5. Ordenar um arquivo para obter todos os registros em ordem de acordo com a chave.

6. Ajuntar dois arquivos para formar um arquivo maior.

A operação 5 foi objeto de estudo no Capítulo 4. A operação 6 demanda a utilização de técnicas sofisticadas e não será tratada neste texto.

Um nome comumente utilizado para descrever uma estrutura de dados para pesquisa é dicionário. Um **dicionário** é um **tipo abstrato de dados** com as operações Inicializa, Pesquisa, Insere e Retira. Em uma analogia com um dicionário da língua portuguesa, as chaves são as palavras e os registros são as entradas associadas com cada palavra, em que cada entrada contém pronúncia, definição, sinônimos e outras informações associadas com a palavra.

Para alguns dos métodos de pesquisa a serem estudados a seguir, vamos implementar o método como um dicionário, como é o caso das Árvores de Pesquisa (Seção 5.3) e Hashing (Seção 5.5). Para os métodos Pesquisa Sequencial e Pesquisa Digital, vamos implementar as operações Inicializa, Pesquisa e Insere e para o método Pesquisa Binária vamos implementar apenas a operação Pesquisa.

5.1 Pesquisa Sequencial

O método de pesquisa mais simples que existe funciona da seguinte forma: a partir do primeiro registro, pesquise sequencialmente até encontrar a chave procurada; então pare. Apesar de sua simplicidade, a pesquisa sequencial envolve algumas idéias interessantes, servindo para ilustrar vários aspectos e convenções a serem utilizadas em outros métodos de pesquisa a serem apresentados.

Uma forma possível de armazenar um conjunto de registros é por meio do tipo estruturado arranjo, conforme ilustra o Programa 5.1. Cada registro contém um campo chave que identifica o registro. Além da chave, podem existir outros componentes em um registro, os quais não têm influência muito grande nos algoritmos. Por exemplo, os outros componentes de um registro podem ser substituídos por um apontador contendo o endereço de um outro local que os contenha.

Uma possível implementação para as operações Inicializa, Pesquisa e Insere é mostrada no Programa 5.2.

A função Pesquisa retorna o índice do registro que contém a chave x; caso não esteja presente, o valor retornado é zero. Observe que esta implementação não suporta mais de um registro com uma mesma chave. Em aplicações com esta característica, é necessário incluir um argumento a mais na função Pesquisa para conter o índice a partir do qual se quer pesquisar, e alterar a implementação de acordo.

Programa 5.1 *Estrutura do tipo dicionário implementado como arranjo*

```
const MAXN = 10;
type TipoRegistro = record
                      Chave: TipoChave;
                      { outros componentes }
                    end;
     TipoIndice   = 0..MAXN;
     Tipotabela   = record
                      Item: array [TipoIndice] of TipoRegistro;
                      n   : TipoIndice;
                    end;
```

Programa 5.2 *Implementação das operações usando arranjo*

```
procedure Inicializa (var T: Tipotabela);
begin
  T.n := 0;
end; { Inicializa }

function Pesquisa (x: TipoChave; var T: Tipotabela): TipoIndice;
var i: integer;
begin
  T.Item[0].Chave := x;
  i := T.n + 1;
  repeat
    i := i - 1;
  until T.Item[i].Chave = x;
  Pesquisa := i;
end; { Pesquisa }

procedure Insere (Reg: TipoRegistro; var T: Tipotabela);
begin
  if T.n = MAXN
  then writeln('Erro: tabela cheia')
  else begin
         T.n := T.n + 1;
         T.Item[T.n] := Reg;
       end;
end; { Insere }
```

Um registro **sentinela** contendo a chave de pesquisa é colocado na posição zero do **array**. Essa técnica garante que a pesquisa sempre termina. Após a chamada da função Pesquisa, se o índice é zero, significa que a pesquisa foi sem sucesso.