



Examen teórico (80% de la calificación total)

Duración total: 3 horas

Ejercicio 1 [3,5 puntos]

La conjetura¹ capicúa plantea el siguiente algoritmo para obtener números capicúas a partir de un número natural: "Al número dado se le suma el que resulta de invertir el orden de sus cifras; se repite el proceso las veces necesarias hasta obtener un capicúa". La conjetura dice que aplicando el proceso descrito anteriormente a un número natural, se obtiene un número capicúa en un número finito de pasos.

Ejemplo: Dado el número 96:

$$96 + 69 = 165; 165 + 561 = 726; 726 + 627 = 1353; 1353 + 3531 = 4884$$

Si hubiéramos partido del número 89, según el proceso anterior, después de 24 pasos se llega al capicúa 8.813.200.023.188

Se pide escribir un programa PASCAL que, dado un número natural introducido interactivamente por el usuario, obtenga un número capicúa de acuerdo al algoritmo presentado. Dado que el número capicúa puede exceder el rango de los enteros se propone utilizar las definiciones que se muestran a continuación.

```
CONST MAXCIFRAS= 1000;  
TYPE tpDigito = 0..9; tpIndice = 1..MAXCIFRAS;  
      tpNumero = record  
                    nDig : 0..MAXCIFRAS;  
                    vDig : array [tpIndice] of tpDigito  
                end;
```

Ejercicio 2 [3 puntos]

Una matriz cuadrada de enteros de dimensiones N*N (máximo N=100) se define como de "máximos crecientes continuos y uniformemente distribuidos" (MCCUD) si cumple:

1. Todos los valores de una fila son distintos entre sí,
2. $max_fila_1 < max_fila_2 < max_fila_3 < \dots < max_fila_N$
donde, max_fila_i , representa el mayor valor entero encontrado en la fila i-ésima de la matriz.
3. $\forall i, (max_fila_{i+1} - max_fila_i = constante)$, es decir, la diferencia entre los máximos de dos filas consecutivas es constante.
4. En cada columna de la matriz sólo puede aparecer el valor máximo de una de las filas.

En la figura se muestra una matriz MCCUD.

5	6	2	-4
6	7	8	3
10	4	-5	9
4	-7	1	12

Utilizando las siguientes declaraciones de datos,

```
CONST MAX = 100;  
TYPE tpIndice = 1..MAX;  
      tpMatriz = record  
                    dimension: tpIndice;  
                    valores: array [tpIndice, tpIndice] of integer  
                end;
```

Se pide escribir una función PASCAL que, dada una matriz cuadrada de tipo *tpMatriz*, determine si es una matriz MCCUD:

```
function esMatrizMCCUD (?: matriz: tpMatriz): boolean;
```

¹ Conjetura: Es el juicio que se forma de las cosas o hechos por indicios y observaciones.



Ejercicio 3 [3,5 puntos]

Leotardo che Pinchi, envidioso coetáneo de un famoso sabio del Renacimiento italiano, decidió crear un sistema propio de codificación de sus escritos que se basaba en escribir las palabras en sentido inverso (de derecha a izquierda) pero manteniendo el sentido de lectura de la frases de izquierda a derecha.

Ejemplo:

Los hombres que no buscan mas que
riquezas y placeres materiales
están completamente privados de
la riqueza de la sabiduría.
Si el alma es mas digna que el
cuerpo, la riqueza del alma es
mas importante que la del cuerpo.

'original.txt'

soL serbmoh euq on nacsab sam euq
sazeuqir y serecalp selairetam
natse etnematelpmoc sodavirp ed
al azeuqir ed al .airudibas
iS le amla se sam angid euq le
,opreuc al azeuqir led amla se
sam etnatropmi euq al led .opreuc

'codificado.txt'

Tarea: Desarrollar un programa PASCAL que permita realizar una codificación según el método de Leotardo.
Para ello,

- 1.- Definir las estructuras de datos necesarias. Se supondrá que no hay palabras de más de 30 letras.
- 2.- Desarrollar el procedimiento

```
procedure Codificar(?? nomFOrig: tpNomFich; ?? nomFCodif: tpNomFich);  
{A partir del fichero de texto con nombre nomFOrig, crea el fichero de texto con nombre nomFCodif cuyo contenido  
ha sido obtenido según el método de codificación descrito en el ejemplo}
```

- 3.- Escribir el programa principal que solicite al usuario los nombres de los ficheros original y resultado, y efectúe la codificación descrita (en el ejemplo, el fichero 'original.txt' se codifica como 'codificado.txt').

Notas: Se supone que el fichero a codificar sólo contiene letras, espacios en blanco, comas y puntos. Comas y puntos aparecen siempre inmediatamente seguidos a las palabras que les preceden, por lo que se les considera parte de dicha palabra. Toda palabra está en una sola línea.

Deben respetarse los espacios en blanco que hay antes, después y entre las palabras del fichero original.