

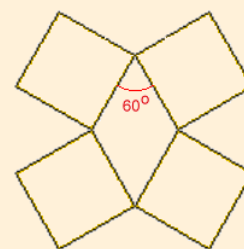
Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Zadanie Posadzka – LOGIA 21 (2020/21), etap 1

Treść zadania

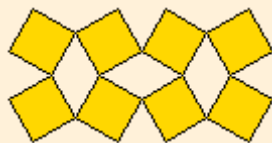
Napisz funkcję **posadzka(n)**, po wywołaniu której powstanie rysunek posadzki złożonej z obróconych i stykających się rogami żółtych kwadratów. Parametr **n** określa liczbę wierszy posadzki i może przyjmować wartości parzyste od **2** do **14**. Liczba kolumn jest zawsze o **2** większa niż liczba wierszy. Długość boku kwadratu wynosi **24**.

Zaznaczony na rysunku pomocniczym kąt pomiędzy kwadratami wynosi **60°**.

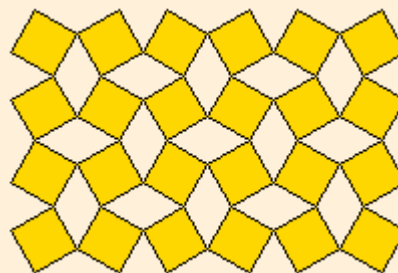


rysunek pomocniczy

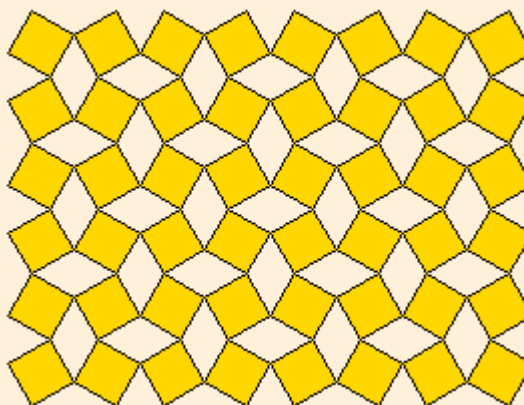
Przykłady:



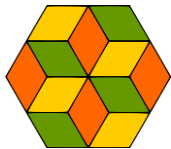
posadzka(2)



posadzka(4)



posadzka(6)



Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Omówienie rozwiązania

Zastanówmy się jak najprościej narysować taką posadzkę. Metod rysowania może być wiele. Jedną z nich, którą tutaj przedstawimy jest powtarzanie elementu składającego się z czterech kwadratów, takiego jak na rysunku pomocniczym. Dla parametru $n = 2$ mamy takie dwie czwórki, dla parametru $n = 3$ mamy sześć czwórek, wreszcie dla $n = 6$ jest dwanaście czwórek.

Przed rozpoczęciem rysowania powinno się przenieść żółwia w takie miejsce, aby dla każdego parametru rysunek był dobrze widoczny. W treści zadania nie ma wymagania aby rysunek był wyśrodkowany, więc nie musimy tego robić.

Warto utworzyć zmienną **bok** i nadać jej wartość **24**, bo taka długość boku jest podana w treści zadania. Wszystkie funkcje pomocnicze będziemy pisać wykorzystując ten parametr, ale wartość będzie przypisana tylko w jednym miejscu.

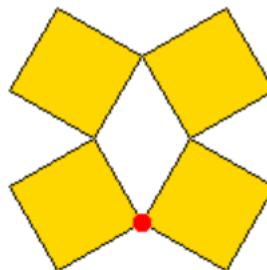
Ponadto z treści zadania wynika, że parametr **n** jest zawsze parzysty, określa liczbę wierszy na rysunku. Nasze rozwiązanie polega na powtarzaniu **2** razy mniejszej liczby czwórek, niż liczba wierszy, dlatego do rysowania możemy zmniejszyć jego wartość dwukrotnie. W takim przypadku należy wykorzystać operację dzielenia całkowitego, ponieważ funkcja **range** musi mieć parametr całkowity.

Warto się zastanowić, z którego miejsce rozpocząć rysowanie, a w którym miejscu skończyć. Początek rysowania czterech kwadratów zaznaczony jest na rysunku poniżej czerwoną kropką. Po zakończeniu rysowania czterech kwadratów żółw stoi w tym samym miejscu, skierowany poziomo, tak jak zaczynał rysowanie. Treść funkcji **posadzka** może być następująca:

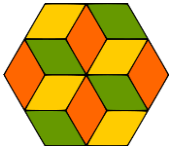
```
def posadzka(n):  
    '''przejdzie, aby rysunek był widoczny  
    na ekranie dla każdego parametru'''  
    pu(); bk(240); lt(90); bk(220); rt(90); pd()  
    # parametry rysunku wg treści zadania  
    n = n // 2  
    bok = 24  
    # rysowanie posadzki  
    for i in range(n + 1):  
        kolumna(bok, n)  
        pu(); fd(bok * sqrt(3) + bok); pd()
```

Pomocnicza funkcja rysowania czterech kwadratów takich, jak na rysunku poniżej, może wyglądać następująco:

```
def czwórka(bok):  
    lt(60)  
    for i in range(2):  
        kwadrat(bok)  
        fd(bok); lt(60)  
        kwadrat(bok)  
        fd(bok); lt("120")  
    rt(60)
```



Kolejna funkcja pomocnicza może służyć do rysowania kolumny składającej się z **n** takich czwórek. Po narysowaniu kolumny, żółw wraca na dół do tego miejsca, gdzie zaczynał rysowanie kolumny.



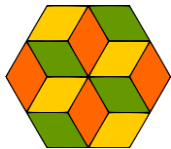
Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

```
def kolumna(bok, n):  
    for k in range(n):  
        czworoka(bok)  
        #przejście  
        pu(); lt(60); fd(bok); lt(60)  
        fd(bok); rt(30)  
        fd(bok); rt(90); pd()  
    #przemieszczenie żółwia  
    pu(); lt(90); bk(n * (bok * sqrt(3) + bok))  
    rt(90); pd()
```



Rozwiązanie w języku Python

```
1. from turtle import *  
2. from math import sqrt  
3.  
4. def kwadrat(bok):  
5.     fillcolor("gold")  
6.     begin_fill()  
7.     for i in range(4):  
8.         fd(bok); rt(90)  
9.     end_fill()  
10.  
11. def czworoka(bok):  
12.     lt(60)  
13.     for i in range(2):  
14.         kwadrat(bok)  
15.         fd(bok); lt(60)  
16.         kwadrat(bok)  
17.         fd(bok); lt(120)  
18.     rt(60)  
19.  
20. def kolumna(bok, n):  
21.     for k in range(n):  
22.         czworoka(bok)  
23.         # przejście  
24.         pu(); lt(60); fd(bok); lt(60)  
25.         fd(bok); rt(30)  
26.         fd(bok); rt(90); pd()  
27.     # przemieszczenie żółwia  
28.     pu(); lt(90); bk(n * (bok * sqrt(3) + bok))  
29.     rt(90); pd()  
30.  
31. def posadzka(n):  
32.     '''przejście, aby rysunek był widoczny  
33.     na ekranie dla każdego parametru'''  
34.     pu(); bk(240); lt(90); bk(220); rt(90); pd()  
35.     # parametry rysunku wg treści zadania  
36.     n = n // 2  
37.     bok = 24  
38.     # rysowanie posadzki  
39.     for i in range(n + 1):  
40.         kolumna(bok, n)  
41.         pu(); fd(bok * sqrt(3) + bok); pd()
```



Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Testy

Chcąc sprawdzić prawidłowość rysunków warto uruchomić funkcję dla parametrów 2, 4 i 6. Rysunki powinny być takie, jak w treści zadania. Warto wywołać funkcję dla największego parametru 14, efekt powinien być taki, jak na rysunku poniżej. Cały rysunek mieści się w oknie Python Turtle Graphics.

