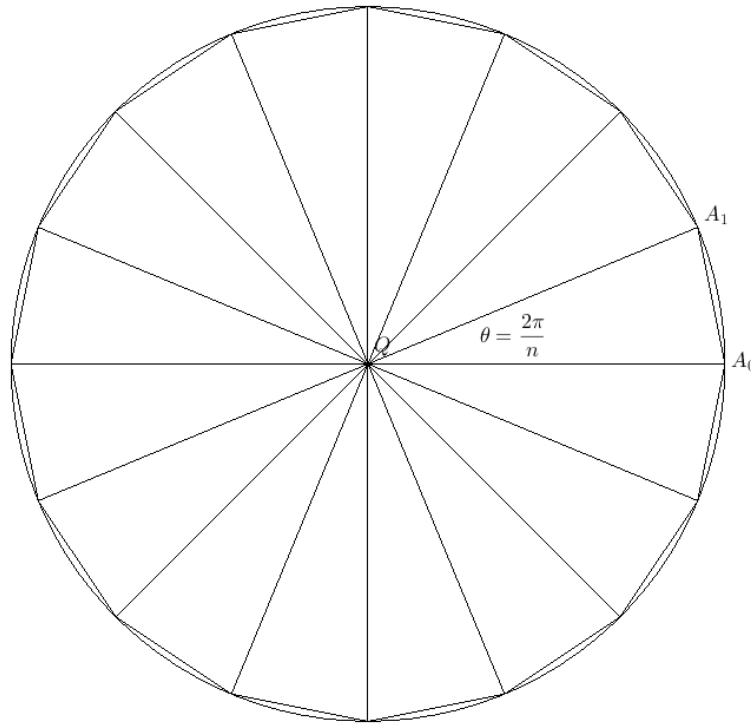


## Tema 8

### Poligoane regulate

Numim *poligon regulat* un poligon cu toate unghiurile de aceeași măsură și toate laturile de aceeași lungime. Orice poligon regulat cu  $n$  laturi se obține prin împărțirea unui cerc în  $n$  arce egale. Dacă parcurgem cercul mereu în același sens și unim în ordine cele  $n$  puncte obținem un *poligon regulat convex* cu  $n$  laturi, numit și  $n$ -gon regulat, iar dacă le unim pe sărite, din  $p$  în  $p$ , cu  $p$  prim cu  $n$ , obținem un *poligon stelat* cu  $n$  laturi, mai precis un  $n$ - $p$ -gon regulat.



Fie  $Q$  centrul cercului circumscris unui poligon regulat convex cu  $n$  laturi și fie  $A_0$  primul vârf al poligonului. Notăm cu litere mici,  $q$  și  $a_0$ , afixele lor. Unghiul la centru sub care sunt văzute laturile este  $\theta = \frac{2\pi}{n}$ . Celelalte vârfuri ale poligonului se obțin rotind pe  $A_0$  în jurul lui  $Q$  pe rând cu unghiurile  $\theta, 2\theta, 3\theta, \dots$ , deci afixele lor sunt

$$a_k = q + \varepsilon^k (a_0 - q), k = 1, 2, \dots, n-1;$$

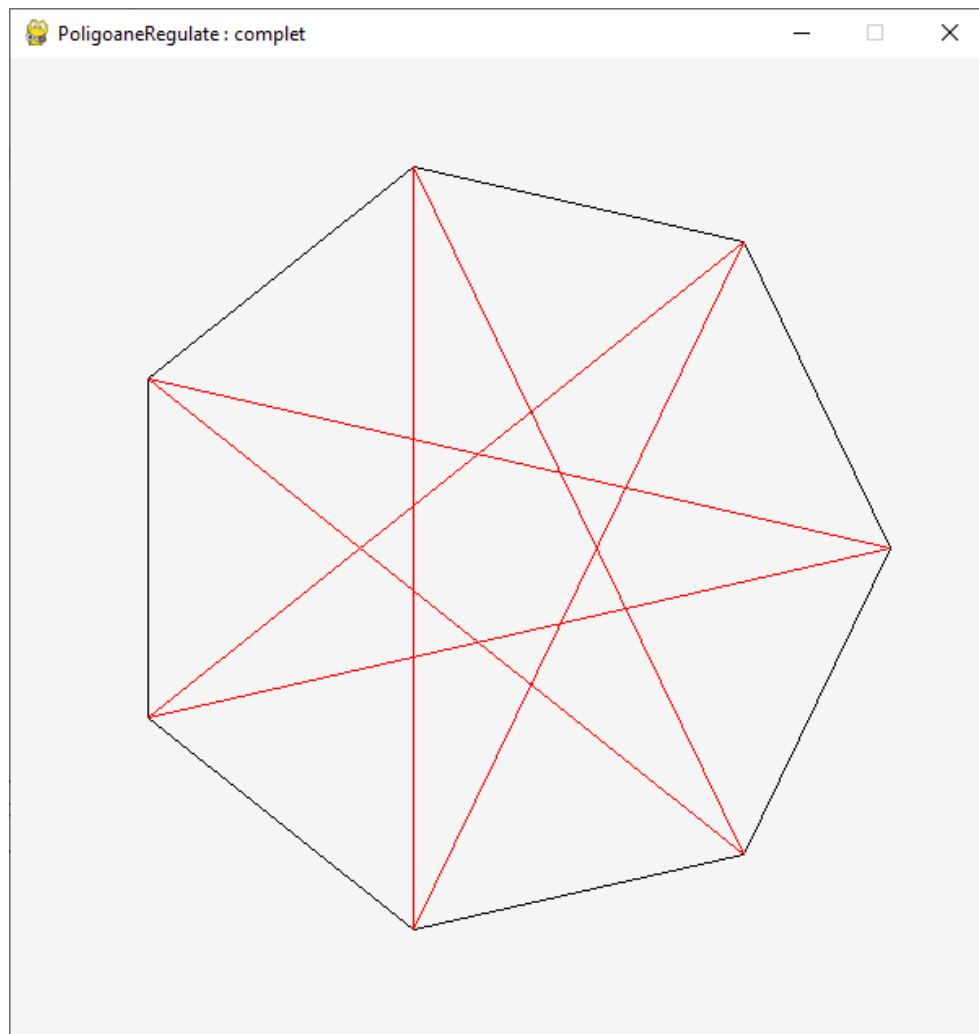
unde  $\varepsilon$  este prima rădăcină de ordin  $n$  a unității,  $\varepsilon = \cos \theta + i \sin \theta$ , și, prin urmare,

$$\varepsilon^k = \cos k\theta + i \sin k\theta.$$

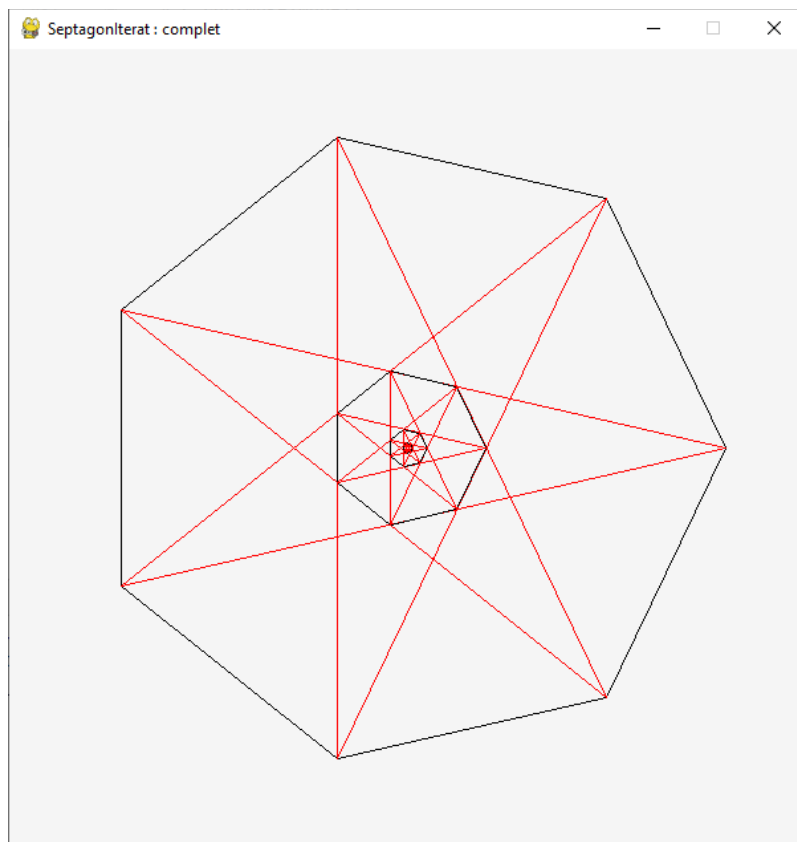
Poligonul stelat “din  $p$  în  $p$ ”,  $n$ - $p$ -gon-ul regulat, se obține considerând  $\theta = p \cdot \frac{2\pi}{n}$ .

**Exercițiul 1.** Urmărirea funcției trasează cu negru un heptagon regulat convex și cu culoarea roșie 7-3-gon-ul stelat corespunzător:

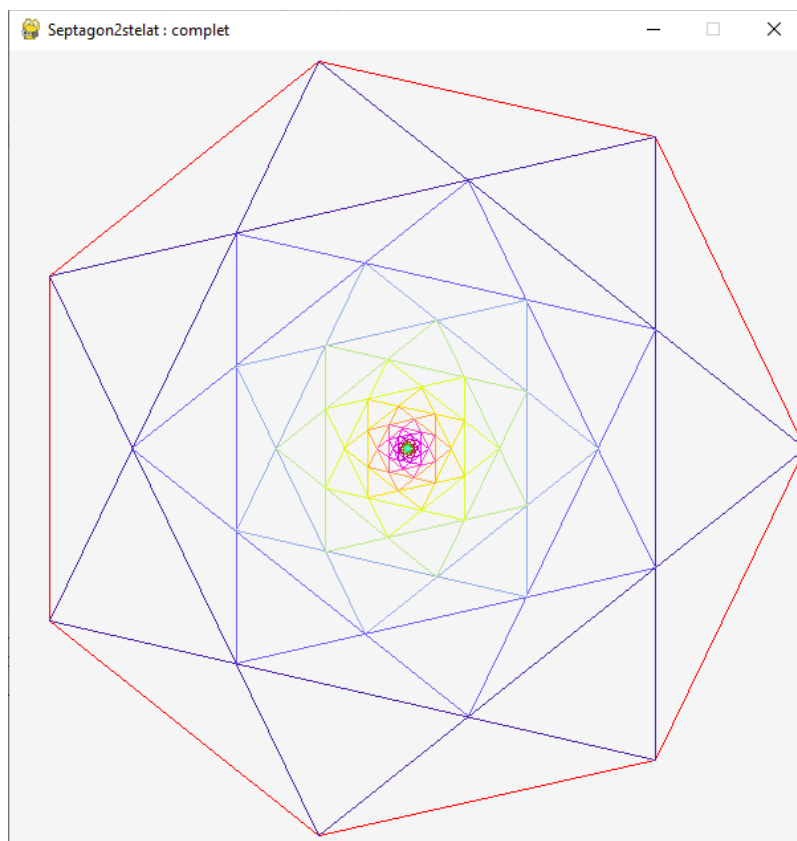
```
def PoligoaneRegulate():  
    def npGonQA(q, a, n, p=1):  
        # returneaza n-p-gonul cu centru q si primul varf a  
        theta = p * 2.0 * math.pi / n  
        return [q + C.fromRhoTheta(1, k * theta) * (a - q) for k in range(n)]  
  
    C.setXminXmaxYminYmax(-10, 10, -10, 10)  
    C.fillScreen()  
    q = 0  
    a = 8  
    n = 7  
    C.drawNgon(npGonQA(q, a, n), Color.Black)  
    C.drawNgon(npGonQA(q, a, n, 3), Color.Red)
```



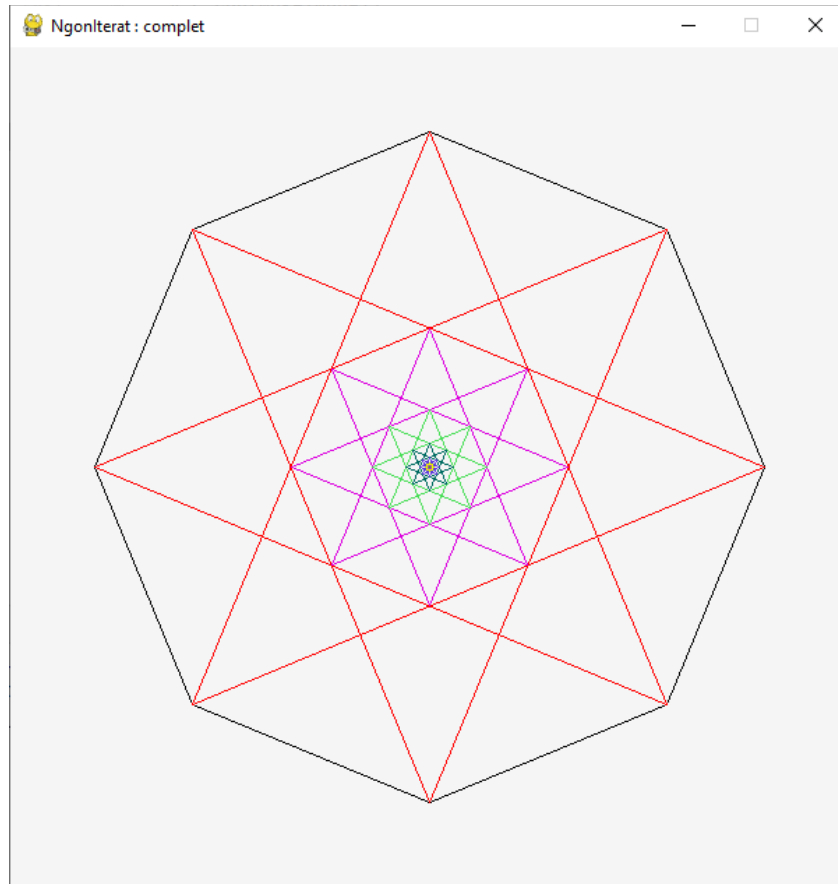
Repetăți această construcție în heptagonul convex format în interiorul celui stelat:



**Exercițiul 2.** Iterați acum un 7-2-gon:



**Exercițiul 3.** Încercați și cu un octogon regulat:



**Exercițiul 4.** Programul următor construiește pentagoane regulate pe laturile unui heptagon regulat;

```
import ComplexPygame as C
import Color
import math
def HeptaPentagon():
    def npGonQA(q, a0, n, p=1):
        theta = p * 2.0 * math.pi / n
        return [q + C.fromRhoTheta(1, k * theta) * (a0 - q) for k in range(n)]

    def bazaApex(zB, zC, uA, peStg=True):
        # calculeaza apexul zA al triunghiului isoscel zB zA zC
        omegaA = C.fromRhoTheta(1, uA) if peStg else C.fromRhoTheta(1, -uA)
        zA = (zC - omegaA * zB) / (1 - omegaA)
        return zA

    C.setXminXmaxYminYmax(-10, 10, -10, 10)
    C.fillScreen()
```

```

q = 0
a = 2
nInt = 7 # poligonul interior este un heptagon
pInt = npGonQA(q, a, nInt)
pInt.append(pInt[0])

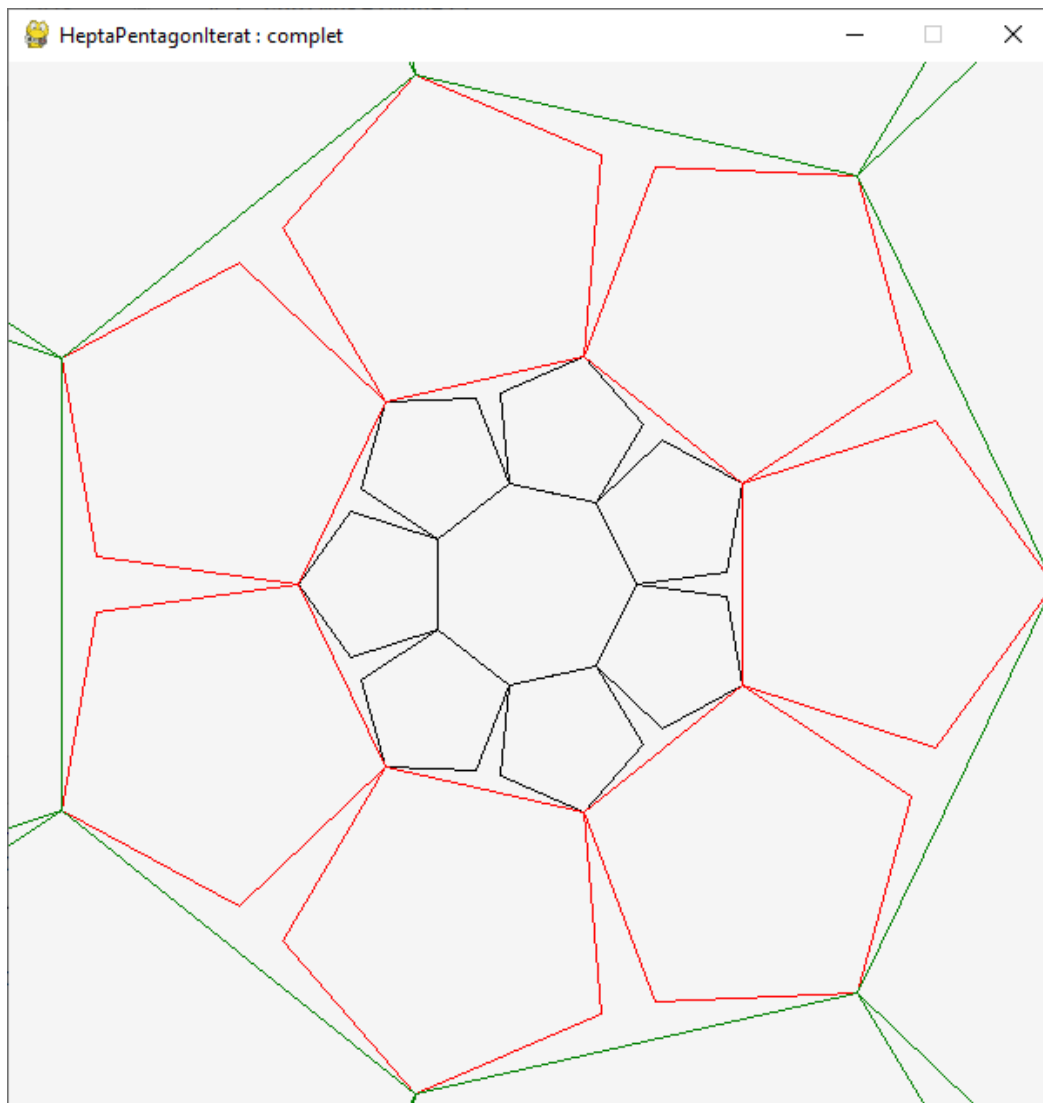
nExt = 5 # la exterior construim pentagoane
thetaExt = 2 * math.pi / nExt

for k in range(nInt):
    qk = bazaApex(pInt[k], pInt[k + 1], thetaExt, False)
    C.drawNgon(npGonQA(qk, pInt[k], nExt), Color.Black)

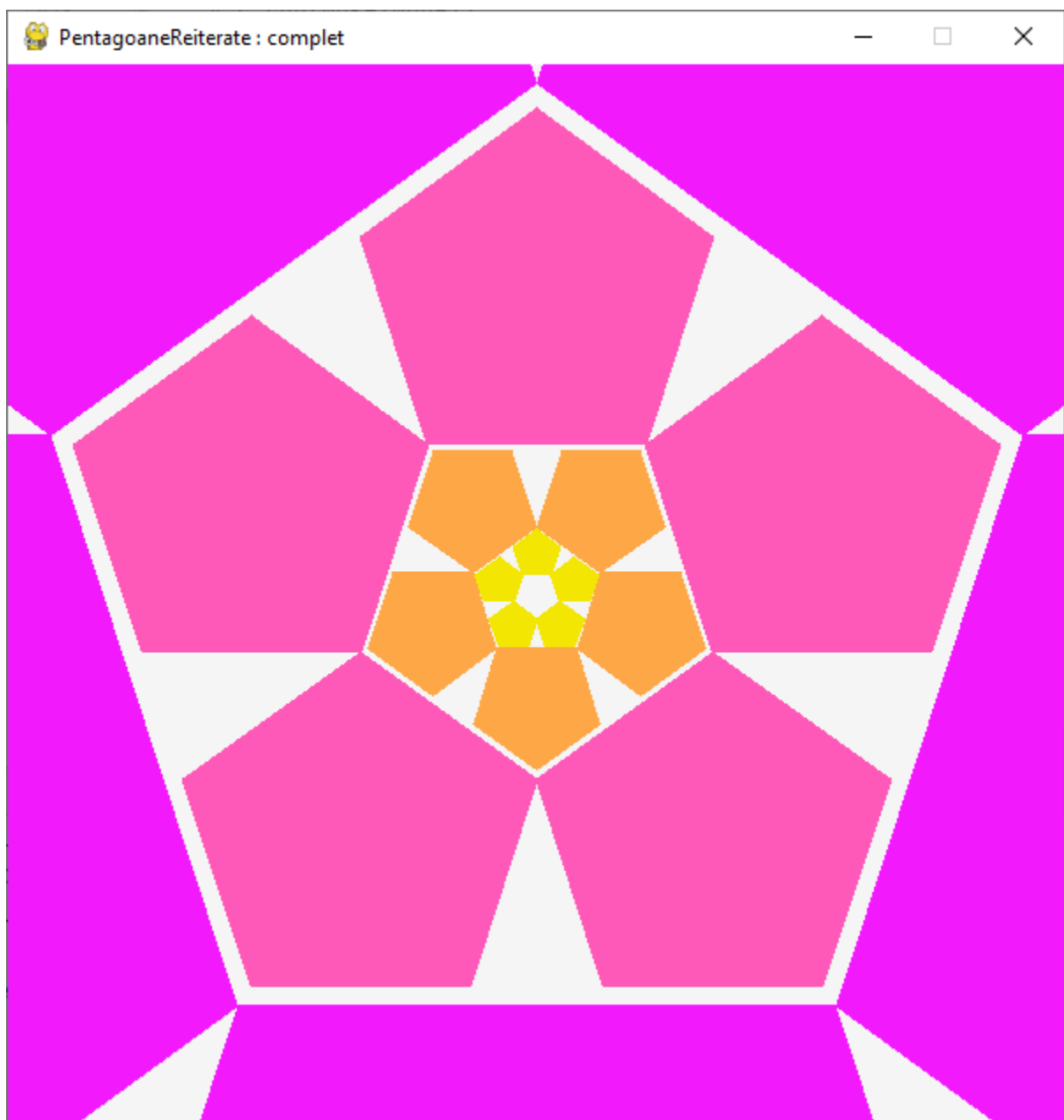
if __name__ == '__main__':
    C.initPygame()
    C.run(HeptaPentagon)

```

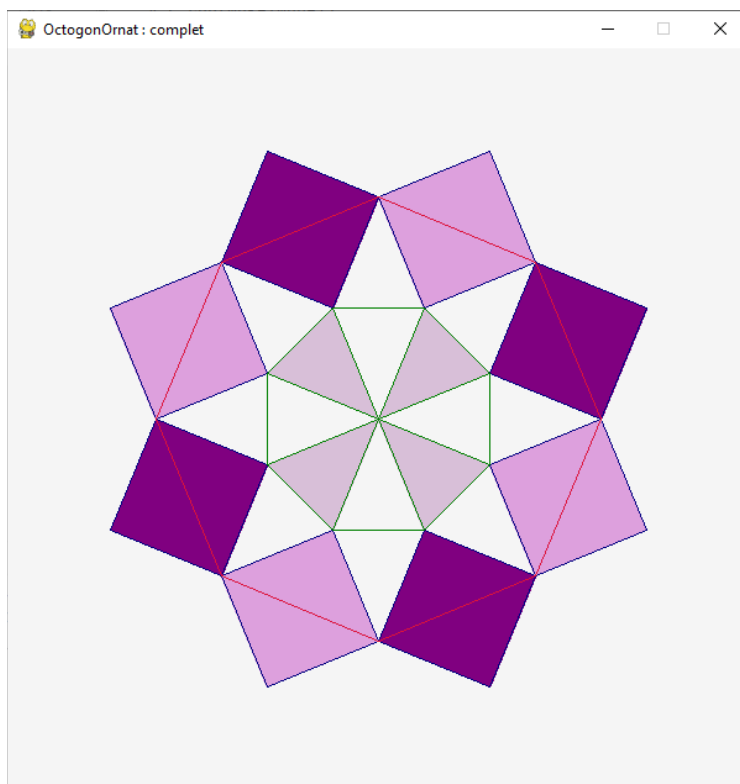
Completați programul pentru a repeta de câteva ori construcția în exterior:



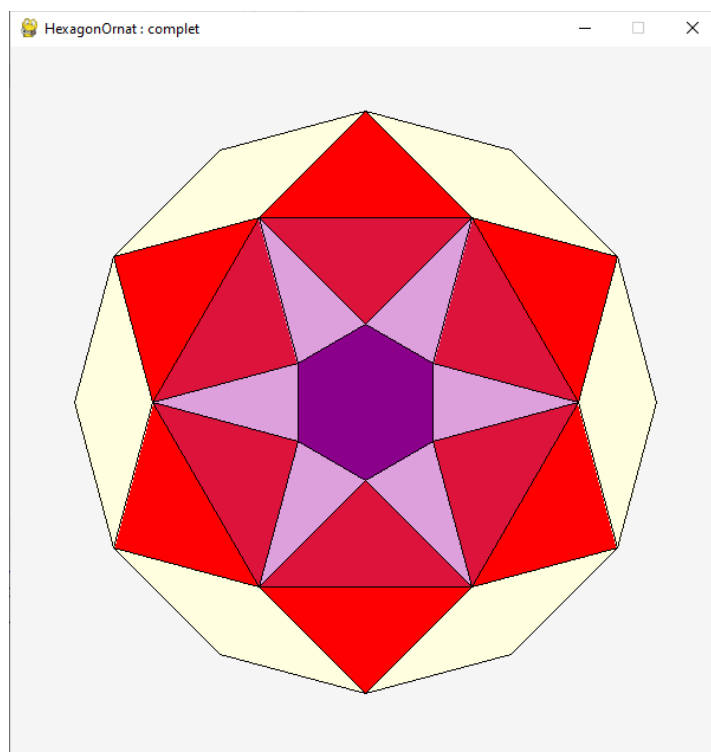
**Exercițiul 5.** Desenați pentagoane în jurul unui pentagon:



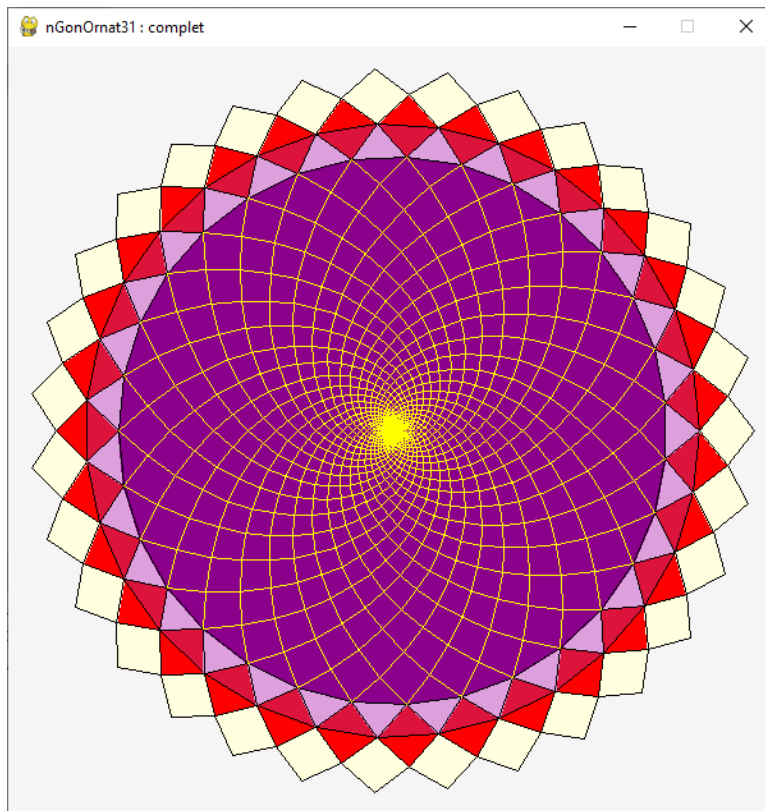
**Exercițiul 6.** Desenul următor a fost obținut ornând cu pătrate octogonul roșu și cu triunghiuri octogonul verde. Incercați și voi.



**Exercițiul 7.** Ornați mai întâi un hexagon regulat



și apoi un 31-gon regulat, pentru a obține o pălărie de floarea-soarelui:

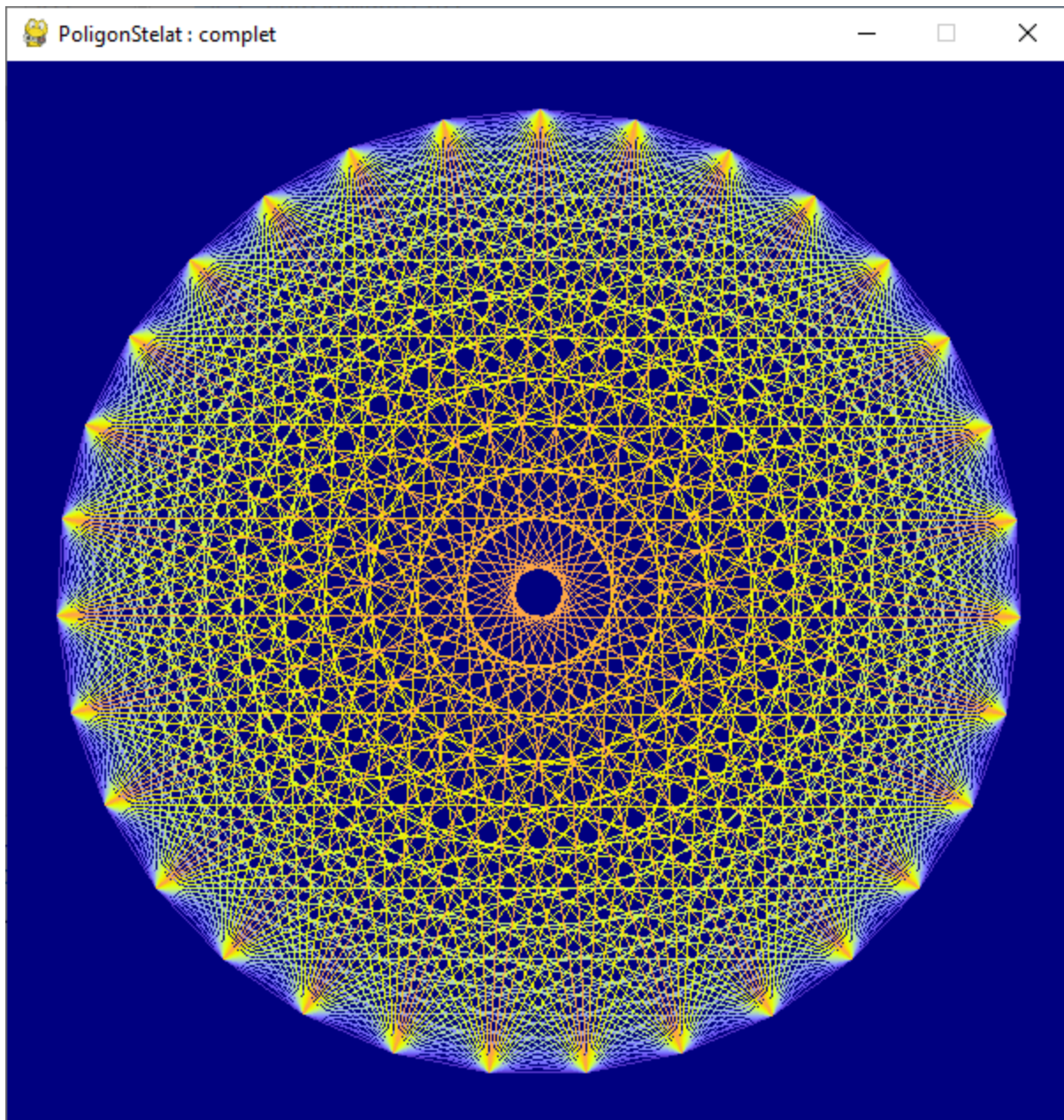


*Observație:* spiralele interioare au fost obținute cu următoarea metodă, care micșorează și rotește vârfurile poligonului  $p$  în jurul centrului  $q$

```
def roteste(q, p, col):
    omega = C.fromRhoTheta(0.9, 0.1)
    pp = list(p) # shallow copy : suficient
    # p este o lista de nr. complexe
    # deci are elementele nemodificabile
    for k in range(100):
        for j in range(len(p)):
            a = q + omega * (p[j] - q)
            C.drawLine(a, p[j], col)
            p[j] = a
            aa = q + omega.conjugate() * (pp[j] - q)
            C.drawLine(aa, pp[j], col)
            pp[j] = aa
    return
```

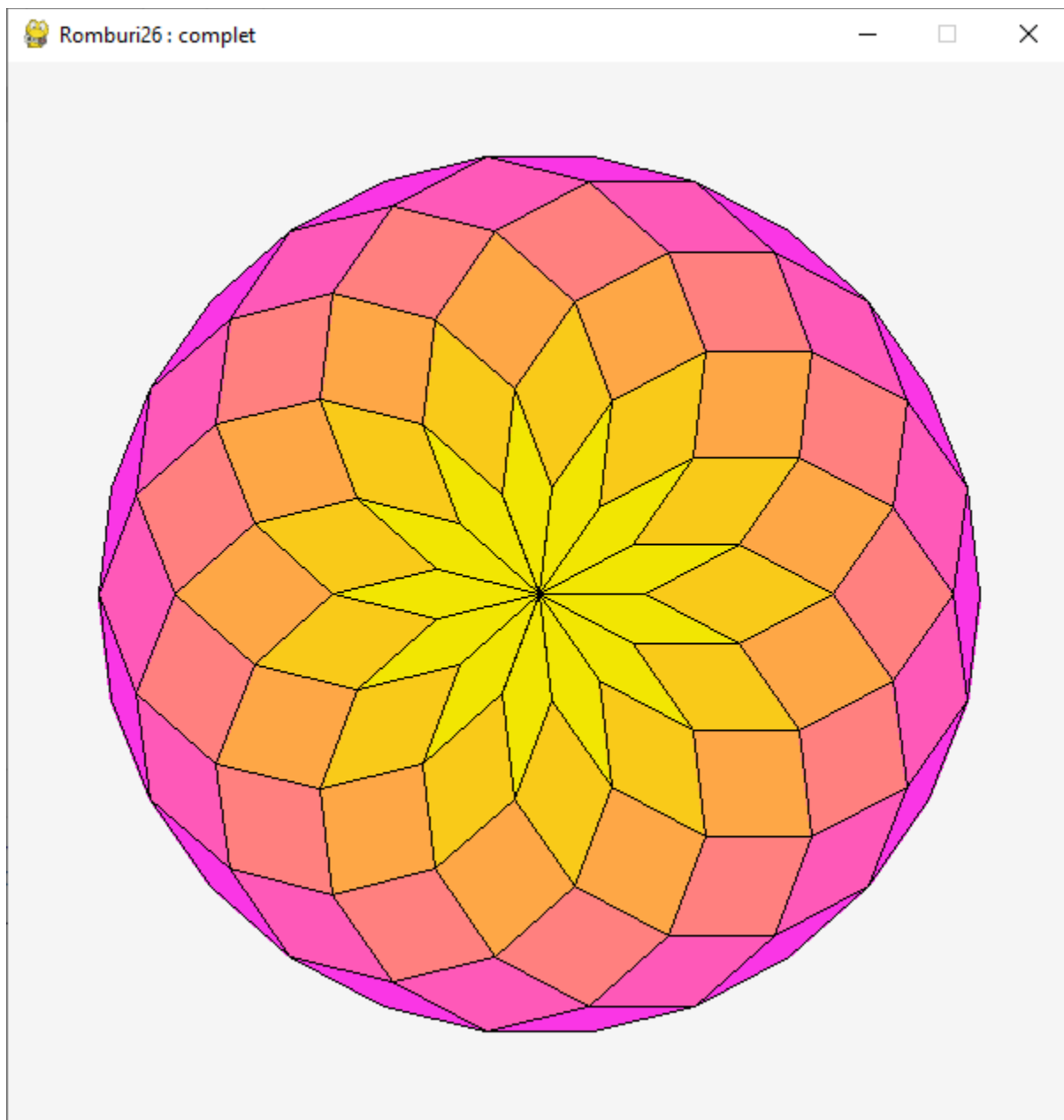


**Exercițiul 8.** Următorul desen a fost obținut trasând toate diagonalele unui  $n$ -gon regulat, cu  $n=31$ . Încercați și voi:

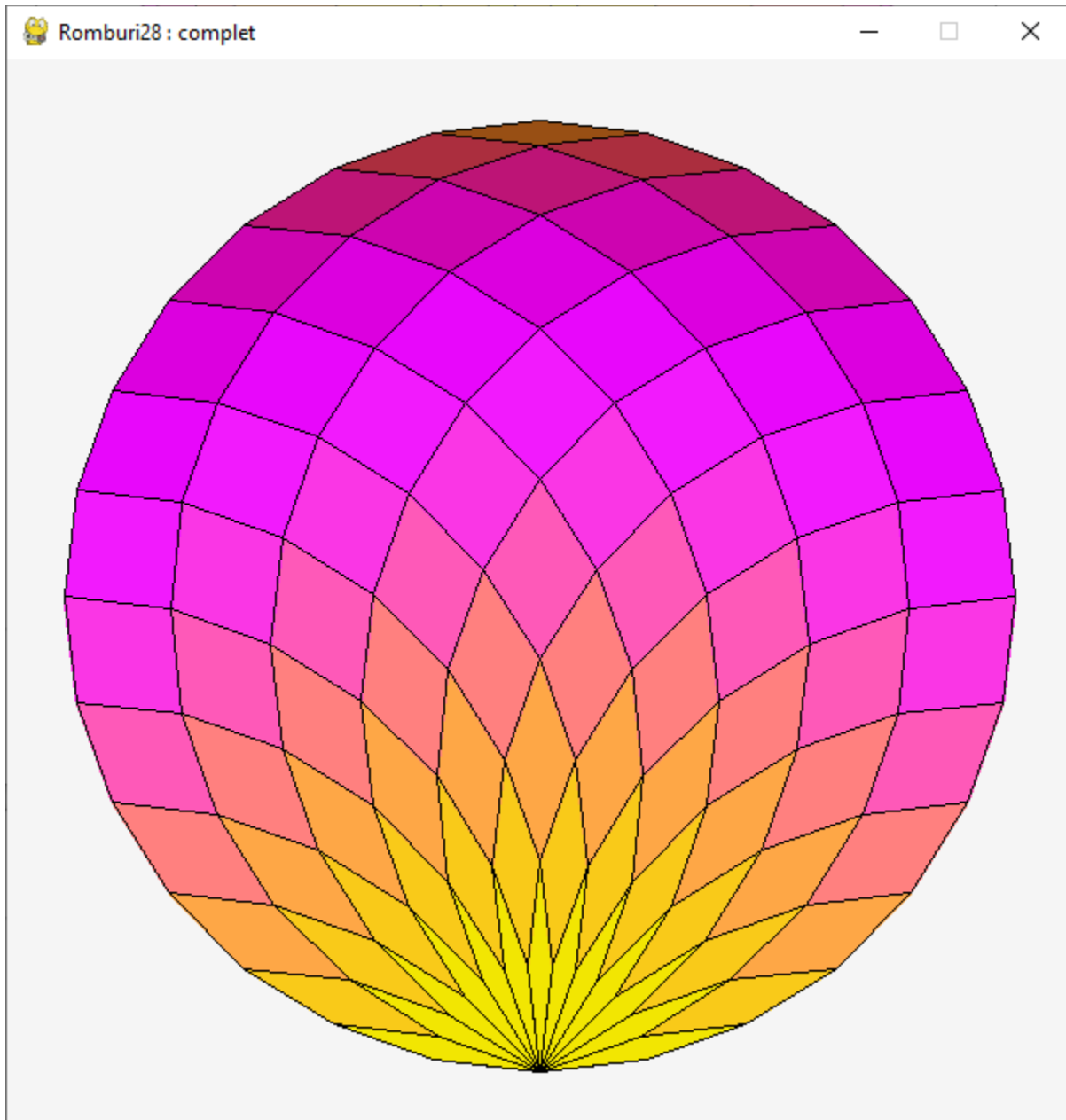


Pentru cazul  $n=32$  vezi: <http://benice-equation.blogspot.ro/2011/10/thirty-two-pointed-star-polygons.html>

**Exercițiul 9.** a) Următoarea disecție în romburi a unui 26-gon regulat a fost trasată plecând de la cele 13 romburi cu vârful în centru. Încercați și voi:



b) Secționați și un 28-gon regulat:



Link: [https://en.wikipedia.org/wiki/Regular\\_polygon](https://en.wikipedia.org/wiki/Regular_polygon)

*Indicație:* Un triunghi  $abc$  se completează până la un paralelogram  $abcd$  alegând  $d = a + c - b$ .