```
import matplotlib.pyplot as plt
import cmath
from cmath import *
import numpy as np
from numpy import exp, sin, cos

def vecteur(p1x,p1y,p2x,p2y):
    return [p2x-p1x,p2y-p1y]

def norme(x):
    return sqrt(x[0]*x[0]+x[1]*x[1])

def intersecte(x, y, z, t):
    a1 = (y.imag-x.imag)/(y.real-x.real)
    b1 = x.imag-a1*x.real
    a2 = (t.imag-z.imag)/(t.real-z.real)
    b2 = z.imag-a2*z.real
    return (b2-b1)/(a1-a2),((b2-b1)/(a1-a2))*a1+b1

plt.figure(figsize = (10,10))

u = np.linspace(0,1,100)
miniexp = np.exp(2*pi*1j*u)
plt.plot(miniexp.real,miniexp.imag)

alpha =122*2*pi/360
beta = 175*2*pi/360

T1bout1 = exp(1j*alpha/3)
I = exp(1j*alpha/3)
T1bout2 = exp(1j*beta)
T2bout1 = exp(1j*2*alpha/3)
E = exp(1j*2*alpha/3)
T2bout2 = exp(1j*beta)
B = exp(1j*beta)
T3bout1 = exp(1j*alpha)
A = exp(1j*alpha)
T3bout2 = exp(1j*(2*beta+2*pi)/3)
D = exp(1j*(2*beta+2*pi)/3)
T4bout1 = exp(1j*alpha)
T4bout2 = exp(1j*(beta+4*pi)/3)
F = exp(1j*(beta+4*pi)/3)
T5bout1 = 1
C = 1
T5bout2 = exp(1j*(2*alpha+beta)/3)
G = exp(1j*(2*alpha+beta)/3)
T6bout1 = 1
T6bout2 = exp(1j*(alpha+2*beta)/3)
H = exp(1j*(alpha+2*beta)/3)

T1 = plt.axline([T1bout1.real,T1bout1.imag],
[T1bout2.real,T1bout2.imag],color='red',alpha=0.5)
T2 = plt.axline([T2bout1.real,T2bout1.imag],
[T2bout2.real,T2bout2.imag],color='red',alpha=0.5)
T3 = plt.axline([T3bout1.real,T3bout1.imag],
[T3bout2.real,T3bout2.imag],color='blue',alpha=0.5)
T4 = plt.axline([T4bout1.real,T4bout1.imag],
[T4bout2.real,T4bout2.imag],color='blue',alpha=0.5)
T5 = plt.axline([T5bout1.real,T5bout1.imag],
[T5bout2.real,T5bout2.imag],color='green',alpha=0.5)
T6 = plt.axline([T6bout1.real,T6bout1.imag],
[T6bout2.real,T6bout2.imag],color='green',alpha=0.5)
plt.xlim(-1.5,1.5)
plt.ylim(-1.5,1.5)
```

```
plt.scatter(T1bout1.real,T1bout1.imag,color='green',alpha=0.8)
plt.scatter(T1bout2.real,T1bout2.imag,color='green',alpha=0.8)
plt.scatter(T2bout1.real,T2bout1.imag,color='red',alpha=0.8)
plt.scatter(T2bout2.real,T2bout2.imag,color='red',alpha=0.8)
plt.scatter(T3bout1.real,T3bout1.imag,color='blue',alpha=0.8)
plt.scatter(T3bout2.real,T3bout2.imag,color='blue',alpha=0.8)
plt.scatter(T4bout1.real,T4bout1.imag,color='orange',alpha=0.8)
plt.scatter(T4bout2.real,T4bout2.imag,color='orange',alpha=0.8)
plt.scatter(T5bout1.real,T5bout1.imag,color='yellow',alpha=0.8)
plt.scatter(T5bout2.real,T5bout2.imag,color='yellow',alpha=0.8)
plt.scatter(T6bout1.real,T6bout1.imag,color='cyan',alpha=0.8)
plt.scatter(T6bout2.real,T6bout2.imag,color='cyan',alpha=0.8)
Nreal, Nimag = intersecte(E,B,A,D)
plt.scatter(Nreal,Nimag,color='black',alpha=0.8)
Mreal, Mimag = intersecte(C,G,A,F)
plt.scatter(Mreal,Mimag,color='black',alpha=0.8)
Lreal, Limag = intersecte(C,H,I,B)
plt.scatter(Lreal,Limag,color='black',alpha=0.8)
plt.plot([A.real,B.real,C.real,A.real],
[A.imag,B.imag,C.imag,A.imag],color='green',alpha=0.8)
plt.plot([Lreal,Mreal,Nreal,Lreal],
[Limag,Mimag,Nimag,Limag],color='brown',alpha=0.8)
print('Norme cote 1 ',norme(vecteur(Lreal,Limag,Mreal,Mimag)))
print('Norme cote 2 ',norme(vecteur(Mreal,Mimag,Nreal,Nimag)))
print('Norme cote 3 ',norme(vecteur(Nreal,Nimag,Lreal,Limag)))
print('8 sin(angle_en_A/3)*sin(angle_en_B/3)*sin(angle_en_C/3) =
',8*sin(alpha/3)*sin(beta/3))
plt.axis('equal')
plt.show()
```