

```

1: unit Insercao;
2: interface
3: uses DefDados;
4:
5: type
6:     Tipo_do_Incremento = record
7:         Num_Incrementos : TamLista;
8:         Valores          : array[TamLista] of longint;
9:     end;
10:
11: procedure InsercaoDireta(var Lista : Tipo da Lista);
12: procedure Shell(var Lista      : Tipo_da_Lista;
13:                var Incremento : Tipo_do_Incremento);
14:
15: implementation
16:
17: (*****)
18:
19: procedure InsercaoDireta(var Lista : Tipo_da_Lista);
20: var
21:     Atual      : Tipo_do_Dado;
22:     InicioParte2, J : TamLista;
23: begin
24:     with Lista do
25:         for InicioParte2 := 2 to N do
26:             begin
27:                 J      := InicioParte2;
28:                 Atual := A[J];
29:
30:                 { Coloca os maiores para a direita ate achar o ponto
31:                   de insercao }
32:                 while (J > 1) and (Atual.Chave < A[J - 1].Chave) do
33:                     begin
34:                         A[J] := A[J - 1];
35:                         dec(J);
36:                     end;
37:
38:                 A[J] := Atual
39:             end;
40: end; { InsercaoDireta }
41:
42: (*****)
43:
44: procedure Shell(var Lista      : Tipo_da_Lista;
45:                var Incremento : Tipo_do_Incremento);
46: var
47:     Atual      : Tipo_do_Dado;
48:     Passo, Incr,
49:     I, J, K      : TamLista;
50: begin
51:     for Passo := 1 to Incremento.Num_Incrementos do
52:         begin
53:             Incr := Incremento.Valores[Passo];
54:
55:             for K := 1 to Incr do
56:                 begin
57:                     I := K;
58:
59:                     with Lista do
60:                         while I + Incr <= N do
61:                             begin

```

```

62:         inc(I, Incr);
63:         J := I;
64:         Atual := A[J];
65:
66:         { Coloca os maiores para a direita ate achar o ponto
67:           de insercao }
68:         while (J > K) and (Atual.Chave < A[J - Incr].Chave) do
69:         begin
70:             A[J] := A[J - Incr];
71:             dec(J, Incr);
72:         end;
73:
74:         A[J] := Atual
75:     end;
76: end;
77: end;
78: end; { Shell }
79:
80: end.

```