

# ZAMIEŃ SŁOWO NA OBRAZEK – ROZWIĄZYWANIE ZADAŃ ALGORYTMICZNYCH W PYTHONIE

Agnieszka Borowiecka  
Ośrodek Edukacji Informatycznej i Zastosowań Komputerów  
w Warszawie, ul. Raszyńska 8/10  
agnieszka.borowiecka@oeiizk.waw.pl

*Abstract. Learning should be interesting. The teacher has to choose the tasks to intrigue the student. For this purpose, you can use tasks from IT competitions about coding drawings and text encryption.*

## 1. Wprowadzenie

Programowanie przestaje być wiedzą tylko dla wtajemniczonych. Już od najmłodszych lat uczniowie poznają pojęcia związane z programowaniem i algorytmiką, tworzą własne aplikacje i rozwiązują niebanalne problemy. Zadaniem nauczyciela jest uczyć tak, by nie zniechęcić, a wręcz zaciekawić ucznia. Należy przy tym zwracać uwagę na uczniów uzdolnionych i właściwie motywować ich do dalszego pogłębiania zdobytej wiedzy. Podstawowym problemem wydaje się dobór odpowiednich zadań – dostosowanych do poziomu i umiejętności uczniów, rozwijających i interesujących, zachęcających do pozytywnej rywalizacji.

W czasie warsztatu sięgniemy do skarbnicy ciekawych zadań związanych z szyfrowaniem i kodowaniem pochodzących z archiwów konkursów informatycznych LOGIA i miniLOGIA dla uczniów z województwa mazowieckiego. Pokażemy, jak rozwiązać przykładowe zadania w języku Python. Podpowiemy, na co zwracać uwagę uczniów, jakie są kolejne kroki przy pisaniu programu i na czym polega testowanie.

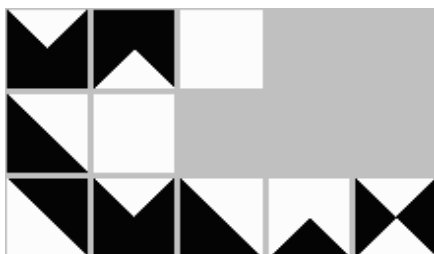
## 2. Konkursy LOGIA i miniLOGIA

Przedmiotowe konkursy informatyczne LOGIA i miniLOGIA były organizowane od 2001 roku przez Ośrodek Edukacji Informatycznej i Zastosowań Komputerów w Warszawie. Zasób zadań dostępnych na stronie konkursu [2] jest imponujący – ponad 350 różnych propozycji. Do tego dochodzą również zadania z warszawskich i mazowieckich konkursów informatycznych, organizowanych wcześniej przez Ośrodek.

Konkurs miniLOGIA był kierowany do uczniów sześcioletnich szkół podstawowych i polegał na tworzeniu różnych rysunków za pomocą grafiki żółwia. Jego ostatnia edycja odbyła się w roku szkolnym 2017-2018. Konkurs Logia przeznaczony pierwotnie do uczniów szkół gimnazjalnych, od bieżącego roku szkolnego stał się konkursem dla uczniów szkół podstawowych. Oba konkursy składają się z trzech etapów: szkolnego, rejonowego i wojewódzkiego. Tematyka konkursu LOGIA obejmuje rozwiązywanie zadań algorytmicznych z zakresu grafiki, definiowania funkcji, przetwarzania napisów oraz list jednopoziomowych i wielopoziomowych. Większość zadań konkursowych jest zbyt trudna dla początkujących, ale doskonale nadaje się dla tych, którzy są naprawdę zainteresowani programowaniem i chcą podnosić swoje umiejętności. Spróbujmy skupić się na zadaniach związanych z kodami i szyfrowaniem.

### 3. Zadania związane z szyfrowaniem

Wiele zadań na drugim etapie konkursu łączy grafikę z analizowaniem liczb i słów. Poszczególne elementy rysunku zależą od pewnego kodu, przy czym nie zawsze zostało to opisane w sposób jawny – uczniowie muszą przeanalizować przykłady dołączone do zadania i wyciągnąć wnioski. W innych zadaniach sposób kodowania rysunku został szczegółowo opisany.



**Rysunek 1.** Zdanie *Ola ma domek* zakodowane na podstawie zdefiniowanego w zadaniu alfabetu (**Kody flagowe**, III etap Mazowieckiego Konkursu Informatycznego 2000)

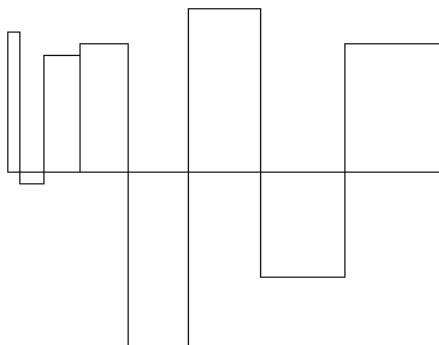
Możemy wyodrębnić kilka grup zadań dotyczących kodowania informacji. Pierwsza z nich jest związana z różnymi własnościami liczb, np. ich parzystością, podzielnością lub zapisem w różnych systemach liczbowych (zadanie *Haft krzyżykowy* z trzeciego etapu konkursu LOGIA06). W innych zadaniach analizujemy poszczególne znaki w wyrazie i na tej podstawie tworzymy rysunek (zadanie *Szyfr Bacona* z drugiego etapu konkursu LOGIA17). Czasem rysunek pełni jedynie rolę pomocniczą, ma ułatwić zrozumienie problemu. Ostatnia grupa zadań związana jest z klasycznymi szyframi, począwszy od szyfrów harcerskich (zadanie *Czekoladka* z drugiego etapu konkursu

LOGIA09), poprzez różne modyfikacje szyfru Cezara, do szyfru Vigenère'a (zadanie Szyfr Vigenère'a z drugiego etapu konkursu LOGIA SP z 2019 roku).

#### 4. Etapy rozwiązywania zadania

Rozwiązywanie zadania należy rozpocząć od przeanalizowania jego treści. Warto zwrócić uwagę uczniów na fakt, że rysunki ilustrujące przykładowy wynik działania funkcji są istotną częścią zadania i powinny być brane pod uwagę przy planowaniu rozwiązania.

W zadaniu Szyfr obrazkowy z drugiego etapu konkursu LOGIA15 definiujemy własny alfabet, a sposób szyfrowania został opisany w treści zdania i polega na rysowaniu prostokątów odpowiedniej wielkości. Położenie prostokątów zależy od tego, czy ilustrowana litera jest samogłoską czy spółgłoską. Wysokość prostokąta odpowiadającego kolejnej literze słowa zależy od jej pozycji w alfabecie, zaś jego szerokość – od pozycji litery w szyfrowanym słowie.



**Rysunek 2.** Wyraz *lajkonik* zaszyfrowany dla alfabetu *abcdefghijklmno*

Po zaplanowaniu rozwiązania, w szczególności podziału na funkcje pomocnicze i ustaleniu komunikacji między funkcjami (znaczenie parametrów), zaczynamy pisać program. Najpierw funkcje pomocnicze, np. rysującą prostokąt o podanych wymiarach, sprawdzając, czy podana litera jest samogłoską, wyliczającą wysokość rysowanego prostokąta. Każdą zdefiniowaną funkcję kilkakrotnie testujemy. Następnie przechodzimy do pisania funkcji stanowiącej rozwiązanie zadania. Warto pamiętać, by przetestować jej działanie nie tylko na przykładowych danych z treści zadania. Na koniec ponownie czytamy treść zadania i sprawdzamy, czy nasze rozwiązanie spełnia wszystkie opisane założenia.

Prezentowane zadanie Szyfr obrazkowy odwołuje się do wiedzy z języka polskiego (kolejność liter w alfabecie, rozpoznawanie samogłosek). Uczniowie muszą skorzystać z instrukcji warunkowej – różne działanie funkcji dla samogłosek i spółgłosek, poznają

również zastosowanie pętli. Elementy grafiki żółwia wykorzystane do uzyskania końcowego efektu uatrakcyjniają zadanie i rozwijają wyobraźnię.

```
from turtle import *

def pr(a, b):
    fd(a); lt(90); fd(b); lt(90)
    fd(a); lt(90); fd(b); lt(90)

def czy_samogloska(z):
    return z in 'aeiouy'

def bok(alfabet, litera):
    return 10*(alfabet.find(litera)+1)
```

Rysunek 3. Przykładowe funkcje pomocnicze

## 5. Podsumowanie

Pełną listę zadań związanych z szyfrowaniem z konkursów LOGIA (bez tegorocznej edycji) znajdziemy w artykule **Programowanie dla zainteresowanych** [1]. Należy pamiętać, że mimo iż zadania z konkursu informatycznego mogą być zbyt trudne dla przeciętnego ucznia, to większość z nich można znacząco uprościć, aby dopasować je do poziomu całej klasy. Jednocześnie stanowią one doskonały przykład nauczania problemowego – uczniowie widzą, w jakim celu uczą się instrukcji warunkowej i pętli, rozumieją potrzebę definiowania funkcji pomocniczych, poznają kolejne etapy rozwiązywania problemu.

## Literatura

1. Borowiecka A., *Programowanie dla zainteresowanych, W cyfrowej szkole* Nr 1/2019, OEliZK 2019.
2. *Konkursy Informatyczne LOGIA organizowane przez OEliZK*, <http://logia.oeizk.waw.pl>, ostatni dostęp 26.05.2019 roku