

Lista de exercícios com vetores e listas encadeadas

Essa lista contém um conjunto de 6 exercícios que exploram vetores e listas encadeadas para interações (operações de inclusão/exclusão) numa base de registros com nomes (*char* de 30 posições) e matrículas (de tipo inteiro). Para todos os exercícios, considerar que o usuário fará, no máximo, 20 interações nos elementos da base de dados (mantida em memória).

As Tabelas 1 a 3 apresentam exemplos de entradas e saídas que os programas devem manipular. Abaixo estão listados os códigos de entrada e os respectivos significados de cada um deles (vide tabelas para compreender o uso desses códigos):

- *IU nome matrícula* – refere-se à inclusão do registro composto pelo nome e matrícula de um aluno. Após a inserção, o programa deve exibir a base de dados na ordem informada pelo usuário.
- *IM nome matrícula* – refere-se à inclusão do registro composto pelo nome e matrícula de um aluno. Nesse caso, a inserção deve ser feita considerando a ordem de matrícula do aluno. Assumir que não existam alunos com matrículas iguais (o programa não precisa testar essa condição, mas apenas considerar que as repetições não ocorrerão). Após a inserção, o programa deve exibir os registros em ordem crescente de matrícula.
- *EX nome matrícula* – refere-se à exclusão do registro composto pelo nome e matrícula do aluno. Caso esse registro não exista, o programa deve imprimir ENTRADA INVALIDA. Caso exista, o programa deve imprimir os registros da base de dados na ordem em que estava até o momento da referida exclusão, ou a mensagem BASE VAZIA, se não houver registros na base depois dessa exclusão.
- *IN nome matrícula* – refere-se à inclusão do registro composto pelo nome e pela matrícula de um aluno. A inserção deve ser feita por ordem de nome, considerando apenas a primeira letra do nome para ordenação. Após a inserção, o programa deve exibir os registros em ordem crescente de nome. Observar que, para uma mesma letra, a impressão deve obedecer a ordem de inclusão; ou seja, se Abadia for inserida depois de Ana, a impressão deve mostrar primeiro o nome de Ana e depois o nome de Abadia.

Em cada programa (escrito em linguagem C/Linux, compilador gcc), considerar que a saída deve corresponder à configuração da base de dados após a aplicação do comando demandado pelo usuário e devem terminar com uma quebra de linha.

Observações gerais

- Procure gerar programas bem escritos (código identado, variáveis significativas, comentários, etc.), sem erros de compilação, preferencialmente considerando as diretivas assumidas no ejudge (não pode haver *warnings*, declarar variáveis nos locais adequados, evitar declarar variáveis que não serão usadas, usar constantes para melhorar legibilidade, etc.)
- Embora essas questões não necessitem ser submetidas ao ejudge, sugere-se testá-las ao máximo, para garantir que os programas consigam cumprir o que foi especificado.

Tabela 1 – Inserções na ordem de entrada do usuário

Entradas	Saídas
IU Carlos 15	Carlos,15
EX Carlos 15	BASE VAZIA
IU Carlos Alberto 22	Carlos Alberto, 22
IU Jose 12	Carlos Alberto, 22; Jose, 12
EX Maria Jose 62	ENTRADA INVALIDA
IU Manoel 44	Carlos Alberto, 22; Jose, 12; Manoel, 44
IU Diego 10	Carlos Alberto, 22; Jose, 12; Manoel, 44; Diego, 10
IU Daniel 51	Carlos Alberto, 22; Jose, 12; Manoel, 44; Diego, 10; Daniel, 51
EX Jose 12	Carlos Alberto, 22; Manoel, 44; Diego, 10; Daniel, 51
IU Amelia Alves 30	Carlos Alberto, 22; Manoel, 44; Diego, 10; Daniel, 51; Amelia Alves, 30
EX Diego 10	Carlos Alberto, 22; Manoel, 44; Daniel, 51; Amelia Alves, 30
IU Felipe de Sa 28	Carlos Alberto, 22; Manoel, 44; Daniel, 51; Amelia Alves, 30; Felipe de Sa, 28

1. Implementar um programa que consiga simular as entradas e saídas exemplificadas na Tabela 1, considerando o uso de um vetor.
2. Implementar um programa que consiga simular as entradas e saídas exemplificadas na Tabela 1, considerando o uso de uma lista encadeada.

Tabela 2 – Inserções em ordem de matrícula

Entradas	Saídas
IM Carlos 15	Carlos,15
EX Carlos 15	BASE VAZIA
IM Carlos Alberto 22	Carlos Alberto, 22
IM Jose 12	Jose, 12; Carlos Alberto, 22
EX Maria Jose 62	ENTRADA INVALIDA
IM Manoel 44	Jose, 12; Carlos Alberto, 22; Manoel, 44
IM Diego 10	Diego, 10; Jose, 12; Carlos Alberto, 22; Manoel, 44
IM Daniel 51	Diego, 10; Jose, 12; Carlos Alberto, 22; Manoel, 44; Daniel, 51
EX Jose 12	Diego, 10; Carlos Alberto, 22; Manoel, 44; Daniel, 51
IM Amelia Alves 30	Diego, 10; Carlos Alberto, 22; Amelia Alves, 30; Manoel, 44; Daniel, 51
EX Diego 10	Carlos Alberto, 22; Amelia Alves, 30; Manoel, 44; Daniel, 51
IM Felipe de Sa 28	Carlos Alberto, 22; Felipe de Sa, 28; Amelia Alves, 30; Manoel, 44; Daniel, 51

3. Implementar um programa que consiga simular as entradas e saídas exemplificadas na Tabela 2, considerando o uso de um vetor.
4. Implementar um programa que consiga simular as entradas e saídas exemplificadas na Tabela 2, considerando o uso de uma lista encadeada.

Tabela 3 – Inserções em ordem de nome

Entradas	Saídas
IN Carlos 15	Carlos,15
EX Carlos 15	BASE VAZIA
IN Carlos Alberto 22	Carlos Alberto, 22
IN Jose 12	Carlos Alberto, 22; Jose, 12
EX Maria Jose 62	ENTRADA INVALIDA
IN Manoel 44	Carlos Alberto, 22; Jose, 12; Manoel, 44
IN Diego 10	Carlos Alberto, 22; Diego, 10; Jose, 12; Manoel, 44
IN Daniel 51	Carlos Alberto, 22; Diego, 10; Daniel, 51; Jose, 12; Manoel, 44
EX Jose 12	Carlos Alberto, 22; Diego, 10; Daniel, 51; Manoel, 44
IN Amelia Alves 30	Amelia Alves, 30; Carlos Alberto, 22; Diego, 10; Daniel, 51; Manoel, 44
EX Diego 10	Amelia Alves, 30; Carlos Alberto, 22; Daniel, 51; Manoel, 44
IN Felipe de Sa 28	Amelia Alves, 30; Carlos Alberto, 22; Daniel, 51; Felipe de Sa, 28; Manoel, 44

5. Implementar um programa que consiga simular as entradas e saídas exemplificadas na Tabela 3, considerando o uso de um vetor.
6. Implementar um programa que consiga simular as entradas e saídas exemplificadas na Tabela 3, considerando o uso de uma lista encadeada.

Bons estudos!