A. Solución de ecuaciones

Resolver las siguientes ecuaciones:

1. $\dfrac{x-3}{2}+7=x-\dfrac{5-x}{4}$

2. $x^2+9=6x$

3. $1-(x-2)^2=1$

4. $x^6-19x^3-216=0$ (sugerencia: hacer cambio de variables $x^3=t$)

En los siguientes sistemas de ecuaciones, utilizar Processing para graficar las rectas correspondientes y el punto *(x, y)* solución

4. $\begin{cases} 2x+4y=10 \\ 4x+y=13 \end{cases}$

5. $\begin{cases} 2x-5y=-12 \\ 7x-2y=-11 \end{cases}$

6. $\begin{cases} x-5y=-3 \\ \frac{x}{4}+\frac{y}{3}=4 \end{cases}$

7. $\begin{cases} 2x-3y=-14 \\ 4\cdot(3x-y)=0 \end{cases}$

8. $\begin{cases} 7x+4z=80 \\ 5x-6z=4 \end{cases}$

9. $\begin{cases} 6x+12y=66 \\ 2000x-2000y+4000=0 \end{cases}$

10. En este siguiente sistema de ecuaciones ¿que figuras geométricas representa cada ecuación?

$$\left.\begin{array}{l} x^2+y^2=45 \\ x-y=3 \end{array}\right\}$$

Graficarlas con Processing, así como también el punto solución.

B. Crear una composición en Processing utilizando solamente la función *line()* donde haya *12* intersecciones entre líneas.

C. ¿Cual la ecuación de la circunferencia de radio *150* y centrada en el punto *(200, 200)*?
Escribir un programa en Processing que grafique este circunferencia, la línea que va desde el origen de coordenadas hasta la posición del mouse *(mouseX, mouseY)*, y el punto de interesección entre la circunferencia y la línea (si existe).

D. Crear una composición en Processing con al menos *5* elipses tal que la distancia entre las mismas sea *30*.

E. Se tiene dos rectángulos: uno con su esquina superior izquierda centrada en *(x0, y0)* y de alto y ancho *(w0, h0)*, y el segundo con su esquina superior izquierda centrada en *(x1, y1)* y de alto y ancho *(w1, h1)*. ¿Cuales son las inecuaciones que un punto *(x, y)* tiene que satisfacer para estar dentro de la intersección entre los rectángulos?

F*. En clase vimos que la ecuación de una recta general es:

$$ax + by + c = 0$$

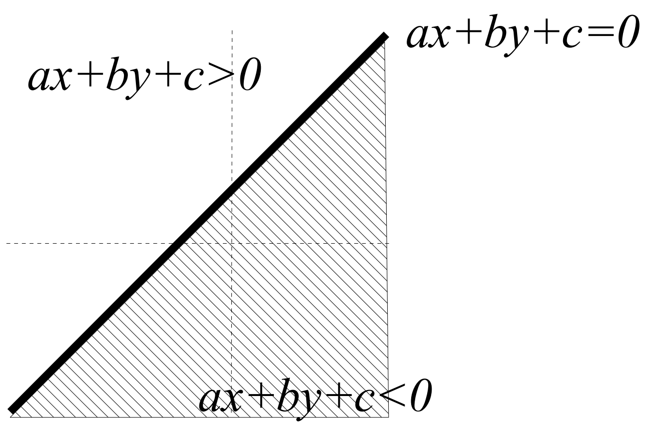
Es decir, los puntos *(x, y)* que verifican la ecuación determinan una recta en el plano. Para los demás puntos *(x, y)* que no están en la recta, es decir que no verifican la ecuación, hay dos posibilidades:

$$ax + by + c < 0$$

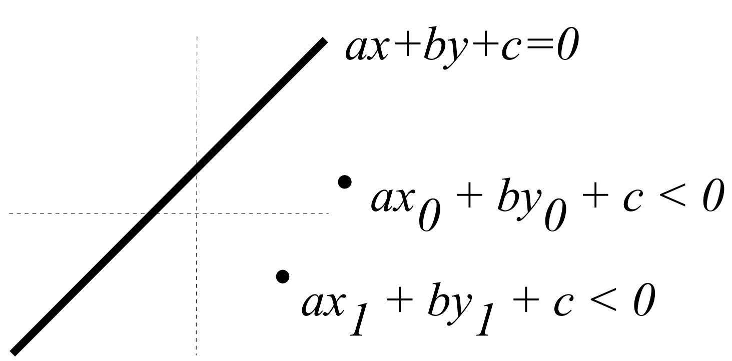
ó:

$$ax + by + c > 0$$

Geométricamente esos dos grupos de puntos son las regiones arriba y abajo de la recta:



Qué región es arriba y cual abajo de la recta depende de los signos de *a*, *b* y *c*. Lo que si podemos decir es que si tenemos dos puntos (x_0, y_0) y (x_1, y_1) tales que $ax_0 + by_0 + c < 0$ y $ax_1 + by_1 + c < 0$, entonces los dos están sobre el mismo lado de la recta:

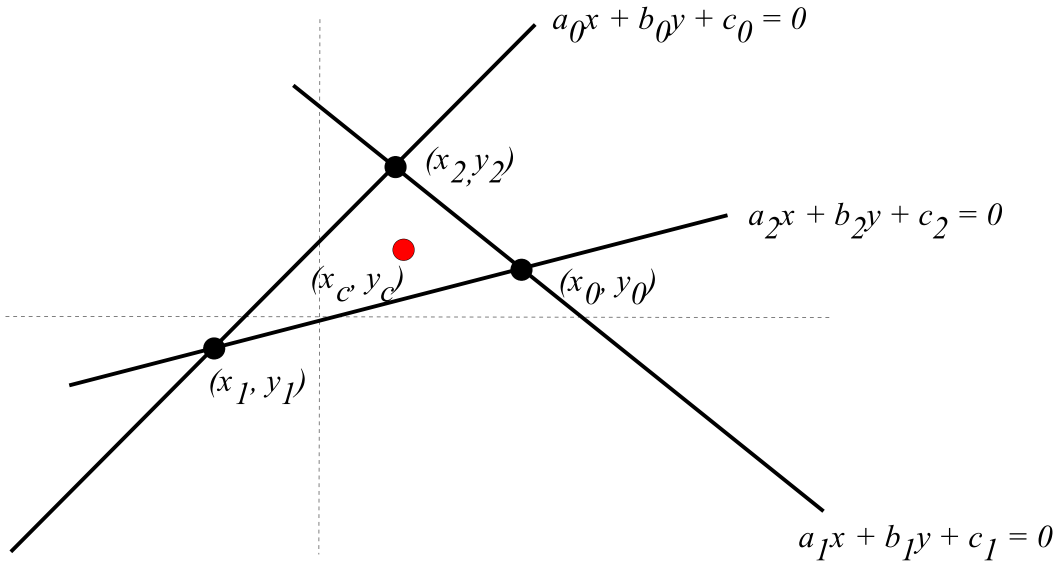


1. Graficar en Processing la recta que pasa entre los puntos *(0, 0)* y *(200, 300)*, y encontrar su ecuación en forma $ax+by+c=0$. Para cada posición del mouse $(x, y) = (mouseX, mouseY)$, imprimir el signo de $ax+by+c$.

2. Esta observación con respecto al signo de $ax+by+c$ nos permite obtener un algoritmo general para determinar si un punto está dentro de un triángulo:
Consideremos un triángulo determinado por puntos (x_0, y_0), (x_1, y_1) y (x_2, y_2). Utilizando las coordenadas de estos puntos construimos las ecuaciones de las rectas correspondientes:

$a_0x + b_0y + c_0 = 0$, la recta que une (x_1, y_1) y (x_2, y_2)
$a_1x + b_1y + c_1 = 0$, la recta que une (x_0, y_0) y (x_2, y_2)
$a_2x + b_2y + c_2 = 0$, la recta que une (x_0, y_0) y (x_1, y_1)

es decir:



Entonces si tenemos un punto (x_c, y_c) dentro del triángulo, entonces debe de estar del mismo lado de las tres rectas que los vértices del triángulo opuestos a las mismas. En otras palabras, las tres condiciones que se deben verificar son:

el signo de $a_0x_c + b_0y_c + c_0$ y $a_0x_0 + b_0y_0 + c_0$ es el mismo.
el signo de $a_1x_c + b_1y_c + c_1$ y $a_1x_1 + b_1y_1 + c_1$ es el mismo.
el signo de $a_2x_c + b_2y_c + c_2$ y $a_2x_2 + b_2y_2 + c_2$ es el mismo.

Escribir un programa en Processing que utilize esta idea para determinar si el mouse está dentro de un triángulo arbitrario.

G*. Escribir un programa para pintar los píxeles interiores a un triángulo arbitrario utilizando la función *point()*.