



Ejercicio 1 [2,5 puntos]

Una **progresión aritmética** de números enteros es una sucesión en la que cada término se obtiene a partir del anterior sumándole un número fijo. Una **progresión geométrica** de números enteros es una sucesión en la que cada término se obtiene multiplicando el anterior por una constante denominada *razón* o *factor* de la progresión.

Se pide: Desarrollar un programa PASCAL que lea del teclado una secuencia de enteros, tan larga como se desee y terminada con “fin de línea”, y diga si es una progresión aritmética, geométrica, ambas, o ninguna. Además mostrará el total de números introducidos y, si no se introduce ningún número, o sólo uno, lo indicará de forma oportuna.

Nota: Se puede suponer que tras el último número no hay ningún espacio.

Ejemplos de ejecuciones:

1 2 3 4
4 números; forman progresión aritmética

no hay ningún número

1 2 4 8 16
5 números; forman progresión geométrica

7
sólo hay un número

1 1 1 1 1 1
6 números; forman progresión aritmética
6 números; forman progresión geométrica

1 2 3 4 7
5 números; NO forman progresión aritm. ni geom.

Ejercicio 2 [3,5 puntos]

Para hacer más eficaz el control de pasajeros en los aeropuertos, se ha diseñado un sistema de reconocimiento de huellas dactilares que digitaliza la huella del pulgar de cada viajero y la compara con las almacenadas en un fichero de registros. Cada registro representa la ficha de un criminal y consta, además de una imagen digitalizada de la huella de su dedo pulgar, de su nombre y apellidos. La imagen está compuesta por píxeles (puntos), codificada en blanco y negro (cada píxel sólo puede tomar dos valores: blanco o negro), y su tamaño es de 1024x1024 píxeles. La imagen de la huella tomada en el aeropuerto se codifica de manera idéntica.

Si al realizar la comparación de la huella del pasajero con alguna huella del fichero se obtiene que coinciden (los píxeles correspondientes tienen el mismo valor) en más del 99% de los píxeles, se considera que se ha detectado un posible *criminal* y se le niega la entrada al país. Si coinciden entre un 90% y un 99% se considera *sospechoso* y ha de ser interrogado antes de entrar al país, y si coinciden menos del 90% se le deja pasar sin más (está *limpio*).

Se pide:

- 1) Diseñar las estructuras de datos más adecuadas para el problema planteado.
- 2) Desarrollar y completar el siguiente procedimiento:

procedure procesarHuella (??? h: tpHuella; ??? fCrim: tpFichCriminales);

{compara la huella del pulgar del viajero, h, con las existentes en el fichero de registros fCrim y muestra por pantalla el resultado de la búsqueda: criminal, sospechoso o limpio. En los dos primeros casos muestra también el nombre y apellidos del criminal cuya huella tiene mayor número de píxeles coincidentes con h, junto con el porcentaje de píxeles coincidentes}

Nota: Se deberá tener en cuenta que en el fichero nunca puede haber más de una huella con coincidencia de más del 99% de los píxeles, aunque sí puede haber varias con coincidencias entre el 90% y 99%

ejemplos de salida por pantalla:

criminal
Osama Bin Laden 99.4%

sospechoso
Ester Rorista Malo 92.0%

limpio

Ejercicio 3

[4 puntos]

La compañía de telecomunicaciones con la que tenemos contratada una línea telefónica nos suministra todos los meses un fichero de texto, 'llamadas.txt', con información acerca de las llamadas realizadas desde nuestro número de teléfono. En cada línea de dicho fichero aparece la información de una llamada, que consiste en:

- el número de teléfono de destino,
- la duración de la llamada en segundos,
- el importe de la llamada en euros.

976123456	180	1.5
976252525	55	0.65
976123456	245	0.93
976481516	45	0.09
976481516	120	1.2

ejemplo de fichero 'llamadas.txt'

Si se supone que el número de teléfonos distintos a los que se ha llamado no es superior a 200,

Se pide:

- 1) 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 (correspondiente a un fichero 'llamadas.txt') y la devuelve a través del registro tpDatosLlamada, dejando el fichero preparado para la lectura de la siguiente línea}

- 2) Escribir un programa PASCAL que, a partir de la información que hay en el fichero 'llamadas.txt', muestre por pantalla, para cada número al que se ha llamado, el número total de llamadas realizadas, el importe total y la duración total (en horas, minutos y segundos). Además, en la última línea se mostrará la suma total de estos valores para todos los teléfonos incluidos en el fichero.

Como ejemplo, a continuación se muestra lo que sacaría por pantalla dicho programa para el fichero de llamadas anterior:

Telefono	Llamadas	Importe	Duracion
-----	-----	-----	-----
976123456	2	2.43	0h 7m 5s
976252525	1	0.65	0h 0m 55s
976481516	2	1.29	0h 2m 45s
TOTAL	5	4.37	0h 10m 45s