

INF1007 - PROGRAMAÇÃO II

LISTA DE EXERCÍCIOS 7

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 busca binária e obrigatoriamente você deve implementar o processo de comparação em uma função auxiliar.

- `busca_codigo` – a função recebe como parâmetro um vetor de estruturas do tipo `Produto` ordenado pelo código de barras, o tamanho do vetor e também um código de barras a ser buscado. A função deve implementar um algoritmo de busca binária e retornar o índice onde o código de barras buscado está armazenado. Se o código de barras não for encontrado no vetor, a função deve retornar -1.
- `busca_descricao` – a função recebe como parâmetro um vetor de estruturas do tipo `Produto` ordenado alfabeticamente pela descrição, o tamanho do vetor e também a descrição de um produto a ser buscado. A função deve implementar um algoritmo de busca binária e retornar o índice onde o produto buscado está armazenado. Se a descrição não for encontrada, a função deve retornar -1.

- `busca_validade` – a função recebe como parâmetro um vetor de estruturas do tipo `Produto` ordenado pela data de validade dos produtos, o tamanho do vetor e também uma data a ser buscada. A função deve implementar um algoritmo de busca binária e retornar o índice do primeiro produto com a data de validade buscada. Se nenhum produto for encontrado, a função deve retornar -1.

Em seguida, implemente a função principal do programa que permita ao usuário fazer uma busca pelo código de barras, seguido de uma busca pela descrição e outra por uma data de validade. Em ambas as buscas, o programa deve exibir todas as informações sobre o produto caso ele seja encontrado. Caso o produto não seja encontrado, o programa deve exibir uma mensagem informando que nenhum produto foi encontrado.

Para testar o seu programa utilize os seguintes vetores:

```

Produto produtos1[] = {{1215462, "Biscoito", 3.0, {4, 2015}, 50},
                      {2348651, "Coca-Cola", 3.0, {8, 2015}, 200},
                      {3464842, "Suco", 4.0, {4, 2014}, 20},
                      {4269813, "Salgado", 3.50, {4, 2014}, 20},
                      {5621978, "Pacoca", 0.50, {10, 2014}, 50},
                      {6521874, "Bolo", 3.50, {4, 2014}, 10},
                      {7516842, "Batata Frita", 4.00, {4, 2014}, 20},
                      {8361512, "Toddynho", 2.50, {9, 2014}, 50},
                      {9561234, "Cafe", 3.50, {4, 2014}, 50}};

Produto produtos2[] = {{7516842, "Batata Frita", 4.00, {4, 2014}, 20},
                      {1215462, "Biscoito", 3.0, {4, 2015}, 50},
                      {6521874, "Bolo", 3.50, {4, 2014}, 10},
                      {9561234, "Cafe", 3.50, {4, 2014}, 50},
                      {2348651, "Coca-Cola", 3.0, {8, 2015}, 200},
                      {5621978, "Pacoca", 0.50, {10, 2014}, 50},
                      {3464842, "Suco", 4.0, {4, 2014}, 20},
                      {4269813, "Salgado", 3.50, {4, 2014}, 20},
                      {8361512, "Toddynho", 2.50, {9, 2014}, 50}};

Produto produtos3[] = {{7516842, "Batata Frita", 4.00, {4, 2014}, 20},
                      {6521874, "Bolo", 3.50, {4, 2014}, 10},
                      {9561234, "Cafe", 3.50, {4, 2014}, 50},
                      {3464842, "Suco", 4.0, {4, 2014}, 20},
                      {4269813, "Salgado", 3.50, {4, 2014}, 20},
                      {8361512, "Toddynho", 2.50, {9, 2014}, 50},
                      {5621978, "Pacoca", 0.50, {10, 2014}, 50},
                      {1215462, "Biscoito", 3.0, {4, 2015}, 50},
                      {2348651, "Coca-Cola", 3.0, {8, 2015}, 200}};

```

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 `bsearch` da biblioteca `stdlib.h` para implementar a busca binária nas suas funções.

- `livroDisponivel` – a função recebe como parâmetro um vetor de ponteiros de estruturas do tipo `Livro` ordenado pelo nome do livro, o tamanho do vetor e também um nome de livro a ser buscado. A função deve buscar o livro e retornar 1 caso ele esteja disponível para empréstimo ou 0 caso não esteja disponível. Se o livro não for encontrado, a função deve retornar -1.
- `estadoAluno` – a função recebe como parâmetro um vetor de ponteiros de estruturas do tipo `Aluno` ordenado pela matricula do aluno, o tamanho do vetor e também uma matricula a ser buscada. A função deve buscar pelo aluno e retornar 1 caso o aluno tenha 3 ou mais livros emprestados ou 0 caso o aluno tenha menos de 3 livros emprestados. Se o aluno não for encontrado, a função deve retornar -1.

Em seguida, implemente a função principal do programa que permita ao usuário verificar se um determinado livro está disponível para empréstimo e em seguida verificar se o aluno que deseja emprestar o livro pode empresta-lo (tem menos de 3 livros emprestados). Em ambos os casos, se o livro/aluno não for encontrado, o programa deve exibir uma mensagem informando que nenhum livro/aluno foi encontrado. Para testar o seu programa utilize os seguintes vetores de ponteiros:

```
Livro *livros[6];
Aluno *alunos[6];

livros[0] = (Livro*)malloc(sizeof(Livro));
strcpy(livros[0]->nome, "Artificial Intelligence: A Modern Approach");
strcpy(livros[0]->autor, "Russell and Norvig");
livros[0]->disponivel = 1;

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

livros[2] = (Livro*)malloc(sizeof(Livro));
strcpy(livros[2]->nome, "C - A Linguagem de Programacao");
strcpy(livros[2]->autor, "Kernighan");
livros[2]->disponivel = 1;

livros[3] = (Livro*)malloc(sizeof(Livro));
strcpy(livros[3]->nome, "Introducao a Estruturas de Dados");
strcpy(livros[3]->autor, "Waldemar Celes");
livros[3]->disponivel = 0;

livros[4] = (Livro*)malloc(sizeof(Livro));
strcpy(livros[4]->nome, "Machine Learning");
strcpy(livros[4]->autor, "Mitchell");
livros[4]->disponivel = 0;

livros[5] = (Livro*)malloc(sizeof(Livro));
strcpy(livros[5]->nome, "Prolog Programming for Artificial Intelligence");
strcpy(livros[5]->autor, "Bratko");
livros[5]->disponivel = 1;

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

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

alunos[2] = (Aluno*)malloc(sizeof(Aluno));
alunos[2]->matricula = 45612681;
strcpy(alunos[2]->nome, "Maria");
alunos[2]->emprestados = 3;

alunos[3] = (Aluno*)malloc(sizeof(Aluno));
alunos[3]->matricula = 51364125;
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;
```