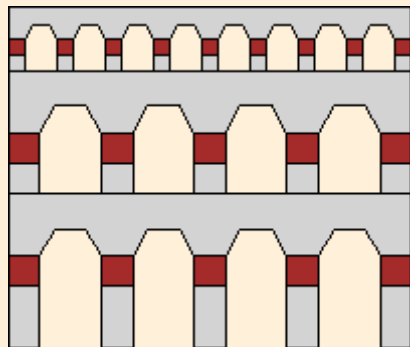


Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

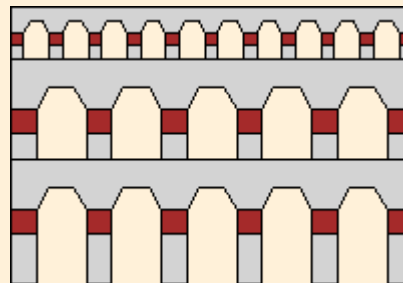
Zadanie Akwedukt – LOGIA 23 (2022/23), etap 1

Treść zadania

W Loglandii zaplanowano budowę nowych akweduktów prowadzących do stolicy. Znany architekt Logian zaprojektował trzypoziomowe akwedukty. Na poziomie trzecim jest dwa razy więcej otworów między podporami niż na pozostałych poziomach.

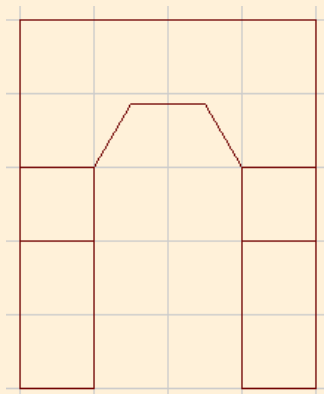


akwedukt(5)

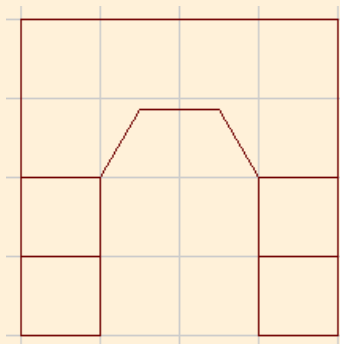


akwedukt(6)

Napisz funkcję **akwedukt(n)**, po wywołaniu której powstanie na środku ekranu rysunek akweduktu taki, jak powyżej. Parametr **n** oznacza liczbę podpór na najniższym poziomie i może przyjmować wartości od **5** do **20**. Długość akweduktu wynosi **500**. Proporcje poszczególnych elementów odczytaj z rysunku pomocniczego.



Rysunek pomocniczy – poziom 1

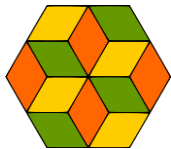


Rysunek pomocniczy – poziomy 2 i 3

Omówienie rozwiązania

W zależności od poziomu akweduktu rysowane są dłuższe lub krótsze podpory. Na pierwszym poziomie każda podpora składa się z jasnoszarego prostokąta o proporcji boków 2 x 1 oraz brązowego kwadratu. Na poziomach drugim i trzecim rysowane są podpory złożone z dwóch kwadratów – jasnoszarego i brązowego. Nad podporami dorysowujemy jasnoszary cokół, przy czym wycięta część cokołu pomiędzy sąsiednimi podporami to połowa sześciokąta foremnego. Szerokość podpory na pierwszym i drugim poziomie wyliczamy w następujący sposób:

$$a = 500 / (3 * n - 2)$$



Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

ponieważ po każdej podporze poza ostatnią następuje odstęp dwa razy większy od szerokości podpory. Nieco trudniejsze jest wyliczenie szerokości podpór w trzecim poziomie akweduktu. Z treści zadania wynika, że znajduje się tam dwa razy więcej otworów między podporami, niż na niższych poziomach. Jeśli na pierwszy poziomie mamy n podpór, to pomiędzy nimi jest $n - 1$ otworów. Zatem na poziomie trzecim mamy $2 * (n - 1)$ otworów, czyli $2 * n - 1$ podpór. Szerokość podpory wyliczamy podobnie jak na niższych poziomach:

$$b = 500 / (3 * (2 * n - 1) - 2) = 500 / (6 * n - 3 - 2) = 500 / (6 * n - 5)$$

Ponieważ rysunek powinien być na środku ekranu, trzeba na początku przemieścić żółwia w lewo o 250 i w dół o połowę wysokości całego akweduktu, czyli o $9 * a / 2 + 4 * b / 2$.

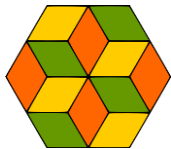
Rozwiązanie w języku Python

Potrzebne będą cztery funkcje – funkcja główna `akwedukt()` oraz pomocnicze `prostokat()`, `cokol()` oraz `pas()`.

W funkcji `prostokat(a, b, kolor)` będziemy rysować zamalowany prostokąt o podanej szerokości a , wysokości b oraz kolorze `kolor`. Funkcja `cokol(a, n)` ma dwa parametry – pierwszy to długość odcinka odpowiadającego miejscu mocowania do podpory oraz kolejnych odcinków rysowanych nad otworem między sąsiednimi podporami tworzących połowę sześciokąta foremnego, a drugi – liczba podpór pod cokół. Funkcja `pas(a, n, m)` rysuje dwa rzędy n prostokątów – pierwszy jasnoszary, powyżej drugi brązowy oraz na samej górze jasnoszary cokół. Parametry a i m określają wielkość rysowanych prostokątów.

Funkcja `akwedukt(n)` składa się z trzech części: wyliczenia szerokości podpory na pierwszych dwóch poziomach i szerokości podpory na najwyższym poziomie, przesunięcia żółwia w lewy dolny róg, narysowania trzech poziomów akweduktu.

```
1 from turtle import *
2
3 def prostokat(a, b, kolor):
4     fillcolor(kolor)
5     begin_fill()
6     for i in range(2):
7         fd(a); lt(90)
8         fd(b); lt(90)
9     end_fill()
10
11 def cokol(a, n):
12     fillcolor('lightgray')
13     begin_fill()
14     for i in range(n - 1):
15         fd(a); lt(60)
16         for i in range(3):
17             fd(a); rt(60)
18         lt(120)
19     fd(a)
20     lt(90); fd(2 * a); lt(90)
21     fd(n * 3 * a - 2 * a)
22     lt(90); fd(2 * a); lt(90)
23     end_fill()
24
```



Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

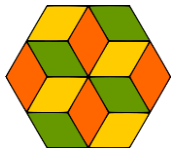
```
25 def pas(a, n, m):
26     for i in range(n):
27         prostokat(a, m * a, 'lightgray')
28         pu(); fd(3 * a); pd()
29     pu(); bk(n * 3 * a); lt(90); fd(a * m); rt(90); pd()
30     for i in range(n):
31         prostokat(a, a, 'brown')
32         pu(); fd(3 * a); pd()
33     pu(); bk(n * 3 * a); lt(90); fd(a); rt(90); pd()
34     cokol(a, n)
35     pu(); lt(90); fd(2 * a); rt(90); pd()
36
37 def akwedukt(n):
38     a = 500 / (3 * n - 2)
39     b = 500 / (6 * n - 5)
40     pu(); bk(500 / 2); rt(90); fd(9 * a / 2 + 2 * b); lt(90); pd()
41     pas(a, n, 2)
42     pas(a, n, 1)
43     pas(b, 2 * n - 1, 1)
```

Testy

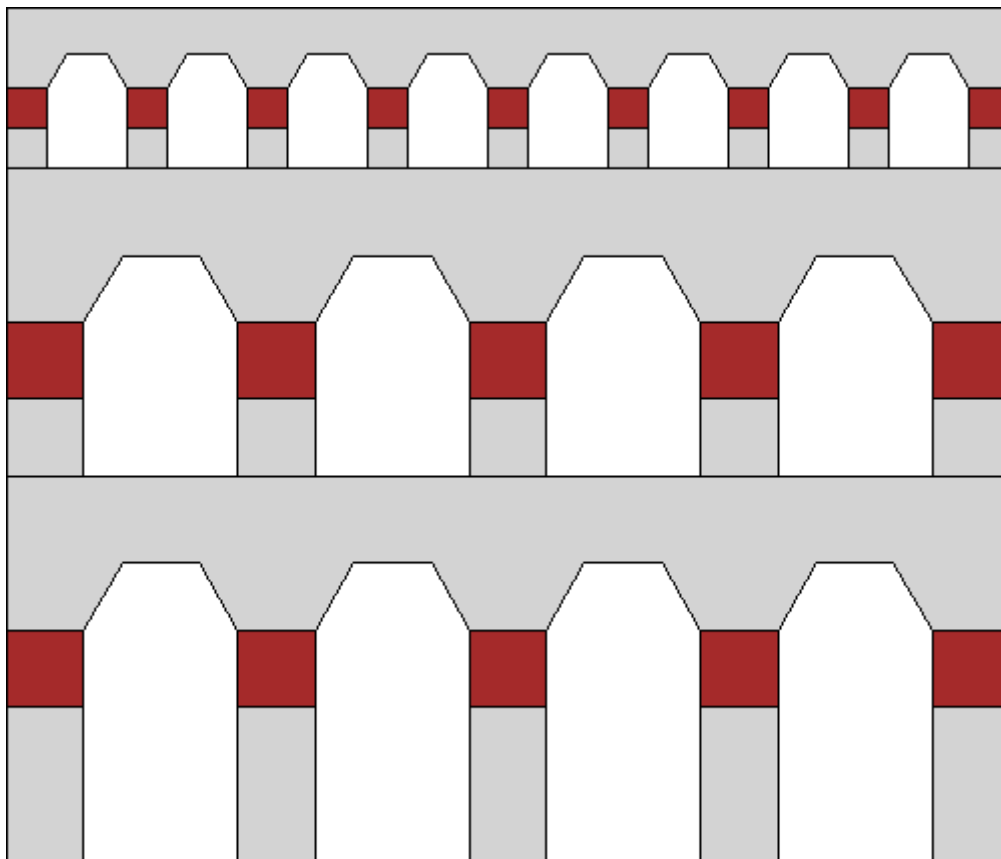
Ponieważ wartość parametru n może przyjmować wartości od 5 do 20, można wykonać testy dla wszystkich wartości. Na konkursie zadanie było testowane dla następujących wartości parametru: 5, 7, 10, 19, 20. Należy zwrócić uwagę na schemat kolorów oraz sprawdzić, czy rysunek jest na środku ekranu. Błędy mogą się też pojawić w proporcjach poszczególnych elementów.

Aby przyspieszyć tworzenie rysunku przez żółwia, stosujemy wywołanie złożone z funkcji `tracer()` – rysownie w pamięci, właściwego wywołania funkcji rysującej `akwedukt` i na końcu uaktualniamy ekran za pomocą funkcji `update()`.

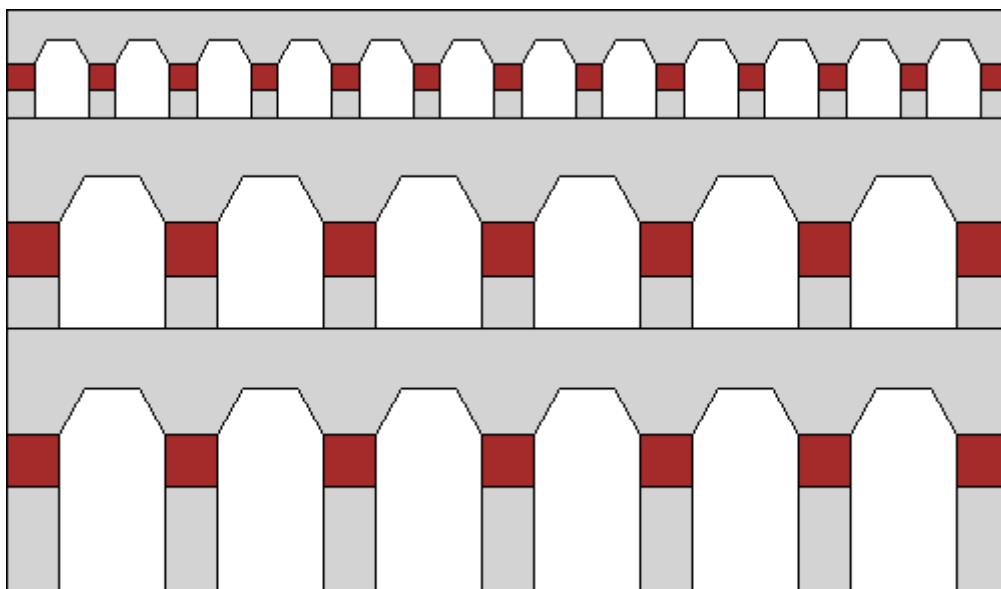
```
tracer(0); akwedukt(5); update()
```



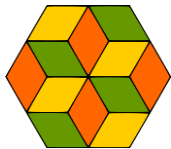
Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty



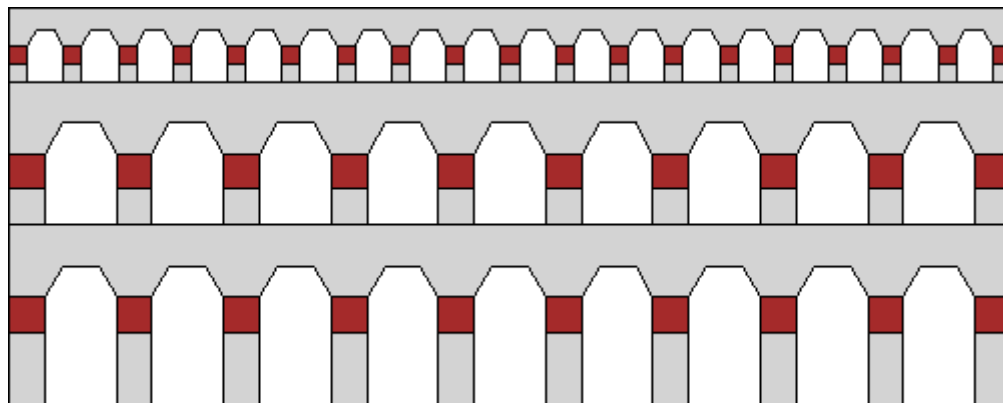
akwedukt(5)



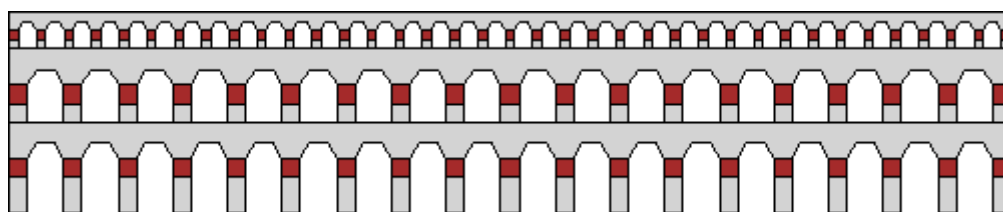
akwedukt(7)



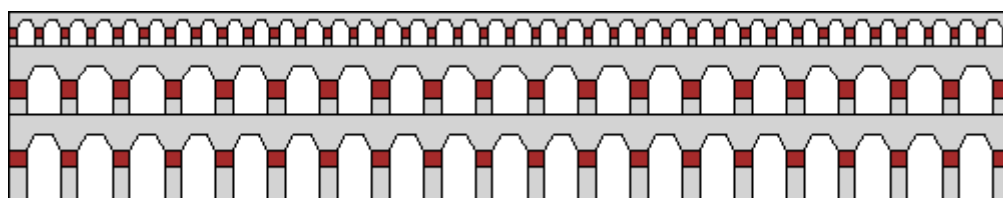
Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty



akwedukt(10)



akwedukt(19)



akwedukt(20)