

INF1007 - PROGRAMAÇÃO II

LISTA DE EXERCÍCIOS 8

1. O Bar das Freiras deseja automatizar o seu sistema de cadastro de produtos e você é o encarregado de desenvolver alguns algoritmos para esse sistema. O registro de cada produto no sistema que está sendo desenvolvido possui as seguintes informações:

- Código de Barras – número identificando o código de barras do produto;
- Descrição – cadeia de caracteres descrevendo o nome do produto (máximo 40 caracteres);
- Preço – valor numérico representando o preço do produto;
- Validade – data de validade do produto (contendo mês e ano);
- Estoque - valor numérico representando a quantidade em estoque do produto;

Essas informações podem ser representadas em dois tipos estruturados: `Produto` e `Data`.

Data
Mês
Ano

Produto
Código de Barras
Descrição
Preço
Validade
Estoque

Escreva um programa que defina os novos tipos estruturados `Produto` e `Data`, como especificados acima. E em seguida, implemente as seguintes funções:

Atenção: todas as funções devem utilizar o algoritmo de ordenação rápida (Quick Sort).

- `ordena_codigo` – a função recebe como parâmetro um vetor de estruturas do tipo `Produto` e o tamanho deste vetor. A função deve implementar o algoritmo Quick Sort para ordenar o vetor em ordem crescente pelo código de barras dos produtos.
- `ordena_descricao` – a função recebe como parâmetro um vetor de estruturas do tipo `Produto` e o tamanho deste vetor. A função deve implementar o algoritmo Quick Sort para ordenar o vetor em ordem alfabética pela descrição dos produtos.
- `ordena_validade` – a função recebe como parâmetro um vetor de estruturas do tipo `Produto` e o tamanho deste vetor. A função deve implementar o algoritmo Quick Sort para ordenar o vetor em ordem cronológica pelo data de validade dos produtos.

Em seguida, implemente a função principal do programa para ordenar o vetor abaixo pelo código de barras, descrição e data de validade dos produtos. Para cada ordenação, o seu programa deve exibir o vetor ordenado no critério especificado.

```
Produto produtos[] = {{4269813, "Salgado", 3.50, {4, 2014}, 20},
                      {9561234, "Cafe", 3.50, {4, 2014}, 50},
                      {5621978, "Pacoca", 0.50, {10, 2014}, 50},
                      {3464842, "Suco", 4.0, {4, 2014}, 20},
                      {8361512, "Toddyinho", 2.50, {9, 2014}, 50},
                      {6521874, "Bolo", 3.50, {4, 2014}, 10},
                      {7516842, "Batata Frita", 4.00, {4, 2014}, 20},
                      {2348651, "Coca-Cola", 3.0, {8, 2015}, 200},
                      {1215462, "Biscoito", 3.0, {4, 2015}, 50}};
```

2. A biblioteca da PUC-Rio está atualizando o seu sistema de gerenciamento de livros e você é o encarregado de implementar algumas funções para o novo sistema.

O sistema utiliza duas estruturas para armazenar as informações sobre livros e alunos:

Livro	Aluno
Nome	Matricula
Autor	Nome
Disponível	Livros Emprestados

A estrutura Livro é composta por:

- Nome – cadeia de caracteres com o nome do livro (máximo 60 caracteres);
- Autor – cadeia de caracteres com o nome do autor (máximo 60 caracteres);
- Disponível – número (0 ou 1) indicando se o livro está disponível (1) ou não (0);

A estrutura Aluno é composta por:

- Matricula – número inteiro representado a matricula do aluno;
- Nome – cadeia de caracteres com o nome do aluno (máximo 60 caracteres);
- Livros Emprestados – número de livros emprestados pelo aluno;

Escreva um programa que defina os novos tipos estruturados `Livro` e `Aluno`, como especificados acima. E em seguida, implemente as seguintes funções:

Atenção: obrigatoriamente você deve utilizar a função `qsort` da biblioteca `stdlib.h` para implementar o Quick Sort nas suas funções.

- `ordena_livros` – a função recebe como parâmetro um vetor de ponteiros de estruturas do tipo `Livro` e o tamanho do vetor. A função deve ordenar o vetor em ordem alfabética pelo nome do autor do livro e usando como critério de desempate o título do livro.

- `ordena_emprestados` – a função recebe como parâmetro um vetor de ponteiros de estruturas do tipo `Aluno` e o tamanho do vetor. A função deve ordenar o vetor em ordem decrescente pelo número de livros emprestados pelos alunos e usando como critério de desempate o nome do aluno em ordem alfabética.

Em seguida, implemente a função principal do programa para ordenar os vetores abaixo utilizando as funções que você criou. Para cada ordenação, o seu programa deve exibir o vetor ordenado no critério especificado.

```
Livro *livros[7];
Aluno *alunos[7];

livros[0] = (Livro*)malloc(sizeof(Livro));
strcpy(livros[0]->nome, "Introducao a Estruturas de Dados");
strcpy(livros[0]->autor, "Waldemar Celes");
livros[0]->disponivel = 0;

livros[1] = (Livro*)malloc(sizeof(Livro));
strcpy(livros[1]->nome, "Machine Learning");
strcpy(livros[1]->autor, "Mitchell");
livros[1]->disponivel = 0;

livros[2] = (Livro*)malloc(sizeof(Livro));
strcpy(livros[2]->nome, "Artificial Intelligence for Games");
strcpy(livros[2]->autor, "Millington");
livros[2]->disponivel = 1;

livros[3] = (Livro*)malloc(sizeof(Livro));
strcpy(livros[3]->nome, "Prolog Programming for Artificial Intelligence");
strcpy(livros[3]->autor, "Bratko");
livros[3]->disponivel = 1;

livros[4] = (Livro*)malloc(sizeof(Livro));
strcpy(livros[4]->nome, "C - A Linguagem de Programacao");
strcpy(livros[4]->autor, "Kernighan");
livros[4]->disponivel = 1;

livros[5] = (Livro*)malloc(sizeof(Livro));
strcpy(livros[5]->nome, "Artificial Intelligence: A Modern Approach");
strcpy(livros[5]->autor, "Russell and Norvig");
livros[5]->disponivel = 1;

livros[6] = (Livro*)malloc(sizeof(Livro));
strcpy(livros[6]->nome, "Game Development: Using Unity and C#");
strcpy(livros[6]->autor, "Millington");
livros[6]->disponivel = 0;

alunos[0] = (Aluno*)malloc(sizeof(Aluno));
alunos[0]->matricula = 35641215;
strcpy(alunos[0]->nome, "Ana");
alunos[0]->emprestados = 1;

alunos[1] = (Aluno*)malloc(sizeof(Aluno));
alunos[1]->matricula = 2654951;
strcpy(alunos[1]->nome, "Pedro");
alunos[1]->emprestados = 0;
```

```
alunos[2] = (Aluno*)malloc(sizeof(Aluno));
alunos[2]->matricula = 51364125;
strcpy(alunos[2]->nome, "Maria");
alunos[2]->emprestados = 3;

alunos[3] = (Aluno*)malloc(sizeof(Aluno));
alunos[3]->matricula = 45612681;
strcpy(alunos[3]->nome, "Joao");
alunos[3]->emprestados = 1;

alunos[4] = (Aluno*)malloc(sizeof(Aluno));
alunos[4]->matricula = 62151578;
strcpy(alunos[4]->nome, "Bruno");
alunos[4]->emprestados = 2;

alunos[5] = (Aluno*)malloc(sizeof(Aluno));
alunos[5]->matricula = 82135123;
strcpy(alunos[5]->nome, "Julia");
alunos[5]->emprestados = 3;

alunos[6] = (Aluno*)malloc(sizeof(Aluno));
alunos[6]->matricula = 2685415;
strcpy(alunos[6]->nome, "Silvio");
alunos[6]->emprestados = 1;
```