

Ejercicio 1 [3,5 puntos]

Publicado por el Instituto Geográfico Nacional, se ha obtenido de internet el fichero de texto 'provincias2012.txt' con los datos demográficos de población y superficie de las provincias españolas. En su primera línea contiene una cabecera con los nombres de los datos que aparecen en cada columna del fichero (esta línea es irrelevante para el problema). Cada una de las líneas siguientes contiene el código de la provincia (entero), número de habitantes (entero), superficie en km² (entero) y el nombre de la provincia (cadena de caracteres). Los datos están separados por espacios.

Se pide:

- 1) Definir el tipo de datos `tpTablaDatos` y todos los elementos necesarios para almacenar en memoria los datos suministrados y resolver los procedimientos del apartado siguiente.
- 2) Implementar los siguientes procedimientos:

```
procedure cargarTabla(??? fTxt: tpFichDat; ??? tblDat: tpTablaDatos);  
{ devuelve en tblDat la tabla con los datos de las provincias que hay en el fichero de texto. El formato del fichero  
es el descrito en el enunciado del problema y en el ejemplo }
```

```
procedure mostrarDensPobl(??? CP: tpCodP; ??? tblDat: tpTablaDatos);  
{ muestra por pantalla la densidad de población de la provincia con código CP a partir de la información  
almacenada en la tabla tblDat. El formato de la salida será el del ejemplo que se muestra a continuación }
```

- 3) Desarrollar un programa que solicite un código de provincia y muestre por pantalla la densidad de población de dicha provincia con el formato utilizado en el ejemplo.

Ejemplo de ejecución:

código INE de provincia: 50
densidad de Población de Zaragoza: 973325/17274 = 56.35 hab/km2

fichero 'provincias2012.txt':

Cod. INE	Población	Superf. (km2)	Provincia
01	319227	3037	Araba / Álava
02	402318	14924	Albacete
03	1934127	5817	Alicante / Alacant
• • •			
50	973325	17274	Zaragoza
51	82376	19	Ceuta
52	78476	13	Melilla

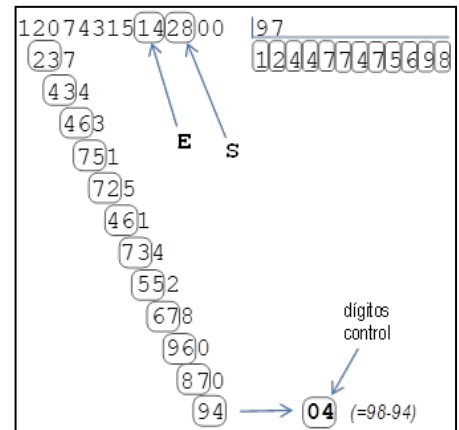
Nota: Se puede suponer que el contenido del fichero responde exactamente al formato especificado; es decir, no hay errores a la hora de crear el fichero.

Ejercicio 2 [3 puntos]

A partir de mañana entra en vigor la normativa europea por la que todas las transacciones bancarias deben incluir el código IBAN (International Bank Account Number) de las cuentas bancarias involucradas. El código IBAN consta de un máximo de 34 caracteres alfanuméricos: los dos primeros son letras que identifican el país (ES para España), los dos siguientes son dígitos de control que permiten validar el IBAN, y los restantes caracteres son el número de la cuenta bancaria, que pueden ser dígitos numéricos o letras mayúsculas de la A a la Z. En España, el número de cuenta consta de 20 dígitos numéricos. El procedimiento de obtención de los dos dígitos de control es el siguiente:

- 1) Se colocan a continuación del número de cuenta las dos letras del país y dos ceros. Después se sustituyen cada una de las letras que hay, por el número de dos dígitos a que equivalen (A →10, B →11, ..., Z →35)
- 2) Se calcula el resto de dividir el número anterior por 97
- 3) Los dos dígitos de control son el resultado de restar a 98 el resto obtenido.

Dado que el número del que se quiere calcular el resto es muy grande y no es representable con ninguno de los tipos enteros habituales, hay que realizar la división tal y como se hace "a mano"; es decir, añadiendo un dígito del dividendo en cada paso y calculando un nuevo resto. En la figura se muestra la obtención de los dos dígitos de control (04) correspondientes al número de cuenta 12074315 de España. El código IBAN de dicha cuenta es: ES0412074315



Se pide: Escribir un programa PASCAL que lea del teclado una línea con un número de cuenta, seguido de las dos letras que identifican el país, y muestre por la salida estándar los dos dígitos de control del código IBAN.

Ejemplo de ejecución:

número de cuenta y letras del país: 12074315ES
 dígitos de control: 04

Nota: Para resolver el problema no se necesitan tipos de datos estructurados.

Ejercicio 3 [3,5 puntos]

En un reactor químico hay dos sondas de temperatura, cada una de las cuales produce una medida de la temperatura del interior del reactor cada cierto tiempo (un minuto, o más). Cada medición se guarda en un registro de tipo `tp_medida` que contiene el tiempo transcurrido desde el inicio de las mediciones, expresado en minutos (entero) y el valor de la temperatura (real). Las mediciones efectuadas por cada sonda se van guardando en un fichero de registros de tipo `tp_medida`. El instante en que se inicia la medición del tiempo es el mismo para las dos sondas.

Se pide:

- 1) Definir los tipos de datos `tp_medida` y `tpFich_med`, y todo lo necesario para resolver el procedimiento del apartado siguiente.
- 2) Desarrollar el siguiente procedimiento:

```
procedure consolidarMedidas(??? fS1,fS2: tpFich_med; ??? fMC: tpFich_med);
{ devuelve en el fichero fMC la consolidación de las mediciones que hay en los ficheros fS1 y fS2 }
```

Para realizar la consolidación de las medidas de las dos sondas en un único fichero se tendrán en cuenta las siguientes especificaciones:

- a) Los parámetros `fS1` y `fS2` contienen las mediciones de la sondas 1 y 2, respectivamente, ordenadas por tiempo (minutos transcurridos desde el inicio).
- b) El procedimiento devuelve a través de `fMC` un fichero de mediciones, también ordenado por tiempo, que integra las mediciones de `fS1` y `fS2`.
- c) Si dos mediciones se han realizado en el mismo instante de tiempo (los minutos transcurridos para la sonda 1 son los mismos que para la sonda 2), se incluirá en la lista final una única medición cuyo valor sea la media de las dos mediciones de temperatura (a lo sumo puede haber dos mediciones con el mismo tiempo, una de cada sonda). Ejemplo:

	tiempo(min.)	temperatura		tiempo(min.)	temperatura		tiempo(min.)	temperatura
fS1	235	84.3	fS2	235	84.5	fMC	235	84.4
	237	84.0		238	84.2		237	84.0
	239	83.7		241	83.9		238	84.2
	241	83.5					239	83.7
	...						241	83.7