

## EJERCICIO 1

Dado el siguiente programa, escribir cuál será su salida cuando la variable x se carga de la entrada estándar con el último dígito de su CI (antes del dígito verificador). Por ejemplo, si su CI es 1.234.567-8, el último dígito es 7:

<pre><b>Program</b> Febrero;  <b>var</b> x, y, z: integer;  <b>Procedure</b> uno (a : integer; <b>var</b> b : integer);   <b>Function</b> tres (a, b : integer) : integer;     <b>var</b> i : integer;     <b>begin</b>       <b>for</b> i := 1 <b>to</b> 3 <b>do</b>         a := a + b;       tres := a     <b>end</b>;   <b>begin</b>     b := tres ( b, b);     a := x * 3;     writeln (a, b);     z := a + 3   <b>end</b>;</pre>	<pre><b>Function</b> dos (b : boolean): integer; <b>begin</b>   <b>if</b> b <b>then</b>     dos := 5   <b>else</b>     dos := 3   <b>end</b>;  <b>begin</b>   readln(x);   y := dos ( x &gt; 5 );   z := dos ( y &gt; 3 );   uno (x, y);   writeln(x, y, z) <b>end</b>.</pre>
--	---

## EJERCICIO 2

Dadas las siguientes declaraciones:

```
const MAX = ...; (* entero mayor que 0 *)
type Numeros = array [1 .. MAX] of integer;
```

Escribir la siguiente función:

```
function repetidos(nums : Numeros) : boolean;
```

que devuelve TRUE si el arreglo *nums* contiene valores repetidos y FALSE en caso contrario.

### EJERCICIO 3

Sean las siguientes declaraciones:

```
CONST
  MAX_EST = ...; {valor mayor que 0}
  CANT_RESP = ...; {valor mayor que 0}
TYPE
  TipoPrueba = (escrita,oral);
  Respuestas = array[1..CANT_RESP] of Char;
  Prueba = record
      ci: Integer;
      case tipo : TipoPrueba of
          escrita : (resp: Respuestas);
          oral : (aprobo: Boolean);
      end;
  Pruebas = record
      pruebas: array[1 .. MAX_EST] of Prueba;
      tope: 0..MAX_EST;
  end;
```

En el arreglo de tipo Pruebas se almacenan las pruebas realizadas por alumnos de un curso, que pueden ser orales o escritas. Si son escritas se guardan las respuestas del estudiante en el arreglo **resp**. Se asume que las respuestas posibles son 'a', 'b', 'c', 'd' o el carácter 'v' que indica respuesta vacía.

Implementar la función:

```
function puntaje(pc: Pruebas; ci: Integer; correctas: Respuestas):Real;
```

que retorna cuántos puntos sacó el alumno de cédula ci en la prueba escrita.

- En el parámetro **correctas** se tienen las respuestas correctas para cada pregunta.
- Cada respuesta correcta del alumno suma 2 puntos, las incorrectas restan 0.25, las vacías no afectan el puntaje.
- Si no existe un alumno con la cédula dada, o existe pero realizó una prueba oral, la función retorna el valor especial -1000.

Cada alumno aparece a lo sumo una vez en el arreglo pruebas.

## EJERCICIO 4

Dadas las siguientes declaraciones:

```
TYPE
  ListaInt = ^Celda;
  Celda = RECORD
    info: Integer;
    sig: ListaInt
  END;
```

Escribir el procedimiento:

```
procedure p (var l : ListaInt; n : Integer);
```

que elimina las repeticiones consecutivas del entero n, dejando una única ocurrencia de n en l. Asuma que todas las ocurrencias de un número n aparecen en forma consecutiva en la lista l.

### Solución

```
procedure p (var l : ListaInt; n : Integer);

var ls, aux : ListaInt;

begin
  ls := l;
  while (ls <> nil) and (ls^.info <> n) do
    ls := ls^.sig;
  if (ls <> nil) then
    begin
      while (ls^.sig <> nil) and (ls^.sig^.info = n) do
        begin
          aux := ls^.sig;
          ls^.sig := aux^.sig;
          dispose(aux)
        end
      end
    end
  end;
```