

Exercício

O objetivo do exercício é gerenciar um sistema de histórico escolar “limpo” de uma dada instituição. O sistema armazena as seguintes informações: ID do aluno, sigla da disciplina, nome do aluno, nome da disciplina, média e frequência. A chave primária é composta pela composição “ID do aluno+Sigla da disciplina”. O arquivo a ser criado deve ser de registros e campos de tamanho variável, com um inteiro (4 bytes) no início do registro indicando o tamanho do registro, e com campos separados pelo caractere ‘#’.

ID do aluno	Sigla da disciplina	Nome do aluno	Nome da disciplina	Média	Frequência
3 caracteres (fixo)	3 caracteres (fixo)	50 caracteres (máximo)	50 caracteres (máximo)	float (a ser bufferizado)	float (a ser bufferizado)

Ex.: 53001#ED2#Paulo da Silva#Estruturas de Dados 2#7.3#75.4

O programa conterá as seguintes opções:

- Inserir
- Listar os dados de todos os alunos cadastrados em suas respectivas disciplinas
- Listar os dados de um aluno específico em uma dada disciplina

Inserção (a)

A cada novo cadastro o “ID do aluno+Sigla da disciplina” deverá ser adicionado ao índice primário estruturado como uma árvore-B. Portanto, a cada nova inserção as seguintes mensagens deverão ser mostradas (note que mais de uma pode aparecer):

- “Divisão de nó” deve ser impressa sempre que um nó for dividido;
- “Chave X promovida” deve ser impressa sempre que uma chave for promovida. X é o valor da chave promovida;
- “Chave X inserida com sucesso” deve ser impressa ao final da inserção indicando sucesso da operação;
- “Chave X duplicada” deve ser impressa ao final da inserção e indica que a operação de inserção não foi realizada.

Observação: antes de inserir um registro no arquivo principal certifique-se de que a chave não existe no índice.

Exemplo de Inserção

C
Chave C inserida com sucesso
S
Chave S inserida com sucesso
D
Chave D inserida com sucesso
T
Divisão de nó
Chave S promovida
Chave T inserida com sucesso
S
Chave S duplicada

Listar os dados de todos os alunos cadastrados em suas respectivas disciplinas (b)

Nessa opção o índice árvore-B deverá ser percorrido em-ordem e a cada “ID do aluno+Sigla da disciplina” encontrado listar os dados associados ao mesmo. Desse modo, essa opção deverá imprimir os dados de todos os alunos cadastrados em suas respectivas disciplinas.

Listar os dados de um aluno específico em uma dada disciplina (c)

Dado um “ID do aluno+Sigla da disciplina” o programa retorna os dados do respectivo aluno em uma dada disciplina. Para tanto, a busca deve ser feita na árvore-B. Além disso, as seguintes mensagens deverão ser exibidas em relação à busca na árvore:

- “Chave X encontrada, página Y, posição Z” indica que a Chave X foi encontrada e encontra-se na página Y, na posição Z da página. Após a exibição dessa mensagem, os dados referentes ao aluno deverão ser recuperados do arquivo principal;
- “Chave X não encontrada” indica que a Chave X não está presente na árvore-B e, conseqüente, no arquivo principal.

Exemplo Pesquisa

C

Chave C encontrada, página 0, posição 0

Z

Chave Z não encontrada

OBSERVAÇÃO

A fim de facilitar os testes, serão fornecidos dois arquivos: (a) “insere.bin” e (b) “busca.bin”. O primeiro (a) conterá os dados a serem inseridos durante os testes (não necessariamente todos os dados serão inseridos). Para tanto, uma sugestão é carregar o arquivo em memória (um vetor de struct) e ir acessando cada posição conforme as inserções vão ocorrendo. Note que é possível encerrar a execução e recomeçar a execução, sendo necessário marcar, de algum modo, quantos registros já foram utilizados do mesmo.

Em relação a (b), o arquivo conterá uma lista de “ID do aluno+Sigla da disciplina” a serem utilizados durante a pesquisa (opção (c)). A ideia é a mesma já descrita, ou seja, carregar o arquivo em memória (um vetor de struct) e ir acessando cada posição conforme as buscas vão ocorrendo. Note que é possível encerrar a execução e recomeçar a execução, sendo necessário marcar, de algum modo, quantos registros já foram utilizados do mesmo.

Observações

TODOS OS ARQUIVOS DEVERÃO SER MANIPULADOS EM MEMÓRIA SECUNDÁRIA.

Não criar os arquivos toda vez que o programa for aberto (fazer verificação). O seu programa deve realizar as operações sobre uma árvore-B de ordem 4 (ou seja, no máximo 3 chaves). Para padronizar, sempre promovam, quando houver overflow, a chave de índice 2, começando em zero.

a	b	c
---	---	---

Inserção de d causa overflow

a	b	c	d
---	---	---	---

Promova c (índice 2)

Para auxiliar o desenvolvimento do trabalho é fornecido um código que insere chaves em uma árvore-B de ordem 5. Vocês devem utilizar esse código como base. Entretanto, algumas alterações serão necessárias para que o mesmo funcione corretamente. Vocês deverão estudar e entender o código para que consigam fazer as alterações necessárias. No caso do procedimento de pesquisa básica, tome como base o pseudocódigo discutido em sala de aula. Em relação ao procedimento de percurso em ordem, o mesmo deverá ser desenvolvido por vocês.