Continuação do Programa 5.21

```
begin
  if t = nil
  then Insere := CriaNodoExt (k)
  else begin
       p := t;
       while not EExterno (p) do
         begin
         if Bit (p^.Index, k) = 1 then p := p^.Dir else p := p^.Esq;
         end;
       i := 1; { acha o primeiro bit diferente }
       while (i <=D) and (Bit (i, k) = Bit (i, p^.Chave)) do i := i+1;
       if i > D
       then begin
            writeln ('Erro: chave ja esta na arvore'); Insere := t;
            end
       else Insere := InsereEntre (k, t, i);
  end;
end; { Insere }
```

5.5 Transformação de Chave (*Hashing*)

Os métodos de pesquisa apresentados anteriormente são baseados na comparação da chave de pesquisa com as chaves armazenadas na tabela, ou na utilização dos *bits* da chave de pesquisa para escolher o caminho a seguir. O método de transformação de chave (ou *hashing*) é completamente diferente: os registros armazenados em uma tabela são diretamente endereçados a partir de uma transformação aritmética sobre a chave de pesquisa. De acordo com o *Webster's New World Dictionary*, a palavra *hash* significa: (i) fazer picadinho de carne e vegetais para cozinhar; (ii) fazer bagunça. Como veremos a seguir, o termo *hashing* é um nome apropriado para o método.

Um método de pesquisa com o uso da transformação de chave é constituído de duas etapas principais:

1. Computar o valor da **função de transformação** (também conhecida por **função *hashing***), a qual transforma a chave de pesquisa em um endereço da tabela;
2. Considerando que duas ou mais chaves podem ser transformadas em um mesmo endereço da tabela, é necessário existir um método para lidar com **colisões**.

Se porventura as chaves fossem inteiros de 1 a n, então poderíamos armazenar o registro com chave i na posição i da tabela, e qualquer registro poderia ser imediatamente acessado a partir do valor da chave. Por outro lado, vamos supor uma tabela capaz de armazenar $M = 97$ chaves, em que cada chave pode ser

um número decimal de quatro dígitos. Nesse caso, existem $N = 10.000$ chaves possíveis, e a **função de transformação** não pode ser um para um: mesmo que o número de registros a serem armazenados seja muito menor do que 97, qualquer que seja a função de transformação, algumas **colisões** irão ocorrer fatalmente, e tais colisões têm de ser resolvidas de alguma forma.

Mesmo que se obtenha uma função de transformação que distribua os registros de forma uniforme entre as entradas da tabela, existe alta probabilidade de haver colisões. O **paradoxo do aniversário** (Feller, 1968, p. 33) diz que em um grupo de 23 ou mais pessoas, juntas ao acaso, existe uma chance maior do que 50% de que 2 pessoas comemorem aniversário no mesmo dia. Isso significa que, se for utilizada uma função de transformação uniforme que enderece 23 chaves randômicas em uma tabela de tamanho 365, a probabilidade de que haja **colisões** é maior do que 50%. A probabilidade p de se inserirem N itens consecutivos sem colisão em uma tabela de tamanho M é:

$$p = \frac{M-1}{M} \times \frac{M-2}{M} \times \ldots \times \frac{M-N+1}{M} = \prod_{i=1}^{N} \frac{M-(i-1)}{M} = \prod_{i=1}^{N} 1 - \frac{(i-1)}{M}.$$

Sempre que $(i-1)/M \ll 1$, podemos utilizar nossos conhecimentos de cálculo para aproximar $1 - \frac{(i-1)}{M}$ por $e^{-\frac{(i-1)}{M}}$. Substituindo na equação acima, obtemos:

$$p \approx \prod_{i=1}^{N} e^{-\frac{(i-1)}{M}} = e^{-\frac{0+1+2+\cdots+(N-1)}{M}} = e^{-\frac{N(N-1)}{2M}}.$$

A tabela 5.1 mostra alguns valores de p para diferentes valores de N, onde $M = 365$.

Tabela 5.1 *Diferentes probabilidades para o paradoxo do aniversário*

N	p
10	0,884
22	0,531
23	0,500
30	0,304

5.5.1 Funções de Transformação

Uma função de transformação deve mapear chaves em inteiros dentro do intervalo $[0..M-1]$, no qual M é o tamanho da tabela. A função de transformação ideal é aquela que: (i) seja simples de ser computada; (ii) para cada chave de entrada, qualquer uma das saídas possíveis é igualmente provável de ocorrer.

Considerando que as transformações sobre as chaves são aritméticas, o primeiro passo é transformar as chaves não-numéricas em números. No caso do