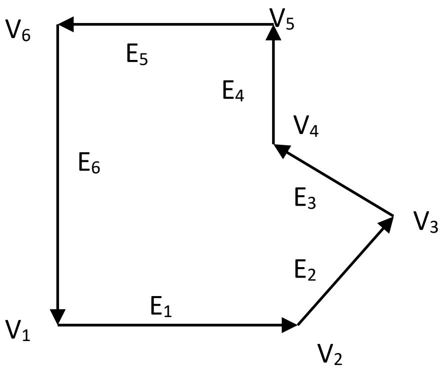
Para identificar si un polígono es convexo o no, se puede proceder calculando la componente ***z*** del producto vectorial de los vectores de aristas sucesivas. Si todos tienen el mismo signo, el polígono es convexo y en caso contrario es cóncavo. Ejemplo:




$(E_1 \wedge E_2)_z > 0$

$(E_2 \wedge E_3)_z > 0$

$(E_3 \wedge E_4)_z < 0$ → El polígono es cóncavo

$(E_4 \wedge E_5)_z > 0$

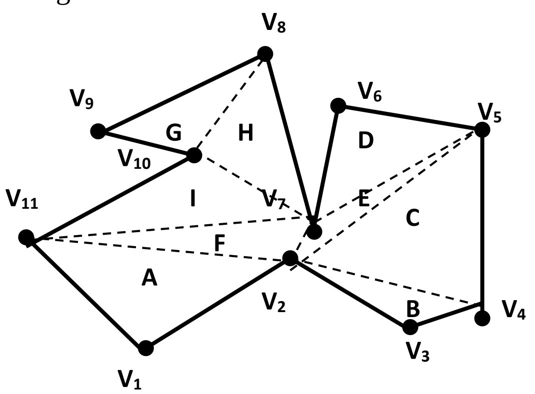
Construir un subprograma para la triangulación de un polígono cualquiera (cóncavo o convexo) utilizando el algoritmo de corte de oreja (algoritmo de Van Gogh). El subprograma aceptará como parámetro de entrada un polígono regular (sin aristas que intersectan) y con sus vértices especificados en sentido anti-horario. A continuación, particionará el polígono en triángulos (triangulación) y finalmente devolverá los resultados a través de un parámetro de salida. Considerar las siguientes tipologías para el parámetro de entrada y el de salida:

```
CONST    MAX=100
TIPOS    tipo_poligono: registro de
                n: entero
                v: tipo_v_p
         Fin_registro
         tipo_v_p: vector[1..MAX] de tipo_punto2d
         tipo_punto2d: registro de
                x,y: real
         Fin_registro

tipo_triangulacion: registro de
        n: entero
                t: tipo_v_t
        Fin_registro
tipo_v_t: vector[1..MAX] de tipo_tri
tipo_tri: vector[1..3] de tipo_punto2d
```

Se define una oreja de un polígono como un triángulo definido por un vértice v y sus vecinos de izquierda y derecha (i y d), tales que el triángulo (i,v,d) está completamente contenido en el interior del polígono. Cada polígono contiene siempre una oreja (al menos 2 si el nº de vértices es mayor que 3).

El algoritmo de corte de oreja consiste en localizar una oreja, formar el triángulo (i,v,d) correspondiente, eliminar el vértice v del polígono y volver a repetir los pasos anteriores hasta que solo queden 3 vértices, que a su vez forman un triángulo.




P: $\{v_1,v_2,v_3,v_4,v_5,v_6,v_7,v_8,v_9,v_{10},v_{11}\}$

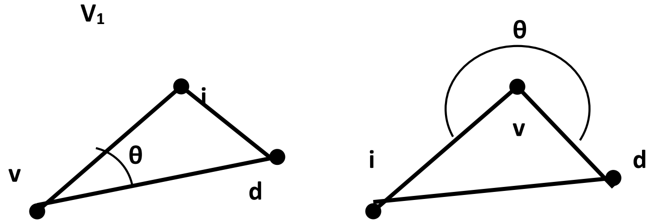
T: {A,B,C,D,E,F,G,H,I}

A: $\{v_{11},v_1,v_2\}$

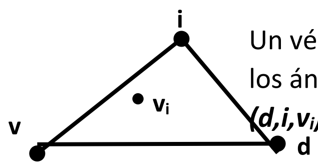
B: $\{v_2,v_3,v_4\}$

C: $\{v_2,v_4,v_5\}$

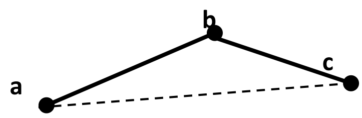
Nota: comprobación de que el triángulo ***(i,v,d)*** está completamente contenido en el polígono:




1) θ<180º (área_triángulo_signo>0)
2) Cualquier vértice $\mathbf{v_i}$ del polígono diferente de **i,v,d**, no está dentro del triángulo

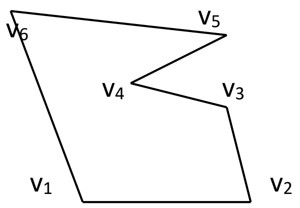



Un vértice $\mathbf{v_i}$ está dentro del triángulo si todos los ángulos de los triángulos ***(i,v,vi)***, ***(v,d,vi)*** y ***(d,i,vi)*** son menores de 180º




Area_triángulo_signo=$(a_x*b_y-a_y*b_x+a_y*c_x-a_x*c_y+b_x*c_y-c_x*b_y)/2.0$

Diseñar un programa que calcule e imprima en pantalla el centroide de un objeto con forma poligonal (centroide: posición del centro de masas para un objeto con densidad uniforme). El número de lados del polígono (***n***) será leído previamente por teclado, comprobándose que es mayor o igual a 3, introduciéndose a continuación en orden la abscisa (***x***) y la ordenada (***y***) de cada uno de los ***n*** vértices del polígono.




$$A = \left[\sum_{k=1}^{n}(x_k*y_{k+1}-x_{k+1}*y_k)\right]/2$$

$$x_{cent} = \left\{\sum_{k=1}^{n}\left[(x_{k+1}+x_k)*(x_k*y_{k+1}-x_{k+1}*y_k)\right]\right\}/(6*A)$$

$$y_{cent} = \left\{\sum_{k=1}^{n}\left[(y_{k+1}+y_k)*(x_k*y_{k+1}-x_{k+1}*y_k)\right]\right\}/(6*A)$$

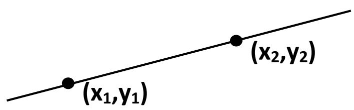
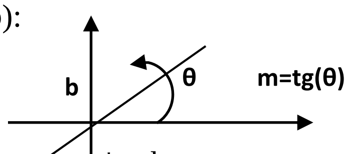
Nota: $(x_{n+1},y_{n+1}) = (x_1,y_1)$

Implementar las estructuras de datos necesarias para representar de forma eficiente los siguientes objetos de información:

Un punto bidimensional: (x,y)

Un triángulo en el plano, dadas las coordenadas de sus tres vértices

Una línea bidimensional, considerando cualquiera de las siguientes posibilidades de especificación

| | |
|---|---|
| 1) Dados dos puntos diferentes de la misma: $\mathbf{p_1}$ y $\mathbf{p_2}$   Ecuación de la recta: $y=[(y_2-y_1)/(x_2-x_1)]*(x-x_1)+y_1$ | 2) Dada la pendiente de la recta (**m**) y la ordenada en el origen (**b**):   Ecuación de la recta: $y=m*x+b$ |
| 3) Dada la pendiente de la recta **m** y un punto $\mathbf{p_1}$ de la misma<br><br>Ecuación de la recta: $y=m*(x-x_1)+y_1$ | 4) Dados los coeficientes **a**, **b** y **c** del polinomio de su representación implícita:<br><br>Ecuación de la recta: $a*x+b*y+c=0$ |
| 5) Dado un valor de abscisa $\mathbf{x_0}$ (para el caso especial de rectas verticales):<br><br>Ecuación de la recta: $x=x_0$ | 6) Dado un valor de ordenada $\mathbf{y_0}$ (para el caso especial de rectas horizontales):<br><br>Ecuación de la recta: $y=y_0$ |
| | |

Diseñar e implementar las siguientes funciones:

- Una función que acepte como argumento una línea bidimensional (en cualquiera de las representaciones anteriores) y la devuelva en representación implícita.
- Una función que calcule y devuelva el punto de intersección de dos líneas dadas como argumentos. Dicha función deberá devolver a través de su identificador un código que indique el resultado del cálculo (1: líneas secantes, 0: líneas paralelas/coincidentes). Nota: dos líneas son paralelas si tienen la misma pendiente.