

Projeto 2

Os critérios de correção deste projeto são:

- (100%) O programa funciona corretamente;
- Só serão corrigidos os trabalhos que apresentarem documentação no código: variáveis, estruturas de dados, sub-rotinas, lógica, etc.

O trabalho deverá ser desenvolvido em dupla. Quaisquer programas similares terão nota zero independente de qual for o original e qual for a cópia. Trabalhos atrasados terão a nota descontada em 1 ponto por dia de atraso. Implementar no Dev-C++ ou compilador compatível.

Em uma clínica veterinária especializada em cães é necessário realizar o controle dos animais vacinados. Para tanto, os proprietários contarão com sua ajuda para que, por meio de um sistema informatizado, essa atividade seja facilitada. As informações que eles precisam armazenar são apresentadas a seguir. O arquivo a ser criado deve ser de registros fixos com campos fixos (fixo-fixo).

Arquivo Principal 1 (AP1) / fixo-fixo

Código de Controle*	Código do Cachorro	Nome da Vacina	Data da Vacinação	Responsável pela Aplicação
int	int	30 caracteres	5 caracteres [Formato: mm/ano]	30 caracteres

*Código de Controle = diferente do projeto 1, a chave deverá ser informada pelo usuário. **NÃO** usem autoincremento!!!

Arquivo Principal 2 (AP2) / fixo-fixo

Código do Cachorro*	Raça	Nome do Cachorro
int	30 caracteres	30 caracteres

Se quiserem, esse código pode ser autoincremento. Sempre que os dados de uma vacinação forem exibidos, esse arquivo deverá ser percorrido para que os dados referentes a cada cachorro sejam exibidos também. Não é necessário um índice sobre esse arquivo. O mesmo pode ser percorrido por busca sequencial ou binária. Se usarem autoincremento, o acesso poderá ser direto!!!

O programa conterá as seguintes opções:

- Inserir
- Listar os dados de todas as vacinações realizadas
- Listar os dados de uma vacinação específica
 - Árvore-B
 - Hash

Inserção (a)

A cada nova vacinação cadastrada no arquivo AP1 o “Código de Controle” da mesma deverá ser adicionado aos índices primários existentes: um estruturado como uma árvore-B e o outro como uma hash. Assim como no Projeto 1, lembre-se que para realizar o cadastro em AP1, é necessário que os dados do cachorro já estejam cadastrados em AP2.

Em relação ao índice árvore-B, a cada nova inserção as seguintes mensagens deverão ser mostradas:

- “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

Em relação ao índice hash, a cada nova inserção as seguintes mensagens deverão ser mostradas:

- “Endereço X”, endereço X gerado para a chave fornecida;
- “Chave X inserida com sucesso” deve ser impressa ao final da inserção indicando sucesso da operação;
- “Colisão”, sempre que um *home address* não estiver livre, gerando uma colisão;
- “Tentativa X”, X é o número da tentativa para tratar a colisão.

Exemplo de Inserção

1234567891234

Endereço 5

Chave 1234567891234 inserida com sucesso

4321987654321

Endereço 5

Colisão

Tentativa 1

Chave 4321987654321 inserida com sucesso

Observações:

- Para tratar as colisões utilize Overflow Progressivo;
- Considere uma hash de 11 posições;
- Utilize o Método da Divisão Inteira para encontrar o endereçamento de uma dada chave (função hash);
- Cada endereçamento contém duas chaves (está sendo utilizado o conceito de Bucket);
- Lembre-se que o arquivo hash é um arquivo de registros fixos que contém, no mínimo, duas informações: Chave + RRN. Campos adicionais podem ser acrescentados se necessário.

Listar os dados de todas as vacinações (b)

Nessa opção o índice árvore-B deverá ser percorrido em-ordem e a cada “Código de Controle” encontrado listar os dados associados ao mesmo. Desse modo, essa opção deverá imprimir os dados de todas as vacinações cadastradas por ordem de “Código de Controle”. Não esqueça de substituir o “Código do Cachorro” pelos dados referente ao mesmo.

Listar os dados de uma vacinação específica (c)

i. Árvore-B

Dado um “Código de Controle” o programa retorna os dados da respectiva vacinação. 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 encontrase na página Y, na posição Z da página. Após a exibição dessa mensagem, os dados

referentes à vacinação deverão ser recuperados do arquivo principal (não esqueça de substituir o “Código do Cachorro” pelos dados referente ao mesmo);

- “Chave X não encontrada” indica que a Chave X não está presente na árvore-B e, consequente, no arquivo principal.

Exemplo Pesquisa

C

Chave C encontrada, página 0, posição 0

Z

Chave Z não encontrada

ii. Hash

Dado um “Código de Controle” o programa retorna os dados da respectiva vacinação. Para tanto, a busca deve ser feita na hash. Além disso, as seguintes mensagens deverão ser exibidas em relação à busca:

- “Chave X encontrada, endereço E, N acessos” indica que a chave X foi encontrada no endereço E e que foram necessários N acessos para recuperar a informação na hash. Após a exibição dessa mensagem, os dados referentes à vacinação deverão ser recuperados do arquivo principal (não esqueça de substituir o “Código do Cachorro” pelos dados referente ao mesmo);
- “Chave X não encontrada” indica que a Chave X não está presente na hash e, consequente, no arquivo principal.

Exemplo Pesquisa

1234567891234

Chave 1234567891234 encontrada, endereço 5, 1 acesso

4321987654321

Chave 4321987654321 encontrada, endereço 6, 2 acessos

1234567899999

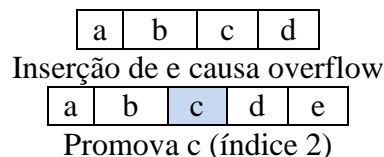
Chave 1234567899999 não encontrada

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 5 (ou seja, no máximo 4 chaves). Para padronizar, sempre promovam, quando houver overflow, a chave de índice 2 (a do meio), começando em zero.



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.