



Entrega:

- Escribe tu nombre completo en **todas** las hojas que entregues como solución del examen.
- Responde a cada ejercicio comenzando **en una hoja diferente**, los ejercicios se entregan por separado.
- La resolución del examen no se puede escribir a lápiz ni con bolígrafo rojo. Lo que esté escrito a lápiz o en rojo será ignorado.

Ejercicio 1

[3.0 puntos]

La compañía de *streaming* de música *Turrify* está implementando un sistema de gestión de música informatizado, para, por ejemplo, poder generar *playlists* de forma automática con ciertos criterios.

Para almacenar la información necesaria sobre la biblioteca de música, se requiere una estructura de datos adecuada. **Define** en Pascal los siguientes tipos de datos:

- **tpTrack**: Almacena la información sobre una canción: su título (una cadena de texto), su duración (en segundos) y el *path* (ruta) del fichero en el disco que almacena la canción (cadena de texto).
- **tpAlbum**: Almacena la información sobre un álbum: su título (una cadena de texto), y la secuencia de *tracks* (pistas o canciones) que contiene. El tamaño de esta secuencia puede cambiar de un álbum a otro, pero es como máximo de 32.
- **tpLibrary**: Almacena la biblioteca de música completa: guarda la lista de álbumes que contiene la biblioteca, que no tiene tamaño fijo (va creciendo con el tiempo) pero sí un máximo número de álbumes de 100.000.

A partir de esta información, se desea poder construir y almacenar listas de reproducción (*playlists*).

Define en Pascal los siguientes tipos de datos:

- **tpTrackRef**: Almacena una referencia a una canción (no una copia, esto es, no un **tpTrack** completo); dicha referencia a la canción está formada por el número de álbum dentro de la biblioteca (un entero) y el número de pista dentro del álbum (otro entero) de una canción determinada.
- **tpPlaylist**: Almacena la información de una lista de reproducción: una secuencia de referencias a canciones (**tpTrackRef**), de tamaño variable, pero con un máximo de 100.

Una vez definidas estas estructuras de datos, **implementa** en Pascal los siguientes subprogramas:

- Una función

function duracionPlaylist(??? lb: tpLibrary; ??? pl: tpPlaylist): **real**;

que devuelva la duración total (en segundos) de una lista de reproducción 'pl' que referencia canciones de la biblioteca 'lb'. Tanto 'pl' como 'lb' son dadas como entrada a la función.

- Un procedimiento

procedure creaPlaylist(??? lb: tpLibrary; maxdur: **real**; ??? pl: tpPlaylist);

que construya y devuelva una lista de reproducción 'pl' con canciones de una biblioteca dada 'lb' que tengan duración inferior a 'maxdur' (en segundos).

Para simplificar nuestro problema, las canciones serán las primeras encontradas en la biblioteca que cumplan ese criterio (en un caso real habría que complicar el método de selección un poco más...). Ten en cuenta que puede que no haya suficientes canciones con esa característica para llenar la lista o que haya más de las que caben.

NOTA: Todas las cadenas de texto a almacenar tienen como máximo 255 caracteres.

Ejercicio 2

[3.5 puntos]

Los trabajadores de una empresa presentan a su supervisor, al acabar el año fiscal, una tabla para calcular las *retribuciones extra* que les deben ser abonadas. Esas retribuciones extra pueden deberse a tres actividades (horas extra, salidas de campo, y festivos trabajados), y pueden aumentarse debido a la antigüedad del trabajador. La tabla tiene una estructura como la siguiente:

	Antigüedad	Horas extra	Salidas de campo	Festivos trabajados
Trabajador 1	15	100	6	1
Trabajador 2	25	249.8	3	20
Trabajador 3	21	300	4	13
Trabajador

El supervisor paga la hora extra a 20€ cada una, la salida de campo a 500€ cada una y los festivos trabajados a 80€ cada uno. Además, piensa pagarles un plus a sus trabajadores por antigüedad de un 0,5 % por año de antigüedad de la cantidad a recibir por las otras actividades. Por ejemplo, para el trabajador número 1, su salario extra se calcularía de la siguiente forma:

extra por actividades : $100 \times 20 + 6 \times 500 + 1 \times 80 = 5080$
porcentaje por antigüedad : $15 \times 0,5 = 7,5$
retribución extra : $5080 + (7,5/100) \times 5080 = 5461$

Se quiere calcular qué gasto total tiene el supervisor en concepto de retribuciones extra.

Para este ejercicio, **se pide**:

- Definir las **estructuras de datos** necesarias para almacenar la información de la tabla de trabajadores, sabiendo que el número puede ser variable, pero nunca mayor que 10.
- **Implementar una función**

```
function gastoTotal(??? tabla: ???): real;
```

que reciba esos datos, y a partir de los mismos calcule y devuelva el total que debe abonar el supervisor a los trabajadores en concepto de retribuciones extra.

Ejercicio 3

[3.5 puntos]

Se dispone de información de la evolución de la cotización del Bitcoin (BTC) respecto al dólar estadounidense (USD), a lo largo de un cierto período de tiempo. La cotización de una moneda frente al dólar es el tipo de cambio, esto es, a cuántos dólares equivale un BTC. Dicha información está contenida en un fichero secuencial de reales, y a partir del mismo se quiere generar un fichero de texto, con la información en un formato que se considera más adecuado.

El mencionado **fichero secuencial de reales** contiene, para cada hora del día, la cotización del BTC. Así, para cada día hay 24 valores reales en el fichero (desde el de las 00:00, hasta el de las 23:00), y en el fichero están consignados datos de un mes (o sea, de 28, 29, 30 o 31 días). Además, los dos primeros valores reales del fichero se utilizan para almacenar el mes y el año al que corresponden las cotizaciones.

El **fichero de texto** que se quiere generar a partir del fichero secuencial tiene el siguiente formato: cada línea corresponde a un día, e incluye, por este orden y separados por espacios en blanco, un valor de cotización para ese día, el texto 'BTC', y la fecha (ver ejemplo de la Fig. 1).

Tal y como se ha explicado, el fichero secuencial tiene cotizaciones por hora, y el fichero de texto tendrá una cotización por día. Hay dos métodos sencillos para obtener una cotización por día a partir de las cotizaciones por hora:

- Método 1: Se toma como cotización por día el valor de cotización de las 00:00 de cada día.
- Método 2: Se toma como cotización por día el promedio diario (esto es, la media de los 24 valores de un día).

Se pide implementar en Pascal un programa que:

1. Pida al usuario el nombre del fichero secuencial de reales del que se deberá leer la información, y el método de cálculo de la cotización por día (1 o 2).
2. A partir de la información contenida en el fichero secuencial, genere el fichero de texto correspondiente con el formato descrito arriba e ilustrado en la Fig. 1. El método de cálculo de la cotización por día será el especificado por el usuario (1 o 2). El fichero de texto se llamará 'cotizaciones.txt'.
3. Calcule y muestre por pantalla la diferencia media entre las cotizaciones por día calculadas con ambos métodos. Esta diferencia media μ_{dif} se calcula como:

$$\mu_{dif} = \frac{1}{N} \sum_{d=1}^{d=N} abs(Z_{d,1} - Z_{d,2}),$$

donde $Z_{d,i}$ es la cotización diaria del día d obtenida con el método i ($i=1$ o $i=2$), y N es el número de días del mes. $abs(\cdot)$ es la función valor absoluto, predefinida en Pascal para datos de tipo real.

Importante: El formato del fichero de texto debe ser análogo al ejemplo de la Fig. 1, y el de interacción y salida por pantalla debe ser como el ejemplo mostrado en la Fig. 2.

Nota 1: Se puede asumir que en el fichero secuencial se han registrado días completos, esto es, que si para un día hay valor de cotización de las 00:00, están todo el resto de cotizaciones del día hasta la de las 23:00 (incluida). Asimismo, se incluye en dicho fichero la información de todos los días del mes.

Nota 2: Se valorará negativamente recorrer el fichero secuencial más veces de lo necesario.

Figura 1: Ejemplo de fichero 'cotizaciones.txt' (se muestra sólo un fragmento del fichero).

```
17438.70 BTC 1-5-2024
17936.70 BTC 2-5-2024
18847.70 BTC 3-5-2024
19922.90 BTC 4-5-2024
20979.70 BTC 5-5-2024
9855.16 BTC 6-5-2024
...
```

Figura 2: Ejemplo de interacción y salida por pantalla (a efectos de ilustrar el formato). Se indican en negrita los datos introducidos por el usuario.

```
Introduzca nombre de fichero cotizaciones por hora:
cotizacionesMayo.dat
Introduzca metodo de calculo (1 = 00:00, 2 = promedio):
2
Fichero cotizaciones.txt generado con exito.
Se ha utilizado el metodo 2 de calculo de cotizacion diaria.
La diferencia media entre ambos metodos es de: 27.89651
```