

Leer con atención

- Todos los programas o fragmentos de programas deben ser escritos en el lenguaje **Pascal** tal como fue dado en el curso. A grandes rasgos este es el Pascal estándar con algunos agregados, a saber:
- Utilización de **else** en la instrucción **case**.
- Evaluación por circuito corto de las operaciones booleanas (**and** y **or**).
- En todos los problemas se evaluará, además de la lógica correcta, la utilización de un buen estilo de programación de acuerdo a los criterios impartidos en el curso. De esta manera se restarán puntos entre otros conceptos por: mala o nula indentación, mala utilización de las estructuras de control, código confuso e innecesariamente largo, programas ineficientes, utilización de variables globales, pasaje incorrecto de parámetros, etc.
- No obstante, por razones prácticas no exigimos que incluya comentarios en los códigos que escriba en la prueba.
- Escriba su nombre completo y cédula en todas las hojas.
- Numere todas las hojas y escriba la cantidad total de hojas entregadas en la primera hoja.
- Escriba de un solo lado de la hoja y comience cada ejercicio en una nueva hoja.

Ejercicio 1

Dadas las siguientes declaraciones que permiten registrar información sobre un conjunto de artículos:

```
CONST
    MAX_ART = ...;
    MAX_DIG = ...;

TYPE
    Comparacion = (menor, igual, mayor);

    Codigo = Array[1..MAX_DIG] of 0..9;

    ListaAutores = ^CeldaAutores;

    CeldaAutores = RECORD
        codAutor : Codigo;
        siguiente : ListaAutores
    END;

    Articulo = RECORD
        codArticulo : Codigo;
        autores : ListaAutores
        CASE eval : Boolean OF
            TRUE : ( evaluacion : Integer);
            FALSE : ()
        END;

    Articulos = RECORD
        elems : array [1 .. MAX_ART] of Articulo;
        tope : 0 .. MAX_ART
    END;
```

Parte a)

Escribir la función `compararCodigos` que, dados dos códigos, determina si el primero es menor, igual o mayor que el segundo.

```
FUNCTION compararCodigos(cod1, cod2 : Codigo) : Comparacion;
```

Parte b)

Escribir el procedimiento `menorArtEval` que, dado un conjunto de artículos, encuentra el artículo de menor código que haya sido evaluado. El procedimiento devuelve el código y la evaluación del artículo encontrado o una indicación de que no se encontró ningún artículo evaluado. Para devolver el resultado se utiliza el tipo `Resultado`, que se declara a continuación:

```
TYPE
    Resultado = RECORD
        CASE encontrado : Boolean OF
            TRUE : (cod : Codigo; eval : Integer);
            FALSE : ()
        END;
```

```
PROCEDURE menorArtEval(arts : Articulos, var result : Resultado);
```

Parte c)

Escribir el procedimiento enteroCodigo que, dado un entero, genera el código correspondiente.

```
PROCEDURE enteroCodigo(n : Integer; var cod : Codigo);
```

Observaciones:

- Si el entero tiene menos dígitos que MAX_DIG, el arreglo cod debe completarse con ceros a la izquierda.
- Si el entero tiene más dígitos que MAX_DIG, se completa cod con los MAX_DIG dígitos de más a la derecha.

Ejemplos para MAX_DIG = 4:

| n | cod |
|----------|---------------|
| 1234 | 1 2 3 4 |
| 123 | 0 1 2 3 |
| 12345678 | 5 6 7 8 |

Parte d)

Escribir el procedimiento agregarAutor que, dado un código de autor y un artículo, agrega el autor al final de la lista de autores del artículo.

```
PROCEDURE agregarAutor(autor : Codigo; var art : Articulo);
```

Ejercicio 2

Dado el siguiente programa, determinar su salida cuando se asigna a la variable b el último dígito de su CI. Por ejemplo, si su CI es 1.368.492-0, el último dígito es 2.

```
PROGRAM alcance;

CONST a = 5;
VAR b : Integer;

PROCEDURE proc(m: Integer; var n: Integer);
VAR a: Integer;

FUNCTION func(x, y : Integer) : Boolean;
VAR h : integer;
BEGIN
    h:= x+y;
    writeln(h);
    func := (h mod 2) = 1
END;

BEGIN
    m := n+2;
    a := m+n;
    writeln(a);
    writeln(m);
    if func(m, a) then n:= m else n:= a
END;

BEGIN
    readln(b);
    proc(a, b);
    writeln(b)
END.
```