



**Universidad
Zaragoza**

Proyecto

Fundamentos de Informática

Grado en Ingeniería de Tecnologías Industriales

Curso 2015-2016

Universidad de Zaragoza

Escuela de Ingeniería y Arquitectura

Departamento de Informática e Ingeniería de Sistemas

Área de Lenguajes y Sistemas Informáticos

OBJETIVOS

Los objetivos de esta práctica son los siguientes:

- Resolver ejercicios a nivel de los que aparecen en el examen, y poder afrontarlos y resolverlos con tiempo y calma.
- Introducirse en el manejo de ficheros.

1. Ficheros y César

En la práctica 3 construiste un programa que calculaba el cifrado César de una frase, dado un desplazamiento. Para descifrar un texto cifrado, es necesario conocer el desplazamiento con el que se codificó para recuperar la letra. No obstante, si no se conoce dicho desplazamiento, se puede deducir. Si el texto está en español se sabe que, estadísticamente, la letra que más se utiliza es la letra “e” (tanto mayúscula como minúscula). Por tanto, se puede asumir que la letra que más aparece en el texto cifrado es la representación de la “e” y con ello deducir el desplazamiento.

Escriba un programa PASCAL que descifre un fichero de texto (cuyo nombre es introducido por el usuario) utilizando la técnica anteriormente expuesta y lo muestre por pantalla, junto con la letra que más se repite y el desplazamiento usado, tal y como se muestra en el siguiente ejemplo:

<i>Contenido del fichero 'texto.txt'</i>	<i>Salida del programa</i>
Kr sgy hxkbbk jkyvoyzk ky kr zktkx ktikxxgjuy ruy sktkyzkxky jkr jkyzotu.	Introduce nombre de fichero: texto.txt La letra que mas aparece es la: k El desplazamiento estimado es de: 6 texto.txt descifrado es: El mas breve despiste es el tener encerrados los menesteres del destino.

Para poder descifrar el texto, debes haber deducido primero el desplazamiento y, para ello, debes haber buscado la letra que más aparece, y compararla con la “e” para calcular el desplazamiento. Puede que te sea útil recordar alguno de los ejercicios de la práctica 4.

Nombre del fichero: **descifrado.pas**

2. Ficheros de imagen

En la práctica 4 desarrollaste un programa que te permitía aplicar el filtro de caja a una imagen representada en forma de matriz. En esta práctica deberás transformar ese programa para que, en lugar de pedir los datos píxel a píxel al usuario, los lea de un fichero de tipo imagen. En este caso utilizaremos el formato ‘pgm’, que es un formato de fichero de texto con la siguiente estructura:

- Primero los caracteres “P2”, en una línea.
- Después un comentario en una línea, que empieza por el carácter “#” y que puede ser ignorado. A partir de este comentario los saltos de línea aparecen indistintamente.
- Después la anchura y la altura de la imagen, como números enteros.
- Después el valor máximo de la imagen (el valor del píxel de color blanco), al que llamaremos p_w . Cuando leamos el fichero deberemos guardarnos este valor para normalizar cada píxel leído con respecto a este. Cuando escribamos el fichero, nuestro valor máximo siempre será 255 y eso será lo que escribamos en esta posición del fichero.

- Después los valores de todos los píxeles consecutivos. Al escribirlos, se escribe el número tal cual, mientras que al leerlos deberás dividir el valor leído por p_w y multiplicarlo por 255 para que el color blanco siempre esté a 255.

Ejemplo de fichero:

```
P2
# imagen.pgm
24 7
15
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 3 3 3 3 0 0 7 7 7 7 0 0 11 11 11 11 0 0 15 15 15 15 0
0 3 0 0 0 0 0 7 0 0 0 0 0 11 0 0 0 0 0 15 0 0 15 0
0 3 3 3 0 0 0 7 7 7 0 0 0 11 11 11 0 0 0 15 15 15 15 0
0 3 0 0 0 0 0 7 0 0 0 0 0 11 0 0 0 0 0 15 0 0 0 0
0 3 0 0 0 0 0 7 7 7 7 0 0 11 11 11 11 0 0 15 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Sobre el programa “filtrocaja.pas” desarrollado en la práctica 4, añade los siguientes procedimientos:

Procedure CargarImagen(??? nombrefichero:string; ??? imagen: tpImagen);

{ Carga una imagen en formato pgm desde el fichero de nombre nombrefichero y la guarda en la estructura de datos de imagen }

Procedure GuardarImagen(??? nombrefichero:string; ??? imagen: tpImagen);

{ Guarda una imagen en formato pgm en el fichero de nombre nombrefichero desde la estructura de datos de imagen }

Después, cambia el programa principal para que en lugar de pedirle al usuario los datos píxel a píxel y mostrarlos píxel a píxel los lea de un fichero (de nombre introducido por el usuario) y los escriba en otro fichero (de nombre, también, introducido por el usuario). Adicionalmente, antes de finalizar, el programa mostrará el valor mínimo y máximo tanto de la imagen original como de la filtrada.

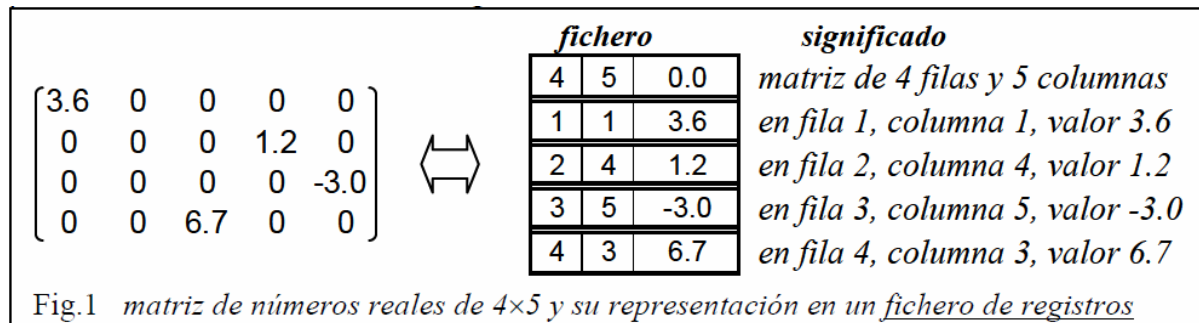
Ejemplo de interacción:

```
Fichero de origen:  imagen.pgm
Fichero de destino: filtrada.pgm
Mínimo y máximo de imagen.pgm:  0  255
Mínimo y máximo de filtrada.pgm:  0  198
```

Nombre del fichero: **filtrocajaficheros.pas**

3. Ficheros secuenciales

En la resolución de diversos problemas de ingeniería se utilizan **matrices cuasivacías** (MCV), que son aquellas que contienen un pequeño número de elementos no nulos. Un modo muy simple de describirlas consiste en indicar sus dimensiones y, a continuación, todos los valores no nulos junto con su posición (pueden estar en cualquier orden). Para almacenar de forma permanente la matriz se puede utilizar un fichero de registros, como se muestra a continuación:



Se pide desarrollar en PASCAL el siguiente procedimiento (previamente se realizarán todas las declaraciones de tipos de datos que sean necesarias).

procedure analizar_MCV(??? m: tpMCV);

{ lee la matriz cuasivacia m, almacenada en un fichero de registros, y muestra por pantalla:

- número de filas y columnas de la matriz
- las filas con todos sus elementos nulos, o la indicación de que ninguna fila es nula
- las columnas con todos sus elementos nulos, o la indicación de que ninguna columna es nula }

La ejecución de este procedimiento con la MCV de la figura, mostraría por pantalla:

Matriz de 4 x 5
Ninguna fila es nula
La columna 2 es nula

Observaciones:

- 1) Se puede suponer que la mayor dimensión de una MCV es de 4000 x 4000.
- 2) No es posible almacenar en memoria todos los elementos de la matriz.

Añade un programa principal que: i) cree un fichero secuencial que contenga una matriz cuasivacia pidiéndole los datos al usuario (primero se pedirá el nombre del fichero, a continuación el número de filas y de columnas, luego la cantidad de datos no nulos y finalmente los valores con el formato: fila columna valor); y ii) ejecute el procedimiento analizar_MCV.

Nombre del fichero: **mcv.pas**

PARA ENTREGAR

Se tienen que entregar los archivos de código fuente .pas pedidos en los ejercicios (“descifrado.pas”, “filtrocajaficheros.pas”, “mcv.pas”).

Todos estos archivos deberán ser incluidos en un único archivo, llamado “proyecto.tar” (sin espacios ni acentos, respetando mayúsculas y minúsculas). Ese archivo es el que deberás someter (y el único que deberás someter).

FECHA DE ENTREGA: LUNES 6 DE JUNIO