

## Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

### Zadanie Szyfr trójkątny – LOGIA 23 (2022/23), etap 2

#### Treść zadania

Adaś postanowił kolejne cyfry systemu dziesiętnego przedstawiać za pomocą czterech bitów (cyfr 0 lub 1). Kolejne cyfry od 0 do 9 przedstawione w ten sposób to:

0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001.

Kolejny etap szyfru Adasia polega na przedstawieniu każdej cyfry od 0 do 9 w postaci czterech trójkątów, zamalowanych kolorem białym lub zielonym, w zależności od tego, czy dany trójkąt reprezentuje 0 lub 1 w cyfrze dziesiętnej przedstawionej na czterech bitach.

cyfra 0 – 0000	cyfra 1 – 0001	cyfra 2 – 0010	cyfra 3 – 0011	cyfra 4 – 0100
cyfra 5 – 0101	cyfra 6 – 0110	cyfra 7 – 0111	cyfra 8 – 1000	cyfra 9 – 1001

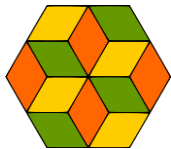
Napisz funkcję **pasek(n)**, po wywołaniu której powstanie rysunek paska powstałego po zaszyfrowaniu liczby całkowitej z zakresu od 0 do  $10^{10}$ . Kolejne cyfry liczby podanej jako parametr przedstawione są na rysunku w postaci czterech trójkątów reprezentujących każdą z cyfr. Długość boku trójkąta wynosi 26. Rysunek powinien być jednakowo odległy od lewej i prawej strony ekranu.

Przykłady:

<b>pasek (4)</b>	<b>pasek (95)</b>
<b>pasek (1203)</b>	

#### Omówienie rozwiązania

Można zauważyć, że cyfry systemu dziesiętnego są przedstawione za pomocą czterech bitów. Warto napisać funkcję zamiany cyfry w systemie dziesiętnym na liczbę binarną. Wynik funkcji – liczba binarna musi zawsze mieć 4 cyfry niezależnie od jej wartości.



## Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

```
1 def szyfruj(cyfra):
2     dw = ''
3     for i in range(4):
4         dw = str(cyfra % 2) + dw
5         cyfra = cyfra // 2
6     return dw
```

Analizując zapis binarny cyfry dziesiętnej rysujemy kolejne trójkąty zamalowane kolorem białym w przypadku zera i kolorem zielonym w przypadku cyfry jeden.

```
1 def jedna(bok, cyfra):
2     d = szyfruj(cyfra)
3     tr(bok, d[0]); fd(bok); lt(60)
4     tr(bok, d[1]); rt(60)
5     tr(bok, d[2]); fd(bok); lt(60)
6     tr(bok, d[3]); rt(60)
```

Wykorzystujemy funkcję pomocniczą rysowania trójkąta zamalowanego kolorem podanym jako drugi parametr (pierwszym parametrem jest długość boku trójkąta).

```
1 def tr(bok, kolor):
2     if kolor == '0':
3         fillcolor('white')
4     else:
5         fillcolor('green')
6     begin_fill()
7     for i in range(3):
8         fd(bok); lt(120)
9     end_fill()
```

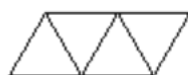
W funkcji głównej **pasek(liczba)** zaczynamy od zamiany liczby na napis, co ułatwi przeglądanie podanej jako parametr liczby. Następnie deklarujemy zmienną **bok** i ustalamy zgodnie z treścią zadania jej wartość na 26.

Ustawiamy żółwia w takim miejscu, aby po narysowaniu paska rysunek był jednakowo odległy od lewej i prawej strony ekranu. Zauważmy, że szerokość rysunku to  $2 * \text{liczba cyfr liczby podanej jako parametr} * \text{długość boku trójkąta} + \text{połowa długości boku trójkąta}$ .

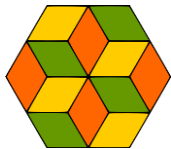
```
1 def pasek(liczba):
2     liczba = str(liczba)
3     bok = 26
4     pu(); bk(bok * len(liczba) + bok / 4); pd()
5     for i in range(len(liczba)):
6         jedna(bok, int(liczba[i]))
```

### Testy

Sprawdzamy czy rysunek paska jest prawidłowy dla podanych przykładów z treści zadania. Podczas konkursu rozwiązanie było testowane dla poniższych danych.



pasek(0)



## Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

---



pasek(12)



pasek(345)



pasek(6789)



pasek(88888888)



pasek(9876504312)