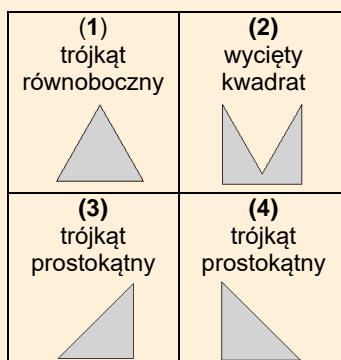


Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

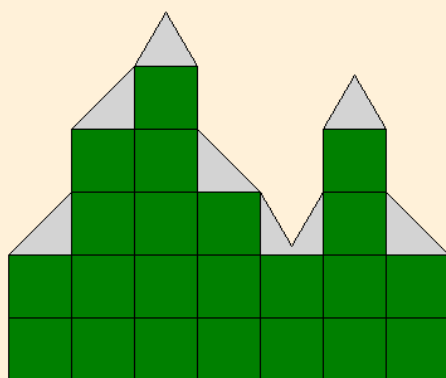
Zadanie góry – LOGIA 21 (2020/21), etap 2

Treść zadania

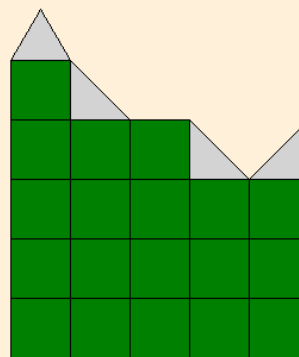
Janek chodzi po górach i oznacza liczbami naturalnymi od 1 do 9 kolejno odwiedzane wierzchołki proporcjonalnie do ich wysokości. Potem rysuje schemat, którym ilustruje zanotowane dane. Jeśli wierzchołek jest oznaczony liczbą n , to rysuje kolumnę n zielonych kwadratów o długości boku 50. Ponadto dodaje szare zakończenie: trójkąt równoboczny (1), gdy sąsiednie wysokości są niższe, kwadrat z wyciętym trójkątem równobocznym (2), gdy są wyższe, trójkąt prostokątny (3), gdy tylko następna wysokość jest wyższa i trójkąt prostokątny (4), gdy tylko poprzednia wysokość jest wyższa. Jeśli żaden z tych warunków nie zachodzi, to nie rysuje szarego zakończenia. Trasę Janek może pokonywać wielokrotnie, więc „lewym sąsiadem” dla pierwszego wierzchołka jest ostatni, a „prawym sąsiadem” dla ostatniego pierwszy.



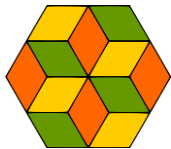
Pomóż Jankowi i napisz funkcję **schemat(lista)** rysującą schemat według opisanych zasad. Parametrem funkcji jest lista o długości od 3 do 20 liczb w zakresie od 1 do 9. Rysunek powinien być jednakowo oddalony od lewej i prawej krawędzi ekranu.



schemat([2, 4, 5, 3, 2, 4, 2])



schemat([5, 4, 4, 3, 3])



Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Omówienie rozwiązania

Rozwiązanie zadanie polega na przejrzaniu wszystkich elementów parametru `lista` i narysowaniu wskazanej liczby kwadratów wraz z zakończeniem. Liczbę kwadratów w danej kolumnie ustalamy na podstawie wartości rozpatrywanego elementu listy, a zakończenie również na podstawie wartości sąsiednich elementów. Należy zadbać o rozpatrywanie warunków w kolejności zgodnej z podaną w treści zadania: obaj sąsiedzi niżsi, obaj wyżsi, gdy tylko następna wysokość jest wyższa i gdy tylko poprzednia wysokość jest wyższa. Jeśli żaden z tych warunków nie zachodzi, to nie rysujemy zakończenia.

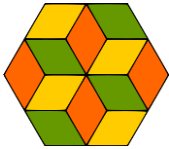
Pewną trudność może też stanowić pierwszy i ostatni element listy, gdyż należy właściwie określić ich sąsiadów. Warto skorzystać z operacji modulo. Dokładniej sąsiadami dla elementu `i` poza pierwszym i ostatnim jest element `i-1` oraz `i+1`. Natomiast dla elementu `0` jest to `n-1`, gdzie `n` oznacza długość listy oraz `i+1`, a dla elementu `n-1`, to `n-2` oraz `0`. Można ogólnie zapisać, że sąsiadami dla elementu `i` są odpowiednio elementy `(i-1) % n` oraz `(i+1) % n`.

Rozwiązanie w języku Python

Przy definiowaniu głównej funkcji `schemat(lista)` warto skorzystać z funkcji pomocniczych: `kwadrat(bok)` oraz `elt1(bok)`, ..., `elt4(bok)`, które będą rysowały kwadrat oraz cztery rodzaje elementów kończących. Do prawidłowego działania skryptu należy zaimportować moduły `turtle` i funkcję `sqrt` z modułu `math`. Jeśli zaczynamy rysowanie figury, która będzie zamalowana należy skorzystać z polecenia `begin_fill()`, gdy kończymy `end_fill()`. Wcześniej jednak ustawiamy kolor obwódki i kolor zamalowania.

Rysunek powinien być jednakowo oddalony od lewej i prawej krawędzi ekranu. Skoro zaczynamy od lewego dolnego rogu, to na początku należy przesunąć żółwia o połowę szerokości rysunku, czyli `n*50/2`. W rozwiązaniu zastosowana została instrukcja `if ... elif`, która pozwala wygodnie zapisać kilka warunków.

```
1. from turtle import *
2. from math import sqrt
3.
4. def kwadrat(bok):
5.     begin_fill()
6.     for i in range(4):
7.         fd(bok); rt(90)
8.     end_fill()
9.
10. def elt1(bok):
11.     begin_fill()
12.     rt(30)
13.     for i in range(3):
14.         fd(bok); rt(120)
15.     lt(30)
16.     end_fill()
17.
18. def elt2(bok):
19.     begin_fill()
20.     fd(bok); rt(150)
21.     fd(bok); lt(120)
22.     fd(bok); rt(150)
23.     fd(bok); rt(90)
24.     fd(bok); rt(90)
25.     end_fill()
26.
```



Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

```
27. def elt3(bok):
28.     begin_fill()
29.     rt(45)
30.     fd(bok * sqrt(2)); rt(135)
31.     fd(bok); rt(90)
32.     fd(bok); rt(90)
33.     end_fill()
34.
35. def elt4(bok):
36.     begin_fill()
37.     fd(bok); rt(135)
38.     fd(bok * sqrt(2)); rt(135)
39.     fd(bok); rt(90)
40.     end_fill()
41.
42. def schemat(lista):
43.     a = 50
44.     n = len(lista)
45.     # wyśrodkowanie rysunku
46.     pu(); bk(n * 50 / 2); lt(90); bk(200); rt(90); pd()
47.     for i in range(n):
48.         lt(90)
49.         color("black","green")
50.         for j in range(lista[i]):
51.             kwadrat(a)
52.             fd(a)
53.         color("black","lightgray")
54.         if lista[(i - 1) % n] < lista[i] > lista[(i + 1) % n]:
55.             elt1(a)
56.         elif lista[(i - 1) % n] > lista[i] < lista[(i + 1) % n]:
57.             elt2(a)
58.         elif lista[i] < lista[(i + 1) % n]:
59.             elt3(a)
60.         elif lista[i] < lista[(i - 1) % n]:
61.             elt4(a)
62.         fd(-a * (lista[i]))
63.         rt(90); fd(a)
```

Testy

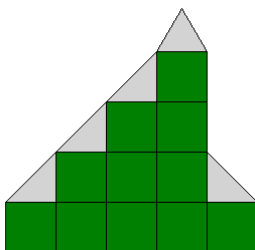
Program testujemy wydając sekwencję poleceń:

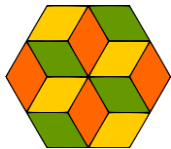
```
tracer(0); schemat(lista); update()
```

gdzie parametr `lista` będzie przyjmować różne wartości.

Przykładowe testy powinny obejmować różne długości tras, a w ramach jednej trasy różne przypadki zakończeń elementów wewnątrz listy i na jej brzegach. Poniżej przykładowe wywołania.

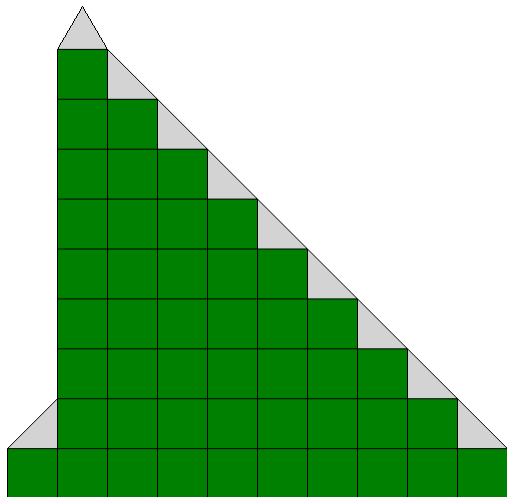
```
schemat([1, 2, 3, 4, 1])
```



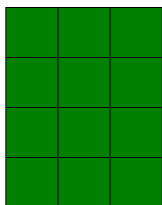


Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

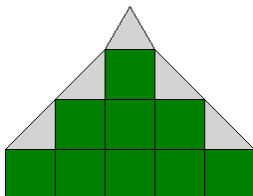
schemat ([1, 9, 8, 7, 6, 5, 4, 3, 2, 1])



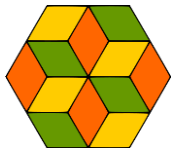
schemat([4, 4, 4])



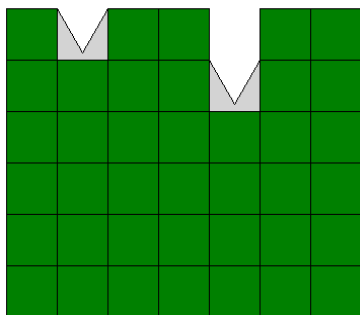
schemat([1, 2, 3, 2, 1])



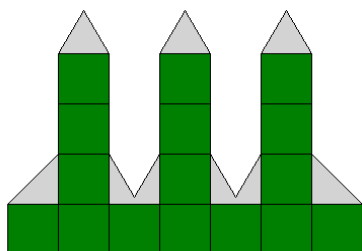
schemat([6, 5, 6, 6, 4, 6, 6])



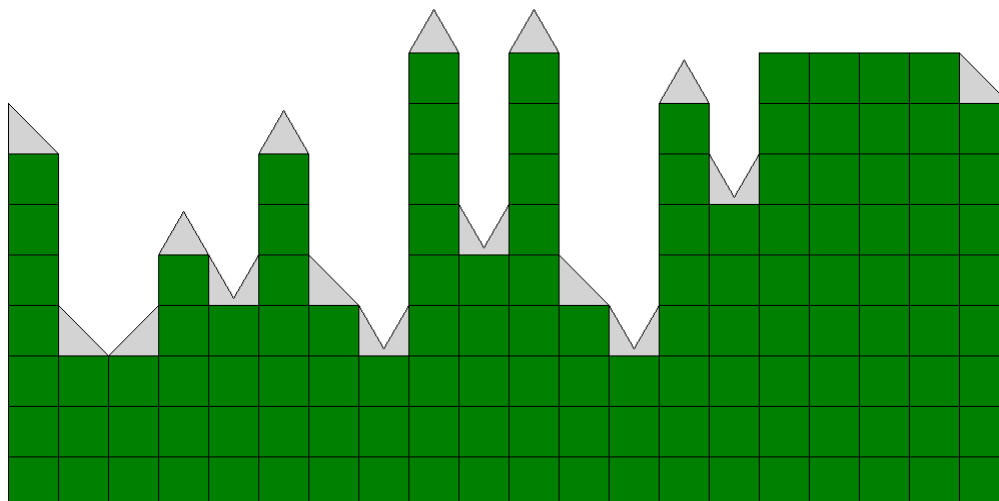
Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty



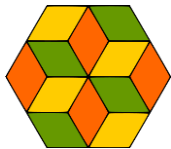
schemat([1, 4, 1, 4, 1, 4, 1])



schemat([7, 3, 3, 5, 4, 7, 4, 3, 9, 5, 9, 4, 3, 8, 6, 9, 9, 9, 9, 8])



schemat([3, 6, 6, 3, 5, 9, 5, 3, 6, 5, 5, 3, 4, 7, 6, 3, 7, 5, 6, 3])



Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

