



Ejercicio 1

[3 puntos]

```
1 program primoEquilibradoRango;
2
3 function esPrimo(n: integer): boolean;
4 var
5     i: integer;
6     loEs: boolean;
7 begin
8     loEs := (n <> 1); { el 1 no es primo }
9     i := 2;
10    while (i < n) and loEs do begin
11        if (n mod i = 0) then loEs := false;
12        i := i + 1;
13    end;
14    esPrimo := loEs;
15 end;
16
17 var
18     p_ant, p, p_sig: integer;
19     esPrEq: boolean;
20     lInf, lSup: integer;
21     i: integer;
22 begin
23     writeln('Introduce limites inf y sup del rango: ');
24     readln(lInf, lSup);
25     for p:=lInf to lSup do begin
26         esPrEq := false; { falso hasta que se demuestre lo contrario }
27         if esPrimo(p) then { solo se entra a comprobar ant y sig si num es primo }
28             begin
29                 p_ant := 0;
30                 p_sig := 0;
31                 { Buscar p_ant }
32                 i := p - 1;
33                 while not esPrimo(i) and (i > 0) do
34                     i := i - 1;
35                 p_ant := i;
36                 { Buscar p_sig }
37                 i := p + 1;
38                 while not esPrimo(i) do
39                     i := i + 1;
40                 p_sig := i;
41                 { Comprobar si se cumple la propiedad }
42                 esPrEq := (p = ((p_ant + p_sig) / 2));
43             end;
44         if esPrEq then write(p, ' ');
45     end;
46 end.
```

```
1  program horarios;
2
3  type
4      tpHueco = record
5          ocupado : Boolean;
6          descripcion: String;
7      end;
8      tpHorario = array[1..5,0..23] of tpHueco;
9
10
11 function indiceDia(c:Char):Integer;
12 begin
13     if (c='L') then indiceDia := 1
14     else if (c='M') then indiceDia := 2
15     else if (c='X') then indiceDia := 3
16     else if (c='J') then indiceDia := 4
17     else indiceDia := 5;
18 end;
19
20 procedure mostrarDia(i:Integer);
21 begin
22     if (i=1) then write('lunes')
23     else if (i=2) then write('martes')
24     else if (i=3) then write('miercoles')
25     else if (i=4) then write('jueves')
26     else write('viernes');
27 end;
28
29 var      t:text; dia: char;
30          d, h, horaIni, horaFin: integer;
31          descripcion: String;
32          horario: tpHorario;
33 begin
34     for d := 1 to 5 do
35         for h := 0 to 23 do
36             horario[d,h].ocupado := false;
37
38 assign(t,'horarios.txt');
39 reset(t);
40 while not eof(t) do begin
41     readln(t, dia, horaIni, horaFin, descripcion);
42     d := indiceDia(dia);
43     for h := horaIni to (horaFin - 1) do begin
44         if horario[d,h].ocupado then begin
45             write('Conflicto los '); mostrarDia(d); writeln(' de ',h,'h a ',(h+1),'h : ');
46             writeln(' ',descripcion);
47             writeln(' ',horario[d,h].descripcion);
48         end else begin
49             horario[d,h].ocupado := true;
50             horario[d,h].descripcion := descripcion;
51         end;
52     end;
53 end;
54 close(t);
55 end.
```

```
1  const
2    MAXPAT = 1000;
3    MAXNIVEL = 5;
4  type
5    tpPatinador = record
6      nombre: string;
7      nivel: integer;
8      votos: array[1..3] of boolean;
9    end;
10   tpGrupo = record
11     numpat: integer;
12     patinadores: array[1..MAXPAT] of tpPatinador;
13   end;
14   tpNiveles = array[1..MAXNIVEL] of tpGrupo;
15
16 procedure SeparaPorNiveles(const todos: tpGrupo; var niveles: tpNiveles);
17 var
18   n,i : integer;
19 begin
20   for i:=1 to MAXNIVEL do
21     niveles[i].numpat := 0;
22   for i:=1 to todos.numpat do
23     begin
24       n := todos.patinadores[i].nivel;
25       niveles[n].numpat := niveles[n].numpat+1;
26       niveles[n].patinadores[niveles[n].numpat] := todos.patinadores[i];
27     end;
28 end;
29
30 function Aprueba(const p: tpPatinador): boolean;
31 var
32   ok,i : integer;
33 begin
34   ok := 0;
35   for i:=1 to 3 do
36     if p.votos[i] then ok := ok+1;
37   Aprueba := (ok>=2);
38 end;
39
40 procedure ActualizaTabla(const original: tpNiveles;
41   var nueva: tpNiveles);
42 var
43   ni,nf : integer; { Niveles inicial y final tras el test }
44   i,j : integer;
45   pos : integer;
46 begin
47   { Inicio }
48   for nf:=1 to MAXNIVEL-1 do
49     nueva[nf].numpat := 0;
50   nueva[MAXNIVEL] := original[MAXNIVEL];
51
52   { Patinadores de nivel inicial < MAXNIVEL }
53   for ni:=1 to MAXNIVEL-1 do
54     begin
55       for i:=1 to original[ni].numpat do
```

```

56  begin
57      { Nivel final }
58      if Aprueba(original[ni].patinadores[i])
59      then  nf := ni+1
60      else  nf := ni;
61      { Se añade al grupo de su nivel final }
62      nueva[nf].numpat := nueva[nf].numpat+1;
63      pos := nueva[nf].numpat;
64      nueva[nf].patinadores[pos] := original[ni].patinadores[i];
65      nueva[nf].patinadores[pos].nivel := nf;
66      { Anula votos }
67      for j:=1 to 3 do
68          nueva[nf].patinadores[pos].votos[j] := false;
69      end;
70  end;
71 end;
72
73 begin
74 end.

```