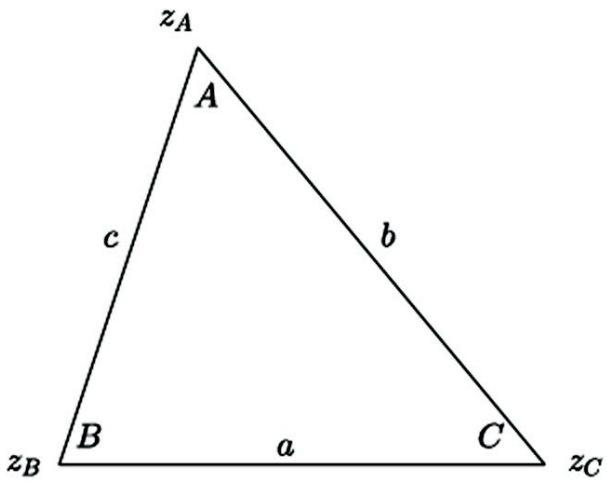
Tema 7

Puncte importante în triunghi

Ne vom ocupa aici de punctele remarcabile întâlnite în geometria triunghiului: centru de greutate G, centrul cercului circumscris O, etc.



Vom nota şi acum cu z_A, z_B şi z_C afixele vârfurilor triunghiului ΔABC, cu a, b şi c lungimile laturilor BC, CA şi AB, iar cu A, B şi C mărimea unghiurilor triunghiului, măsurată în radiani şi cuprinsă strict în intervalul $(0,\pi)$.

Amintim că numerele reale α, β şi γ formează un set de coordonate baricentrice omogene pentru un punct Q din planul triunghiului ΔABC dacă afixul său se calculează cu relaţia

$$z_Q = \frac{\alpha z_A + \beta z_B + \gamma z_C}{\alpha + \beta + \gamma}.$$

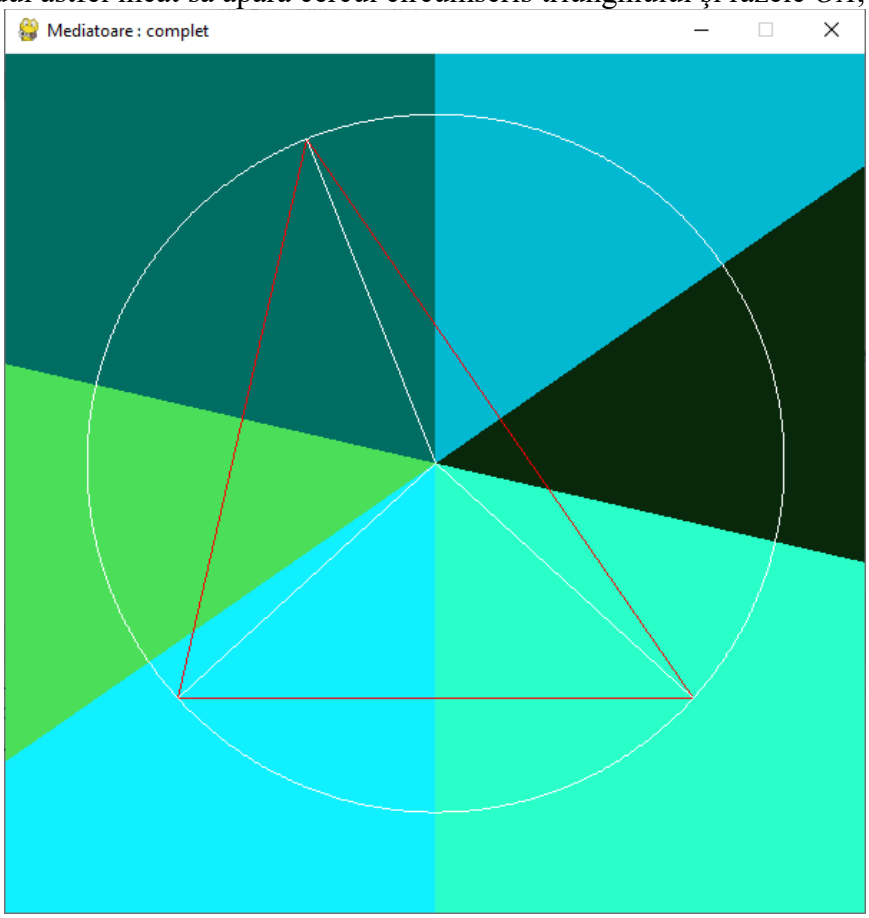
Pentru a afla cum se determină aceste coordonate, vezi tutorialul *Coordonate baricentrice*.

Exerciţiul 1. **Centrul cercului circumscris triunghiului.** Următorul program pune în evidenţă punctul de intersecţie al mediatoarelor laturilor unui triunghi care, după cum se ştie, este punctul O, centrul cercului circumscris:

```
import ComplexPygame as C
import Color
import math


def unCercQA(q, a, N):
    theta = 2 * math.pi / N
    return [q + C.fromRhoTheta(1, k * theta) * (a - q) for k in range(N)]
```

```
def Mediatoare():
    C.setXminXmaxYminYmax(0, 10, 0, 10)
    zA = 2 + 2.5j
    zB = 8 + 2.5j
    zC = 3.5 + 9j
    for z in C.screenAffixes():
        za = C.rho(z - zA)
        zb = C.rho(z - zB)
        zc = C.rho(z - zC)
        k = 0
        if za < zb:
            k += 1
        if zb < zc:
            k += 2
        if zc < za:
            k += 4
        C.setPixel(z, Color.Index(600 + 50 * k))
    C.drawNgon([zA, zB, zC], Color.Red)


if __name__ == '__main__':
    C.initPygame()
    C.run(Mediatoare)
```

Completaţi codul astfel încât să apară cercul circumscris triunghiului şi razele *OA*, *OB* şi *OC*:



Indicaţie: *Metoda I*. Avem $z_O=(\alpha z_A+\beta z_B+\gamma z_C)/(\alpha+\beta+\gamma)$ unde ponderele α, β şi γ sunt date de relaţiile

$$\alpha=a^2(b^2+c^2-a^2),\beta=b^2(c^2+a^2-b^2),\gamma=c^2(a^2+b^2-c^2),$$

sau de relaţiile

$$\alpha=\sin 2A,\beta=\sin 2B,\gamma=\sin 2C.$$

Metoda a II-a. Se foloseşte triunghiul ΔBOC care este isoscel cu unghiul $\widehat{BOC}=2A$.

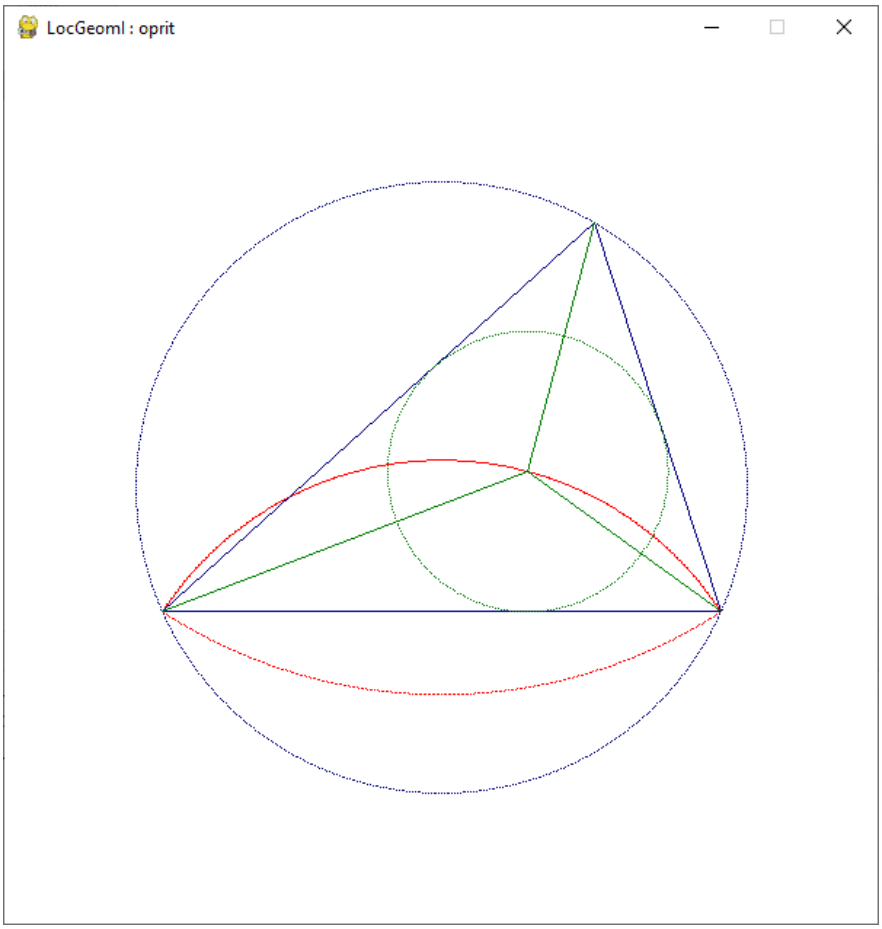
Exerciţiul 2. Centrul cercului înscris în triunghi. Următorul program pune în evidenţă, printr-o animaţie grafică, locul geometric al punctului I, centrul cercului înscris în triunghiul ΔABC, atunci când vârful A se mişcă pe un cerc dat iar B şi C sunt fixate arbitrar pe cerc.

Funcţia
`cercInscris(zA, zB, zC)`
calculează afixul lui I cu formula $z_I=(a z_A+b z_B+c z_C)/(a+b+c)$ unde a, b şi c sunt lungimile laturilor BC, CA şi, respectiv, AB. Notând cu p semiperimetrul $p=(a+b+c)/2$, raza r a cercului înscris se găseşte din relaţia $S=pr$ unde aria S a triunghiului este calculată cu formula lui Heron $S=\sqrt{p(p-a)(p-b)(p-c)}$.

```
import ComplexPygame as C
import Color
import math


def LocGeomI():
    def unCercQR(q, r, N):
        alfa = 2 * math.pi / N
        return [q + C.fromRhoTheta(r, k * alfa) for k in range(N)]

    def cercInscris(zA, zB, zC):
        # returneaza zI si r pentru cercul inscris
        a = C.rho(zB - zC)
        b = C.rho(zC - zA)
        c = C.rho(zA - zB)
        p = (a + b + c) / 2
        S = math.sqrt(p * (p - c) * (p - b) * (p - a))
        zI = (a * zA + b * zB + c * zC) / (a + b + c)
        r = S / p
        return zI, r

    C.setXminXmaxYminYmax(-10, 10, -10, 10)
    q = 0
    R = 7
    nrPuncte = 720
    delta = 2 * math.pi / nrPuncte
    nB = nrPuncte // 2 + nrPuncte // 15
    nC = nrPuncte - nrPuncte // 15
    zB = C.fromRhoTheta(R, nB * delta)
    zC = C.fromRhoTheta(R, nC * delta)
```

```
    for n in range(10 * nrPuncte):
        C.fillScreen(Color.White)
        C.setNgon(unCercQR(q, R, nrPuncte), Color.Navy)
        zA = C.fromRhoTheta(R, n * delta)
        C.drawNgon([zA, zB, zC], Color.Navy)
        zI, r = cercInscris(zA, zB, zC)
        C.setNgon(unCercQR(zI, r, 300), Color.Green)
        C.drawNgon([zA, zI, zB, zI, zC, zI], Color.Green)
        if C.mustClose():
            return


if __name__ == '__main__':
    C.initPygame()
    C.run(LocGeomI)
```

Adăugaţi codul necesar pentru a pune în evidenţă, colorate în roşu, cele două arce de cerc care formează locul geometric al lui *I* (demonstraţi!), şi care să rămână nemişcate în timpul animaţiei:



Exerciţiul 3. Ortocentrul. Vrem să evidenţiem acum locul geometric al ortocentrului H al triunghiului ΔABC, atunci când vârful A se mişcă pe un cerc dat iar B şi C sunt fixate arbitrar pe cerc. În programul următor trebuie implementată funcția

```
ortocentru(zA, zB, zC)
```

astfel încât aceasta să calculeze afixul lui H cu formula $z_H = (\alpha z_A + \beta z_B + \gamma z_C)/(\alpha+\beta+\gamma)$

unde ponderele α, β şi γ sunt date de relaţiile

$$\alpha = (c^2+a^2-b^2)(a^2+b^2-c^2)$$

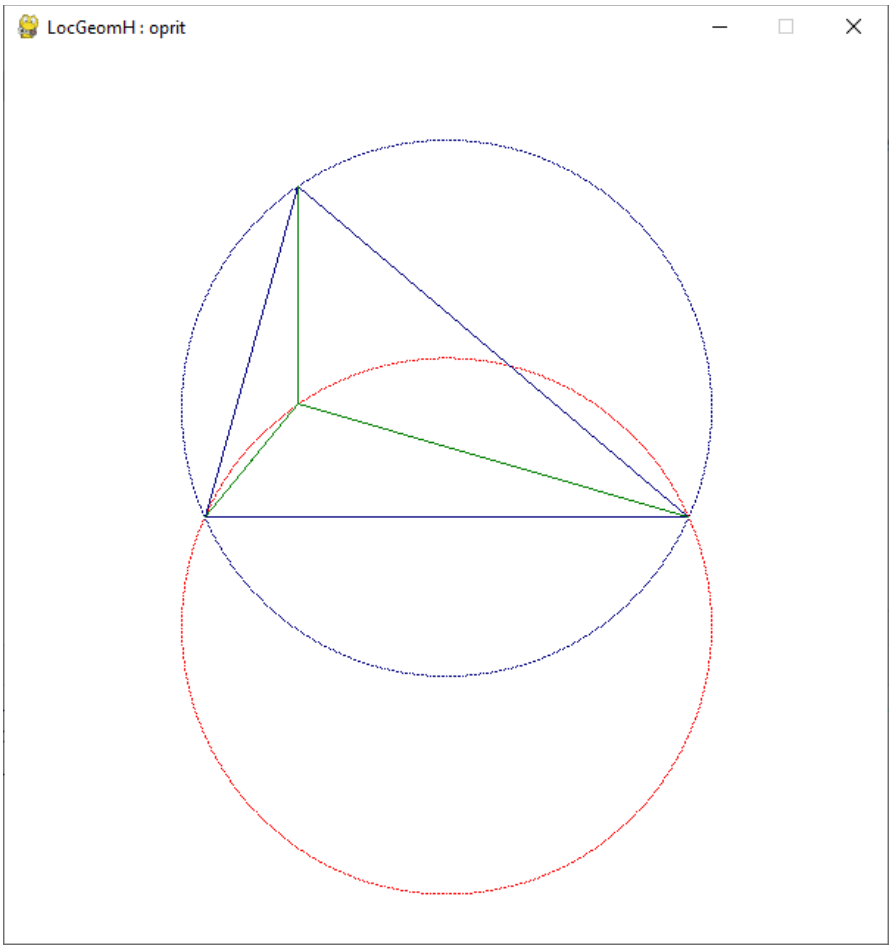
$$\beta = (a^2+b^2-c^2)(b^2+c^2-a^2),$$

$$\gamma = (b^2+c^2-a^2)(c^2+a^2-b^2).$$

```
import ComplexPygame as C
import Color
import math
def LocGeomH():
    def unCercQR(q, r, N):
        alfa = 2 * math.pi / N
        return [q + C.fromRhoTheta(r, k * alfa) for k in range(N)]

    def ortocentru(zA, zB, zC):
        # returneaza afixul zH al ortocentrului
        zH = 0 #  trebuie corectat!
        return zH

    C.setXminXmaxYminYmax(-10, 10, -12, 8)
    q = 0
    R = 6
    nrPuncte = 720
    delta = 2 * math.pi / nrPuncte
    nB = nrPuncte // 2 + nrPuncte // 15
    nC = nrPuncte - nrPuncte // 15
    zB = C.fromRhoTheta(R, nB * delta)
    zC = C.fromRhoTheta(R, nC * delta)
    for n in range(10 * nrPuncte):
        if n % nrPuncte == nB or n % nrPuncte == nC:
            continue
        C.fillScreen(Color.White)
        C.setNgon(unCercQR(q, R, nrPuncte), Color.Navy)
        zA = C.fromRhoTheta(R, n * delta)
        C.drawNgon([zA, zB, zC], Color.Navy)
        zH = ortocentru(zA, zB, zC)
        C.drawNgon([zA, zH, zB, zH, zC, zH], Color.Green)
        if C.mustClose():
            return
```

```
if __name__ == '__main__':
    C.initPygame()
    C.run(LocGeomH)
```

Adăugaţi şi codul necesar pentru a pune în evidenţă, colorat în roşu, cercul care formează locul geometric al lui H (demonstraţi!), şi care să rămână nemişcat pe parcursul animaţiei:



Exerciţiul 4. Punctul lui Gergonne. Fie A', B' şi C' punctele de contact cu laturile BC, CA şi, respectiv, AB ale cercului înscris în triunghiul ΔABC. Notăm cu p semiperimetrul

$$p=(a+b+c)/2.$$

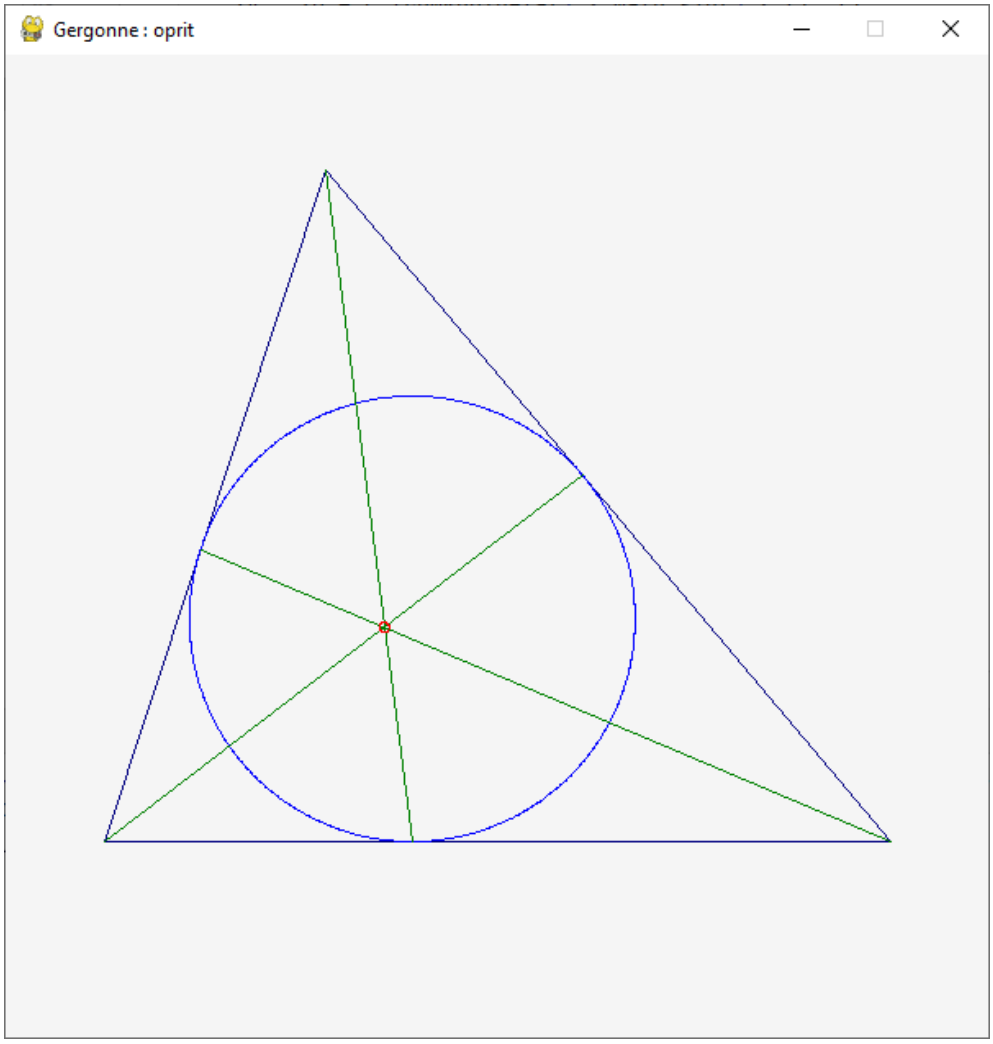
a) arătaţi că $BA'=BC'=p-b$, $CB'=CA'=p-c$, $AC'=AB'=p-a$,

b) demonstraţi că dreptele AA', BB', CC' sunt concurente într-un punct J, numit *punctul lui Gergonne.*

c) aflaţi coordonatele baricentrice ale punctului J şi corectaţi programul următor astfel încât cerculeţul roşu să fie centrat mereu în intersecţia celor trei ceviene verzi:

```
import ComplexPygame as C
import Color
import math


def Gergonne():
    def unCercQR(q, r, N):
        alfa = 2 * math.pi / N
        return [q + C.fromRhoTheta(r, k * alfa) for k in range(N)]

    def pctGergonne(zA, zB, zC):
        # returneaza afixul zJ al punctului Gergonne
        alfa = 1  # de corectat!
        beta = 1  # de corectat!
        gama = 1  # de corectat!
        zJ = (alfa * zA + beta * zB + gama * zC) / (alfa + beta + gama)
        return zJ

    def deseneazaTriunghi(zA, zB, zC):
        a = C.rho(zB - zC)
        b = C.rho(zC - zA)
        c = C.rho(zA - zB)
        p = (a + b + c) / 2
        S = math.sqrt(p * (p - c) * (p - b) * (p - a))
        zI = (a * zA + b * zB + c * zC) / (a + b + c)
        r = S / p

        zAprim = ((p - b) * zC + (p - c) * zB) / a
        zBprim = ((p - c) * zA + (p - a) * zC) / b
        zCprim = ((p - a) * zB + (p - b) * zA) / c
        C.drawNgon([zA, zB, zC], Color.Navy)
        C.setNgon(unCercQR(zI, r, 1000), Color.Blue)
        C.drawLine(zA, zAprim, Color.Green)
        C.drawLine(zB, zBprim, Color.Green)
        C.drawLine(zC, zCprim, Color.Green)

    C.setXminXmaxYminYmax(0, 10, 0, 10)
    zB = 1 + 2j
    zC = 9 + 2j
    zQ = 5 + 8j
    t = 0
    while t < 1000:
        C.fillScreen()
        zA = zQ + C.fromRhoTheta(2 * math.sin(3 * t), t)
        deseneazaTriunghi(zA, zB, zC)
        zJ = pctGergonne(zA, zB, zC)
        C.setNgon(unCercQR(zJ, 0.05, 100), Color.Red)
        if C.mustClose():
            return
```

```
        t += 0.001

if __name__ == '__main__':
    C.initPygame()
    C.run(Gergonne)
```



Indicaţie: a) Se foloseşte faptul că tangentele dintr-un punct la un cerc dat sunt egale.
b) Se aplică teorema lui Ceva.
c) Cu notaţiile standard, avem

$$\frac{\sigma_c}{\sigma_b}=\frac{p-b}{p-c}\Rightarrow\frac{\sigma_c}{\frac{1}{p-c}}=\frac{\sigma_b}{\frac{1}{p-b}}$$

de unde deducem următorul set de coordonate omogene pentru punctul lui Gergonne:

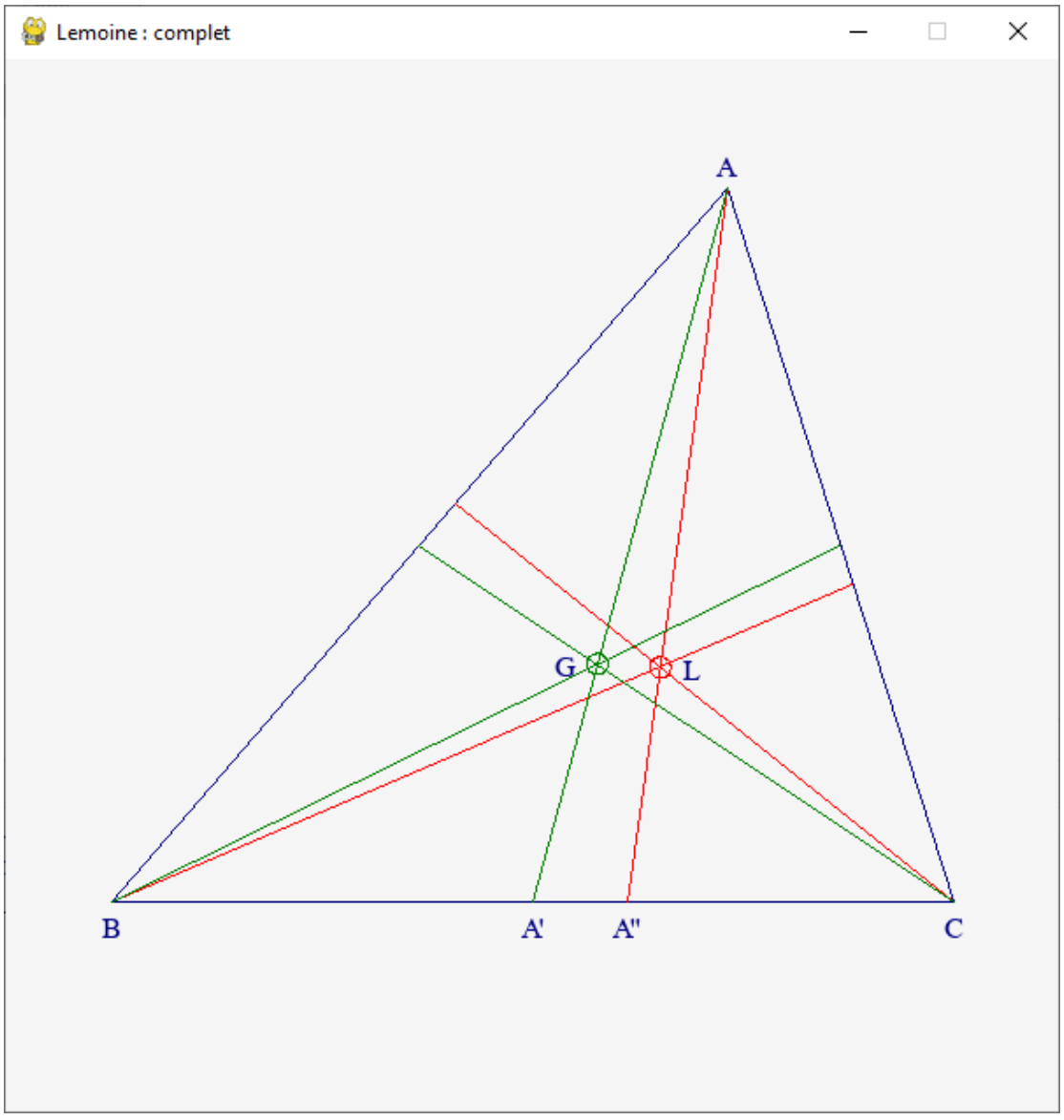
$$\frac{1}{p-a}:\frac{1}{p-b}:\frac{1}{p-c}$$

Exerciţiul 5. Punctul lui Lemoine. Două ceviene AA' şi AA'' duse prin vârful A al unui triunghi ΔABC se numesc *izogonale* dacă unghiul făcut de una dintre ele cu o latură este egal cu unghiul făcut de cealaltă ceviană cu cealaltă latură a unghiului A, mai precis, dacă $\widehat{BAA'} = \widehat{A\,AC}$. Izogonala unei mediane se numeşte *simediană*.

a) arătaţi că simediana împarte latura opusă într-un raport egal cu pătratul raportului celorlalte două laturi ale triunghiului.

b) demonstraţi că simedianele unui triunghi sunt concurente într-un punct L, numit *punctul lui Lemoine*.

c) Puneţi în evidenţă această proprietate (printr-o animaţie grafică, eventual) marcând cu un cerculeţ verde centrul de greutate şi cu unul roşu punctul lui Lemoine al unui triunghi oarecare:



Indicaţie: a) Fie AA'' simediana corespunzătoare medianei AA'. Notăm $\beta = \widehat{BAA'} = \widehat{A''AC}$ şi $\gamma = \widehat{CAA'} = \widehat{A''AB} = A - \beta$ şi exprimăm cu arii rapoartele în care cevienele AA'' şi AA' împart latura BC. Avem

$$\frac{BA''}{A''C} = \frac{\sigma_{\Delta BAA''}}{\sigma_{\Delta CAA''}} = \frac{\frac{1}{2} AB \cdot AA'' \cdot \sin \widehat{BAA''}}{\frac{1}{2} AC \cdot AA'' \cdot \sin \widehat{CAA''}} = \frac{c \sin \gamma}{b \sin \beta}$$

şi

$$1 = \frac{BA'}{A'C} = \frac{\sigma_{\Delta BAA'}}{\sigma_{\Delta CAA'}} = \frac{\frac{1}{2} AB \cdot AA' \cdot \sin \widehat{BAA'}}{\frac{1}{2} AC \cdot AA' \cdot \sin \widehat{CAA'}} = \frac{c \sin \beta}{b \sin \gamma}.$$

Înmulţind aceste două rapoarte, obţinem relaţia căutată:

$$\frac{BA''}{A''C} = \frac{c^2}{b^2}.$$

b) Se aplică teorema lui Ceva, folosind punctul a).

c) Punctul A'' se poziţionează cunoscând raportul în care acesta împarte segmentul BC. Pentru aflarea afixului punctului L se folosesc coordonatele lui baricentrice, care se determină analog cu coordonatele centrului cercului înscris (vezi tutorialul *Coordonate baricentrice*).

Exerciţiul 6. a) Fie Q un punct în interiorul triunghiului ΔABC şi fie AA', BB', CC' cevienele prin Q, conform figurii. Determinaţi poziţia lui Q cunoscând poziţiile vârfurilor triunghiului şi rapoartele

$$v_B = \frac{AB'}{AC}, v_C = \frac{AC'}{AB}.$$

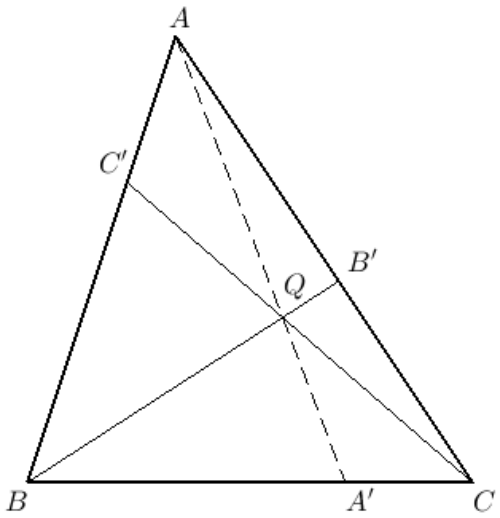
Rezolvare. *Metoda I*, aflarea afixului z_Q prin determinarea coordonatelor baricentrice ale lui Q este lăsată cititorului (vezi tutorialul).

Metoda a II-a. Pentru a poziţiona punctul Q pe AA' vom folosi relaţia lui Van Aubel

$$\frac{AQ}{QA'} = \frac{AB'}{B'C} + \frac{AC'}{C'B}$$

iar pentru al poziţiona pe A' pe BC vom folosi teorema lui Ceva:

$$\frac{AC'}{C'B}\cdot\frac{BA'}{A'C}\cdot\frac{CB'}{B'A}=1.$$



Avem următoarele calcule

$$\frac{AB'}{B'C}=\frac{AB'}{AC-AB'}=\frac{\frac{AB'}{AC}}{1-\frac{AB'}{AC}}=\frac{v_B}{1-v_B},\frac{AC'}{C'B}=\frac{v_C}{1-v_C}.$$

Prin urmare

$$\lambda_Q=\frac{AQ}{QA'}=\frac{v_B}{1-v_B}+\frac{v_C}{1-v_C}$$

şi

$$\lambda_A=\frac{BA'}{A'C}=\frac{C'B}{AC'}\cdot\frac{AB'}{B'C}=\frac{1-v_C}{v_C}\cdot\frac{v_B}{1-v_B}.$$

Mai departe, afixele punctelor A' şiQ se determină astfel

$$\lambda_A=\frac{z_{A'}-z_B}{z_C-z_{A'}}\Rightarrow z_{A'}=\frac{z_B+\lambda_A z_C}{1+\lambda_A}$$

şi

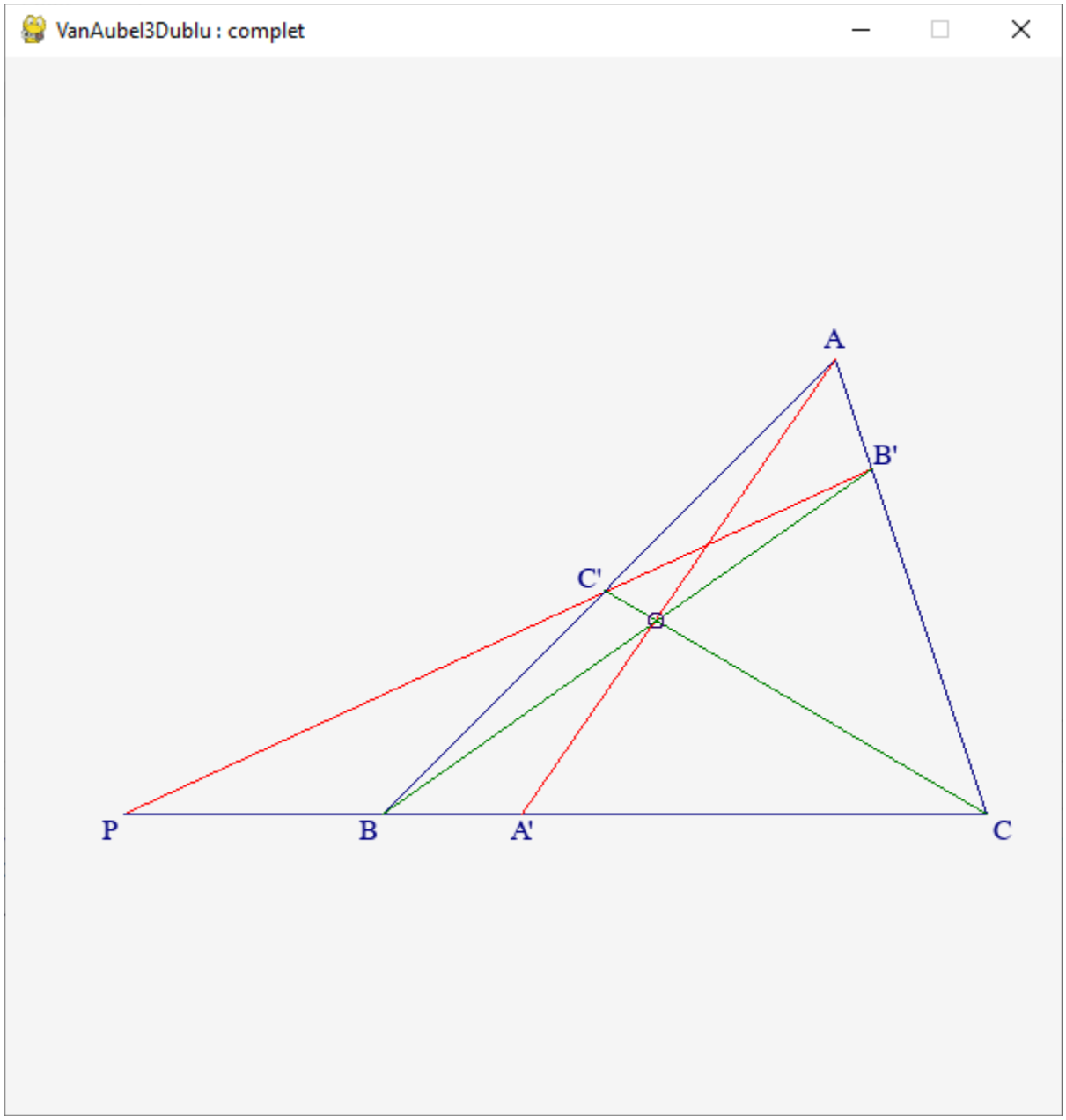
$$\lambda_Q=\frac{z_Q-z_A}{z_{A'}-z_Q}\Rightarrow z_Q=\frac{z_A+\lambda_Q z_{A'}}{1+\lambda_Q}$$

b) Următorul program pune în evidenţă locul geometric al punctului Q atunci când segmentul $B'C'$se mişcă rămânând tot timpul paralel cu BC.

```
import ComplexPygame as C
import Color
import math
```

```
def VanAubel3():
    def unCercQR(q, r, N):
        alfa = 2 * math.pi / N
        return [q + C.fromRhoTheta(r, k * alfa) for k in range(N)]

    def intersectie(zA, zB, zC, niuB, niuC):
        # niuB=AB'/AC     niuC=AC'/AB
        lambdaA = niuB * (1 - niuC) / (niuC * (1 - niuB))
        # lambdaA =BA'/A'C
        zAprim = (zB + lambdaA * zC) / (1 + lambdaA)
        lambdaQ = niuB / (1 - niuB) + niuC / (1 - niuC)
        # lambdaQ=AQ/QA'
        zQ = (zA + lambdaQ * zAprim) / (1 + lambdaQ)
        return zAprim, zQ

    C.setXminXmaxYminYmax(0, 10, 0, 10)
    zA, zB, zC = 7 + 8j, 1 + 2j, 9 + 2j
    t = 0
    while t < 10:
        t += 0.01
        niuCprim = 0.05 + 0.45 * (1 + math.sin(t))
        niuBprim = niuCprim
        zCprim = zA + niuCprim * (zB - zA)
        zBprim = zA + niuBprim * (zC - zA)
        zAprim, zQ = intersectie(zA, zB, zC, niuBprim, niuCprim)

        C.fillScreen()
        C.drawLine(zA, zAprim, Color.Red)
        C.drawLine(zB, zBprim, Color.Green)
        C.drawLine(zC, zCprim, Color.Green)
        C.drawNgon(unCercQR(zQ, 0.1, 100), Color.Red)
        C.drawLine(zCprim, zBprim, Color.Red)
        C.drawNgon([zA, zB, zC], Color.Navy)

        C.setText("A", zA)
        C.setText("B", zB - 0.2 - 0.1j)
        C.setText("C", zC + 0.2 - 0.1j)
        C.setText("A'", zAprim - 0.4j)
        C.setText("B'", zBprim + 0.2 - 0.1j)
        C.setText("C'", zCprim - 0.2 - 0.1j)
        C.wait(10)
        if C.mustClose():
            break


if __name__ == '__main__':
    C.initPygame()
    C.run(VanAubel3)
```

Rulaţi programul şi apoi modificaţi-l pentru a arăta locul geometric al punctului Q atunci când dreapta $B'C'$ trece tot timpul printr-un punct fix P situat pe dreapta BC, în exteriorul triunghiului.



Indicaţie. Fixaţi punctul Ppe BCalegând un raport $PB/BC=2/3$, de exemplu. Lăsaţi punctul B' să se mişte periodic pe ACca în programul dat, determinaţi mai întâi poziţia lui C' ca intersecţie de două ceviene în triunghiul ΔCPA (sau, prin Teorema lui Menelaus, ca intersecţie dintre secanta PB' cu latura ABa triunghiuluiΔABC) şi apoi determinaţi poziţia lui Q ca intersecţie de două ceviene în ΔABC.

Exerciţiul 7. Pe laturile triunghiului ΔABC se construiesc în exterior triunghiurile $\Delta AC'B$, $\Delta BA'C$, şi $\Delta CB'A$ astfel încât $\widehat{BAC'}=\widehat{CAB'}=\alpha, \widehat{ABC'}=\widehat{CBA'}=\beta, \widehat{BCA'}=\widehat{ACB'}=\gamma$. Puneţi în evidenţă printr-o animaţie grafică (şi, eventual demonstraţi) că dreptele AA', BB', CC' sunt concurente. De exemplu: ţineţi fix triunghiul ΔABC şi variaţi mărimea lui $\alpha \in \left(0, \frac{\pi}{2}\right)$.

Indicaţie : Exprimăm cu arii rapoartele în care cevienele AA',BB',CC' împart laturile triunghiu-lui:

$$\frac{BA''}{A''C}=\frac{\sigma_{\Delta ABA'}}{\sigma_{\Delta ACA'}}=\frac{\frac{1}{2}AB\cdot BA'\cdot\sin\widehat{ABA'}}{\frac{1}{2}AC\cdot CA'\cdot\sin\widehat{ACA'}}=\frac{c}{b}\cdot\frac{BA'}{CA'}\cdot\frac{\sin(B+\beta)}{\sin(C+\gamma)}=\frac{c}{b}\cdot\frac{\sin\gamma}{\sin\beta}\cdot\frac{\sin(B+\beta)}{\sin(C+\gamma)}$$

Aici am folosit teorema sinusurilor în $\Delta BA'C$

$$\frac{BA'}{\sin\gamma}=\frac{CA'}{\sin\beta}\Rightarrow\frac{BA'}{CA'}=\frac{\sin\gamma}{\sin\beta}.$$

Mai departe se aplică teorema lui Ceva.