

# Soluciones – Examen Febrero 2016

## Ejercicio 1

a)

```
function area(f: Figura) : Real;
begin
    case f.tipo of
        triangulo : area := 0.5 * f.baseT * f.alturaT;
        cuadrado  : area := f.lado * f.lado;
        rectangulo: area := f.baseR * f.alturaR
    end
end;

procedure sumasAreas(figs : Figuras; var sumaT, sumaC, sumaR : Real);
var i : integer;
begin
    sumaT := 0; sumaC := 0; sumaR := 0;
    for i := 1 to figs.tope do
        begin
            case figs.elementos[i].tipo of
                triangulo : sumaT := sumaT + area(figs.elementos[i]);
                cuadrado  : sumaC := sumaC + area(figs.elementos[i]);
                rectangulo: sumaR := sumaR + area(figs.elementos[i]);
            end
        end
    end
end;
```

b)

```
function esMayor(figs : Figuras; f: Figura) : Boolean;

var i : integer;
    areaF : real;
    hayMayorOIgual : Boolean;

begin
    (* inicializacin *)
    areaF := area(f);
    i := 1;
    hayMayorOIgual := FALSE;

    (* busco alguna figura del mismo tipo que f con area mayor o igual *)
    while (i <= figs.tope) and not hayMayorOIgual do
        begin
            if (figs.elementos[i].tipo = f.tipo) and
                (area(figs.elementos[i]) >= areaF) then
                hayMayorOIgual := TRUE
            else
                i := i + 1
            end;
        end;

    esMayor := not hayMayorOIgual
end;
```

## Ejercicio 2

```
procedure moverAlFinal(VAR l : Lista);
var p : Lista;
begin
    if (l <> nil) and (l^.sig <> nil) then
    begin
        p := l^.sig;
        while p^.sig <> nil do
            p := p^.sig;
        end;
        p^.sig := l;
        l := l^.sig;
        p^.sig^.sig := nil;
    end;
end;
```

## Ejercicio 3

La primera columna es el dato de entrada.

0	T	21	38	21	F	63
1	T	9	18	9	F	27
2	F	19	34	19	T	57
3	F	7	16	7	T	21
4	T	18	32	18	F	54
5	T	6	14	6	F	18
6	F	16	30	16	T	48
7	F	4	10	4	T	12
8	T	15	28	15	F	45
9	T	3	8	3	F	9

## Ejercicio 4

```
function areaCurva(a, b : Real; n : Integer) : Real;
var base, acum: real;
    i : integer;
begin
    base := (b - a) / n;
    acum := 0;
    for i := 1 to n do
    begin
        acum := acum + f(a);
        a := a + base;
    end;
    areaCurva := acum * base;
end;
```