

Nome: Filipe Damasceno Coelho.

Algoritmo em python.

```
'''
Created on 31 de mai de 2016

@author: Filipe Damasceno
'''

def colocar_origem(ponto1,ponto2):
    return ponto1, [0,0],[ponto2[0]-ponto1[0],ponto2[1]-ponto1[1]]

def voltar_posicao(ponto,val):
    return [ponto[0]+val[0],ponto[1]+val[1]]

def transformar(ponto,posicao):
    if posicao == 2:
        return [ponto[1],ponto[0]]
    elif posicao == 3:
        return [-ponto[1],ponto[0]]
    elif posicao == 4:
        return [-ponto[0],ponto[1]]
    elif posicao == 5:
        return [-ponto[0],-ponto[1]]
    elif posicao == 6:
        return [-ponto[1],-ponto[0]]
    elif posicao == 7:
        return [ponto[1],-ponto[0]]
    elif posicao == 8:
        return [ponto[0],-ponto[1]]
    else:
        return ponto

def posicao(ponto1,ponto2):
    m = (ponto2[1] - ponto1[1])/(ponto2[0] - ponto1[0])
    if ponto1[0] < ponto2[0] and 0 <= m <= 1:
        return 1
    elif ponto1[0] < ponto2[0] and 0 > m >= -1:
        return 8
    elif ponto1[0] > ponto2[0] and 0 >= m >= -1:
        return 4
    elif ponto1[0] > ponto2[0] and 0 < m <= 1:
        return 5
    elif ponto1[1] < ponto2[1] and m < -1:
        return 3
    elif ponto1[1] < ponto2[1] and m > 1:
        return 2
    elif ponto1[1] > ponto2[1] and m < -1:
        return 7
    elif ponto1[1] > ponto2[1] and m > 1:
        return 6

def bresenham(ponto1,ponto2):
    x = ponto1[0]
    y = ponto1[1]
    m = (ponto2[1]-ponto1[1])/(ponto2[0]-ponto1[0])
    e = m - 0.5
    pontos = []
    for _ in range(ponto2[0]):
        pontos.append([x,y])
        while e >= 0:
            y += 1
            e -= 1
        x += 1
        e += m
    pontos.append([x,y])
    return pontos
```

```

def desenhar(pontos = []):

    pontos.append(pontos[0])

    retas = []

    for i in range(len(pontos)-1):
        p1,p2 = pontos[i],pontos[i+1]
        print("pontos: << P1: ",p1," P2: ",p2,">>")

        val,p1,p2 = colocar_origem(p1, p2)
        print("ponto de subtracao: ",val,"novos pontos: P1: ",p1," P2: ",p2)

        pos = posicao(p1, p2)
        print("Posicao no octeto: ",pos)
        p1 = transformar(p1, pos)
        p2 = transformar(p2, pos)
        if p2[0] < 0 > p2[1]:
            p2[0],p2[1] = p2[0]*-1,p2[1]*-1
        print("Pontos transformados: P1: ",p1," P2: ",p2)
        ptos = bresenham(p1, p2)
        print("Reta: ",ptos)
        pf = []
        for i in ptos:
            pf.append(voltar_posicao(transformar(i, pos), val))
        print("Reta na posicao de Origem: ",pf)
        print("+=+=+=*10)
        retas.append(pf)
    return retas

if __name__ == '__main__':
    pontos = [[-4,3],[0,5],[3,3],[5,0],[0,-3],[-5,-1]]
    for i,v in enumerate(desenhar(pontos)):
        print("Reta[" ,i,"]: ",v)

```

[illegible]