



# Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

## Zadanie Szyfr Bacona – LOGIA 17 (2016/17), etap 2

### Treść zadania

Szyfr Bacona polega na zastępowaniu liter alfabetu łacińskiego pięciorazowymi ciągami złożonymi z liter a i b zgodnie z poniższą tabelą:

A	B	C	D	E	F	G	H	I oraz J	K	L	M
aaaaa	aaaab	aaaba	aaabb	aabaa	aabab	aabba	aabbb	abaaa	abaab	ababa	ababb

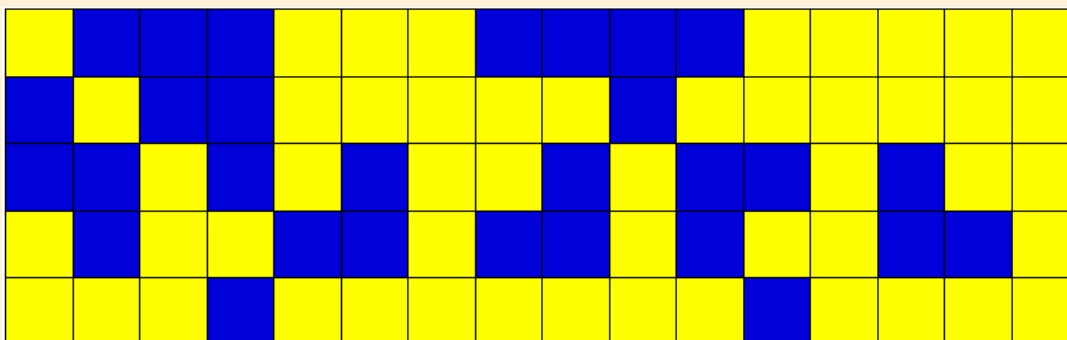
N	O	P	Q	R	S	T	U i V	W	X	Y	Z
abbaa	abbab	abbba	abbbb	baaaa	baaab	baaba	baabb	babaa	babab	babba	babbb

Adaś postanowił szyfrować kolejne litery słowa. Zamiast liter a i b użył odpowiednio zamalowanych kwadratów:

A	B	C	D	...	X	Y	Z
				...			

Napisz jednoparametrową procedurę/funkcję szyfruj, po wywołaniu której na środku ekranu powstanie rysunek zaszyfrowanego słowa (od lewej do prawej). Parametrem jest niepuste słowo złożone z wielkich liter alfabetu łacińskiego o maksymalnej długości 70. Szerokość rysunku wynosi 700 lub wysokość wynosi 400. Do zamalowania użyj dwóch dowolnych kolorów różnych od koloru krawędzi kwadratów.

Przykład:



efekt wywołania:

w Logo – szyfruj "GODZINAKODOWANIA"

w Pythonie – szyfruj ('GODZINAKODOWANIA')



# Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

## Omówienie rozwiązania

Mamy do czynienia z szyfrem podstawieniowym. Każdej literze w danej odpowiada „słupek” składający się z odpowiednio zamalowanych kwadratów. Wystarczy tylko przejrzeć całe słowo litera po literze i rysować słupki odpowiadające aktualnej literze.

Najpierw warto zapamiętać sposób kodowania poszczególnych liter. W tym celu tworzymy zmienną **alfabet**, której wartością jest lista 26 kodów kolejnych liter. Litera **A** została zakodowana jako **aaaaa**, litera **B** jako **aaaab**, litera **C** jako **aaaba** itd. Ostatniej literze alfabetu łacińskiego **Z**, odpowiada kod **babbb**.

Kody ASCII wielkich liter są liczbami z zakresu od 65 (kod ASCII litery A) do 90 (kod ASCII litery Z). Kod ASCII aktualnej litery danego słowa pomniejszony o 65 jest indeksem zmiennej **alfabet**. Przeglądając kolejne znaki słowa podanego jako parametr znajdujemy jego kod. Następnie przeglądamy znaki kodu i sprawdzamy jaką wartość reprezentują. Jeśli napotkamy **b**, rysujemy kwadrat zamalowany kolorem niebieskim, jeśli **a**, to rysujemy kwadrat zamalowany na żółto. Kolejne kwadraty rysujemy jeden nad drugim, po zakodowaniu jednej litery przesuwamy żółtą do początkowej pozycji kolejnego słupka.

W treści zadania znajduje się informacja, że rysunek ma mieć szerokość 700 lub wysokość 400. Dla wszystkich wartości parametrów rysunek musi mieścić się na ekranie. Wytyczne do rozwiązywania zadań określają wymiary ekranu, które obowiązują podczas trwania zawodów. Każdy rysunek musi się mieścić na stronie (w Logomocji), przy czym nie wolno zmieniać standardowego rozmiaru strony lub w prostokącie o szerokości 796 i wysokości 499 oraz o środku w punkcie (0,0) (w Pythonie).

Samo wyliczenie długości boku jednego kwadratu jako iloraz 700 przez długość danego słowa nie wystarczy, trzeba sprawdzić, czy słupek pięciu kwadratów zmieści się w pionie na ekranie.

## Rozwiązanie w języku Python

```
1. from turtle import *
2.
3. def kw(a, kolor):
4.     fillcolor(kolor)
5.     begin_fill()
6.     for i in range(4):
7.         fd(a); rt(90)
8.     end_fill()
9.
10. def szyfruj(s):
11.     # obliczenia wielkości boku jednego kwadratu
12.     a = 700 / len(s)
13.     # sprawdzenie, czy rysunek mieści się w pionie na ekranie
14.     # jeśli nie, to obliczmy wielkość boku zakładając, że wysokość rysunku = 400
15.     if a * 5 > 498:
16.         a = 400 / 5
17.     # przejście do punktu rozpoczęcia rysowania
18.     pu(); bk(a * len(s)/2); lt(90); bk(a * 5 / 2); pd()
19.
20.     # definiowanie zakodowanych liter
21.     alfabet = ["aaaaa", "aaaab", "aaaba", "aaabb", "aabaa", "aabab", "aabba",
22. "aabbb", "abaaa", "abaab", "ababa", "ababb", "abbab", "abbba",
23. "abbbb", "baaaa", "baaab", "baaba", "baabb", "babaa", "babab", "babba",
24. "babbb"]
25.     # przeglądanie kolejnych liter danego słowa w pętli for
26.     for litera in s:
27.         # kod litery
```



## Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

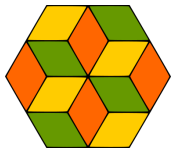
```
25.     x = alfabet[ord(litera) - 65]
26.     # rysowanie słupka danej litery
27.     # przeglądanie kolejnych znaków kodu litery w pętli for
28.     for zn in x:
29.         if zn == "b":
30.             kw(a, "blue")
31.             fd(a)
32.         else:
33.             kw(a, "yellow")
34.             fd(a)
35.     # powrót po narysowaniu słupka litery
36.     pu(); bk(5*a); rt(90); fd(a); lt(90); pd()
```

### Rozwiązanie w języku Logo

```
1.  oto kw :a :kolor
2.  ukm :kolor
3.  wielokąt [powtórz 4[np :a pw 90]]
4.  już
5.
6.  oto szyfruj :s
7.  ; obliczenia wielkości boku jednego kwadratu
8.  niech "a 700 / długość :s
9.  ; sprawdzenie, czy rysunek mieści się w pionie na ekranie
10. ; jeśli nie, to obliczymy wielkość boku zakładając, że wysokość rysunku = 400
11. jeśli :a * 5 > 498 [niech "a 400 / 5]
12. ; przejście do punktu rozpoczęcia rysowania
13. pod pw 90 ws (:a * długość :s) / 2
14. lw 90 ws 5 * :a / 2 opu
15. ; definiowanie zakodowanych liter
16. niech "alfabet [aaaaa aaaab aaaba aaabb aabaa aabab aabba aabbb aaaaa
17.               abaaa abaab ababa ababb abbaa abbab abbba abbbb baaaa
18.               baaaab baaba baabb baabb babaa babab babba bbbbb]
19. ; przeglądanie kolejnych liter danego słowa w pętli powtórz
20. powtórz długość :s[
21.     niech "x element ((ascii pierw :s) - 64) :alfabet
22.     ; numerowanie elementów zmiennej alfabet zaczyna się od 1
23.     ; dlatego odejmujemy 64, a nie 65 jak było w Pythonie
24.     ; rysowanie słupka odpowiadającego jednej literze
25.     ; przeglądanie kolejnych znaków kodu litery
26.     powtórz 5[jeżeli pierw :x = "b[kw :a "jasnoniebieski np :a]
27.               [kw :a "żółty np :a]
28.     niech "x bp :x]
29.     niech "s bp :s
30.     ; przejście do rysowania kolejnego słupka
31.     pod ws 5 * :a pw 90 np :a lw 90 opu
32. ]
33. już
```

### Testy

Testowanie rozwiązania powinno obejmować wszystkie litery i przypadki szczególne. Pierwszy test może obejmować cały alfabet. Tutaj należy sprawdzić prawidłowość zaszyfrowania każdej litery. Należy zwrócić uwagę na to, że zaszyfrowane litery I i J są identyczne. To samo dotyczy liter U i V.



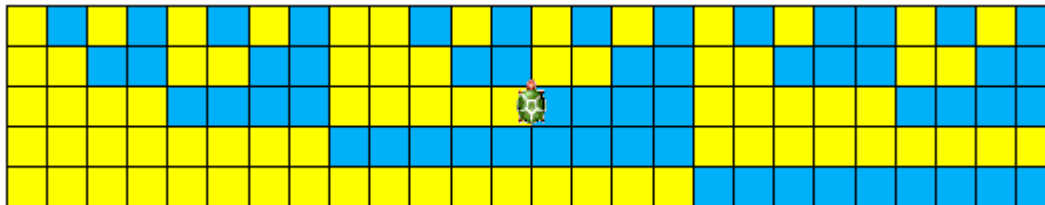
# Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Python

Logo

```
szyfruj("ABCDEFGHIJKLMNOPQRSTUVWXYZ")
```

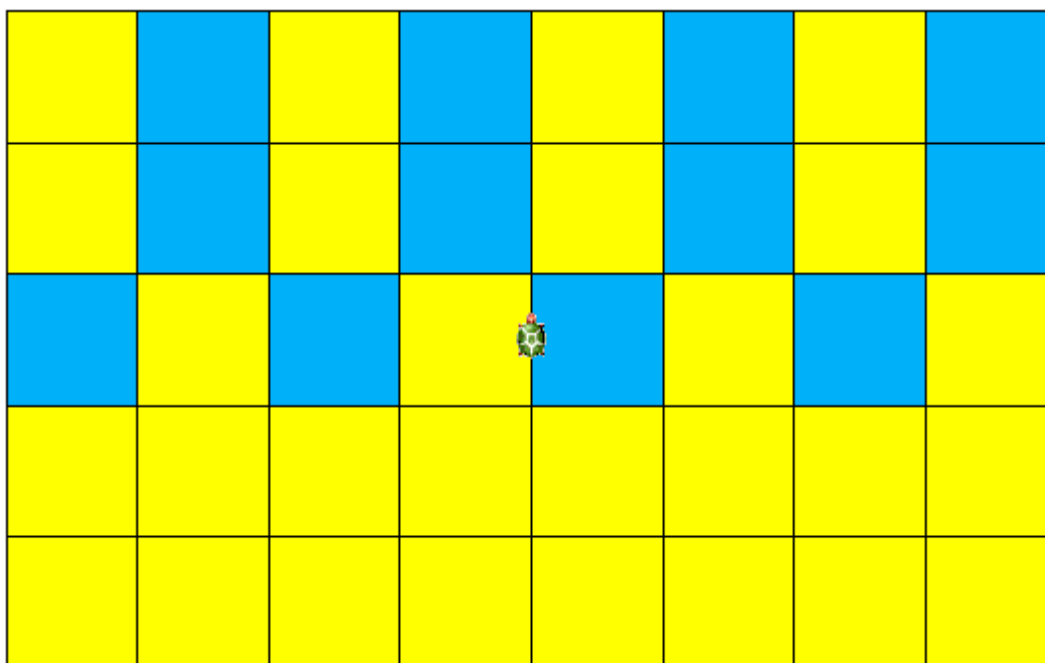
```
szyfruj "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
```



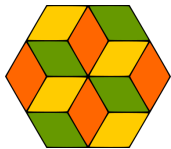
*Uwaga! Na rzucie żółtów stoi na środku ekranu. Sprawdzamy w ten sposób wyśrodkowanie rysunku. Dane, to cały alfabet. Ten rysunek ma szerokość 700.*

```
szyfruj("EDEDEDED")
```

```
szyfruj "EDEDEDED"
```



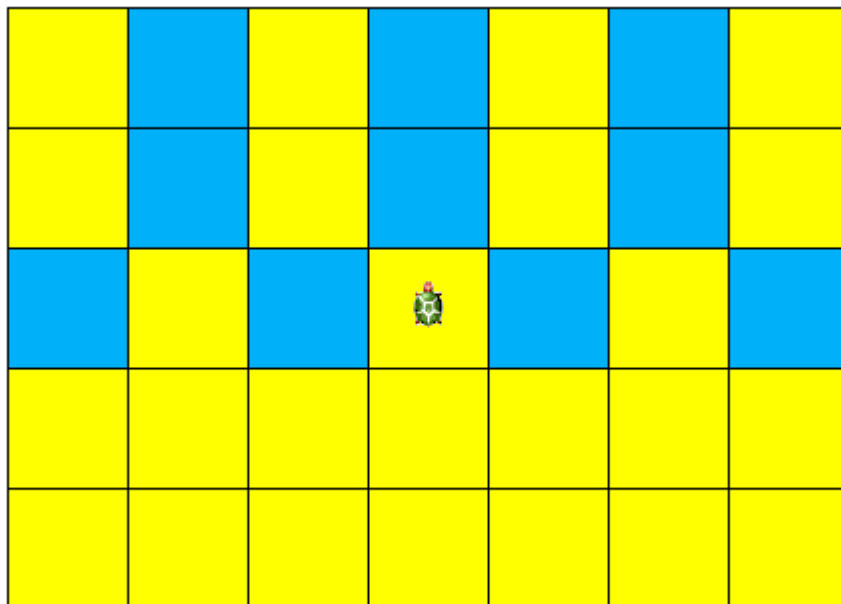
*Uwaga! Ten rysunek ma szerokość 700.*



## Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

szyfruj ("EDEDEDE")

szyfruj "EDEDEDE"

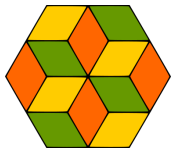


*Uwaga! Ten rysunek ma wysokość 400, ponieważ gdyby miał szerokość 700, to nie mieściłby się na ekranie w pionie.*

szyfruj ("ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ")  
szyfruj "ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ"



*Uwaga! Ten rysunek ma szerokość 700.*



## Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

---

szyfruj ("S")

szyfruj "S



*Uwaga! Ten rysunek ma wysokość 400, ponieważ gdyby miał szerokość 700, to nie mieściłby się na ekranie w pionie.*