

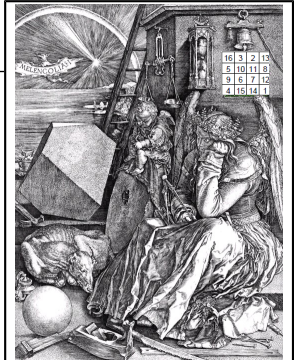
### Ejercicio 1 [3 puntos]

Un “cuadrado mágico” de dimensión  $n$  es una tabla de  $n \times n$  números que contiene todos los números naturales del 1 al  $n^2$ , dispuestos en  $n$  filas de  $n$  números cada una, de forma que la suma de los números de cualquier fila, columna y diagonal es siempre la misma. En la fig. 1 se muestra el cuadrado mágico de dimensión 4 que aparece en el grabado “melancolía” realizado por Durero en 1514. Es fácil comprobar que la suma de cualquier fila, columna o diagonal es 34.

Se pide desarrollar en PASCAL las siguientes funciones (previamente se realizarán las declaraciones de tipos de datos necesarias para trabajar con cuadrados mágicos de hasta  $10 \times 10$ ):

```
function estanTodos(CM: tpCuadMagic; n: tpDim): boolean;
{ devuelve true si el cuadrado mágico CM, de dimensión n, contiene todos los naturales de
  1 a  $n^2$  y false en caso contrario}

function esMagico(CM: tpCuadMagic; n: tpDim): boolean;
{ devuelve true si CM es un cuadrado mágico de dimensión n (están todos los números de
  1 a  $n^2$  y es igual la suma de los números de cada fila, columna o diagonal) }
```



16	3	2	13
5	10	11	8
9	6	7	12
4	15	14	1

fig. 1 Grabado “Melancolía”, y detalle del “cuadrado mágico”

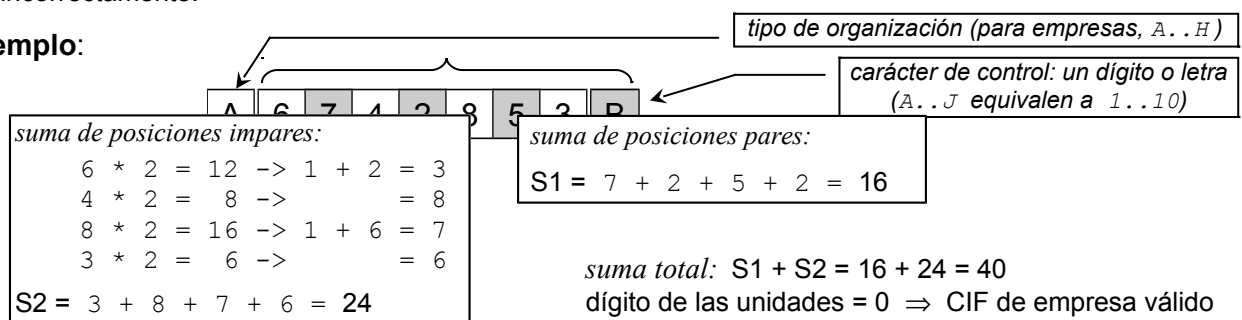
### Ejercicio 2 [3 puntos]

El CIF (Código de Identificación Fiscal) es un elemento de identificación administrativa de organizaciones que consta siempre de 9 caracteres. El primero de ellos es una letra que sirve para identificar el tipo de organización (en el caso de una empresa es una letra de la A a la H), que va seguida de 7 dígitos decimales y de un último carácter, que es un dígito para las empresas españolas, o una letra de la A a la J para las empresas extranjeras.

La verificación se realiza del siguiente modo:

- 1) Se suman todos los dígitos que están en posición par (7, 2, 5 y B en el ejemplo). Si el último carácter es una letra, se sustituye previamente por su valor numérico ( $A \rightarrow 1$ ,  $B \rightarrow 2$ , ...,  $J \rightarrow 10$ ).
- 2) Se suman todos los dígitos que están en posición impar multiplicados por 2 (6, 4, 8 y 3 en el ejemplo), pero si el resultado de multiplicar el dígito por 2 es mayor que 9, se suman sus dígitos.
- 3) Si el dígito de las unidades de la suma de las cantidades anteriores es distinto de 0, el CIF está construido incorrectamente.

**Ejemplo:**



Se pide escribir un programa en Pascal que lea del teclado una secuencia de caracteres terminada con un espacio y muestre por pantalla si es, o no, un CIF válido de una empresa.

**Observación:** Se deben procesar los caracteres a medida que son leídos (**NO se deberán utilizar estructuras de datos de tipo array o string**). Se supondrá que las letras están siempre en mayúscula.

### Ejercicio 3 [4 puntos]

La compañía de telecomunicaciones con la que nuestra empresa tiene contratadas las comunicaciones telefónicas nos suministra todos los meses un fichero de texto, 'llamadas.txt', con información acerca de las llamadas realizadas desde cada una de las extensiones telefónicas que hay en la empresa. En cada línea de dicho fichero aparece la información de una llamada, que consiste en:

- el número de extensión, que es de 3 dígitos,
- la duración de la llamada en segundos,
- el importe de la llamada en euros.

456	180	1.5
525	55	0.65
456	245	0.93
516	45	0.09
456	237	0.45
516	120	1.2

*ejemplo de 'llamadas.txt'*

Para controlar el gasto telefónico de cada extensión, se dispone de un fichero de registros, 'acumulado.dat', en el que se guarda el consumo anual total de cada extensión, pero sólo de las extensiones que han sido usadas para hacer llamadas (su tiempo o importe son > 0).

#### Se pide:

1) Definir todos los tipos de datos necesarios para resolver el problema planteado.

2) Desarrollar el siguiente procedimiento:

```
procedure leerLlamada ( ?? f : text; ?? datosLlamada : tpDatosLlamada);  
{ lee la información de la llamada que hay en la línea actual del fichero de texto f y la devuelve a través del  
registro datosLlamada, dejando el fichero preparado para leer la siguiente llamada }
```

3) Escribir un programa PASCAL que, a partir de la información que hay en el fichero 'llamadas.txt',

- muestre por pantalla, para cada extensión, el número total de llamadas realizadas, el importe total y la duración total (en horas, minutos y segundos) y
- actualice el fichero 'acumulado.dat' con los datos obtenidos de 'llamadas.txt'.

Como ejemplo, a continuación se muestra lo que sacaría por pantalla dicho programa para el fichero de llamadas anterior, y el contenido del fichero 'acumulado.dat' (si es la primera vez que se actualiza):

extension	llamadas	importe	duracion
-----	-----	-----	-----
456	3	2.88	0h 11m 2s
516	2	1.29	0h 2m 45s
525	1	0.65	0h 0m 55s

*Resultado en pantalla de la ejecución del programa con el ejemplo anterior.*

456	662	2.88
516	165	1.29
525	55	0.65

*Contenido de 'acumulado.dat' tras actualizarlo, por primera vez, con los datos del ejemplo.*

**Nota:** Se podrá suponer que toda la información de todas las posibles extensiones se puede almacenar en la memoria del computador.