

Fundamentos de la programación y la informática

Test parcial

19 de Enero de 2023

Grados en ingeniería aeroespacial. Turno de tarde

Universidad Rey Juan Carlos

- Este examen consta de dos ejercicios, uno por cada cara del enunciado.
- Ejecuta en un terminal la orden
`~mortuno/prepara`
- Comprueba que esto ha dejado en tu cuenta el fichero
`~/fpi.enero.23/parcial.TULOGIN.txt`, donde deberás escribir la solución de ambos ejercicios ¹.

Ejercicio 1 (3 puntos)

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

```
e1 := not ( not (a and b) or (not c and d) );
```

```
e2 := not (x <= 5) and (y > 0);
```

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 := (a and b) and (c or not d)
```

```
s2 := (x > 5) and (y > 0)
```

```
n1 := (not a or not b ) or ( not c and d)
```

```
n2 := (x <= 5) or (y <= 0)
```

¹TULOGIN será tu nombre de usuario en el laboratorio

Ejercicio 2 (7 puntos)

Necesitamos un programa que, a partir de la nota de un examen, si es un aprobado (normalmente 5), la bonifique multiplicando por cierto factor. Por ejemplo 1.1. Si el resultado es mayor que el máximo (normalmente 10), se deja en exactamente 10. Esto serían tres ejemplos distintos de ejecución:

```
Nota inicial:5.50
Nota bonificada:6.05
```

```
Nota inicial:3.00
Nota bonificada:3.00
```

```
Nota inicial:9.6
Nota bonificada:10
```

A continuación se muestra un programa que intenta, sin éxito, obtener estos resultados. 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 ej_bonifica;
2
3  const
4      Aprobado = 5;
5      NotaMaxima = 10;
6      FactorBonus = 1.1;
7
8  var
9      nota: integer;
10
11 procedure bonifica(x:real);
12 begin
13     if (x > Aprobado) then
14         x := x * FactorBonus;
15     // A los aprobados se les añade el bonus
16
17     if (x > MaximaNota ) then
18         x = MaximaNota;
19     // Si la nota es mayor que el máximo,
20     // la dejamos en el máximo
21
22 end;
23
24 begin
25     nota := 5.5;
26     write('Nota inicial:');
27     writeln( nota:0:2);
28     nota := bonifica(nota);
29     write('Nota bonificada:')
30     writeln(nota:0:2);
31 end;
```

Solución

- En la línea 5 la constante se llama *NotaMaxima*, mientras que en las líneas 17 y 18 el nombre es *MaximaNota*. Cualquiera de los dos identificadores serían correctos, con tal de que fueran iguales.

- Línea 9. La variable *nota* se declara global. Esto es un fallo muy severo, debería ser una variable local del cuerpo del programa principal.
- Línea 9. La variable *nota* se declara entera, pero las notas son números reales.
- Línea 11. El parámetro *x* tal y como está es de entrada. Falta declararlo de entrada salida anteponiendo *var*.
- Línea 13. El operador de comparación debería ser *mayor o igual*, no *mayor*. De lo contrario, a los aprobados con la nota mínima (típicamente 5), no se les bonifica.
- Línea 18. El operador de asignación es *:=* (dos puntos igual) , no *=* (igual).
- Línea 28. Llama al subprograma *bonifica* como si fuera una función, pero es un procedimiento².
- Línea 29. La sentencia debería acabar en punto y coma.
- Línea 31. El programa tiene que acabar en punto, no en punto y coma.

Por tanto, la versión correcta es la siguiente:

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

```
program ej_bonifica;

const
  Aprobado = 5;
  MaximaNota = 10;
  FactorBonus = 1.1;

procedure bonifica(var x:real);
begin
  if (x >= Aprobado) then
    x := x * FactorBonus;
  // A los aprobados se les añade el bonus

  if (x > MaximaNota ) then
    x := MaximaNota;
  // Si la nota es mayor que el máximo,
  // la dejamos en el máximo

end;

var
  nota: real;

begin
  nota := 5.5;
  write('Nota inicial:');
  writeln( nota:0:2);
  bonifica(nota);
  write('Nota bonificada:');
  writeln( nota:0:2);
end.
```

²Alternativamente, también podríamos declarar el subprograma como función en la línea 11, para que se corresponda con la llamada