

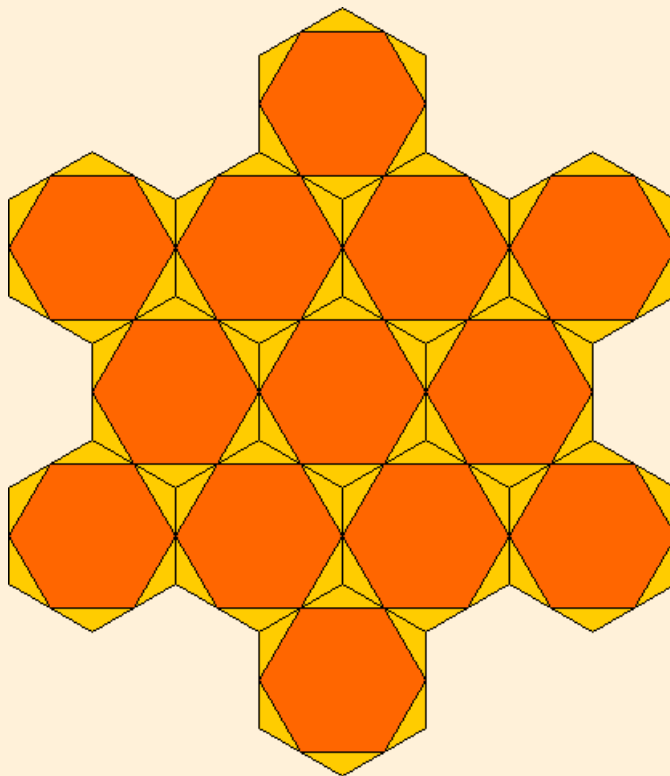


# Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

## Zadanie Motyw – LOGIA 24 (2023/24), etap 1

### Treść zadania

Napisz funkcję **motyw()**, po wywołaniu której powstanie na środku ekranu rysunek taki, jak poniżej. Wysokość rysunku wynosi **480**.



### Omówienie rozwiązania

Rysunek, który ma być wynikiem wywołania funkcji **motyw()** składa się z dwóch rodzajów sześciokątów foremnych. Różnią się one wielkością i kolorem zamalowania. Warto zdefiniować funkcję pomocniczą **szesc** z dwoma parametrami – długość boku i kolor zamalowania. Rysunek jest wyśrodkowany, co ułatwi wpisanie pomarańczowego sześciokąta w żółty sześciokąt.

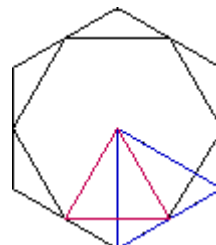
```
1 def szesc(bok, kolor):
2     pu()
3     lt(30); fd(bok); rt(120)
4     pd()
5     fillcolor(kolor)
6     begin_fill()
7     for i in range(6):
8         fd(bok); rt(360 / 6)
9     end_fill()
10    pu()
11    lt(120); bk(bok); rt(30)
12    pd()
```



## Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

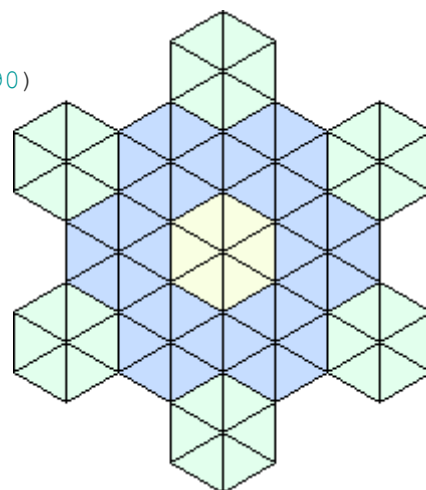
Drugą funkcją pomocniczą będzie **element** z jednym parametrem, oznaczającym długość boku żółtego sześciokąta. Kolory sześciokątów są reprezentowane w kodzie szesnastkowym – żółty '#ffcc00' i pomarańczowy '#ff6600'. Sześciokąt foremny składa się z sześciu trójkątów równobocznych. Bok mniejszego sześciokąta stanowi wysokości trójkąta równobocznego wchodzącego w skład większego sześciokąta. Mniejszy sześciokąt jest obrócony o kąt 30 stopni względem większego sześciokąta.

```
1 def element(bok):
2     szesc(bok, '#ffcc00')
3     rt(30)
4     bok = bok * sqrt(3) / 2
5     szesc(bok, '#ff6600')
6     lt(30)
```



Długości boku żółtego sześciokąta to 1/8 wysokości rysunku. Całość rysunku stanowi wewnętrzny element (na rysunku pomocniczym oznaczony kolorem żółtym), korona złożona z sześciu sześciokątów (kolor niebieski) oraz sześć zewnętrznych wypustek (kolor zielony). Żółt tworząc koronę lub wypustki przemieszcza się po bokach niewidocznych trójkątów równobocznych. Po narysowaniu elementu wraca do środka rysunku i obraca się o 60 stopni.

```
1 def motyw():
2     bok = 60
3     element(bok)
4     for i in range(6):
5         pu()
6         lt(90); fd(bok); rt(60); fd(bok); rt(30)
7         pd()
8         element(bok)
9         pu()
10        lt(30); bk(bok); lt(60); bk(bok); rt(90)
11        pd()
12        rt(60)
13
14    for i in range(6):
15        pu()
16        lt(90); fd(3*bok); rt(30)
17        pd()
18        element(bok)
19        pu()
20        lt(30); bk(3*bok); lt(-90)
21        pd()
22        rt(60)
```





# Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

## Rozwiązanie w języku Python

```
1 from turtle import *
2
3 def szesc(bok, kolor):
4     pu(); lt(30); fd(bok); rt(120); pd()
5     fillcolor(kolor)
6     begin_fill()
7     for i in range(6):
8         fd(bok); rt(360 / 6)
9     end_fill()
10    pu(); lt(120); bk(bok); rt(30); pd()
11
12 def element(bok):
13     szesc(bok, '#ffcc00')
14     rt(30)
15     bok = bok * sqrt(3) / 2
16     szesc(bok, '#ff6600')
17     lt(30)
18
19 def motyw():
20     bok = 60
21     element(bok)
22     for i in range(6):
23         pu()
24         lt(90); fd(bok); rt(60); fd(bok); rt(30)
25         pd()
26         element(bok)
27         pu()
28         lt(30); bk(bok); lt(60); bk(bok); rt(90)
29         pd()
30         rt(60)
31
32     for i in range(6):
33         pu()
34         lt(90); fd(3*bok); rt(30)
35         pd()
36         element(bok)
37         pu()
38         lt(30); bk(3*bok); lt(-90)
39         pd()
40         rt(60)
```

## Testy

Funkcja **motyw()** jest bezparametrowa, dlatego wystarczy raz przetestować jej działanie. Sprawdzamy prawidłowość rysunku, jego zgodność w treścią zadania, rozmiar i wyśrodkowanie. W języku Python, aby przyspieszyć tworzenie rysunku przez żółwia, stosujemy wywołanie złożone z funkcji `tracer()` – rysownie w pamięci, właściwego wywołania funkcji **motyw()** i na końcu uaktualniamy ekran za pomocą funkcji `update()`.

```
tracer(0); motyw(); update()
```