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] \rightarrow ponto2[0] and 0 \rightarrow= m \rightarrow= -1:
        return 4
    elif ponto1[0] \rightarrow 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] \rightarrow ponto2[1] and m < -1:
        return 7
    elif ponto1[1] > ponto2[1] and m > 1:
        return 6
def bresenham(ponto1, ponto2):
    x = pontol[0]
    y = pontol[1]
   m = (ponto2[1]-ponto1[1])/(ponto2[0]-ponto1[0])
    e = m - 0.5
    pontos = []
         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 = \overline{\text{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)
```

Saida:

```
pontos: << P1: [-4, 3] P2: [0, 5] >>
ponto de subtracao: [-4, 3] novos pontos: P1: [0, 0] P2: [4, 2]
Posicao no octeto: 1
Pontos transformados: P1: [0, 0] P2: [4, 2]
Reta: [[0, 0], [1, 1], [2, 1], [3, 2], [4, 2]]
Reta na posicao de Origem: [[-4, 3], [-3, 4], [-2, 4], [-1, 5], [0, 5]]
pontos: << P1: [0, 5] P2: [3, 3] >>
ponto de subtracao: [0, 5] novos pontos: P1: [0, 0] P2: [3, -2]
Posicao no octeto: 8
Pontos transformados: P1: [0, 0] P2: [3, 2]
Reta: [[0, 0], [1, 1], [2, 1], [3, 2]]
Reta na posicao de Origem: [[0, 5], [1, 4], [2, 4], [3, 3]]
pontos: << P1: [3, 3] P2: [5, 0] >>
ponto de subtracao: [3, 3] novos pontos: P1: [0, 0] P2: [2, -3] Posicao no octeto: 7
Pontos transformados: P1: [0, 0] P2: [3, 2]
Reta: [[0, 0], [1, 1], [2, 1], [3, 2]]
Reta na posicao de Origem: [[3, 3], [4, 2], [4, 1], [5, 0]]
pontos: << P1: [5, 0] P2: [0, -3] >>
ponto de subtracao: [5, 0] novos pontos: P1: [0, 0] P2: [-5, -3]
Posicao no octeto: 5
Pontos transformados: P1: [0, 0] P2: [5, 3]
Reta: [[0, 0], [1, 1], [2, 1], [3, 2], [4, 2], [5, 3]]
Reta na posicao de Origem: [[5, 0], [4, -1], [3, -1], [2, -2], [1, -2], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [0, -1], [
3]]
pontos: << P1: [0, -3] P2: [-5, -1] >>
ponto de subtracao: [0, -3] novos pontos: P1: [0, 0] P2: [-5, 2]
Posicao no octeto: 4
Pontos transformados: P1: [0, 0] P2: [5, 2] Reta: [[0, 0], [1, 0], [2, 1], [3, 1], [4, 2], [5, 2]]
Reta na posicao de Origem: [[0, -3], [-1, -3], [-2, -2], [-3, -2], [-4, -1],
[-5, -1]
pontos: << P1: [-5, -1] P2: [-4, 3] >> ponto de subtracao: [-5, -1] novos pontos: P1: [0, 0] P2: [1, 4]
Posicao no octeto: 2
Pontos transformados: P1: [0, 0] P2: [4, 1]
Reta: [[0, 0], [1, 0], [2, 1], [3, 1], [4, 1]]
Reta na posicao de Origem: [[-5, -1], [-5, 0], [-4, 1], [-4, 2], [-4, 3]]
Reta[ 0 ]: [[-4, 3], [-3, 4], [-2, 4], [-1, 5], [0, 5]]
Reta[ 1 ]: [[0, 5], [1, 4], [2, 4], [3, 3]]
Reta[ 2 ]: [[3, 3], [4, 2], [4, 1], [5, 0]]
Reta[ 3 ]: [[5, 0], [4, -1], [3, -1], [2, -2], [1, -2], [0, -3]]
Reta[ 4 ]: [[0, -3], [-1, -3], [-2, -2], [-3, -2], [-4, -1], [-5, -1]]
Reta[ 5 ]: [[-5, -1], [-5, 0], [-4, 1], [-4, 2], [-4, 3]]
```