

```

from math import *
import matplotlib.pyplot as plt

def vecteur(point1, point2):
    return [y - x for x, y in zip(point1, point2)]

def add(vecteur1, vecteur2):
    return [x + y for x, y in zip(vecteur1, vecteur2)]

def norme(vecteur):
    return sqrt(prodscal(vecteur, vecteur))

def prodscal(vecteur1, vecteur2):
    return sum([x*y for x, y in zip(vecteur1, vecteur2)])

def determ(vecteur1, vecteur2):
    return vecteur1[0]*vecteur2[1]-vecteur1[1]*vecteur2[0]

def angle(vecteur1, vecteur2):
    cosinus=prodscal(vecteur1, vecteur2)/(norme(vecteur1)*norme(vecteur2))
    sinus=determ(vecteur1, vecteur2)/(norme(vecteur1)*norme(vecteur2))
    return atan2(sinus,cosinus)

def rotation(u, theta):
    return [u[0]*cos(theta)-u[1]*sin(theta),u[0]*sin(theta)+u[1]*cos(theta)]

def intersekte(x, y, z, t):
    a1 = (y[1]-x[1])/(y[0]-x[0])
    b1 = x[1]-a1*x[0]
    a2 = (t[1]-z[1])/(t[0]-z[0])
    b2 = z[1]-a2*z[0]
    return [(b2-b1)/(a1-a2),((b2-b1)/(a1-a2))*a1+b1]

a=[-0.5, 0]
b=[0, 2]
c=[2.5, 2.5]
d=[3.48,0]
print('a = (',a[0],',',a[1],')')
print('b = (',b[0],',',b[1],')')
print('c = (',c[0],',',c[1],')')
print('d = (',d[0],',',d[1],')')
ab = vecteur(a,b) ; ac = vecteur(a,c)
ba = vecteur(b,a) ; bc = vecteur(b,c)
ca = vecteur(c,a) ; cb = vecteur(c,b)
anglea=angle(ab,ac) ; angleb=angle(bc,ba) ; anglec=angle(ca,cb) ;
aprime = add(a, rotation(ab,anglea/3.0)) ; aseconde = add(a, rotation(ab,anglea*2.0/3.0))
bprime = add(b, rotation(bc,angleb/3.0)) ; bseconde = add(b, rotation(bc,angleb*2.0/3.0))
cprime = add(c, rotation(ca,anglec/3.0)) ; cseconde = add(c, rotation(ca,anglec*2.0/3.0))
n=intersekte(a,aseconde,c,cprime)
o=intersekte(c,cseconde,b,bprime)
x=intersekte(b,bseconde,a,aprime)
print('n = (',n[0],',',n[1],')')

```

```

print('o = ('o[0],',',o[1],'.')
print('x = ('x[0],',',x[1],'.')
print('Normes des cotes %3.15f '% norme(vecteur(n,o)))
print('Normes des cotes %3.15f '% norme(vecteur(o,x)))
print('Normes des cotes %3.15f '% norme(vecteur(x,n)))

bc = vecteur(b,c) ; bd = vecteur(b,d)
cd = vecteur(c,d) ; cb = vecteur(c,b)
db = vecteur(d,b) ; dc = vecteur(d,c)
angleb=angle(bc,bd) ; anglec=angle(cd,cb) ; angled=angle(db,dc) ;
bprime = add(b, rotation(bc,angleb/3.0)) ; bseconde = add(b, rotation(bc,angleb*2.0/3.0))
cprime = add(c, rotation(cd,anglec/3.0)) ; cseconde = add(c, rotation(cd,anglec*2.0/3.0))
dprime = add(d, rotation(db,angled/3.0)) ; dseconde = add(d, rotation(db,angled*2.0/3.0))
p=intersecte(b,bseconde,d,dprime)
q=intersecte(d,dseconde,c,cprime)
r=intersecte(c,cseconde,b,bprime)
print('p = ('p[0],',',p[1],'.')
print('q = ('q[0],',',q[1],'.')
print('r = ('r[0],',',r[1],'.')
print('Normes des cotes %3.15f '% norme(vecteur(p,q)))
print('Normes des cotes %3.15f '% norme(vecteur(q,r)))
print('Normes des cotes %3.15f '% norme(vecteur(r,p)))

cd = vecteur(c,d) ; ca = vecteur(c,a)
da = vecteur(d,a) ; dc = vecteur(d,c)
ac = vecteur(a,c) ; ad = vecteur(a,d)
anglec=angle(cd,ca) ; angled=angle(da,dc) ; anglea=angle(ac,ad) ;
cprime = add(c, rotation(cd,anglec/3.0)) ; cseconde = add(c, rotation(cd,anglec*2.0/3.0))
dprime = add(d, rotation(da,angled/3.0)) ; dseconde = add(d, rotation(da,angled*2.0/3.0))
apprime = add(a, rotation(ac,anglea/3.0)) ; aseconde = add(a, rotation(ac,anglea*2.0/3.0))
s=intersecte(c,cseconde,a,apprime)
t=intersecte(a,aseconde,d,dprime)
u=intersecte(d,dseconde,c,cprime)
print('s = ('s[0],',',s[1],'.')
print('t = ('t[0],',',t[1],'.')
print('u = ('u[0],',',u[1],'.')
print('Normes des cotes %3.15f '% norme(vecteur(s,t)))
print('Normes des cotes %3.15f '% norme(vecteur(t,u)))
print('Normes des cotes %3.15f '% norme(vecteur(u,s)))

da = vecteur(d,a) ; db = vecteur(d,b)
ab = vecteur(a,b) ; ad = vecteur(a,d)
bd = vecteur(b,d) ; ba = vecteur(b,a)
angled=angle(da,db) ; anglea=angle(ab,ad) ; angleb=angle(bd,ba) ;
dprime = add(d, rotation(da,angled/3.0)) ; dseconde = add(d, rotation(da,angled*2.0/3.0))
apprime = add(a, rotation(ab,anglea/3.0)) ; aseconde = add(a, rotation(ab,anglea*2.0/3.0))
bprime = add(b, rotation(bd,angleb/3.0)) ; bseconde = add(b, rotation(bd,angleb*2.0/3.0))
v=intersecte(d,dseconde,b,bprime)
w=intersecte(b,bseconde,a,apprime)
z=intersecte(a,aseconde,d,dprime)
print('v = ('v[0],',',v[1],'.')
print('w = ('w[0],',',w[1],'.')

```

```

print('z = ('z[0],',',z[1],'.)')
print('Normes des cotes %3.15f '% norme(vecteur(s,t)))
print('Normes des cotes %3.15f '% norme(vecteur(t,u)))
print('Normes des cotes %3.15f '% norme(vecteur(u,s)))

fig = plt.figure()
ax = fig.add_subplot(111)
plt.plot([a[0],b[0],c[0],d[0],a[1],b[1],c[1],d[1],a[1]], 'black', alpha=0.7)
plt.plot([n[0],o[0],x[0],n[0]], [n[1],o[1],x[1],n[1]], 'blue', alpha=0.7)
plt.plot([p[0],q[0],r[0],p[0]], [p[1],q[1],r[1],p[1]], 'red', alpha=0.7)
plt.plot([s[0],t[0],u[0],s[0]], [s[1],t[1],u[1],s[1]], 'orange', alpha=0.7)
plt.plot([v[0],w[0],z[0],v[0]], [v[1],w[1],z[1],v[1]], 'green', alpha=0.7)
ax.set_aspect('equal')
plt.annotate('A',xy=(a[0]+0.03,a[1]-0.03)) ; plt.annotate('B',xy=(b[0]+0.03,b[1]-0.03)) ;
plt.annotate('C',xy=(c[0]+0.03,c[1]-0.03)) ; plt.annotate('D',xy=(d[0]+0.03,d[1]-0.03))
plt.annotate('N',xy=(n[0]+0.03,n[1]-0.03)) ; plt.annotate('O',xy=(o[0]+0.03,o[1]-0.03)) ;
plt.annotate('X',xy=(x[0]+0.03,x[1]-0.03))
plt.annotate('P',xy=(p[0]+0.03,p[1]-0.03)) ; plt.annotate('Q',xy=(q[0]+0.03,q[1]-0.03)) ;
plt.annotate('R',xy=(r[0]+0.03,r[1]-0.03))
plt.annotate('S',xy=(s[0]+0.03,s[1]-0.03)) ; plt.annotate('T',xy=(t[0]+0.03,t[1]-0.03)) ;
plt.annotate('U',xy=(u[0]+0.03,u[1]-0.03))
plt.annotate('V',xy=(v[0]+0.03,v[1]-0.03)) ; plt.annotate('W',xy=(w[0]+0.03,w[1]-0.03)) ;
plt.annotate('Z',xy=(z[0]+0.03,z[1]-0.03))

C1x = (n[0]+o[0]+x[0])/3 ; C1y = (n[1]+o[1]+x[1])/3 ; C1 = [C1x,C1y]
C2x = (p[0]+q[0]+r[0])/3 ; C2y = (p[1]+q[1]+r[1])/3 ; C2 = [C2x,C2y]
C3x = (s[0]+t[0]+u[0])/3 ; C3y = (s[1]+t[1]+u[1])/3 ; C3 = [C3x,C3y]
C4x = (v[0]+w[0]+z[0])/3 ; C4y = (v[1]+w[1]+z[1])/3 ; C4 = [C4x,C4y]
plt.scatter(C1x,C1y,color='blue')
plt.scatter(C2x,C2y,color='red')
plt.scatter(C3x,C3y,color='green')
plt.scatter(C4x,C4y,color='orange')
print(angle(vecteur(C1,C2),vecteur(C1,C4))*360/(2*pi))
print(angle(vecteur(C2,C3),vecteur(C2,C1))*360/(2*pi))
print(angle(vecteur(C3,C4),vecteur(C3,C2))*360/(2*pi))
print(angle(vecteur(C4,C1),vecteur(C4,C3))*360/(2*pi))
plt.plot([C1x,C2x,C3x,C4x,C1x],[C1y,C2y,C3y,C4y,C1y], 'gray', alpha=0.7)
plt.show()

```