

Leer con atención

- Todos los códigos deben ser escritos en el lenguaje **Pascal** tal como fue dado en el curso. A grandes rasgos, este es el Pascal estándar con los siguientes agregados: utilización de `else` en la instrucción `case` y 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.**
- **Escriba de un solo lado de la hoja y comience cada ejercicio en una nueva hoja.**

Ejercicio 1

Dada la siguiente declaración de tipos, que representa una matriz de números enteros:

```
const MAX = ...; (* NUMERO MAYOR A CERO *)  
type matriz = array[1 .. MAX, 1 .. MAX] of integer;
```

Implemente las siguientes funciones:

Parte a)

```
function sumatotal(m: matriz) : integer;
```

Que devuelve la suma de todos los números de la matriz `m`.

Parte b)

```
function primerPositivo(m: matriz) : integer;
```

Que devuelve el primer elemento positivo de la matriz `m`. Si no existe ningún valor positivo en la matriz, retorna `-1`.

Definimos una relación de orden entre las posiciones según las siguientes reglas:

$(i,j) < (i+k, r)$

$(i,j) < (i, j+k)$

Para ambos casos se supone que los valores para fila y columna están en el rango apropiado y que $k > 0$.

Por ejemplo, si `MAX` es 3, el resultado de llamar a `primerPositivo` con la siguiente matriz es 8, que se encuentra en la posición (1,3):

-1	-6	8
2	-5	4
10	3	-48

Ejercicio 2

Dadas las siguientes declaraciones:

```
const MaxDig = ...; (* NUMERO MAYOR A CERO *)
type ArrChars = record
    elems : array [1 .. MaxDig] of Char;
    tope : 0 .. MaxDig
end;

Resultado = record
    case numValido : Boolean of
        true: (suma : Integer);
        false: (car: Char;
                ind : Integer)
    end;
```

Escribir el procedimiento:

```
procedure suma0Error(arr : ArrChars; var res : Resultado);
```

Que, dado un arreglo con tope de caracteres, devuelve en el parámetro **res** el resultado siguiente:

- Si todos los caracteres son dígitos, devuelve el campo **numValido** en TRUE y en el campo **suma** la suma de todos los dígitos. Si el arreglo con tope está vacío, se devuelve el valor 0 como suma.
- Si existe algún carácter que no sea un dígito, devuelve el campo **numValido** en FALSE, en el campo **car** el primer carácter no dígito encontrado, y en el campo **ind** el índice de dicho carácter.

Ejemplos (suponiendo que **MaxDig** es 5):

arr	res
elems=['2','7','8',?,?], tope = 3	numValido = true, suma = 17
elems=[?,?,?,?,?], tope = 0	numValido = true, suma = 0
elems=['2','=', '8', '+',?], tope = 4	numValido = false, char = '=', ind = 2

Ejercicio 3

Dada la siguiente declaración de tipos:

```
type Lista = ^Nodo;
    Nodo = record
        elem : Integer;
        sig : Lista
    end;
```

Escribir un procedimiento que, dada una lista, inserte al final una nueva celda que contenga la suma de todos sus elementos. Realice una única recorrida de la lista.

Si la lista es vacía, debe insertar el elemento 0.

```
procedure sumaTotal(var l :Lista);
```

Ejemplos:

l inicial	l resultante
[2, 7, 3]	[2, 7, 3, 12]
[2]	[2, 2]
[]	[0]

Ejercicio 4

Determinar la salida del siguiente programa, cuando se le da como entrada el último dígito antes del guión de su número de cédula. Por ejemplo, si su cédula es 1234567-8, la entrada será 7.

```
program g20;
var a,b,c: integer;

function merkel(a: integer) : integer;
var b: integer;
  procedure macron(a: integer; var b: integer);
  begin
    b := a + c;
  end;
begin
  macron(a,b);
  c := b;
  writeln(c);
  merkel := b - a;
end;

function trump(var a: integer; b: integer) : integer;
function macri() : integer;
begin
  macri := -1;
end;
begin
  a := c + macri();
  writeln(a);
  trump := b - c;
end;

begin
  readln(a);
  b := 9 - a;
  c := b div 2;
  writeln(merkel(trump(a,b)))
end.
```