

**Ejercicio 1****[3 puntos]**

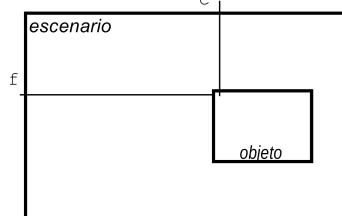
Se dispone de un fichero de números reales, 'datos.dat', que contiene los valores proporcionados por un sensor de temperatura durante un intervalo de tiempo, y se quiere saber si la representación de estos valores tiene forma de montículo o no. Una secuencia de números tiene forma de montículo si los números forman una secuencia estrictamente creciente hasta que alcanzan un máximo (que es único) y, a partir de entonces, forman una secuencia estrictamente decreciente. Para que se considere montículo es necesario que, sin contar el valor máximo, tanto la subsecuencia estrictamente creciente como la subsecuencia estrictamente decreciente sean no vacías.

Se pide escribir un programa en Pascal que informe en la salida estándar (pantalla) si los datos reales contenidos en el fichero 'datos.dat' corresponden, o no, a un montículo y, en caso afirmativo, muestre el valor máximo. Se podrá suponer que el fichero siempre contiene al menos un dato, aunque el número máximo de datos que puede almacenar no está acotado. A continuación se muestran nueve posibles ejemplos de ejecución:

<i>fichero 'datos.dat'</i>	<i>salida estandar</i>
3.7	NO es monticulo
4.1 7.0	NO es monticulo
1.1 2.0 3.21	NO es monticulo
8.3 6.0 4.17	NO es monticulo
1.8 2.0 1.59 1.59	NO es monticulo
1.8 2.0 2.0 1.59	NO es monticulo
1.8 2.0 1.59	SI es monticulo. Max. = 2.00
1.8 2.0 2.1 1.59	SI es monticulo. Max. = 2.10
1.8 3.14 2.1 1.59	SI es monticulo. Max. = 3.14

Ejercicio 2**[3 puntos]**

Un problema muy frecuente en el mundo de las imágenes digitales consiste en localizar una imagen dentro de otra: la cara de una persona dentro de una fotografía de grupo, una pieza dentro de una cinta transportadora, etc. El problema se puede formular de forma abstracta como sigue: tenemos una imagen grande (el “escenario”) y una imagen pequeña (el “objeto”) y nos interesa encontrar la posición (fila y columna) del escenario en que mejor encaja el objeto.



Para medir este encaje se define la “distancia” entre la imagen objeto y la subimagen del escenario localizada en el punto (f,c) (véase la figura) como la suma de los valores absolutos de las diferencias entre los valores de los píxeles correspondientes. Cuanto menor sea esta distancia, más parecido será el objeto a la subimagen (en el caso de que sean idénticos, esta suma será 0). Las características de las imágenes son las siguientes: **1)** tanto los escenarios como los objetos tienen, como máximo, 500×500 píxeles; **2)** las imágenes están en escala de grises con 256 niveles: cada píxel es un valor entero entre 0 (completamente negro) y 255 (completamente blanco).

Se pide:

- 1) Definir una estructura de datos `tpImagen`, que permita almacenar una imagen en escala de 256 niveles de gris de hasta 500×500 píxeles.
- 2) Diseñar la siguiente función que, dados un escenario `iEsc`, un objeto `iObj` y una posición de píxel en el escenario (`f, c`) en la que se puede situar el objeto (`iObj` cabe dentro de `iEsc` en esa posición), devuelve el valor de la “distancia” entre la imagen del objeto y la subimagen correspondiente del escenario.

Function distancia(??? `iEsc`: `tpImagen`; ??? `iObj`: `tpImagen`; ??? `f, c`: `integer`): `integer`;

- 3) Diseñar el siguiente procedimiento que, dados un escenario `iEsc` y un objeto `iObj`, cuyo tamaño no es mayor que el del escenario, devuelve la posición del píxel del escenario en que se localiza la subimagen con menor distancia al objeto, así como el valor de dicha distancia. Si hubiera varias posiciones con el mismo valor mínimo, puede devolver cualquiera de ellas.

Procedure localizar(??? `iEsc`: `tpImagen`; ??? `iObj`: `tpImagen`;
??? `f, c`: `integer`; ??? `distancia`: `integer`);

Ejercicio 3

[4 puntos]

Las Islas Canarias son un archipiélago volcánico, por lo que es necesario mantener una vigilancia extrema sobre los fenómenos sísmicos que se producen en ellas. Por este motivo, el Instituto Geográfico Nacional dispone de una red de vigilancia para observar dichos fenómenos, cuyas mediciones se pueden ver en internet (algunas son en tiempo real), en forma de tablas. Una de estas tablas es el listado de los terremotos de los últimos 10 días en las Islas Canarias, de magnitud igual o superior a 1.5. Más adelante se muestra un ejemplo con 9 de estos eventos sísmicos, pero en periodos de actividad puede haber centenares de ellos en un solo día.

Supongamos que nos pasan una de estas tablas en un fichero de texto con el siguiente formato:

- La primera línea contiene las cabeceras de las columnas.
- En cada una de las siguientes líneas aparecen los datos de un evento: el número de evento (*entero*), fecha (*dd mm aaaa*), hora (*hh mm ss*), longitud y latitud (*reales*), profundidad en km (*entero*) y la magnitud (*real*). Los datos están separados por espacios en blanco.

Se pide:

- 1) Definir la estructura de datos `tpEvento` que permita almacenar en memoria uno de estos eventos sísmicos.
- 2) Desarrollar el procedimiento

```
Procedure leerEvento(??? f:text; ??? elEvento: tpEvento);  
{ f está asociado a un fichero con la estructura descrita y situado al principio de una línea. Lee de f los  
datos de un evento y los devuelve en elEvento }
```

- 3) Desarrollar el procedimiento

```
Procedure elMasIntenso(??? f:text; ??? elEvento: tpEvento);  
{ f está asociado a un fichero con la estructura descrita; lee todos los eventos en f y devuelve en elEvento  
los datos del evento de mayor magnitud que es más reciente }
```

- 4) Desarrollar un programa PASCAL que, utilizando todo lo anterior, muestre por pantalla la información del evento de mayor magnitud a partir del nombre del fichero introducido interactivamente desde el teclado (entrada estándar).

Nota: Los eventos del fichero están ordenados por fecha y hora en sentido decreciente, y el fichero no contiene errores de formato y es no vacío.

Ejemplo: Si el nombre de fichero introducido es `Terremotos0907.txt`, la salida debería ser.

```
Evento de mayor magnitud:  
Magnitud 3.5  
Día 5 de julio de 2013 a las 17h 55m 00s  
Latitud 29.4315 Longitud -16.3004 Profundidad 55 km
```

fichero Terremotos0907.txt

Evento	Fecha	Hora (GMT)	Latitud	Longitud	Prof (km)	Magnitud
1220406	08 07 2013	12 35 20	27.8018	-18.1193	22	3.1
1220249	07 07 2013	21 51 04	27.7220	-18.0863	12	2.0
1219864	05 07 2013	17 55 00	29.4315	-16.3004	55	3.5
1219814	05 07 2013	01 22 33	27.6995	-18.0674	19	2.3
1219672	04 07 2013	06 22 28	27.8052	-18.2462	11	1.8
1219479	03 07 2013	04 57 53	27.7981	-18.2349	18	1.8
1219449	02 07 2013	23 56 45	27.7452	-18.1945	21	1.9
1219054	30 06 2013	18 39 04	27.6851	-18.0998	33	1.9
1219011	30 06 2013	14 15 25	27.7224	-18.0262	12	1.8

Examen de **Fundamentos de Informática**
Grado de Tecnologías Industriales
Escuela de Ingeniería y Arquitectura
segunda convocatoria: 2 de Septiembre de 2013



**Departamento de
Informática e Ingeniería
de Sistemas**

Universidad Zaragoza

Duración total del examen: 3 horas