



Ejercicio 1

[3 puntos]

```
1 function sumaDigitos(n: integer): integer;
2 var
3     suma: integer;
4 begin
5     suma := 0;
6     while (n > 0) do
7         begin
8             suma := suma + n mod 10;
9             n := n div 10;
10        end;
11    sumaDigitos := suma;
12 end;
13
14 function valorSheldon(n: integer; lim: integer): integer;
15 var
16     suma: integer;
17 begin
18     if (n <= 0) then
19         valorSheldon := -1
20     else
21         begin
22             suma := n;
23             while (suma >= lim) do
24                 begin
25                     suma := sumaDigitos(suma);
26                 end;
27             valorSheldon := suma;
28         end;
29 end;
```

SOLUCIÓN 1: Con vectores, mejor

```
1 program contabilidad;
2
3 type tpCuentasB = array[0..999] of Boolean;
4
5 var fContabilidad, fContabilidadA, fContabilidadB, fCuentasB : Text;
6     cuentasB:tpCuentasB;
7     debe, haber, cuenta: Integer;
8     cantidad: Real;
9     concepto: String;
10
11 begin
12     { Primero identificamos todas las cuentas del fichero cuentasB.txt
13       y las metemos en un vector de booleanos para facilitar el acceso. }
14     for cuenta := 0 to 999 do cuentasB[cuenta] := false;
15     assign(fCuentasB,'cuentasB.txt');
16     reset(fCuentasB);
17     while not eof(fCuentasB) do begin
18         readln(fCuentasB,cuenta);
19         cuentasB[cuenta] := true;
20     end;
21     close(fCuentasB);
22
23     { Empezamos a leer el fichero de contabilidad y a escribir en A y B}
24     assign(fContabilidad,'contabilidad.txt'); reset(fContabilidad);
25     assign(fContabilidadA,'contabilidadA.txt'); rewrite(fContabilidadA);
26     assign(fContabilidadB,'contabilidadB.txt'); rewrite(fContabilidadB);
27
28     { Leemos asiento a asiento, guardándolo en el fichero correspondiente }
29     while not eof(fContabilidad) do begin
30         readln(fContabilidad, debe, haber, cantidad, concepto);
31         { Miramos si la cuenta del debe o de haber es cuenta tipo B }
32         if cuentasB[debe] or cuentasB[haber] then
33             writeln(fContabilidadB, debe, ' ', haber, ' ',
34                 cantidad:8:2, ' ', concepto)
35         else
36             writeln(fContabilidadA, debe, ' ', haber, ' ',
37                 cantidad:8:2, ' ', concepto);
38     end;
39
40     close(fContabilidad);
41     close(fContabilidadA);
42     close(fContabilidadB);
43 end.
```

SOLUCIÓN 2: Sin vectores, peor

```
1  program contabilidad;
2
3  function esB(cuenta: Integer): Boolean;
4  { Esta funcion comprueba si una cuenta esta en el fichero cuentasB. }
5  var t:Text; i: Integer;
6  begin
7      assign(t,'cuentasB.txt');
8      reset(t);
9      esB := false;
10     while not eof(t) do begin
11         readln(t,i);
12         if (i = cuenta) then esB := true;
13         end;
14         close(t);
15     end;
16
17
18     var fContabilidad, fContabilidadA, fContabilidadB : Text;
19         debe, haber: Integer;
20         cantidad: Real;
21         concepto: String;
22
23     begin
24         { Empezamos a leer el fichero de contabilidad y a escribir en los de A y B}
25         assign(fContabilidad,'contabilidad.txt');  reset(fContabilidad);
26         assign(fContabilidadA,'contabilidadA.txt'); rewrite(fContabilidadA);
27         assign(fContabilidadB,'contabilidadB.txt'); rewrite(fContabilidadB);
28
29         { En este bucle leemos asiento a asiento. }
30         while not eof(fContabilidad) do begin
31             readln(fContabilidad, debe, haber, cantidad, concepto);
32             { Miramos si la cuenta del debe o de haber es cuenta tipo B }
33             if esB(debe) or esB(haber) then
34                 writeln(fContabilidadB, debe, ' ', haber, ' ',
35                     cantidad:8:2, ' ', concepto)
36             else
37                 writeln(fContabilidadA, debe, ' ', haber, ' ',
38                     cantidad:8:2, ' ', concepto);
39             end;
40
41             close(fContabilidad);
42             close(fContabilidadA);
43             close(fContabilidadB);
44         end.
```

Ejercicio 3

[3.5 puntos]

```
1  const MAXPALGRUPO  = 20; MAXCONCEPTOS = 1000;
2  type
3    tpGrupoPalabras = record
4      palabras  : array [1..MAXPALGRUPO] of string;
5      npalabras : integer;
6    end;
7    tpConcepto = record
8      grupoa : tpGrupoPalabras;
9      grupob : tpGrupoPalabras;
10   end;
11   tpTesauro = record
12     conceptos : array [1..MAXCONCEPTOS] of tpConcepto;
13     nconceptos : integer;
14   end;
15
16 function estaEnGrupo(palabra : string; grupo : tpGrupoPalabras) : boolean;
17 var encontrada : boolean; i : integer;
18 begin
19   encontrada := false; i := 1;
20   while (i<=grupo.npalabras) and (not encontrada) do begin
21     encontrada := (grupo.palabras[i] = palabra);
22     { encontrada := (CompareStr(grupo.palabras[i],palabra) = 0); }
23     i := i+1;
24   end;
25   estaEnGrupo := encontrada;
26 end;
27 procedure escribeGrupo(nombre: string; grupo: tpGrupoPalabras);
28 var i : integer;
29 begin
30   write(nombre, ': ');
31   for i:=1 to grupo.npalabras do write(grupo.palabras[i], ' ');
32   writeln;
33 end;
34 procedure muestraAlternativas(palabra : string; tesauro : tpTesauro);
35 var encontrada : boolean; i : integer;
36 begin
37   encontrada := false;
38   for i:=1 to tesauro.nconceptos do begin
39     if estaEnGrupo(palabra,tesauro.conceptos[i].grupoa) then begin
40       encontrada := true;
41       escribeGrupo('Sinonimos',tesauro.conceptos[i].grupoa);
42       escribeGrupo('Antonimos',tesauro.conceptos[i].grupob);
43     end;
44     if estaEnGrupo(palabra,tesauro.conceptos[i].grupob) then begin
45       encontrada := true;
46       escribeGrupo('Sinonimos',tesauro.conceptos[i].grupob);
47       escribeGrupo('Antonimos',tesauro.conceptos[i].grupoa);
48     end;
49   end;
50   if not encontrada then writeln(palabra,': no encontrda');
51 end;
```