Examen de **Fundamentos de Informática**
Grado de Tecnologías Industriales
Escuela de Ingeniería y Arquitectura
Primera convocatoria: 15 de Junio de 2011

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

*Duración total del examen: **3 horas***

Ejercicio 1 [3.5 puntos]

Una imagen digital no es más que una matriz bidimensional de elementos llamados pixels, cuyos valores numéricos indican el color de la correspondiente región de la foto. En el caso de una imagen en blanco y negro, 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 imagen 1a).

Las técnicas de tratamiento digital se basan en alterar los datos de la matriz que representa la imagen para obtener distintos resultados. Por ejemplo, la *reducción de ruido en* una imagen consiste en aplicar un filtro que consiga reducir o eliminar la *nieve digital* que se produce por diferentes motivos al tomar la imagen. Uno de los filtros se llama *filtro de mediana*, que en su forma más simple consiste en sustituir el valor original de cada pixel de la imagen por el que se obtiene de la siguiente forma:

Se toman los valores de ese pixel y los de sus vecinos a izquierda y derecha, se ordenan por valor decreciente de sus valores, y se selecciona el valor central, que será el que sustituye al original en la imagen mejorada. En el caso de los pixeles de la primera columna, y en los de la última columna, se toma duplicado el valor de ésta (váse el ejemplo de la figura 2).

Sabiendo que una imagen tiene un tamaño cualquiera inferior a 640x480, se pide:

1) Definir la estructura de datos `tpImagen`, que permita guardar una imagen en memoria para su tratamiento posterior.
2) Si las imágenes se guardan en ficheros de texto de modo que en su primera línea figuran los dos enteros que indican el tamaño (nfilas, ncolumnas), y en cada línea posterior los valores de los pixeles de cada fila de la matriz, desarrollar el procedimiento:

   ```
   procedure cargarImagen(??? f: text; ??? imagen: tpImagen);
   ```
   {lee los datos de una imagen del fichero f y los guarda en imagen}

3) Desarrollar los siguientes procedimientos y funciones:

   ```
   function mediana3(??? a,b,c:tpPixel): tpPixel;
   ```
   {Devuelve la mediana de los tres números a, b y c}.

   ```
   procedure quitarRuido(??? original: tpImagen; ??? mejorada: tpImagen);
   ```
   {Aplica el filtro de mediana a la imagen original, devolviendo la imagen mejorada}

a) imagen original *b) imagen mejorada*

Figura 1 *Ejemplo de mejora*

Imagen original

```
[  1   4 211  13]
[ 93 255   5  56]
[ 67 128   0 202]
```

Imagen mejorada

```
[  1   4  13  13]
[ 93  93  56  56]
[ 67  67 128 202]
```

La segunda fila se ha obtenido así:

```
mejorada[2,1] = mediana( 93, 93,255) = 93
mejorada[2,2] = mediana( 93,255,  5) = 93
mejorada[2,3] = mediana(255,  5, 56) = 56
mejorada[2,4] = mediana(  5, 56, 56) = 56
```

Figura 2 *Ejemplo de cálculo del filtro de la mediana*

Ejercicio 2 [3 puntos]

El cifrado César es un tipo de cifrado que sustituye una letra del texto original por la letra que se encuentra un número fijo de posiciones más adelante en el alfabeto (volviendo a empezar desde el principio del alfabeto al llegar al final). Nótese que esta operación puede hacerse de forma muy eficiente utilizando aritmética modular. Por ejemplo, con un desplazamiento de 6, tendríamos el siguiente cifrado del propio alfabeto:

```
Texto original:   ABCDEFGHIJKLMNOPQRSTUVWXYZ
Texto codificado: GHIJKLMNOPQRSTUVWXYZABCDEF
```

Para descifrar un texto, es necesario conocer el desplazamiento para calcular la letra, pero, si no es así, se puede deducir. Si el texto está en español se sabe que, estadísticamente, la letra que más se utiliza es la letra **"e"**. 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 el fichero de texto *'texto.txt'* (que ha sido cifrado utilizando un cifrado César) 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:

Contenido del fichero 'texto.txt'

```
Kr sgy hxkbk jkyvoyzk ky kr zktkx
ktikxxgjuy ruy sktkyzkxky jkr jkyzotu.
```

Salida del programa

```
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 haber letras mayúsculas y minúsculas. A la hora de contar las apariciones de las letras, se hace independientemente de si la letra es mayúscula o minúscula (una **"K"** es lo mismo que una **"k"**). Sin embargo, al descifrar una letra mayúscula, el resultado es una letra mayúscula, mientras que al descifrar una letra minúscula, el resultado es una letra minúscula. El resto de caracteres (espacios, números, signos de puntuación y saltos de línea) se dejan tal cual están.

Ejercicio 3 [3.5 puntos]

Una empresa de venta de productos por internet decide premiar a los mejores clientes. Para ello guarda en los ficheros de registros `'comprasMes.dat'` y `'totalCompras.dat'` el importe de las compras efectuadas por cada cliente (sólo los que han realizado alguna compra) en el último mes, y en total desde el inicio de la promoción, respectivamente. Cada registro es de tipo *`tpGasto`* y consta de la identificación del cliente (un número entero), *`id_cliente`*, y el importe de las compras efectuadas (un número real), *`importe`*. Sabiendo que los ficheros están ordenados por *`id_cliente`*, en sentido creciente, se pide:

1) Definir los tipos de datos necesarios para representar la información descrita
2) Desarrollar el procedimiento:

```
procedure acumularGastoMes(??? fAcumulado: tpFichGasto;
                           ??? fGastoMes:  tpFichGasto;
                           ??? mejorCliente: tpGasto);
```

{ Acumula sobre el fichero fAcumulado los importes de las compras que hay en fGastoMes y devuelve en mejorCliente la información del cliente con mayor gasto acumulado }