

```

1  (* Ejercicio 1a *)
2
3  function TienenMismaOpcion(e1,e2: Persona) : boolean;
4  begin
5      (* La evaluación por circuito corto asegura que no se accede a campos
6       * que no están definidos para la opción correspondiente
7       *)
8      TieneMismaOpcion:=
9          (e1.opcion = idiomas) and (e2.opcion = idiomas) and (e1.lengua = e2.lengua)
10         or
11         (e1.opcion = pasantia) and (e2.opcion = pasantia) and (e1.codigo = e2.codigo)
12  end;
13
14  procedure misma_opcion (grupo : tipo_grupo, estudiante : Persona);
15  var
16      i:Integer;
17  begin
18      for i:=1 to grupo.cantidad do
19          if TienenMismaOpcion(grupo.personas[i],estudiante) then
20              writeln(grupo.personas[i].NrEst);
21  end;
22
23  (* Ejercicio 1b *)
24
25  procedure insertar_estudiante (estudiante: Persona; var grupo: tipo_grupo);
26  var i, j: integer;
27  begin
28      (* Se asume que hay lugar: grupo.cantidad < Max_NrEst *)
29      with grupo do
30          begin
31              (* búsqueda del lugar *)
32              i:= 1;
33              while (i <= cantidad) and (estudiante.NrEst > personas[i].NrEst) do
34                  i:=i+1;
35
36              (* corrimiento de los elementos que le siguen *)
37              for j:= cantidad downto i do
38                  personas[j+1]:= personas[j];
39
40              (* asignar el nuevo elemento en el lugar i *)
41              personas[i]:= estudiante;
42
43              (* ajustar el tope *)
44              cantidad:= cantidad + 1;
45          end;
46  end;
47
48  (* Ejercicio 2 *)
49
50  cédula 0
51
52  25 10 6 25
53  5 6 25
54  150 60 31 65
55  65 6 150
56
57  cédula 1
58
59  30 12 7 30
60  6 7 30
61  210 70 36 76
62  76 7 210
63
64  cédula 2
65
66  35 14 8 35
67  7 8 35
68  280 80 41 87
69  87 8 280
70
71  cédula 3
72
73  40 16 9 40

```

```

74 8 9 40
75 360 90 46 98
76 98 9 360
77
78 cédula 4
79
80 45 18 10 45
81 9 10 45
82 450 100 51 109
83 109 10 450
84
85 cédula 5
86
87 50 20 11 50
88 10 11 50
89 550 110 56 120
90 120 11 550
91
92 cédula 6
93
94 55 22 12 55
95 11 12 55
96 660 120 61 131
97 131 12 660
98
99 cédula 7
100
101 60 24 13 60
102 12 13 60
103 780 130 66 142
104 142 13 780
105
106 cédula 8
107
108 65 26 14 65
109 13 14 65
110 910 140 71 153
111 153 14 910
112
113 cédula 9
114
115 70 28 15 70
116 14 15 70
117 1050 150 76 164
118 164 15 1050
119
120 (* Ejercicio 3 *)
121
122 function validarLuhn(numero:TNumero):boolean;
123 var
124 i, suma, valor: integer;
125
126 begin
127
128 i:=1;
129 suma := 0;
130
131 while (i<=MAX) and (numero[i]>='0') and (numero[i]<='9') do
132 begin
133 valor := ord(numero[i]) - ord('0');
134 if (i MOD 2 = 0) then
135 begin
136 valor := valor * 2;
137 suma := suma + (valor DIV 10) + (valor MOD 10);
138 end
139 else
140 suma := suma + valor;
141 i:=i+1;
142 end;
143
144 validarLuhn := (i > MAX) and (suma MOD 10 = 0);
145
146 end;

```