

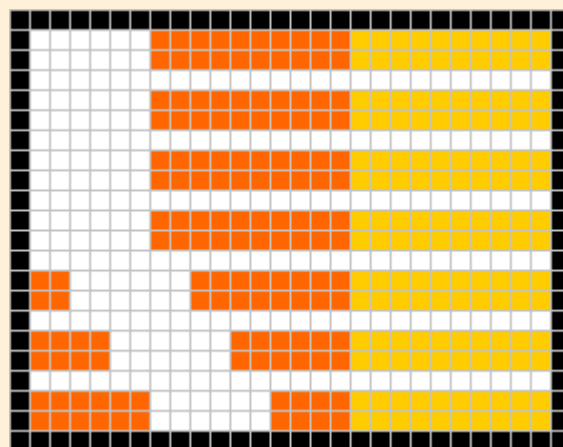
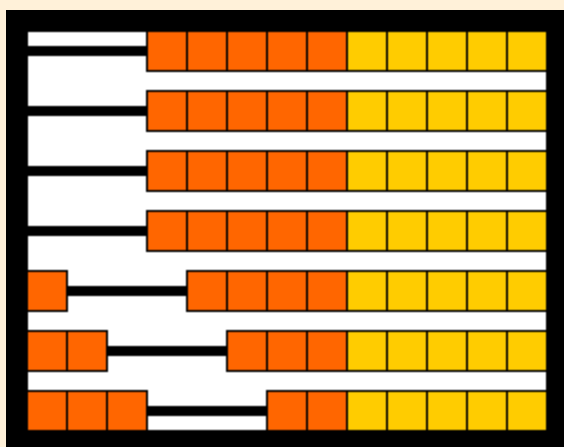


# Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

## Zadanie Liczydło – LOGIA 24 (2023/24), etap 2

### Treść zadania

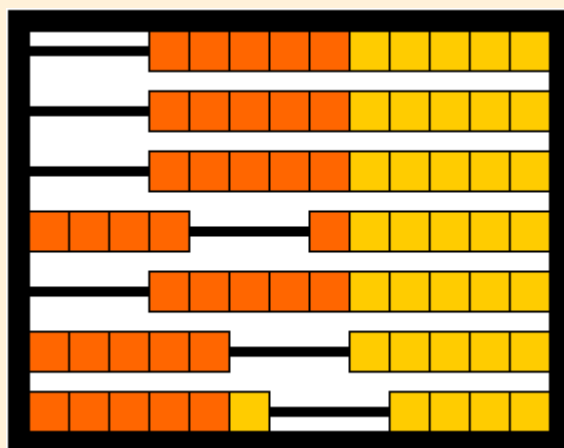
Liczydło to przyrząd służący do wykonywania prostych działań arytmetycznych, zbudowany z kolorowych ruchomych koralików umieszczonych na prętach. Koraliki są w dwóch kolorach pogrupowane po 5, tak jak na rysunkach przykładowych. Każdy poziom odpowiada określonej potęgze liczby 10. Dolny poziom reprezentuje jedności ( $10^0$ ), drugi od dołu dziesiątki ( $10^1$ ), kolejny setki ( $10^2$ ), itd. Liczbę można przedstawić na liczydłe przesuwając wybrane koraliki z prawej strony na lewą, maksymalnie 9 na jednym poziomie. Układ na rysunku poniżej reprezentuje liczbę 123 na liczydłe złożonym z 7 prętów.



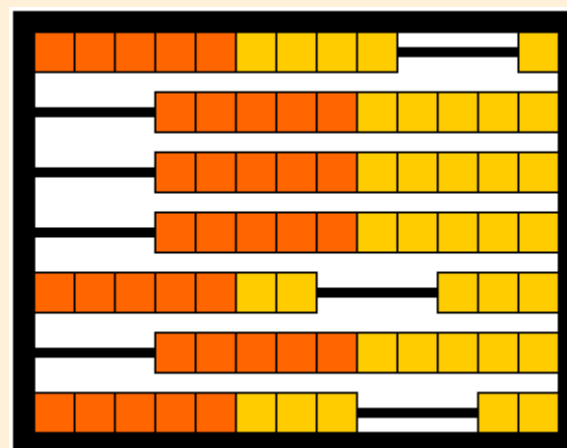
rysunek pomocniczy

Napisz funkcję **ustaw(n)**, po wywołaniu której powstanie na środku ekranu rysunek liczydła złożonego z 7 prętów z ustawionymi koralikami reprezentującymi parametr **n** z zakresu od 0 do 9 999 999. Długość boku koralika wynosi 50, a grubość pręta 10. Pozostałe wymiary odczytaj z rysunku pomocniczego.

Przykłady:



ustaw(4056)



ustaw(9000708)



# Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

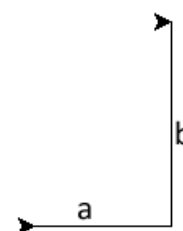
## Omówienie rozwiązania

Podstawowymi elementami liczydła są kolorowe prostokąty, dlatego w rozwiązaniu będzie wykorzystana funkcja pomocnicza **prost** z trzema parametrami – **szerokość** i **wysokość prostokąta** oraz **kolor zamalowania**. Rysunek prostokąta będzie tworzony z dolnego lewego narożnika.

```
1 def prostokat(x, y, kolor):
2     fillcolor(kolor)
3     begin_fill()
4     for i in range(2):
5         fd(x); lt(90); fd(y); lt(90)
6     end_fill()
```

Kolejną funkcję pomocniczą **skok** z dwoma parametrami **a** i **b** warto wykorzystać do przemieszczania żółwia z podniesionym pisakiem po ekranie.

```
1 def skok(a, b):
2     pu()
3     lt(90); fd(b); rt(90); fd(a)
4     pd()
```



Cały rysunek można podzielić na trzy etapy. Rysowanie:

- ramki,
- prętów,
- koralików.

Przyjmijmy za jednostkę długość boku koralika. Ramka składa się z czarnego prostokąta o szerokości  $14 \cdot \text{bok}$  i wysokości  $11 \cdot \text{bok}$  oraz nałożonego na niego białego prostokąta o szerokości  $13 \cdot \text{bok}$  i wysokości  $10 \cdot \text{bok}$ . Prostokąty są współśrodkowe.

```
1 def ramka(bok):
2     prostokat(14 * bok, 11 * bok, "black")
3     skok(0.5 * bok, 0.5 * bok)
4     prostokat(13 * bok, 10 * bok, "white")
```

Pręty są koloru czarnego. Szerokość prętów wynosi  $13 \cdot \text{bok}$  a ich wysokość jest równa  $1/5$  długości boku. Odległości między prętami to  $3/2$  długości boku.

```
1 def prety(bok):
2     for i in range(7):
3         prostokat(13 * bok, bok / 5, "black")
4         skok(0, 1.5 * bok)
```

Ostatni etap odnosi się do kluczowej części zadania – rysowania koralików. Zrobimy to dla pojedynczej cyfry. Należy zauważyć, że zawsze mamy do narysowania 10 koralików. Pierwsze pięć ma kolor pomarańczowy (reprezentacja w kodzie szesnastkowym '#ff6600'), kolejne pięć kolor żółty ('#ffcc00'). Po lewej stronie liczydła rysujemy tyle koralików, jaka jest wartość parametru **cyfra**. Następnie robimy odstęp szerokości trzech koralików i rysujemy pozostałe koraliki. Możemy przy tym posłużyć się regułą:



## Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

jeżeli masz do narysowania zero koralików po lewej stronie liczydła to  
przesuń się o trzy długości boku koralika w prawo  
narysuj koralik i przesuń się o długość boku koralika w prawo  
zmniejsz liczbę koralików do narysowania o jeden

```
1 def koraliki(cyfra, bok):
2     kolor = "#ff6600"
3     po_lewej = cyfra
4     for j in range(10):
5         if j > 4:
6             kolor = "#ffcc00"
7         if po_lewej == 0:
8             skok(3 * bok, 0)
9         prostokat(bok, bok, kolor)
10        skok(bok, 0)
11        po_lewej -= 1
```

Cały układ składa się z siedmiu serii koralików. Idąc od dołu liczydła, serie koralików odpowiadają poszczególnym cyfrom liczby od prawej do lewej. Najniższa seria odpowiada ostatniej cyfrze, druga przedostatniej itd. Ostatnią cyfrę danej liczby wyznaczamy obliczając jej resztę z dzielenia przez 10. Po narysowaniu układu dla danej cyfry liczbę dzielimy całkowicie przez 10. Odcinamy w ten sposób rozpatrzoną cyfrę. Jeśli liczba ma mniej cyfr niż prętów, to na górnych prętach są reprezentacje cyfry 0.

Funkcję **ustaw()** należy uzupełnić o skoki tak, aby rysunek był na środku ekranu oraz poszczególne elementy znajdowały się we właściwych miejscach.

```
1 def ustaw(n):
2     bok = 50
3     #ramka
4     skok(-7 * bok, -5.5 * bok)
5     ramka(bok)
6     #pręty
7     skok(0, 2 * bok / 5)
8     prety(bok)
9     #korale
10    skok(0, -7 * 1.5 * bok - 2 * bok / 5)
11    for i in range(7):
12        cyfra = n % 10
13        n = n // 10
14        koraliki(cyfra, bok)
15        skok(-13 * bok, 1.5 * bok)
```



# Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

## Rozwiązanie w języku Python

```
1 from turtle import *
2 def prostokat(x, y, kolor):
3     fillcolor(kolor)
4     begin_fill()
5     for i in range(2):
6         fd(x); lt(90); fd(y); lt(90)
7     end_fill()
8
9 def skok(a, b):
10    pu()
11    lt(90); fd(b); rt(90); fd(a)
12    pd()
13
14 def ramka(bok):
15    prostokat(14 * bok, 11 * bok, "black")
16    skok(0.5 * bok, 0.5 * bok)
17    prostokat(13 * bok, 10 * bok, "white")
18
19 def prety(bok):
20    for i in range(7):
21        prostokat(13 * bok, bok / 5, "black")
22        skok(0, 1.5 * bok)
23
24 def koraliki(cyfra, bok):
25    kolor = "#ff6600"
26    po_lewej = cyfra
27    for j in range(10):
28        if j > 4:
29            kolor = "#ffcc00"
30            if po_lewej == 0:
31                skok(3 * bok, 0)
32            prostokat(bok, bok, kolor)
33            skok(bok, 0)
34            po_lewej -= 1
35
36 def ustaw(n):
37    bok = 50
38    #ramka
39    skok(-7 * bok, -5.5 * bok)
40    ramka(bok)
41    #prety
42    skok(0, 2 * bok / 5)
43    prety(bok)
44    #korale
45    skok(0, -7 * 1.5 * bok - 2 * bok / 5)
46    for i in range(7):
47        cyfra = n % 10
48        n = n // 10
49        koraliki(cyfra, bok)
50    skok(-13 * bok, 1.5 * bok)
```



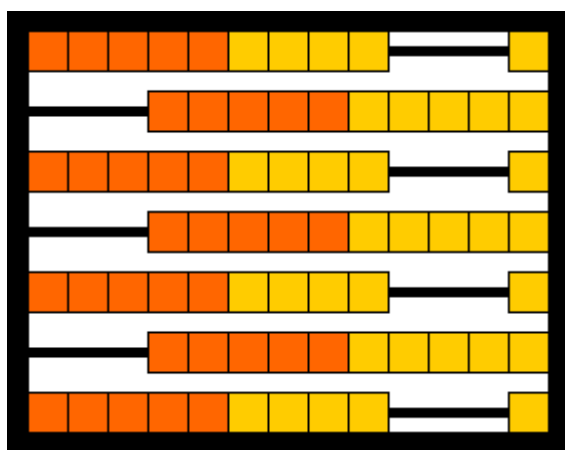
## Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

### Testy

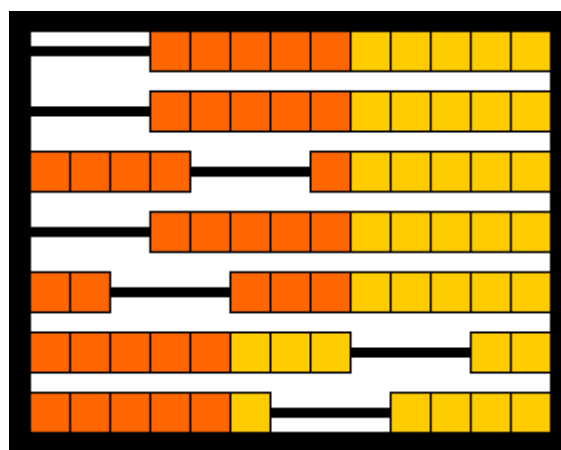
Wywołujemy funkcję **ustaw()** dla różnych parametrów. Liczby wytypowane do testów powinny zawierać wszystkie cyfry od 0 do 9 oraz składać się od 1 do 7 cyfr. Należy uwzględnić liczbę 0 oraz liczbę kończącą się cyfrą 0 (np. 7953100).

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

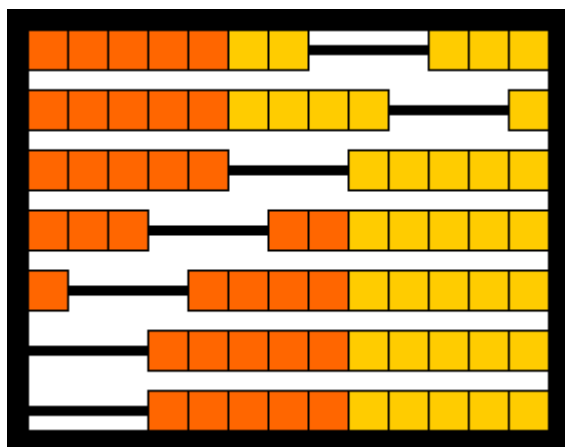
```
tracer(0); ustaw(); update()
```



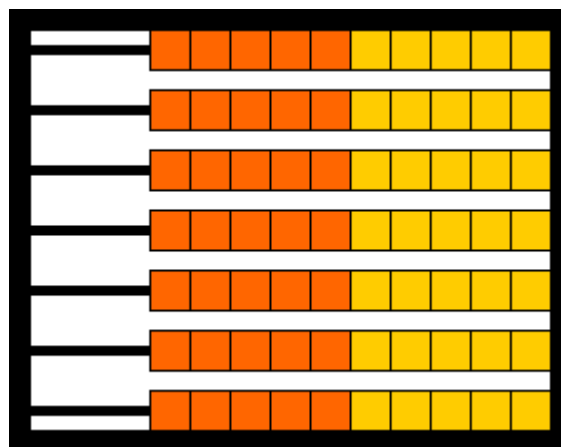
ustaw(9090909)



ustaw(40286)



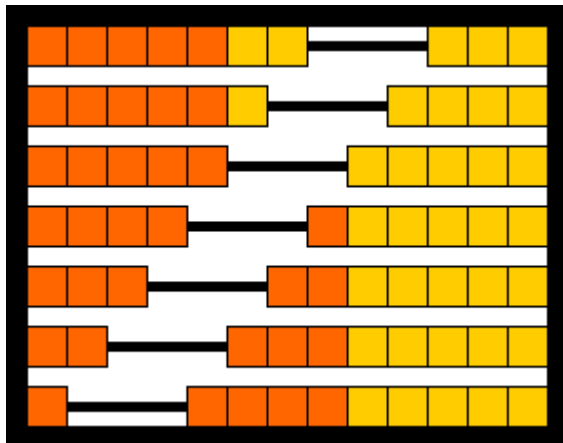
ustaw(7953100)



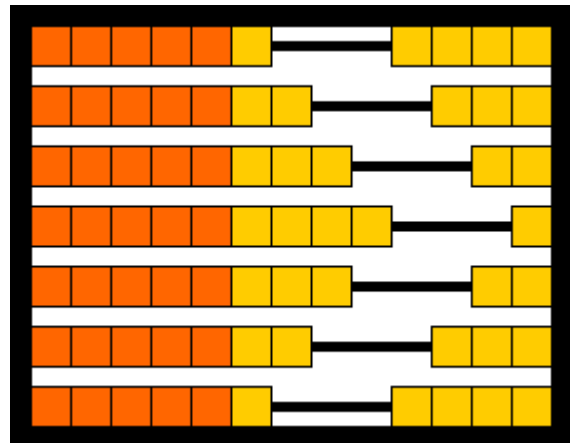
ustaw(0)



## Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty



ustaw(7654321)



ustaw(6789876)