

```

1: unit Busca;
2:
3: interface
4:
5: const
6:   NumMax = 10000;
7:
8: type
9:   Tipo_da_Chave = longint;
10:  FaixaLista    = 1..NumMax;
11:  Tipo do Dado  = longint;
12:  TVetor = array[FaixaLista] of Tipo_do_Dado;
13:
14: function BuscaSequencial(var Vetor: TVetor; Inicio, Fim: longint;
15:   Chave: Tipo_Da_Chave): longint;
16:
17: function BuscaBin(var Vetor: TVetor; Inicio, Fim: longint;
18:   Chave: Tipo_Da_Chave): longint;
19:
20: function BuscaBinRec(var Vetor: TVetor; Inicio, Fim: longint;
21:   Chave: Tipo_Da_Chave): longint;
22:
23: implementation
24:
25: function BuscaSequencial(var Vetor: TVetor; Inicio, Fim: longint;
26:   Chave: Tipo_Da_Chave): longint;
27: {
28:   Objetivo: Efetura a busca pela Chave entre as posicoes Inicio e Fim,
29:   partindo de Inicio sequencialmente ate o Fim.
30:   Se encontrar, retorna a posicao. Caso contrario retorna -1.
31: }
32: var
33:   Pos : longint;
34: begin
35:   BuscaSequencial := -1;
36:   Pos := Inicio;
37:   while Pos <= Fim do
38:     begin
39:       if Chave = Vetor[Pos] then
40:         begin
41:           BuscaSequencial := Pos;
42:           Pos := Fim
43:         end;
44:       inc(Pos)
45:     end
46: end;
47:
48:
49: function BuscaBin(var Vetor: TVetor; Inicio, Fim: longint;
50:   Chave: Tipo_Da_Chave): longint;
51: {
52:   Objetivo: Efetua a busca binaria pela Chave entre as posicoes Inicio e Fim.
53:   Se encontrar, retorna a posicao. Caso contrario retorna -1.
54: }
55: var
56:   Meio : longint;
57: begin
58:   BuscaBin := -1;
59:   repeat
60:     Meio := (Inicio + Fim) div 2;
61:     if (Chave < Vetor[Meio]) then

```

```

62:         Fim:=(Meio - 1)
63:     else
64:         if (Chave > Vetor[Meio]) then
65:             Inicio:=(Meio + 1)
66:         else
67:             begin
68:                 BuscaBin := Meio;
69:                 Inicio := Fim + 1;
70:             end
71:         until (Inicio > Fim)
72:     end;
73:
74:
75: function BuscaBinRec(var Vetor: TVetor; Inicio, Fim: longint;
76:                     Chave: Tipo_Da_Chave): longint;
77: {
78:     Objetivo: Efetua a busca binaria pela Chave entre as posicoes Inicio e Fim.
79:     Se encontrar, retorna a posicao. Caso contrario retorna -1.
80: }
81: var
82:     meio: longint;
83: begin
84:     if Inicio > Fim then
85:         BuscaBinRec := -1
86:     else
87:         begin
88:             meio := (Inicio + Fim) div 2;
89:             if Chave < Vetor[Meio] then
90:                 BuscaBinRec := BuscaBinRec(Vetor, Inicio, Meio-1, Chave)
91:             else
92:                 if Chave > Vetor[Meio] then
93:                     BuscaBinRec := BuscaBinRec(Vetor, Meio+1, Fim, Chave)
94:                 else
95:                     BuscaBinRec := Meio;
96:                 end;
97:             end;
98:         end;
99:     end.

```