

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 (el dígito verificador). Por ejemplo, si su CI es 1.234.567-8, el último dígito es 8:

<pre>program marzo; var x, y, z: integer; procedure proc (a : integer; var b : integer); function fun (a, b : integer) : integer; begin a := a * 2; fun := a + b end; var z : integer; begin (* proc *) z := a + 1; b := fun(a, b); y := z + b; writeln (a, b, z); end; (* proc *)</pre>	<pre> begin (* principal *) readln(x); y := x + 2; z := y + 1; proc (x, y); writeln(x, y, z) end.</pre>
--	--

EJERCICIO 2

Considere los tipos:

```
type
  TIndice = 1..MAX;
  TA      = array [TIndice] of integer;
  TOrden  = (creciente, decreciente, desordenado);
```

Considere que $MAX > 1$ y que todos los elementos del arreglo son diferentes.

Escriba la función:

```
function ordenado (A : TA): TOrden;
```

que determina si los elementos del arreglo A están ordenados en forma creciente, están ordenados en forma decreciente, o están desordenados.

EJERCICIO 3

Se tienen los tipos:

```
type
  PosibleEntero = record
    case vacio : boolean of
      true : ();
      false: (valor : integer);
    end;

  TArreglo = array [1..MAX] of PosibleEntero;

  TArregloConTope = record
    elementos : array [1..MAX] of integer;
    tope      : 0 .. MAX
  end;
```

Escriba el procedimiento:

```
procedure Defrag(arreglo : TArreglo; var resultado : TArregloConTope);
```

Este procedimiento copia todos los enteros que están en arreglo (en las celdas cuyo campo vacio es false) a la estructura resultado. Las celdas de arreglo cuyo campo vacio es true se ignoran. Los enteros en resultado deben aparecer en el mismo orden que aparecían en arreglo.

Ejemplos (con MAX=10):

- arreglo = [(),3,5,(),8,2,3,(),7,4] resultado = (elementos=[3,5,8,2,3,7,4,...], tope = 7)
- arreglo = [(),(),(),(),(),(),(),(),(),()] resultado = (elementos=[...], tope = 0)
- arreglo = [10,3,5,2,8,2,3,19,7,4] resultado = (elementos=[10,3,5,2,8,2,3,19,7,4], tope = 10)

EJERCICIO 4

Dadas las siguientes declaraciones:

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

Escribir la función:

```
function deHastaSin (ini, fin : integer; sin : ListaInt) : ListaInt;
```

que retorna la lista que contiene los enteros en el rango entre ini y fin que no pertenecen a la lista sin. La lista resultante debe quedar ordenada de menor a mayor.

Ejemplos:

- si ini = 1, fin = 4 y sin = (), retorna (1,2,3,4).
- si ini = 1, fin = 4 y sin = (6,5), retorna (1,2,3,4).
- si ini = 1, fin = 4 y sin = (6,2,5), retorna (1,3,4).
- si ini = 1, fin = 4 y sin = (6,2,5,3,4,1), retorna ().
- si ini = 1, fin = 1 y sin = (6,5), retorna (1).
- si ini = 4, fin = 1 y sin = (6,5), retorna ().