Fundamentos de la programación y la informática
Examen de teoría. 19 de diciembre de 2019
Grado en ingeniería aeroespacial en vehículos aeroespaciales
Universidad Rey Juan Carlos

Preparativos

- Ejecuta el script `~mortuno/prepara`
- Esto creará en tu ordenador el fichero `~/fpi.teoria.dic.19/teoria.TULOGIN.txt`, donde TULOGIN es tu nombre de usuario en el laboratorio. Escribe tus respuestas en este fichero.

Ejercicio 1 (2 puntos)

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

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

e2 := not  (x > 15) and not (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

1. `(not b or not c) and (b or d or a)`
2. `(x <= 15 and y <> 0)`
3. `(b and c) or (not b and not d and not a)`
4. `(x > 15) or (y = 0)`

Ejercicio 2 (8 puntos)

Un programador desea resolver el siguiente problema: en una asignatura el profesor da puntos extra a los alumnos que participan activamente en clase. Algunos alumnos tienen hasta 15 puntos, pero el profesor desea que la nota *sature* a partir de 10, esto es, las notas como 11, 12, etc, serán reemplazadas por un 10.

Así que escribe el programa mostrado a continuación para simular esto: genera notas al azar entre 0 y 15 (ambos inclusive), las guarda en un array y luego genera otro array filtrando el array inicial. El programador es novato y escribe el programa muy mal. Como sabes, en un programa mal escrito nos podemos encontrar con errores de compilación, errores de ejecución, errores lógicos y defectos en la claridad del código. Encuentra y describe brevemente todos los errores y todos los defectos que encuentres.

- Deja claro si se trata de un error o un defecto, aunque no es necesario que especifiques si el error es de compilación, ejecución o lógico.
- Obviamente, el número que aparece al principio de cada línea no forma parte del programa, es el número de línea. Para cada error que encuentres, indica en qué linea está.

  Por ejemplo, un error encontrado en el código podría describirse de la siguiente forma: *Línea 2: Error: los programas en Pascal no empiezan por 'Programa' sino por 'Program'*

```
01 {$mode objfpc}{$H-}{$R+}{$T+}{$Q+}{$V+}{$D+}{$X-}{$warnings on}
02 programa filtra_tabla;
03 uses crt;
04 const
05     tamanio_array : 8;
06 type
07     TipoTabla : array[1..tamanio_array] of integer;
08 var
09     tabla_bruto, tabla_filtrada = TipoTabla;
10
11
12 function tira_dado(caras_dado: integer):integer;
13 begin
14     result :=  random( caras_dado ) + 1 ;
15 end;
16
17
18 procedure rellena_tabla(var tabla:TipoTabla; valor_maximo:integer);
19 var
20     i : integer;
21 begin
22     for i := 1 to tamanio_array do
23         tabla[i] := tira_dado(valor_maximo);
24 end;
25
26
27 procedure filtra_tabla(tabla_entrada, tabla_salida:TipoTabla; limite_Saturación:integer);
28 var
29     i : integer;
30 begin
31     for i = 1 to tamanio_array do begin
32         if tabla_entrada[i] < limite_Saturación then
33             tabla_salida[i] := limite_Saturación;
34         else
35             tabla_salida[i] := tabla_entrada[i]
36         end;
37 end;
38
39
40 procedure muestra_tabla(var tabla: TipoTabla);
41 var
42     i : integer;
43 begin
44     for i = 1 to tamanio_array do
45         writeln(tabla[i]);
46 end;
47
48
49 const
50     ValorMaximo = 15;
51     LimiteSaturación = 10;
```

```
52 begin
53     rellena_tabla(tabla_bruto, ValorMaximo);
54     tabla_filtrada := filtra_tabla(tabla_bruto,  LimiteSaturación);
55     writeln('Tabla en bruto:');
56     muestra_tabla(tabla_bruto);
57     writeln('Tabla filtrada:');
58     muestra_tabla(tabla_filtrada);
59 end;
```

Solución:

(En la página siguiente)

Código dado en el examen (mal)

```
programa filtra_tabla;
uses crt;
const
    tamanio_array : 8;
type
    TipoTabla : array[1..tamanio_array] of integer;
var
    tabla_bruto, tabla_filtrada = TipoTabla;


function tira_dado(caras_dado: integer):integer;
begin
    result := random( caras_dado ) + 1 ;
end;


procedure rellena_tabla(var tabla:TipoTabla; valor_maximo:integer);
var
    i : integer;
begin
    for i := 1 to tamanio_array do
        tabla[i] := tira_dado(valor_maximo);
end;


procedure filtra_tabla(tabla_entrada, tabla_salida:TipoTabla; limite_Saturación:integer);
var
    i : integer;
begin
    for i = 1 to tamanio_array do begin
        if tabla entrada[i] < limite Saturación then
            tabla_salida[i] := limite_Saturación;
        else
            tabla_salida[i] := tabla_entrada[i]
        end;
end;


procedure muestra_tabla(var tabla: TipoTabla);
var
    i : integer;
begin
    for i = 1 to tamanio_array do
        writeln(tabla[i]);
end;




const
    ValorMaximo = 15;
    LimiteSaturación = 10;

begin



    rellena tabla(tabla bruto, ValorMaximo);
    tabla filtrada := filtra tabla(tabla_bruto, LimiteSaturación);
    writeln('Tabla en bruto:');
    muestra tabla(tabla bruto);
    writeln('Tabla filtrada:');
    muestra_tabla(tabla_filtrada);
end;
```

Solución (bien)

```
program filtra_tabla;
uses crt;
const
    TamanioArray = 8;
type
    TipoTabla = array[1..TamanioArray] of integer;




function tira_dado(caras_dado: integer):integer;
begin
    result := random( caras_dado ) + 1 ;
end;


procedure rellena_tabla(var tabla:TipoTabla; valor_maximo:integer);
var
    i : integer;
begin
    for i := 1 to TamanioArray do
        tabla[i] := tira_dado(valor_maximo)-1;
end;

procedure filtra_tabla(var tabla_entrada, tabla_salida:TipoTabla; limite_saturacion:integer);

var
    i : integer;
begin
    for i := 1 to TamanioArray do begin
        if tabla entrada[i] > limite saturacion then
            tabla_salida[i] := limite_saturacion
        else
            tabla_salida[i] := tabla_entrada[i]
        end;
end;


procedure muestra_tabla(var tabla: TipoTabla);
var
    i : integer;
begin
    for i := 1 to TamanioArray do
        writeln(tabla[i]);
end;


var
    tabla_bruto, tabla_filtrada : TipoTabla;
const
    ValorMaximo = 16; // Generaremos valores entre 1 y 16, luego restaremos 1
                      // para que vaya de 0 a 15
    LimiteSaturacion = 10;
begin
    delay(1000);
    randomize;

    rellena tabla(tabla bruto, ValorMaximo);
    filtra tabla(tabla bruto, tabla_filtrada, LimiteSaturacion);
    writeln('Tabla en bruto:');
    muestra tabla(tabla bruto);
    writeln('Tabla filtrada:');
    muestra_tabla(tabla_filtrada);
end.
```