve ser chamada no início da execução do programa para carregar dados previamente salvos.
3. O programa deve apresentar um menu simples ao usuário com as opções: Cadastrar, Listar, Salvar, Sair.

Dica: olhe a função `strtok` da biblioteca `string.h`. Usá-la pode lhe ajudar.