



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 puntos]

Una herramienta de diseño CAD permite modelar objetos geométricos bidimensionales en base a secuencias de polígonos. Cada polígono es representado como una **secuencia** de vértices (máximo 20). Cada vértice se representa mediante sus dos coordenadas reales, x e y . El programa CAD guarda todos los polígonos que representan un objeto en un fichero secuencial de polígonos (registros) con la siguiente estructura:

```
tpVertice = record
  x: real;
  y: real;
end;

tpPoligono = record
  vertices: array[1..20] of tpVertice;
  n: integer; { Numero de vertices }
end;

tpObjeto = file of tpPoligono;
```

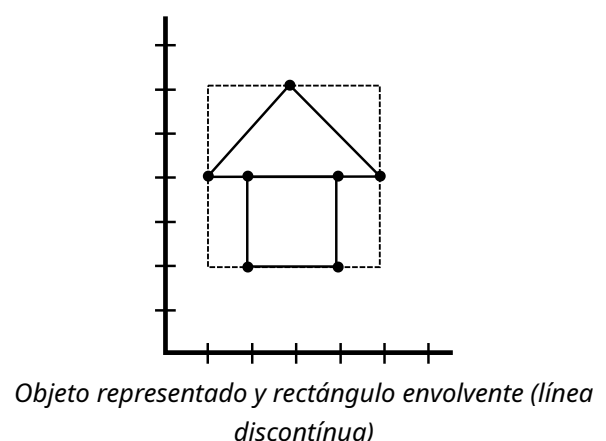
Se pretende extender la herramienta CAD con una nueva funcionalidad. En concreto **se pide** que implementes un programa que, dado el nombre de un fichero secuencial (que representa un objeto) introducido por teclado, muestre por pantalla el **rectángulo envolvente** del objeto: su esquina inferior izquierda (valores mínimos de x e y) y su esquina superior derecha (valores máximos de x e y). El rectángulo envolvente es el rectángulo de mínimo tamaño que contiene todos los vértices de todos los polígonos. Se puede asumir que el objeto tiene como mínimo un polígono.

Ejemplo de ejecución:

```
1 Nombre del fichero: flecha.dat
2 Rectangulo envolvente:
3 1.00, 2.00 - 5.00, 6.00
```

Fichero flecha.dat (registros)

| | | | | |
|-----------|-----|-----|-----|-----|
| vertices: | x=1 | x=5 | x=3 | |
| | y=4 | y=4 | y=6 | ... |
| n: | 3 | | | |
| vertices: | x=2 | x=4 | x=4 | x=2 |
| | y=4 | y=4 | y=2 | y=2 |
| n: | 4 | | | |



Se valorará positivamente que el fichero secuencial se recorra una única vez.

Ejercicio 2

[3.5 puntos]

Una compañía quiere crear un programa para gestionar el rendimiento y el coste del servicio de mensajería que ofrece. Para ello, guarda cada día, en un fichero de texto de nombre 'entregas.txt', la información de las entregas realizadas por cada uno de sus mensajeros a lo largo de ese día (un mensajero por línea).

Cada línea del fichero comienza con el código identificativo del empleado (un número entero); a continuación, para cada entrega efectuada por el mensajero, la línea incluye las horas de inicio y fin del reparto de esa entrega (en formato hhmm, donde hh indica horas y mm minutos), la zona en que se ha realizado (1, 2 ó 3), y la cantidad (en euros) cobrada por el mensajero. El número de entregas efectuadas por cada mensajero puede ser variable, y todos los campos de una línea están separados por espacios en blanco. A partir de esta información se puede obtener el precio efectivo de cada entrega, que se define como la cantidad cobrada por minuto de dicha entrega (y se mide por tanto en euros/min).

Se pide escribir un programa en Pascal que, dado el fichero de texto 'entregas.txt', muestre por pantalla:

- la zona más rentable, esto es, la zona para la que el promedio del precio efectivo es mayor, y
- el mensajero más trabajador, esto es, el que más tiempo ha trabajado en ese día.

La salida deberá mostrarse como en el ejemplo de ejecución.

Ejemplo de fichero 'entregas.txt':

```
1 123 1105 1135 1 15.6 1345 1401 3 25.7 2050 2110 1 11.6
2 89 1415 1445 2 7.1
3 236
4 511 1530 1555 1 10.5 2130 2210 3 31.0
```

Salida por pantalla:

```
1 Mejor precio efectivo: Zona 3 (con un promedio de 1.19 eur/min)
2 Mas trabajador: 123 (66 minutos)
```

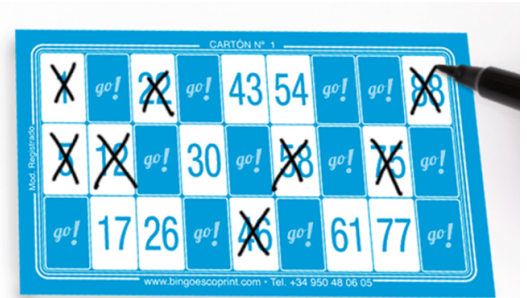
Ejercicio 3

[3.5 puntos]

En el juego del **Bingo** cada jugador dispone de uno o varios cartones, divididos en casillas, organizadas en filas y columnas (normalmente 3x9). Cada casilla puede estar ocupada por un número (del 1 al 89), o por un fondo o logotipo, indicando que está vacía.

Durante el juego se extraen de un bombo una serie de bolas numeradas (del 1 al 89), y cada jugador va marcando los números que aparecen en su cartón. Se obtienen premios por marcar una línea horizontal completa (*línea*) o todos los números del cartón (*bingo*).

Queremos escribir un programa que simule el juego del Bingo, para lo cual **se pide** realizar en Pascal las siguientes tareas:



- Definir los tipos de datos necesarios para representar estos elementos:
 - **tpCasilla:** Una casilla, que puede estar ocupada por un número, que hay que almacenar, o 'vacía' (no hay número), y puede ser marcada (tachada) a lo largo del juego.
 - **tpCarton:** Un cartón de bingo, que contiene 3x9 casillas.
 - **tpBolas:** Una **secuencia** de enteros que contiene los números que han aparecido hasta un momento dado del sorteo, de tamaño variable, con un máximo de 89.
- Implementar las siguientes funciones:
 - Marcar en un cartón las casillas correspondientes a las bolas que han salido en el sorteo, si aparecen en él:


```
procedure marcar(???? carton: tpCarton; ???? bolas: tpBolas);
```

NOTA: puedes utilizar el hecho de que cada columna del cartón contiene números de la misma DECENA.
 - Comprobar si un cartón dado contiene un premio de tipo línea, es decir, existe una fila con todos sus números marcados:


```
function hayLinea(???? carton: tpCarton): boolean;
```
 - Contar cuantos números quedan por marcar en un cartón para conseguir un bingo:


```
function faltanParaBingo(???? carton: tpCarton): integer;
```