

## Exercício

O objetivo do exercício é gerenciar um sistema de streaming de filmes de uma dada empresa. O sistema armazena as seguintes informações: Código do Cliente (CodCli), Código do Filme (CodF), Nome do Cliente, Nome do Filme, e Gênero. A chave primária é composta pela composição “CodCli+CodF”. Para facilitar, suponha que um cliente pode assistir uma única vez um determinado filme. O arquivo a ser criado deve ser de registros fixos com campos fixos (fixo-fixo).

Código do Cliente (CodCli)	Código do Filme (CodF)	Nome do Cliente	Nome do Filme	Gênero
3 caracteres	3 caracteres	50 caracteres	50 caracteres	50 caracteres

O programa conterá as seguintes opções:

- Inserir
- Remover
- Buscar
- Carrega Arquivos (opcional)

### Inserir (a)

A cada inserção o “CodCli+CodF” deverá ser adicionado ao índice primário estruturado como uma hash. Portanto, 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

“001”+“333” (1333)

*Endereço 7*

*Chave inserida com sucesso*

“001”+“346” (1346)

*Endereço 7*

*Colisão*

*Tentativa 1*

*Chave inserida com sucesso*

### Observações:

- Para tratar as colisões utilize Overflow Progressivo;
- Considere uma hash de 13 posições;
- Utilize o Método da Divisão Inteira para encontrar o endereçamento de uma dada chave (função hash). Contudo, é necessário processar a chave antes, como no exemplo acima (“001”+“333” → 1333);
- Cada endereçamento contém uma chave (não 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.

### Remover (b)

Dado um “CodCli+CodF” o programa remove a respectiva chave do índice hash. Não é necessário realizar a remoção no arquivo principal (o que contém os registros, somente no índice). Para tanto, utilize o processo de remoção associado ao Overflow Progressivo.

### Buscar (c)

Dado um “CodCli+CodF” o programa retorna os dados do respectivo registro. 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 ao registro deverão ser recuperados do arquivo principal;
- “Chave X não encontrada” indica que a Chave X não está presente na hash e, conseqüente, no arquivo principal.

#### **Exemplo Pesquisa**

“001”+”333” (1333)

*Chave encontrada, endereço 7, 1 acesso*

“001”+”346” (1346)

*Chave encontrada, endereço 8, 2 acessos*

“001”+”888” (1888)

*Chave não encontrada*

### Carrega Arquivos (d)

A fim de facilitar os testes, serão fornecidos três arquivos: “insere.bin” (para teste da opção (a)); “busca.bin” (para teste da opção (c)); “remove.bin” (para teste da opção (b)). A ideia aqui é a mesma já descrita nos trabalhos anteriores.

### Observações

TODOS OS ARQUIVOS DEVERÃO SER MANIPULADOS EM MEMÓRIA SECUNDÁRIA.

Não criar os arquivos toda vez que o programa for aberto (fazer verificação).