



# Zadania drugiego etapu konkursu Logia18

– przedmiotowego konkursu informatycznego  
dla uczniów gimnazjów województwa mazowieckiego  
11 stycznia 2018 roku

## Zadanie 1 (zawijasy).

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 `ko_duj`, 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 – `ko_duj("fghjastuz")`, w Logo – `ko_duj "fghjastuz"`

## Zadanie 2 (neony).

W Turtlandii przygotowują neon do zawieszenia na dwóch słupach. Słupy stoją w rzędzie, odległość pomiędzy dwoma sąsiednimi wynosi 2. Dział marketingu uzależnia wybór słupów od oceny zdefiniowanej jako suma ich wysokości i odległości między nimi.

Napisz jednoparametrową funkcję `neon`, której wynikiem jest najwyższa możliwa ocena. Parametr jest listą wysokości kolejnych słupów (ma co najmniej 2 elementy, co najwyżej 500). Wysokość każdego słupa wynosi od 1 do 10 000.

*Przykłady:*

w Pythonie – wynikiem `neon([10, 4, 5, 7, 1, 4, 1])` jest 24 (bo najwyższa ocena będzie uzyskana przy wyborze pierwszego i przedostatniego słupa,  $24 = 10 + 4 + 5 \cdot 2$ ), wynikiem `neon([1, 10, 1])` jest 13 ( $13 = 1 + 10 + 1 \cdot 2$ , też  $13 = 10 + 1 + 1 \cdot 2$ ),

w Logo – wynikiem `neon [10 4 5 7 1 4 1]` jest 24 (bo najwyższa ocena będzie uzyskana przy wyborze pierwszego i przedostatniego słupa,  $24 = 10 + 4 + 5 \cdot 2$ ), wynikiem `neon [1 10 1]` jest 13 ( $13 = 1 + 10 + 1 \cdot 2$ , też  $13 = 10 + 1 + 1 \cdot 2$ ).

## Zadanie 3 (cukierki).

Adam ma papierową torebkę. Wrzuca do niej cukierki, a następnie częstuje nimi znajomych. Cukierki są wrzucane i wyciągane pojedynczo. Postępuje tak wielokrotnie: wrzuca i częstuje. Adam zastanawia się, ile razy w torebce było dokładnie *n* cukierków.

Napisz dwuparametrową funkcję `ilerazy`, której pierwszym parametrem jest *n* (od 1 do 10 000). Drugim parametrem funkcji jest parzystej długości lista dodatnich liczb całkowitych, zawierająca na przemian liczby wrzucanych i wyjmowanych cukierków. Na przykład lista zawierająca 4, 1, 3 i 2 oznacza, że kolejno do torebki zostały wrzucone cztery cukierki, wyjęty jeden, wrzucone trzy i wyjęte dwa. Długość listy wynosi od 2 do 1 000. Wynikiem funkcji jest liczba określająca, ile razy w torebce było dokładnie *n* cukierków.

*Przykłady:*

w Pythonie – wynikiem `ilerazy(2, [3, 2])` jest 2 (liczba cukierków w torebce to kolejno 0, 1, 2, 3, 2, 1), wynikiem `ilerazy(4, [5, 3, 5, 3, 2, 1])` jest 4 (liczba cukierków w torebce to kolejno 0, 1, 2, 3, 4, 5, 4, 3, 2, 3, 4, 5, 6, 7, 6, 5, 4, 5, 6, 5),

w Logo – wynikiem `ilerazy 2 [3 2]` jest 2 (liczba cukierków w torebce to kolejno 0, 1, 2, 3, 2, 1), wynikiem `ilerazy 4 [5 3 5 3 2 1]` jest 4 (liczba cukierków w torebce to kolejno 0, 1, 2, 3, 4, 5, 4, 3, 2, 3, 4, 5, 6, 7, 6, 5, 4, 5, 6, 5).