

Nota: Responde cada ejercicio en una hoja diferente. No utilices la misma hoja para dos (o tres) ejercicios.

Ejercicio 1

[3 puntos]

Un **anemómetro** ubicado en una estación meteorológica mide la velocidad del viento a intervalos regulares, guardando los datos en un fichero secuencial de números reales, de nombre 'velocidades.dat'. Debido a la variabilidad de la velocidad del viento, dichos datos son poco útiles para tareas de predicción meteorológica. Por ese motivo, se desean calcular las **medias móviles** (de tres elementos) del fichero, medida más robusta que sí que puede ser utilizada.

Se pide escribir un programa en Pascal que lea la secuencia de velocidades (números reales) del fichero 'velocidades.dat' y calcule las medias móviles de tres elementos, guardándolas en el fichero secuencial 'velocidadesfiltradas.dat'. La media móvil se calcula como la media de los tres elementos contiguos en la posición correspondiente. Por tanto, la primera media móvil es la media de los tres primeros datos del fichero original, la segunda media móvil es la media del segundo, tercero y cuarto y así sucesivamente. Si el fichero de entrada dispone de menos de tres elementos el fichero de salida estará vacío. El tamaño del fichero de entrada no es conocido a priori, ni existe un límite máximo de elementos.

Entrada: 'velocidades.dat'

Salida: 'velocidadesfiltradas.dat'

Ejemplo 1	10.5	11.5	12.5	12.0	8.5	11.5	12.0	11.0
Ejemplo 2	35.2	14.7	40.1			30.0		

Ejercicio 2

[3.5 puntos]

Para poder llevar la cuenta de cuántos personajes mueren en la serie Juego de Tronos, un fan procede a ver de nuevo las 6 temporadas existentes de la serie (hay 10 episodios por temporada) escribiendo en un fichero de texto llamado 'got.txt' cuántos personajes de las casas Stark, Lannister y Baratheon fallecen a lo largo la serie. Conforme va viendo los episodios, el fan añade una línea en el fichero para cada episodio, de forma que pueden faltar episodios en el fichero y además pueden aparecer desordenados. Cada línea contiene varios campos separados por un espacio: la temporada, el número de episodio dentro de la temporada, y por cada personaje importante que muere en ese episodio escribe en dicha línea una letra indicando a qué casa pertenecía: S para Stark, L para Lannister y B para Baratheon (estas letras también están separadas entre sí por espacios). Al escribir el fichero, el fan se ha asegurado de que no aparezcan episodios repetidos, ni valores de temporada o episodio incorrectos.

Ejemplo de fichero 'got.txt':

```
1 5 B L S L
3 9 S S S S S
2 10 L B B
2 1
...
```

- Episodio 1x05: murieron 1 Baratheon, 1 Stark y 2 Lannister
- Episodio 3x09: murieron 5 Stark
- Episodio 2x10: murieron 1 Lannister y 2 Baratheon
- Episodio 2x01: no murió ningún personaje de esas casas

Se pide escribir un programa en Pascal que lea los datos del fichero de texto y escriba por pantalla lo siguiente: cuántos episodios de cada una de las 6 temporadas faltan por ver (es decir, que no aparecen en el fichero), la media del número de personajes muertos por episodio, y cuál es la casa que ha perdido más personajes hasta el momento (si hay empate, con mostrar una cualquiera de las que tengan un mayor número de muertes es suficiente). Pueden definirse nuevas estructuras de datos si se considera necesario.

Ejemplo de salida por pantalla (nota: no corresponde necesariamente al got.txt mostrado, sino que es un ejemplo de cómo se ha de mostrar la salida):

```
Faltan por ver 2 episodios de la temporada 5
Faltan por ver 7 episodios de la temporada 6
Media de muertes por episodio: 2.54
```

Ejercicio 3**[3.5 puntos]**

Una imagen digital no es más que una matriz bidimensional de elementos llamados *píxeles*, cuyos valores numéricos indican el color de la correspondiente región de la foto. En el caso de una imagen en escala de grises, estos valores numéricos van desde 0 (que indica negro) hasta 255 (que indica blanco). Los valores intermedios representan, por tanto, distintas gradaciones de gris (ver Fig. 1a). El **histograma** de una imagen almacena, para cada posible valor de los píxeles (de 0 a 255), el número de píxeles con ese valor que hay en la imagen. El **máximo del histograma** de una imagen es el valor (de 0 a 255) que acumula mayor número de píxeles en dicha imagen.

Las técnicas de tratamiento digital de imagen se basan en alterar los datos de la matriz que representa la imagen para obtener distintos resultados. Por ejemplo, **umbralizar** una imagen consiste en elegir un valor (llamado umbral) y hacer que todos los píxeles de la imagen cuyo valor sea menor o igual a ese valor umbral valgan 0, y todos los que superen el umbral valgan 255 (ver Fig. 1b). Un posible valor para ese umbral es el máximo del histograma (definido arriba, y mostrado en la Fig. 2 para una imagen de ejemplo) correspondiente a esa imagen.

Sabiendo que una imagen se representa con la estructura de datos `tpImagen` (ver ejemplo debajo), **se pide**:

1. Definir una estructura de datos adecuada para almacenar el histograma de una imagen.
2. Desarrollar los siguientes procedimientos (teniendo en cuenta la descripción de los mismos incluida como comentario):

procedure calcularHistograma(??? imagen: tpImagen; ??? hist: tpHistograma);
{Devuelve en hist el histograma correspondiente a la imagen suministrada como parámetro.}

procedure umbralizarSegunMaxHist(??? imagenAUmb: tpImagen);
{Devuelve umbralizada la imagen suministrada como parámetro. Como valor umbral toma el máximo del histograma correspondiente a dicha imagen. Puede utilizar el procedimiento calcularHistograma. }

Nota: Puedes crear procedimientos o funciones adicionales si lo estimas apropiado o necesario.



Fig. 1a
 Imagen original



Fig. 1b
 Imagen umbralizada

11	120	11	11
120	120	120	15
120	120	120	156
0	213	221	120
255	213	9	120

Fig. 2. El máximo del histograma de esta imagen es 120.

Estructuras de datos
(completar lo que falta en
???)

const

MAXFIL = 1024;
MAXCOL = 1024;
MAXNIVEL = 255;

type

tpPixel=0..MAXNIVEL;
tpImagen = **record**
 nFilas: 1..MAXFIL;
 nColumnas: 1..MAXCOL;
 dat:

array[1..MAXFIL,1..MAXCOL
] of tpPixel;

end;

tpHistograma=???;