

MC202GH - ESTRUTURAS DE DADOS

2º Semestre de 2017

Professor: Rafael C. S. Schouery

Monitores: Yulle Glebbyo Felipe Borges (PED)

Victor Andrietta Razoli Silva (PAD)

Lab. 02 - Controle de Distribuição de Livros

Peso 1

1. O Problema

Uma empresa distribuidora de livros é responsável por fornecer material para diversas universidades públicas no país. Esta empresa possui um grande catálogo de livros e universidades clientes, e por isso o controle da distribuição foi comprometido conforme novas universidades foram acrescentadas ao sistema. A Unicamp, como uma das empresas parceiras da distribuidora, se ofereceu para ajudar na implementação de um sistema de controle do seu fluxo que seja sensível a adição de novas universidades e livros. Você faz parte de um projeto inovador da Unicamp, que visa inserir os alunos em atividades voltadas à aplicações reais de estruturas de dados, e foi escolhido para implementar este sistema.

Para modelar o sistema, utilizaremos uma matriz dinâmica. Para cada livro cadastrado, teremos uma linha da matriz dinâmica. Para cada universidade cadastrada, teremos uma coluna da matriz dinâmica. O valor em cada célula $A_{i,j}$ da matriz, refere-se a quantidade de livros do tipo i demandados pela universidade j . Além disso, deve-se manter um vetor dinâmico para armazenar os nomes dos livros, e outro vetor dinâmico para armazenar os nomes das universidades. É importante manter sincronizados os índices entre os vetores dinâmicos e as colunas e linhas da matriz dinâmica, ou então gerenciar essa informação de alguma forma no seu programa.

Livros e universidades devem ser referidos pelos seus nomes no cadastro. O sistema deve suportar a operação de adição de livros, assim como a operação de adição de universidades. Além disso, deve ser possível cadastrar um pedido por uma universidade para múltiplos livros, assim como obter relatórios sobre as demandas dos livros pelas universidades.

Os tamanhos alocados em memória devem estar sempre potência de 2, **começando com 1** e aumentando conforme novos registros são inseridos no sistema. Por exemplo, se tivermos 10 livros cadastrados e 4 clientes, teríamos uma matriz 16×4 alocada, caso um novo cliente seja inserido neste momento, as 16 linhas referentes aos livros devem ser realocadas para terem tamanho no máximo 8 (8 é a próxima potência de 2 após 4), fazendo com que a matriz efetivamente tenha dimensão máxima 16×8 .

As mesmas restrições de alocação de memória devem ser aplicados aos vetores dinâmicos que armazenam os nomes das universidades e livros, para que seus índices permaneçam consistentes com os índices da matriz.

2. Entrada

Ambos os livros e universidades, devem ser cadastrados por seu nome **sem a utilização de espaços** neles, além disso seus nomes terão no máximo 41 caracteres (incluindo o '\0'). Iremos utilizar os nomes como identificadores para fazer todas as operações de consulta. As operações a serem implementadas e suas respectivas entradas são descritas abaixo:

ID	Nome	Descrição
1	Inserção de um livro	Dado um vetor de caracteres representando o nome do livro sem espaços, cadastra o livro, expandindo a memória atual caso necessário.
2	Inserção de uma universidade	Dado um vetor de caracteres representando o nome da universidade sem espaços, cadastra a universidade, expandindo a memória atual caso necessário.
3	Execução de um pedido	Recebe um vetor de caracteres representando o nome de uma universidade já cadastrada e um inteiro q , em seguida serão fornecidas q linhas, cada uma com o nome de um livro já cadastrado e um número de exemplares demandados n . Caso já exista um pedido anterior para uma dada requisição, este pedido deve ser incrementado com a quantidade informada. Caso não exista um pedido anterior, registra o pedido.
4	Relatório de exemplares por universidade	Dado um vetor de caracteres representando o nome de uma universidade já cadastrada, imprime a quantidade total de livros demandados por esta universidade.
5	Relatório de universidades por livro	Dado um vetor de caracteres representando o nome de um livro já cadastrado, imprime o número total de cópias do livro solicitadas pelas universidades.
6	Relatório total de exemplares	Imprime o número total de exemplares solicitados pelas universidades.
0	Finalização	Saia do sistema, liberando corretamente toda a memória utilizada

Exemplo de entrada:

```
1 IntroductionToAlgorithms
1 AlgorithmsInC
1 TheArtOfComputerProgramming
2 Unicamp
2 USP
3
Unicamp 2
IntroductionToAlgorithms 50
TheArtOfComputerProgramming 10
3
USP 1
IntroductionToAlgorithms 75
4
Unicamp
6
0
```

3. Saída

Abaixo, temos as saídas esperadas correspondentes a cada uma das operações:

ID	Saída	Comentário
1	Livro cadastrado com sucesso!	-
2	Universidade cadastrada com sucesso!	-
3	Pedido registrado!	-
4	X	X é um inteiro que representa a soma dos números de exemplares solicitados pela universidade dada como entrada
5	Y	Y é um inteiro que representa a quantidade de cópias do livro dado como entrada solicitadas pelas universidades
6	Z	Z é um inteiro que representa a soma da quantidade de exemplares de todos os livros

		solicitados por todas as universidades cadastradas no sistema
0	Sistema encerrado.	Antes de encerrar o sistema, é necessário liberar toda a memória alocada

Exemplo de saída:

```
Livro cadastrado com sucesso!
Livro cadastrado com sucesso!
Livro cadastrado com sucesso!
Universidade cadastrada com sucesso!
Universidade cadastrada com sucesso!
Pedido registrado!
Pedido registrado!
60
135
Sistema Encerrado.
```

4. Informações

- Este laboratório possui **peso 1**.
- **Não** há um número máximo de submissões.
- No início de cada arquivo a ser submetido, insira um comentário com seu nome, RA e uma breve descrição do conteúdo do arquivo.
- Apenas comentários no formato `/* comentário */` serão aceitos. Comentários com `//` serão acusados como erros pelo SuSy.
- **Não é permitido** o uso da função `realloc` neste laboratório.
- Os cabeçalhos permitidos são:
 - `stdio.h`
 - `stdlib.h`
 - `string.h`
 - `math.h`
- A submissão da sua solução deverá conter múltiplos arquivos:
 - **matriz.h**: interface da matriz dinâmica
 - **matriz.c**: implementação da interface da matriz dinâmica
 - **vetor.h**: interface do vetor dinâmico

- **vetor.c**: implementação da interface do vetor dinâmico
- **lab02.c**: cliente que utiliza as estruturas
- Será disponibilizado um **Makefile** para este trabalho na página do laboratório:
 - Para compilar seu projeto, basta navegar até o diretório do projeto e utilizar o comando 'make' no terminal do Linux.
 - Veja mais instruções na página da disciplina.

5. Critérios de Avaliação

- Seu código deverá passar por todos os casos de teste do SuSy definidos para este laboratório. Caso positivo, utilizaremos os seguintes critérios adicionais:
 - Indentação correta de todo o código. Escolha entre espaços ou tabs para indentar seu código e seja coerente em todos os arquivos. Indentações utilizando espaços e tabs intercalados, ou trechos não indentados corretamente irão resultar em uma penalização de **um ponto** na nota
 - O aumento da memória alocada para a matriz e vetor dinâmicos **deve** ocorrer de acordo com a descrição do enunciado
 - Utilização de bibliotecas ou funções não permitidas pelo enunciado resultará em uma **penalização severa** na nota
 - Toda a memória alocada deve ser liberada adequadamente ao fim do programa