

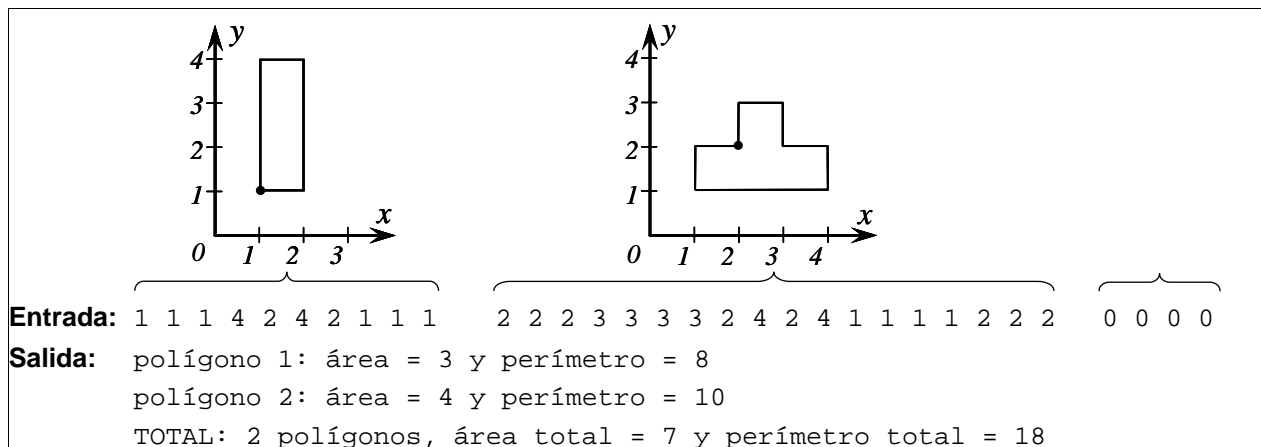
Ejercicio 1 [3,5 puntos]

Un **polígono** plano simple puede ser descrito mediante una secuencia ordenada de todos sus vértices, de modo que dos vértices consecutivos definen un **lado** del polígono. Cada vértice se describe por sus coordenadas (números reales) horizontal o **abscisa** (x) y vertical u **ordenada** (y). Se pide desarrollar un programa PASCAL que lea de la entrada estandar (teclado) una serie de vértices que conforman una secuencia de polígonos planos simples e indique, para cada uno de ellos, tanto su **área** como su **perímetro**. Además mostrará el número total de polígonos, junto con el área y perímetro totales.

Los polígonos tienen las siguientes características:

- Todos los lados son verticales u horizontales (nunca habrá diagonales) y, por tanto, dos lados contiguos forman siempre un ángulo de 90°.
- Los polígonos son figuras cerradas, por lo que el último vértice coincide con el primero. En la secuencia, tras cerrar un polígono se considerará que empieza otro polígono nuevo.

La secuencia finaliza con un polígono de área y perímetro cero (sólo tiene un vértice). A continuación se muestra un ejemplo junto con la salida por pantalla correspondiente:



Notas:

- 1) El perímetro es la suma de las longitudes de todos los lados.
- 2) Si se supone que el recorrido de los puntos se hace en sentido horario, el cálculo del área de uno de estos polígonos puede efectuarse de un modo muy simple: considerando que cada lado horizontal que va de izquierda a derecha añade al área el valor $\ell * y$ (y es la ordenada del lado y ℓ su longitud) y si va de derecha a izquierda resta dicho valor. Así, en el ejemplo anterior, el área del primer polígono es: $1*4 - 1*1 = 3$ y la del segundo polígono es: $1*3 + 1*2 - 3*1 + 1*2 = 4$.

Ejercicio 2 [3 puntos]

Se dispone del fichero de texto 'expresiones.txt' que contiene en cada una de sus líneas una expresión. Diseñar un programa PASCAL que lea dicho fichero y muestre por pantalla, para cada una de las líneas, lo siguiente:

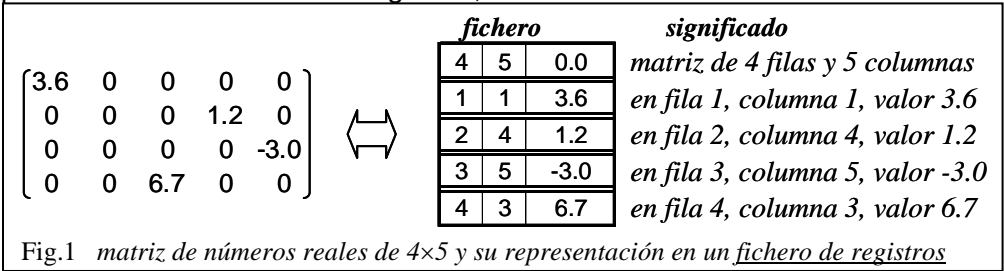
- 1) El número de línea.
- 2) Si la estructura de paréntesis de la expresión que hay en la línea es correcta, o no; es decir, si todo paréntesis '(' tiene el correspondiente ')' y viceversa.
- 3) El mayor nivel de anidamiento de paréntesis.
- 4) Si hay alguna expresion nula entre paréntesis; es decir, si entre un '(' y el ')' correspondiente, a lo sumo hay espacios.

A continuación se muestra un ejemplo de fichero junto con la salida por pantalla correspondiente:

3f(x) +4(5+2(6-y))	1: correcta	max.nivel=2
)a(2: error	max.nivel=0
(4+x(6+y)	3: error	max.nivel=2
X+y)	4: error	max.nivel=0
Ax+b	5: correcta	max.nivel=0
2m+f() *g(x)	6: correcta	max.nivel=1 hay expr.nula

Ejercicio 3 [3,5 puntos]

En la resolución de diversos problemas de ingeniería se utilizan **matrices cuasivacías (MCV)**, que son aquellas que contienen un pequeño número de elementos no nulos. Un modo muy simple de describirlas consiste en indicar sus dimensiones y, a continuación, todos los valores no nulos junto con su posición (pueden estar en cualquier orden). Para almacenar de forma permanente la matriz se puede utilizar un fichero de registros, como se muestra a continuación:



Se pide desarrollar en PASCAL el siguiente procedimiento (previamente se realizarán todas las declaraciones de tipos de datos que sean necesarias).

```
procedure analizar_MCV(??? m: tpMCV);
{ lee la matriz cuasivacia m, almacenada en un fichero de registros, y muestra por pantalla lo siguiente:
- número de filas y columnas de la matriz
- las filas con todos sus elementos nulos, o la indicación de que ninguna fila es nula
- las columnas con todos sus elementos nulos, o la indicación de que ninguna columna es nula }
```

La ejecución de este procedimiento con la MCV de la fig.1 mostraría por pantalla:

Matriz de 4 x 5
ninguna fila es nula
columna 2 nula

Observaciones:

- 1) Se puede suponer que la mayor dimensión de una MCV es de 4000 x 4000.
- 2) No es posible representar en memoria todos los elementos de la matriz, aunque sí es posible representar una fila y una columna.