



Ejercicio 1

[3 puntos]

```
1 program generarenvolverte;
2 type
3     tpVertice = record
4         x: real;
5         y: real;
6     end;
7     tpPoligono = record
8         vertices : array[1..20] of tpVertice;
9         n : Integer; { Numero de vertices }
10    end;
11    tpObjeto = file of tpPoligono;
12
13 var nombre : String;
14     objeto: tpObjeto;
15     poligono : tpPoligono;
16     i : Integer;
17     primero : boolean;
18     xmin, ymin, xmax, ymax : Real;
19 begin
20     write('Nombre del fichero : '); readln(nombre);
21     assign(objeto,nombre);
22     reset(objeto);
23     { De momento no hemos contado el primer vertice }
24     primero := true;
25     while not eof(objeto) do begin
26         read(objeto, poligono);
27         for i:=1 to poligono.n do begin
28             if primero or (xmin > poligono.vertices[i].x)
29                 then xmin := poligono.vertices[i].x;
30             if primero or (ymin > poligono.vertices[i].y)
31                 then ymin := poligono.vertices[i].y;
32             if primero or (xmax < poligono.vertices[i].x)
33                 then xmax := poligono.vertices[i].x;
34             if primero or (ymax < poligono.vertices[i].y)
35                 then ymax := poligono.vertices[i].y;
36             { Ya hemos contado el primer vertice }
37             if primero
38                 then primero:=false;
39         end;
40     end;
41     if primero
42         then writeln('Objeto vacio')
43     else begin
44         writeln('Rectangulo envolvente:');
45         writeln(xmin:1:2, ' ',ymin:1:2, ' - ',xmax:1:2, ' ',ymax:1:2);
46     end;
47     close(objeto);
```

48 **end.**

Ejercicio 2

[3.5 puntos]

```
1  program mensajeros;
2
3  function transcurrido(ti, tf: integer): integer;
4  var
5      minsti, minstf: integer;
6  begin
7      minsti := (ti div 100) * 60 + ti mod 100;
8      minstf := (tf div 100) * 60 + tf mod 100;
9      transcurrido := minstf - minsti;
10 end;
11
12 var
13     t: text;
14     suma_pe_Z1, suma_pe_Z2, suma_pe_Z3, pe: real;
15     contZ1, contZ2, contZ3: integer;
16     prom_pe_Z1, prom_pe_Z2, prom_pe_Z3: real;
17     id, maxId: integer;
18     tAcum, maxTAcum: integer;
19     tIni, tFin, zona: integer;
20     cantidad: real;
21     minutos: integer;
22 begin
23     assign(t, 'reparto.txt');
24     reset(t);
25     maxTAcum := 0; maxId := 0;
26     suma_pe_Z1 := 0; suma_pe_Z2 := 0; suma_pe_Z3 := 0;
27     contZ1 := 0; contZ2 := 0; contZ3 := 0;
28     while not eof(t) do
29         begin
30             read(t, id); tAcum := 0;
31             while not eoln(t) do
32                 begin
33                     read(t, tIni, tFin, zona, cantidad);
34                     minutos := transcurrido(tIni, tFin);
35                     pe := cantidad / minutos;
36                     if (zona = 1) then
37                         begin
38                             suma_pe_Z1 := suma_pe_Z1 + pe; contZ1 := contZ1 + 1;
39                         end
40                     else if (zona = 2) then
41                         begin
42                             suma_pe_Z2 := suma_pe_Z2 + pe; contZ2 := contZ2 + 1;
43                         end
44                     else
45                         begin
46                             suma_pe_Z3 := suma_pe_Z3 + pe; contZ3 := contZ3 + 1;
47                         end;
48                     tAcum := tAcum + minutos;
49                 end;
50                 if tAcum > maxTAcum then
51                     begin
52                         maxTAcum := tAcum;
53                         maxId := id;
54                     end;
55                 end;
56             close(t);
57             prom_pe_Z1 := suma_pe_Z1 / contZ1;
58             prom_pe_Z2 := suma_pe_Z2 / contZ2;
```

```
59  prom_pe_Z3 := suma_pe_Z3 / contZ3;
60  if (prom_pe_Z1 > prom_pe_Z2) and (prom_pe_Z1 > prom_pe_Z3) then
61    writeln('Mejor coste: Zona 1 ( ', prom_pe_Z1:1:2, ' eur/min de media)')
62  else if (prom_pe_Z2 > prom_pe_Z1) and (prom_pe_Z2 > prom_pe_Z3) then
63    writeln('Mejor coste: Zona 2 ( ', prom_pe_Z2:1:2, ' eur/min de media)')
64  else
65    writeln('Mejor coste: Zona 3 ( ', prom_pe_Z3:1:2, ' eur/min de media)');
66  writeln('Mas trabajador: ', maxId, ' ( ', maxTAcum, ' minutos)');
67  end.
```

Ejercicio 3

[3.5 puntos]

```
1  program bingo;
2  const
3      MAXFIL = 4;
4      MAXCOL = 9;
5      MAXBOL = MAXCOL*10-1;
6  type
7      tpCasilla = record
8          ocupada : boolean;
9          numero  : integer;
10         marcada : boolean;
11     end;
12     tpCarton = array[1..MAXFIL,1..MAXCOL] of tpCasilla;
13     tpBolas = record
14         n      : integer;
15         nums   : array[1..MAXBOL] of integer;
16     end;
17
18 procedure marcar(var carton: tpCarton; const bolas: tpBolas);
19 var
20     i,c,f : integer;
21 begin
22     for i:=1 to bolas.n do
23         begin
24             c := (bolas.nums[i] div 10) + 1;
25             for f:=1 to MAXFIL do
26                 if carton[f,c].ocupada and
27                    (carton[f,c].numero = bolas.nums[i]) then
28                     carton[f,c].marcada := true;
29             end;
30         end;
31
32 function hayLinea(const carton: tpCarton): boolean;
33 var
34     haylin,eslin : boolean;
35     f,c : integer;
36 begin
37     f := 1;
38     haylin := false;
39     while (f<=MAXFIL) and not haylin do
40         begin
41             eslin:=true;
42             for c:=1 to MAXCOL do
43                 begin
44                     if carton[f,c].ocupada and not carton[f,c].marcada
45                     then
46                         eslin := false;
47                     end;
48                     if eslin then haylin:=true;
49                     f:=f+1;
50                 end;
51             hayLinea := haylin;
52         end;
53
54 function faltanParaBingo(const carton: tpCarton): integer;
55 var
56     f,c,cnt : integer;
57 begin
58     cnt := 0;
```

```
59  for f:=1 to MAXFIL do
60  begin
61    for c:=1 to MAXCOL do
62    begin
63      if carton[f,c].ocupada and not carton[f,c].marcada
64      then
65        cnt:=cnt+1;
66      end;
67    end;
68    faltanParaBingo := cnt;
69  end;
70
71  begin
72  end.
```