

**Fundamentos de la programación y la informática**  
**Test parcial**  
**19 de Enero de 2022**

Grados en ingeniería aeroespacial. Turno de tarde  
Universidad Rey Juan Carlos

---

- Ejecuta en un terminal la orden  
  `~mortuno/prepara`
- Comprueba que esto ha dejado en tu cuenta el fichero  
  `~/fpi.enero.22/parcial.TULOGIN.txt`, donde deberás resolver el ejercicio 2 <sup>1</sup>.

## Ejercicio 1 (3 puntos)

Sean  $a, b, c, d$  variables booleanas. Sean  $x$  e  $y$  variables reales. Sean las expresiones

$e1 := \text{not} ( (a \text{ and not } b) \text{ and not } (c \text{ or } d) );$

$e2 := \text{not} (x > 3) \text{ or } (y \leq 5);$

1. A partir de  $e1$ , escribe una expresión lógica equivalente, más legible, de nombre  $s1$ .
2. A partir de  $e2$ , escribe una expresión lógica equivalente, más legible, de nombre  $s2$ .
3. Escribe una expresión  $n1$  que sea la negación de  $s1$ .
4. Escribe una expresión  $n2$  que sea la negación de  $s2$ .

## Solución

$s1 := \text{not } a \text{ or } b \text{ or } c \text{ or } d$

$s2 := (x \leq 3) \text{ or } (y \leq 5)$

$n1 := a \text{ and not } b \text{ and not } c \text{ and not } d$

$n2 := (x > 3) \text{ and } (y > 5)$

## Ejercicio 2 (7 puntos)

Indica qué líneas del siguiente programa contienen errores. Describe brevemente cada error, aunque no es necesario que lo clasifiques.

```
{ $mode objfpc } { $H } { $R+ } { $T+ } { $Q+ } { $V+ } { $D+ } { $X- } { $warnings on }
```

```
1 program pts_eur;
2
3 var
4   val_ejemplo : real = 1000;
5   val_convertido : real;
6
7 procedure pts_eur(pts:real; var eur: real );
8 begin
9   euros := pts / 166.386;
10 end;
11
12 begin
13   valor_convertido = pts_eur(valor_ejemplo)
14   write(valor_ejemplo:0:2, ' pts son ')
15   writeln(valor_convertido:0:2, ' euros')
16 end.
```

---

<sup>1</sup>TULOGIN será tu nombre de usuario en el laboratorio

## Solución

```
{ $mode objfpc } { $H- } { $R+ } { $T+ } { $Q+ } { $V+ } { $D+ } { $X- } { $warnings on }
```

```
1  program pts_eur;
2
3  // Normalmente los números deben definirse como constantes, no
4  // usar 'números mágicos'
5  const PtsEur = 166.386;
6
7  // Este subprograma se invoca como función, así que debe ser una
8  // función, no un procedimiento (o si se usa como procedimiento,
9  // invocarlo correctamente)
10 function pts_eur(pts:real): real;
11 begin
12     result := pts / PtsEur;
13 end;
14
15 // Si llamamos al parámetro 'eur', no podemos escribir la expresión
16 // con el identificador 'euros'
17 procedure pts_eur(pts:real; var eur: real );
18 begin
19     eur := pts / PtsEur;
20 end;
21
22 // Las variables deben ser locales
23 var
24     valor_ejemplo : real = 1000;
25     valor_convertido : real;
26 // Las variables deben usarse con el mismo nombre con el que se
27 // han definido
28
29 begin
30     // pts -> eur
31     // El operador de asignación es :=, no =
32     valor_convertido := pts_eur(valor_ejemplo) ;
33
34     // Las sentencias acaban en punto y coma
35     pts_eur(valor_ejemplo, valor_convertido);
36     write(valor_ejemplo:0:2, ' pts son ');
37     writeln(valor_convertido:0:2, ' euros');
38 end.
```