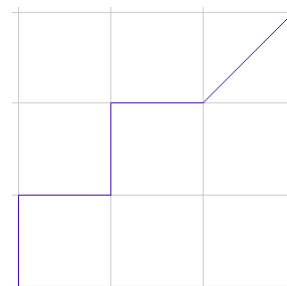


Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Zadanie Tarasy – LOGIA 25 (2024/25), etap 2

Treść zadania

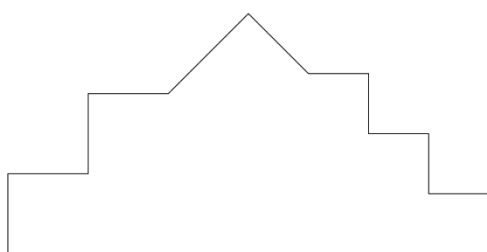
Karol uprawia na zboczach góry dwa rodzaje roślin: gatunki wymagające więcej światła opisuje za pomocą cyfr 2, 4, 6, 8, zaś gatunki rosnące w cieniu oznacza cyframi 1, 3, 5, 7, 9. Zapisał rośliny zaplanowane do posadzenia w najbliższym roku jako cyfry pewnej liczby. Następnie przygotowuje trójkątny przekrój góry z tarasami, dla cyfr parzystych po lewej stronie, dla nieparzystych po prawej. Każda cyfra oznacza roślinę uprawianą na osobnym tarasie. Nad ostatnim tarasem zostawia fragment zbocza o wysokości równej szerokości tarasu.



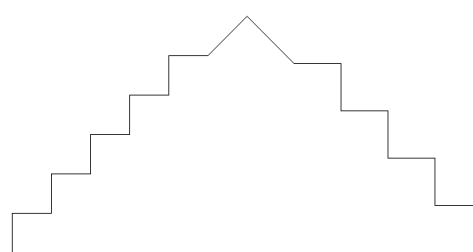
rysunek pomocniczy

Pomóż Karolowi i napisz jednoparametrową funkcję **tarasy()**, po wywołaniu której powstanie rysunek przekroju góry z tarasami. Parametrem funkcji jest liczba całkowita dodatnia maksymalnie 10-cyfrowa nie zawierająca cyfr 0. Wysokość rysunku wynosi 300, a szerokość 600. Każdy taras po danej stronie góry ma taką samą wysokość i szerokość, która zależy od liczby tarasów. Rysunek powinien być na środku ekranu.

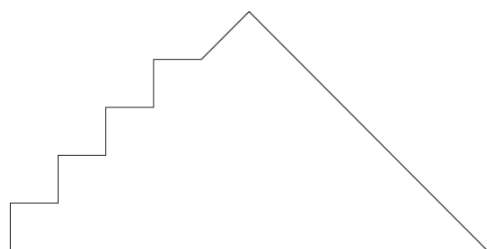
Przykłady:



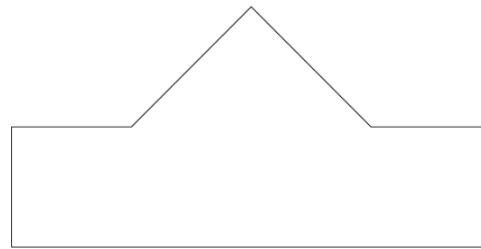
tarasy(14271)



tarasy(124456789)



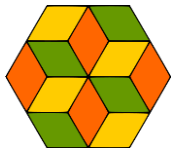
tarasy(2648)



tarasy(12)

Omówienie rozwiązania

Rysunek przedstawiający górę bez tarasów to trójkąt prostokątny równoramienny o wysokości 300 i szerokości 600. Przyprostokątne są w tym przypadku przekątnymi kwadratu o długości boku 300, a ich długość wyliczamy z wzoru na przekątną kwadratu mnożąc pierwiastek z 2 przez długość boku ($\sqrt{2} * 300$). Kluczowym elementem zadania jest policzenie liczby tarasów po każdej ze stron góry oraz



Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

wyliczenie wymiarów każdego z tarasów. Zgodnie z treścią zadania liczba cyfr parzystych, niezależnie od ich wartości, określa liczbę tarasów po lewej stronie rysunku, zaś liczba cyfr nieparzystych – po prawej. Cyfra 0 nie występuje w liczbie i nie bierzemy jej pod uwagę podczas obliczeń.

Możemy wykorzystać jeden z dwóch sposobów na wyliczenie liczby tarasów:

1. wyodrębniać kolejne cyfry liczby korzystając z operacji reszty z dzielenia i dzielenia całkowitego, aby policzyć cyfry parzyste oraz nieparzyste,

```
ile_parzystych ← 0
ile_nieparzystych ← 0
dopóki liczba > 0 wykonuj
  x = liczba mod 10
  jeżeli x mod 2 = 0 to
    ile_parzystych ← ile_parzystych + 1
  w przeciwnym przypadku
    ile_nieparzystych ← ile_nieparzystych + 1
  liczba ← liczba div 10
```

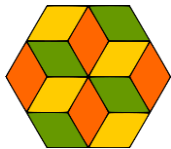
2. zamienić liczbę na napis, a następnie policzyć znaki odpowiadające cyfrom parzystym. Liczbę cyfr nieparzystych można otrzymać jako różnicę długości napisu i liczby cyfr parzystych.

```
liczba ← str(liczba)
ile_parzystych ← 0
dla każdej cyfra w liczba
  jeżeli cyfra in "2468" to ile_parzystych ← ile_parzystych + 1
ile_nieparzystych ← długość(liczba) - ile_parzystych
```

Po wyliczeniu liczby tarasów po lewej i prawej stronie góry możemy policzyć ich wysokości, pamiętając, że nad ostatnim tarasem zostaje nietknięty kawałek zbocza o tej samej wysokości co taras.

```
taras_lewy ← 300 / (ile_parzystych + 1)
taras_prawy ← 300 / (ile_nieparzystych + 1)
```

Znając długości poszczególnych odcinków możemy przystąpić do rysowania, pamiętając o wyśrodkowaniu całego rysunku.



Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Rozwiązanie w języku Python

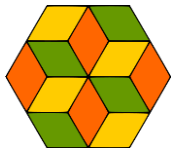
```
1 from turtle import *
2 from math import sqrt
3
4 def policz(liczba):
5     lewe = 0
6     for i in liczba:
7         if i in '2468':
8             lewe += 1
9     return lewe
10
11 def tarasy(liczba):
12     ile_l = policz(str(liczba))
13     ile_p = len(str(liczba)) - ile_l
14     a = 300
15     # wyśrodkowanie i połowa podstawy
16     pu(); rt(90); fd(a / 2); pd(); lt(90); bk(a)
17     #lewe zbocze
18     lt(90)
19     if ile_l > 0:
20         b = a / (ile_l + 1)
21     else:
22         b = a
23     for i in range(ile_l):
24         fd(b); rt(90); fd(b); lt(90)
25     rt(45)
26     fd(b * sqrt(2))
27     #prawe zbocze
28     rt(90)
29     if ile_p > 0:
30         b = a / (ile_p + 1)
31     else:
32         b = a
33     fd(b * sqrt(2)); lt(45)
34     for i in range(ile_p):
35         fd(b); rt(90); fd(b); lt(90)
36     #domknięcie podstawy
37     bk(a)
```

Testy

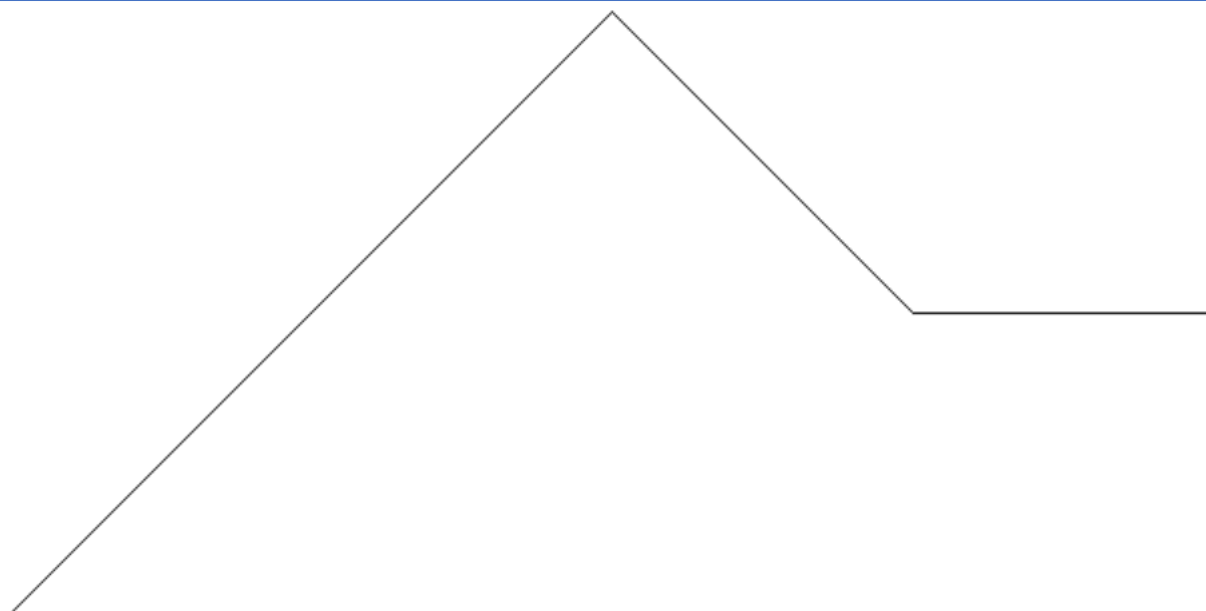
Wywołujemy funkcję **tarasy()** dla różnych parametrów. Liczby wytypowane do testów powinny zawierać wszystkie cyfry od 1 do 9 oraz składać się od 1 do 10 cyfr.

W języku Python, aby przyspieszyć tworzenie rysunku przez żółwia, stosujemy wywołanie złożone z funkcji `tracer()` – rysownie w pamięci, właściwego wywołania funkcji **tarasy()** i na końcu uaktualniamy ekran za pomocą funkcji `update()`.

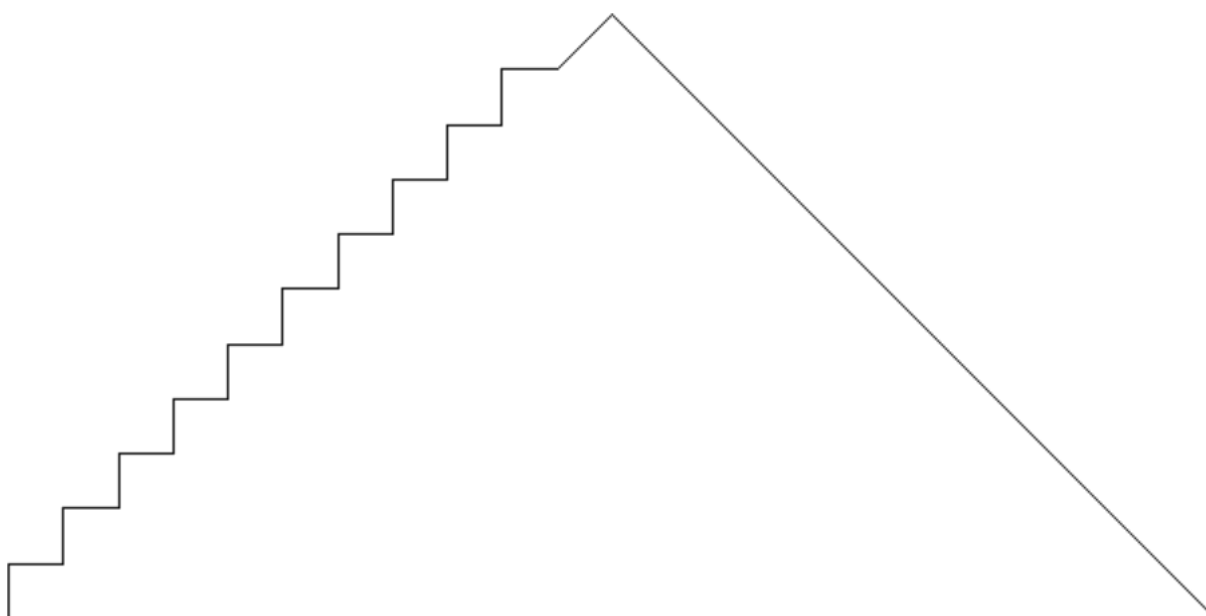
```
tracer(0); tarasy(); update()
```



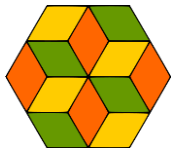
Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty



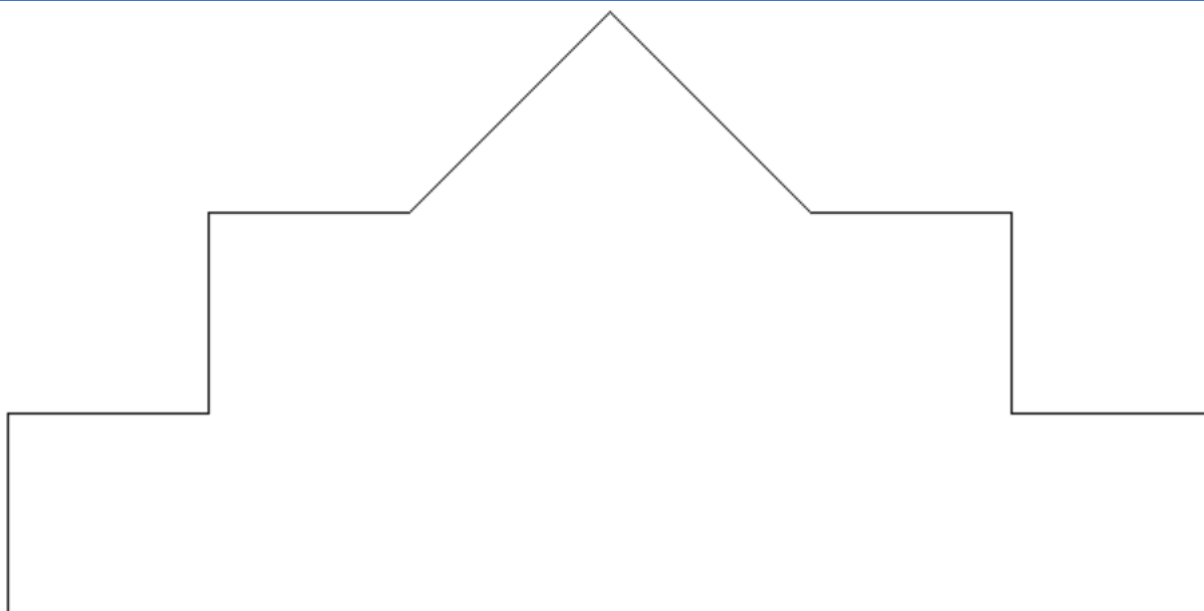
tarasy(1)



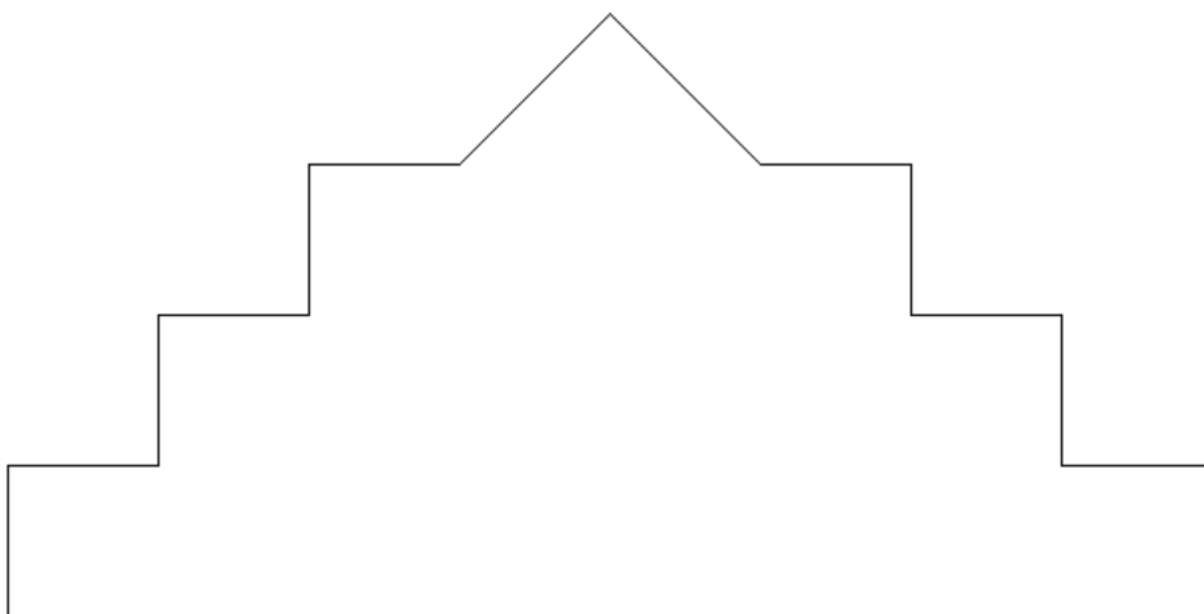
tarasy(2244668828)



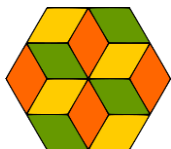
Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty



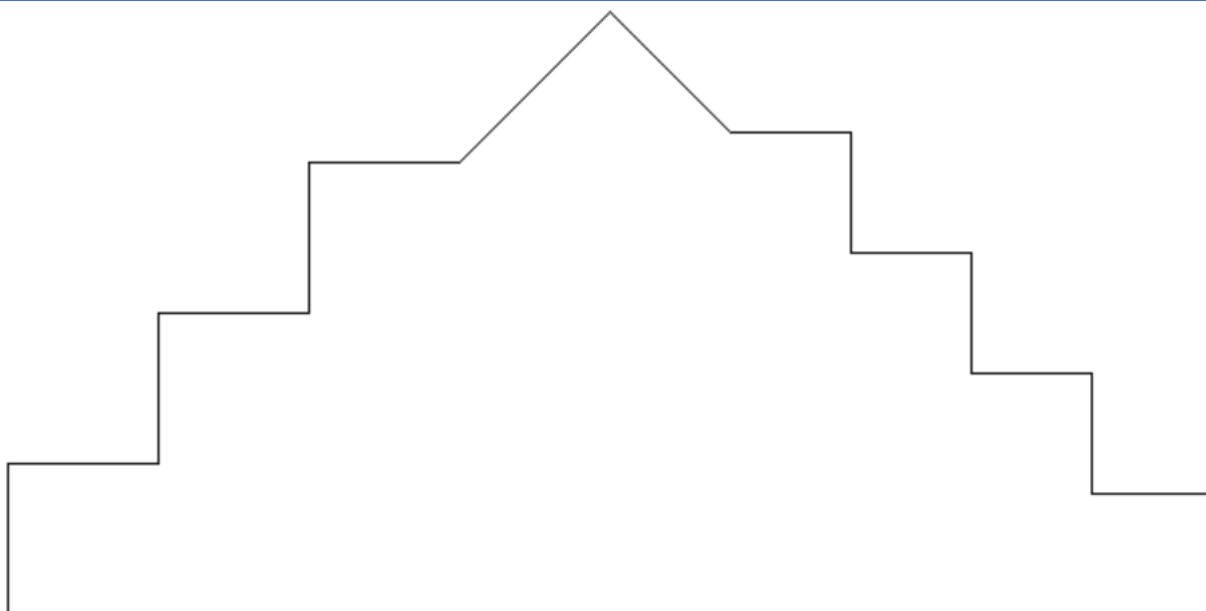
tarasy(1289)



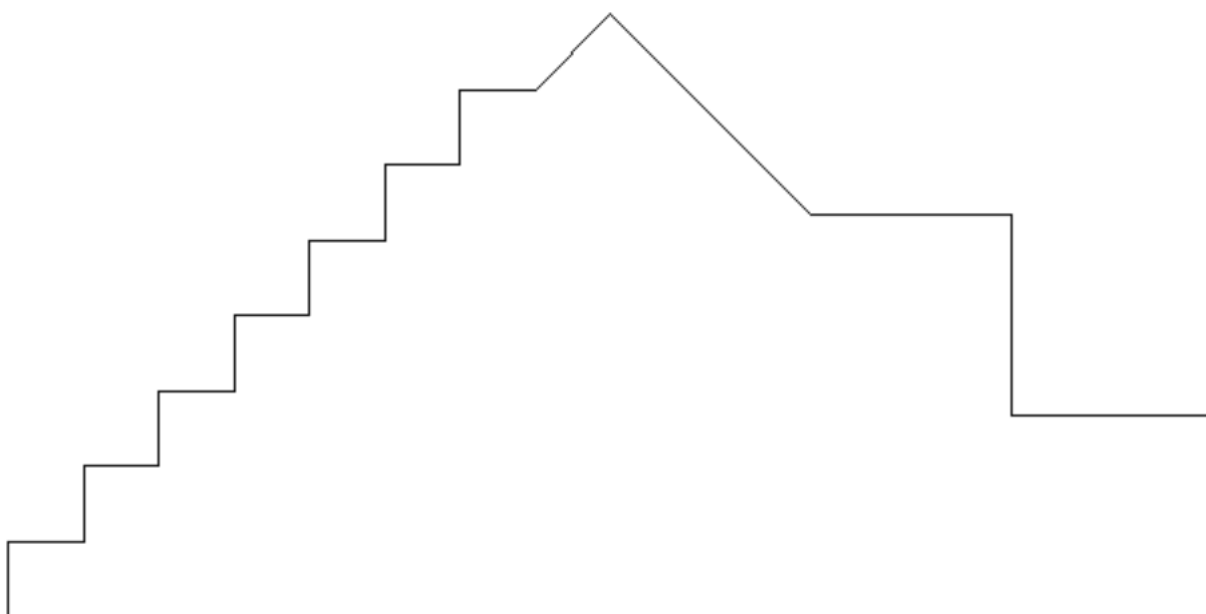
tarasy(135244)



Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty



tarasy(2231578)



tarasy(246824491)