

Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Zadanie Zawijasy – LOGIA 18 (2017/18), etap 2

Treść zadania

Tablica Polibiusza jest kwadratową tabelą zawierającą litery alfabetu łacińskiego. Kolumny numerujemy od 0 do 4, a wiersze od 1 do 5. Kodujemy słowo, zastępując każdą literę sumą numerów wiersza i kolumny, w których stoi. Na przykład litera **s** zostanie zastąpiona liczbą 6.

Napisz jednoparametrową procedurę/funkcję **koduj**, której parametrem jest słowo składające się z co najmniej 2 i co najwyżej 18 małych liter alfabetu łacińskiego. Po jej wywołaniu na środku ekranu powstaje rysunek zakodowanego słowa. Każdą literę słowa zastępujemy liczbą, według zasady podanej powyżej, a następnie rysujemy zawijas stopnia takiego, jak liczba odpowiadająca zakodowanej literze. Pierwszy odcinek zawijasa jest poziomą kreską. Każdy kolejny odcinek zawijasa jest o 4 krótszy od poprzedniego. Odległości między kolejnymi zawijasami są równe $\frac{1}{5}$ długości najdłuższego odcinka. Szerokość rysunku wynosi 780.

	0	1	2	3	4
1	a	b	c	d	e
2	f	g	h	i/j	k
3	l	m	n	o	p
4	q	r	s	t	u
5	v	w	x	y	z



Rysunek pomocniczy – zawijasy stopnia 6, 7 i 9



efekt wywołania: w Pythonie – `koduj ("fghjastuz")`, w Logo – `koduj "fghjastuz"`

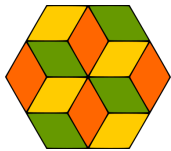
Omówienie rozwiązania

Warto zastanowić się chwilę, jak sprytnie zamienić każdą z liter alfabetu łacińskiego na liczbę będącą sumą numerów jej wiersza i kolumny. Warto skorzystać z operacji obliczania reszty z dzielenia oraz ilorazu całkowitego.

Po zamianie liter na liczby, możemy przystąpić do rysowania zawijasów.

Z treści zadania wynika, że szerokość całego rysunku jest stała, niezależna od liczby kodowanych liter. Dlatego musimy obliczyć szerokość pojedynczego zawijasa. Należy wziąć pod uwagę fakt, że zawijasów jest tyle co liter, a odstępów pomiędzy nimi mniej o 1.

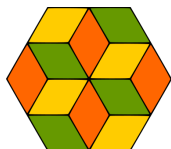
Zgodnie z treścią zadania rysunek ma być wyśrodkowany, dlatego ważna jest wysokość rysunku. Wszystkie litery, oprócz **a**, mają wysokość równą szerokości zawijasa pomniejszonej o 4, bo tyle wynosi drugi odcinek zawijasa. W przypadku litery **a**, zawijas składa się z tylko jednego odcinka poziomego, wtedy jego wysokość wynosi 0 i trzeba to uwzględnić przy wyśrodkowaniu.



Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Rozwiązanie w języku Python

```
1. from turtle import *
2.
3. def liczba(litera):
4.     alfabet = "abcdefghijklmnopqrstuvwxyz"
5.     if (litera == "i"):
6.         return 5
7.     return alfabet.index(litera) % 5 + alfabet.index(litera) // 5 + 1
8.
9. def zawijas(a, ile):
10.    for i in range(ile):
11.        fd(a); lt(90)
12.        a -= 4
13.    for i in range(ile):
14.        a += 4
15.        rt(90); bk(a)
16.
17. def czysamoa(wyraz):
18.    for i in wyraz:
19.        if(i != "a"):
20.            return False
21.    return True
22.
23. def koduj(wyraz):
24.    # szerokość zawijasa
25.    a = 780 / (1.2 * len(wyraz) - 0.2)
26.    pu(); bk(390); pd()
27.    if not czysamoa(wyraz):
28.        pu(); lt(90); bk((a - 4)/2); rt(90); pd()
29.    for i in range(len(wyraz)):
30.        zawijas(a, liczba(wyraz[i]))
31.        pu(); fd(1.2 * a); pd()
```



Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Rozwiązanie w języku Logo

```
1. oto liczba :litera
2. niech "alfabet "abcdefghijklmnopqrstuvwxyz
3. jeśli (:litera = "i) [wy 5]
4. niech "nr (numel :litera :alfabet) - 1
5. wy (reszta :nr 5) + (ilorazc :nr 5) + 1
6. już
7.
8. oto zawijas :a :ile
9. powtórz :ile [np :a lw 90 (zmniejsz "a 4)]
10. powtórz :ile [(zwiększ "a 4) pw 90 ws :a]
11. już
12.
13. oto czysamoa :wyraz
14. powtórz długość :wyraz [jeśli (element npw :wyraz) <> "a [wy "fałsz]]
15. wy "prawda
16. już
17.
18. oto koduj :wyraz
19. ; szerokość zawijasa
20. niech "a 780 / (1.2 * (długość :wyraz) - 0.2)
21. pw 90
22. pod ws 390 opu
23. jeśli nie czysamoa :wyraz[pod lw 90 ws (:a - 4) / 2 pw 90 opu]
24. powtórz długość :wyraz [zawijas :a liczba element npw :wyraz
25.                             pod np 1.2 * :a opu]
26. już
```

Wszystkie zawijasy mają pierwszy odcinek poziomy, dlatego warto w Logo na samym początku procedury głównej żółwia obrócić w prawo o 90°.

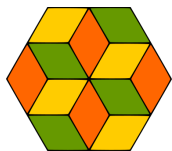
Podczas rozwiązywania zadań, powinno się testować procedury/funkcje pomocnicze. Jak zrobić to w jednej linijce dla wszystkich możliwych wartości parametru? Na przykład, aby sprawdzić, czy funkcja `liczba` daje prawidłowe wyniki dla wszystkich liter alfabetu łacińskiego w Pythonie można napisać taką pętlę:

```
1. for i in range(26):
2.     print(liczba(chr(97 + i)))
```

To samo w Logo będzie to wyglądało następująco:

```
1. powtórz 26 [pokaż liczba znak 96 + npw]
```

Kolejny test, który warto wykonać, to sprawdzenie, czy szerokość rysunku jest prawidłowa.



Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Testy

Testowanie rozwiązania powinno obejmować wszystkie litery i przypadki szczególne.

Python

```
koĉuj ("abcde")
```



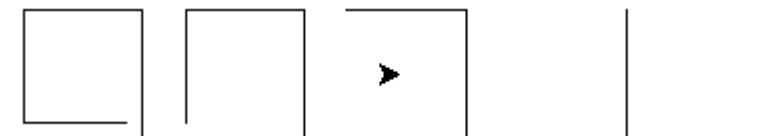
Logo

```
koĉuj "abcde"
```

*Uwaga! Na rzucie żółt stoi na środku ekranu. Sprawdzamy w ten sposób wyśrodkowanie rysunku.
Dane z pierwszego wiersza tablicy Polibiusza.*

```
koĉuj ("vqlfa")
```

```
koĉuj "vqlfa"
```



Dane z pierwszej kolumny tablicy Polibiusza

```
koĉuj ("ngatz")
```

```
koĉuj "ngatz"
```



Dane z przekątnej tablicy Polibiusza

```
koĉuj ("hijkmp")
```

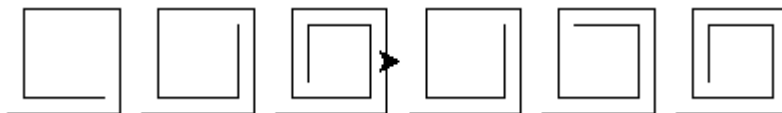
```
koĉuj "hijkmp"
```



*Zwracamy uwagę, czy kody liter i oraz j są identyczne.
Pozostałe dane z tablicy Polibiusza*

```
koĉuj ("rsuwxy")
```

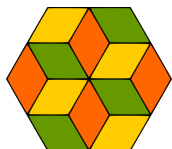
```
koĉuj "rsuwxy"
```



Pozostałe dane z tablicy Polibiusza

```
koĉuj ("aaaaaaa")
```

```
koĉuj "aaaaaaa"
```



Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty



Szczególny przypadek – wszystkie litery a.

kodej ("cccccccccccccccccccc")

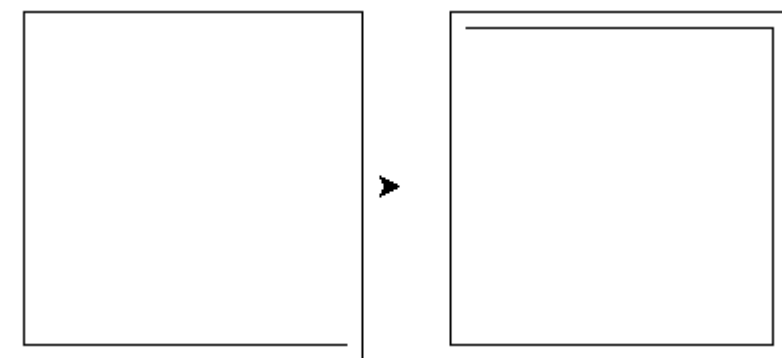
kodej "cccccccccccccccccccc"



Szczególny przypadek – wszystkie jednakowe litery, maksymalna długość danych.

kodej ("rp")

kodej "rp"



Szczególny przypadek – tylko 2 litery.