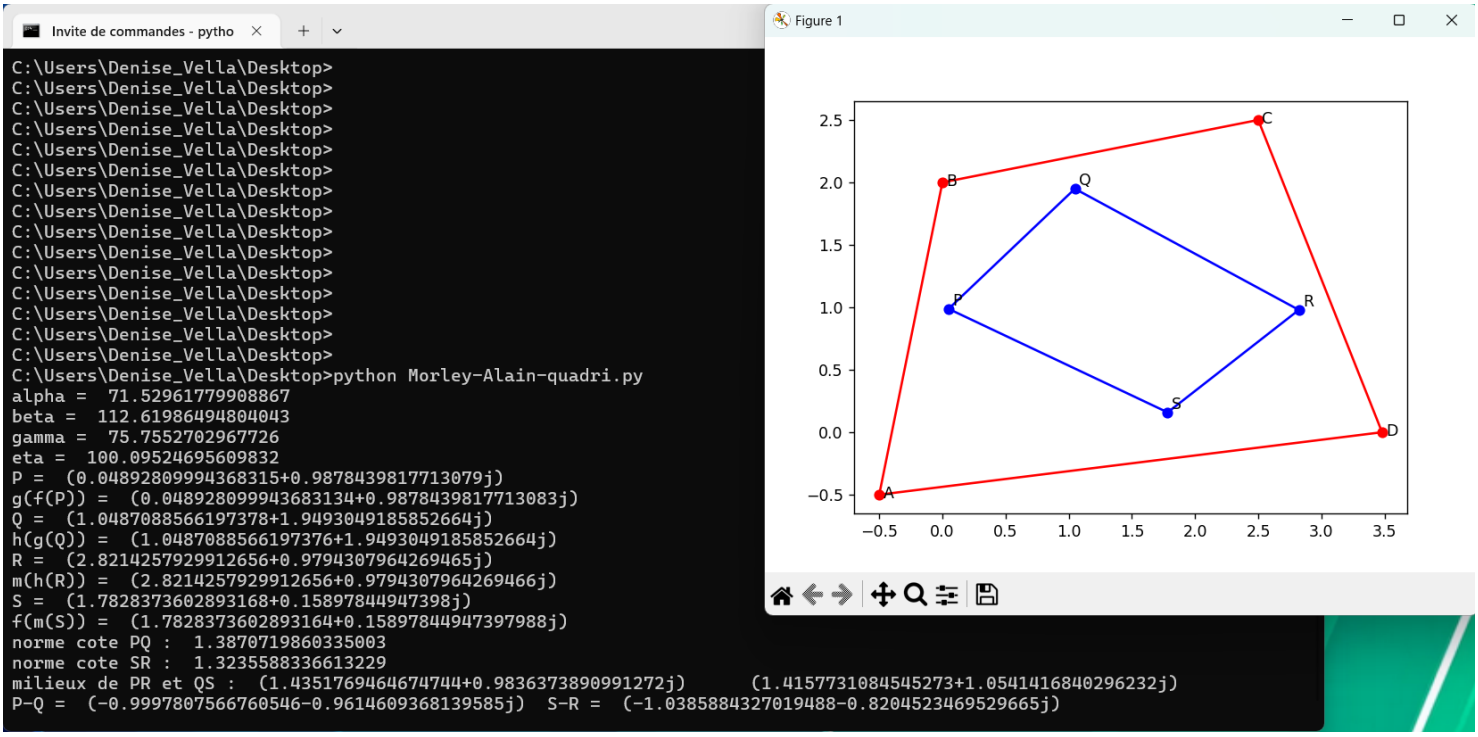
```
import cmath
from cmath import pi, exp,sqrt,acos
import matplotlib.pyplot as plt
import math
from math import atan2

def f(z):
   return(exp(1j*alpha/4)*(z-A)+A)
def g(z):
   return(exp(1j*beta/4)*(z-B)+B)
def h(z):
   return(exp(1j*gamma/4)*(z-C)+C)
def m(z):
   return(exp(1j*eta/4)*(z-D)+D)

def norme(v):
   return((sqrt(v.real*v.real+v.imag*v.imag)).real)

def prodscal(v1, v2):
   res = v1.real*v2.real+v1.imag*v2.imag
   return(res.real)

def determ(v1, v2):
   res = v1.real*v2.imag-v1.imag*v2.real
   return(res.real)

def angle(v1, v2):
   sinus = determ(v1, v2)/(norme(v1)*norme(v2))
   cosinus = prodscal(v1, v2)/(norme(v1)*norme(v2))
   return(-atan2(sinus,cosinus))

A = -0.5-0.5*1j ; B = 2*1j ; C = 2.5+2.5*1j ; D = 3.48+0*1j
AB = B-A ; AD = D-A ; BC = C-B ; BA = A-B ; CD = D-C ; CB = B-C ; DA = A-D ; DC = C-D
alpha = angle(AB,AD) ; beta = angle(BC,BA) ; gamma = angle(DA,DC) ; eta = angle(CD,CB)
print('alpha = ',alpha*360/(2*pi)) ; print('beta = ',beta*360/(2*pi)) ; print('gamma =
',gamma*360/(2*pi)) ; print('eta = ',eta*360/(2*pi))
a1 = exp(1j*alpha/4) ; a2 = exp(1j*beta/4) ; a3 = exp(1j*gamma/4) ; a4 = exp(1j*eta/4)
P = (a1*a2*A-a2*(A-B)-B)/(a1*a2-1) ; Q = (a2*a3*B-a3*(B-C)-C)/(a2*a3-1) ; R = (a3*a4*C-
a4*(C-D)-D)/(a3*a4-1) ; S = (a4*a1*D-a1*(D-A)-A)/(a4*a1-1)
print('P = ',P) ; print('g(f(P)) = ',g(f(P)))
print('Q = ',Q) ; print('h(g(Q)) = ',h(g(Q)))
print('R = ',R) ; print('m(h(R)) = ',m(h(R)))
print('S = ',S) ; print('f(m(S)) = ',f(m(S)))
print('norme cote PQ : ',norme(Q-P)) ; print('norme cote SR : ',norme(R-S))
print('milieux de PR et QS : ',(P+R)/2,'    ',(Q+S)/2)
print('P-Q = ', P-Q, ' S-R = ',S-R)
plt.plot([A.real,B.real,C.real,D.real,A.real],[A.imag,B.imag,C.imag,D.imag,A.imag],color='red')
plt.scatter(A.real,A.imag,color='red') ; plt.scatter(B.real,B.imag,color='red') ;
plt.scatter(C.real,C.imag,color='red') ; plt.scatter(D.real,D.imag,color='red')
plt.annotate('A',xy=(A.real+0.03,A.imag-0.03)) ; plt.annotate('B',xy=(B.real+0.03,B.imag-0.03)) ;
plt.annotate('C',xy=(C.real+0.03,C.imag-0.03)) ; plt.annotate('D',xy=(D.real+0.03,D.imag-0.03))
plt.plot([P.real,Q.real,R.real,S.real,P.real],[P.imag,Q.imag,R.imag,S.imag,P.imag],color='blue')
```

```
plt.scatter(P.real,P.imag,color='blue') ; plt.scatter(Q.real,Q.imag,color='blue') ;
plt.scatter(R.real,R.imag,color='blue') ; plt.scatter(S.real,S.imag,color='blue')
plt.annotate('P',xy=(P.real+0.03,P.imag+0.03)) ; plt.annotate('Q',xy=(Q.real+0.03,Q.imag+0.03)) ;
plt.annotate('R',xy=(R.real+0.03,R.imag+0.03)) ; plt.annotate('S',xy=(S.real+0.03,S.imag+0.03))
plt.show()
```




```
import cmath
from cmath import pi, exp,sqrt,acos
import matplotlib.pyplot as plt
import math
from math import atan2

def f(z):
    return(exp(1j*alpha/5)*(z-A)+A)
def g(z):
    return(exp(1j*beta/5)*(z-B)+B)
def h(z):
    return(exp(1j*gamma/5)*(z-C)+C)
def m(z):
    return(exp(1j*eta/5)*(z-D)+D)
def n(z):
    return(exp(1j*zita/5)*(z-E)+E)

def norme(v):
    return((sqrt(v.real*v.real+v.imag*v.imag)).real)

def prodscal(v1, v2):
    res = v1.real*v2.real+v1.imag*v2.imag
    return(res.real)
```

```
def determ(v1, v2):
    res = v1.real*v2.imag-v1.imag*v2.real
    return(res.real)

def angle(v1, v2):
    sinus = determ(v1, v2)/(norme(v1)*norme(v2))
    cosinus = prodscal(v1, v2)/(norme(v1)*norme(v2))
    return(-atan2(sinus,cosinus))

A = -0.5-0.5*1j ; B = 2*1j ; C = 2.5+2.5*1j ; D = 3.48+0*1j ; E = 1-1j
AB = B-A ; AE = E-A ; BC = C-B ; BA = A-B ; CD = D-C ; CB = B-C ; DE = E-D ; DC = C-D ;
EA = A-E ; ED = D-E
alpha = angle(AB,AE) ; beta = angle(BC,BA) ; gamma = angle(DE,DC) ; eta = angle(CD,CB) ; zita
= angle(EA,ED)
print('alpha = ',alpha*360/(2*pi)) ; print('beta = ',beta*360/(2*pi)) ; print('gamma =
',gamma*360/(2*pi)) ; print('eta = ',eta*360/(2*pi)) ; print('zita = ',zita*360/(2*pi))
a1 = exp(1j*alpha/5) ; a2 = exp(1j*beta/5) ; a3 = exp(1j*gamma/5) ; a4 = exp(1j*eta/5) ; a5 =
exp(1j*zita/5)
P = (a1*a2*A-a2*(A-B)-B)/(a1*a2-1) ; Q = (a2*a3*B-a3*(B-C)-C)/(a2*a3-1) ; R = (a3*a4*C-
a4*(C-D)-D)/(a3*a4-1) ; S = (a4*a5*D-a5*(D-E)-E)/(a4*a5-1) ; T =
(a5*a1*E-a1*(E-A)-A)/(a5*a1-1)
print('P = ',P) ; print('g(f(P)) = ',g(f(P)))
print('Q = ',Q) ; print('h(g(Q)) = ',h(g(Q)))
print('R = ',R) ; print('m(h(R)) = ',m(h(R)))
print('S = ',S) ; print('n(m(S)) = ',n(m(S)))
print('T = ',T) ; print('f(n(T)) = ',f(n(T)))
print('norme grand : ',norme(B-A)+norme(C-B)+norme(D-C)+norme(E-D)+norme(A-E))
print('norme petit : ',norme(Q-P)+norme(R-Q)+norme(S-R)+norme(T-S)+norme(P-T))
plt.plot([A.real,B.real,C.real,D.real,E.real,A.real],
[A.imag,B.imag,C.imag,D.imag,E.imag,A.imag],color='red')
plt.scatter(A.real,A.imag,color='red') ; plt.scatter(B.real,B.imag,color='red') ;
plt.scatter(C.real,C.imag,color='red') ; plt.scatter(D.real,D.imag,color='red') ;
plt.scatter(E.real,E.imag,color='red')
plt.annotate('A',xy=(A.real+0.03,A.imag-0.03)) ; plt.annotate('B',xy=(B.real+0.03,B.imag-0.03)) ;
plt.annotate('C',xy=(C.real+0.03,C.imag-0.03)) ; plt.annotate('D',xy=(D.real+0.03,D.imag-0.03)) ;
plt.annotate('E',xy=(E.real+0.03,E.imag-0.03))
plt.plot([P.real,Q.real,R.real,S.real,T.real,P.real],
[P.imag,Q.imag,R.imag,S.imag,T.imag,P.imag],color='blue')
plt.scatter(P.real,P.imag,color='blue') ; plt.scatter(Q.real,Q.imag,color='blue') ;
plt.scatter(R.real,R.imag,color='blue') ; plt.scatter(S.real,S.imag,color='blue') ;
plt.scatter(T.real,T.imag,color='blue')
plt.annotate('P',xy=(P.real+0.03,P.imag+0.03)) ; plt.annotate('Q',xy=(Q.real+0.03,Q.imag+0.03)) ;
plt.annotate('R',xy=(R.real+0.03,R.imag+0.03)) ; plt.annotate('S',xy=(S.real+0.03,S.imag+0.03)) ;
plt.annotate('T',xy=(T.real+0.03,T.imag+0.03))
plt.show()
```

Invite de commandes - pytho

```
Microsoft Windows [version 10.0.22631.4037]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\Denise_Vella>cd Desktop

C:\Users\Denise_Vella\Desktop>python Morley-Alain-quinti.py
alpha =  97.12501634890181
beta =  112.61986494804043
gamma =  90.55538458997793
eta =  100.09524695609832
zita =  139.60448715698152
P =  (-0.0010055129435253142+0.7984053010512225j)
g(f(P)) =  (-0.0010055129435252441+0.7984053010512222j)
Q =  (1.1556617091113246+2.0009670755104567j)
h(g(Q)) =  (1.1556617091113244+2.0009670755104563j)
R =  (2.805598567285899+1.1042095825340974j)
m(h(R)) =  (2.805598567285898+1.1042095825340974j)
S =  (1.9264775533491043-0.3288241397060157j)
n(m(S)) =  (1.9264775533491043-0.3288241397060161j)
T =  (0.43904556959888125-0.6440706983114082j)
f(n(T)) =  (0.4390455695988813-0.6440706983114082j)
norme grand :  12.039400329093045
norme petit :  8.256209986564848
```

Figure 1