

## LOGIA – KONKURSOWE POTYCZKI W PYTHONIE

Maciej Borowiecki, Agnieszka Samulska  
Ośrodek Edukacji Informatycznej i Zastosowań Komputerów  
w Warszawie, ul Raszyńska 8/10  
{maciej.borowiecki, agnieszka.samulska}@oeiizk.edu.pl

**Abstract.** This article presents the principles of organizing the LOGIA computer science competition for students in grades 4-8 of primary schools in the Mazovian Voivodeship, in the format effective from the 2025/26 school year. It discusses sample graphical, computational, and algorithmic tasks used in the last competition, as well as the principles of testing and evaluating them.

### 1. Wprowadzenie

LOGIA to Przedmiotowy Konkurs Informatyczny dla uczniów klas IV-VIII szkół podstawowych województwa mazowieckiego powoływany przez Mazowieckiego Kuratora Oświaty, a organizowany przez Ośrodek Edukacji Informatycznej i Zastosowań Komputerów w Warszawie [5].

W roku szkolnym 2025/2026 odbyła się 32 edycja konkursu, który miał swoje początki w roku szkolnym 1994/95 jako Warszawski Konkurs Informatyczny. Konkurs polega na samodzielnym rozwiązywaniu testu dotyczącego treści zawartych w podstawie programowej przedmiotu informatyka w szkole podstawowej i zadań algorytmicznych z zakresu grafiki oraz przetwarzania napisów i list w języku programowania Python.

Od tego roku zmieniła się nieco formuła I etapu. Składał się on z dwóch części:

- testu oraz dwóch zadań algorytmicznych z zakresu grafiki oraz definiowania obliczeń, która trwała od 30 października do 18 listopada 2025;
- dwóch zadań algorytmicznych z zakresu grafiki oraz definiowania obliczeń rozwiązywanych w warunkach kontrolowanej samodzielności w macierzystej szkole ucznia w dniu 19 listopada 2025 (60 min.).

### 2. Zadania graficzne i moduł turtle

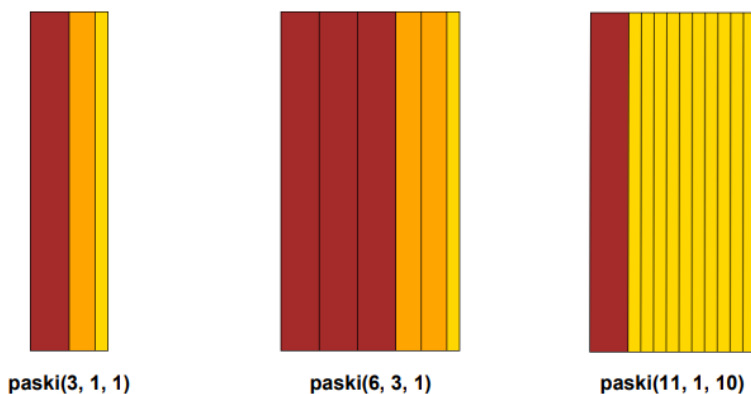
To, co odróżnia LOGIĘ od innych konkursów programistycznych dedykowanych uczniom szkół podstawowych, to zadania wykorzystujące grafikę żółwia. Historycznie uczniowie implementowali rozwiązania w języku Logo (do 2018 roku). Od 2013 roku

otrzymali możliwość rozwiązywania zadań w języku Python. Wybór alternatywnego języka programowania był nieprzypadkowy – język ten posiada moduł *turtle*, oferujący narzędzia do rysowania znane z języka Logo.

Zadania graficzne bazują głównie na prostych figurach takich, jak: wielokąt foremny, prostokąt, romb oraz trójkąt prostokątny równoramienny. Do ich rozwiązania konieczna jest wiedza matematyczna dotycząca m.in.: umiejętności wyznaczania przekątnej w kwadracie, wysokości w trójkącie równobocznym czy podziale kąta na  $n$ -równych części. Często w zadaniach pojawia się konieczność wyznaczenia proporcji elementów, skalowanie rysunków, odnajdywanie elementów podobnych itp. Jest to jedyny typ zadania, w którym uczeń musi się wykazać umiejętnością definiowania funkcji. Rysunek 1. przedstawia zadanie graficzne z części b I etapu tegorocznego konkursu.

Napisz trzyparametrową funkcję **paski(ile, s, w)**, po wywołaniu której powstanie na środku ekranu rysunek prostokąta złożonego z pionowych pasków kolejno w kolorach: brązowym, pomarańczowym i żółtym. Wysokości wszystkich pasków wynoszą 400, zaś ich szerokości są równe odpowiednio 45 – paski brązowe, 30 – paski pomarańczowe i 15 – paski żółte. Parametr **ile** oznacza liczbę wszystkich pasków ( $1 \leq \text{ile} \leq 20$ ), parametr **s** liczbę pasków brązowych, a **w** liczbę pasków żółtych ( $0 \leq s \leq 20$ ,  $0 \leq w \leq 20$ ,  $s + w \leq \text{ile}$ ).

Przykłady:



Rysunek 1. Zadanie Paski LOGIA 26 2025/202 – I etap

Zadania graficzne sprawdzane są półautomatycznie – wykorzystywany jest program, który z pracy ucznia importuje funkcję o nazwie zgodnej z treścią zadania. Wywoływana jest funkcja z odpowiednią liczbą parametrów oraz sprawdzana jest zgodność rysunku ze wzorcem. Ocenie podlegają również wymagania opisane w treści zadania. W zadaniu Paski weryfikowane były: szerokość i wysokość prostokątów, sposób kolorowania, a także umieszczenie rysunku na środku ekranu.

### 3. Obliczenia i symulacje

Od wielu lat na pierwszym etapie konkursu pojawia się zadanie obliczeniowe lub prosta symulacja. W ich przypadku do budowania programu należy wykorzystać podstawowe konstrukcje języka Python: instrukcje wejścia i wyjścia, operacje arytmetyczne i logiczne, instrukcję przypisania, instrukcje warunkowe, pętlę o określonej liczbie powtórzeń i z warunkiem oraz zmienne. Zadania są na tyle proste, aby zachęcić jak największą liczbę uczniów do podjęcia wysiłku, jakim jest start w konkursie. Rysunek 2. przedstawia zadanie obliczeniowe z części b I etapu tegorocznego konkursu.

Tosia buduje piramidę z sześciennych klocków. Obowiązują następujące zasady:

- najniższa warstwa piramidy może mieć dowolną liczbę klocków,
- każda kolejna warstwa znajduje się nad poprzednią i ma co najmniej jeden klocek mniej niż warstwa bezpośrednio pod nią.

Napisz program, który wczyta liczbę klocków do dyspozycji Tosi i wypisze maksymalną wysokość piramidy, jaką może zbudować przy zachowaniu powyższych zasad.

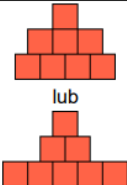
**Wejście:**

liczba całkowita dodatnia określająca liczbę klocków z zakresu od 1 do 10000

**Wyjście:**

liczba całkowita dodatnia – maksymalna wysokość piramidy

Przykłady:

|              |                                                                                     |                                                                                                                                                                                   |                                                                                     |
|--------------|-------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
| Wejście      | 2                                                                                   | 8                                                                                                                                                                                 | 11                                                                                  |
| Wyjście      | 1                                                                                   | 3                                                                                                                                                                                 | 4                                                                                   |
| Uzasadnienie |  | <br>lub<br> |  |

**Rysunek 2.** Zadanie Piramida LOGIA 26 2025/2026 – I etap

W prezentowanym zadaniu zostały opisane zasady budowania piramidy z sześciennych klocków. Aby uzyskać maksymalną wysokość piramidy, każdy jej poziom powinien składać się z możliwie najmniejszej liczby klocków. Stąd wniosek, że najwyższą warstwę najlepiej zbudować z jednego klocka. Warstwa bezpośrednio pod nim powinna składać się z dwóch klocków itd. Układamy kolejną warstwę pod warunkiem, że mamy do dyspozycji wystarczającą liczbę klocków.

Prosta symulacja może wyglądać tak: zacznij od jednego klocka; dopóki liczba klocków jest wystarczająca, buduj kolejną warstwę piramidy. Pamiętaj przy tym, że liczbę klocków należy pomniejszać o już wykorzystane oraz kolejna warstwa ma o jeden klocek więcej.

Pseudokod rozwiązania:

```
wysokość ← 0
warstwa ← 1
dopóki warstwa ≤ klocki wykonuj
    wysokość ← wysokość + 1
    klocki ← klocki - warstwa
    warstwa ← warstwa + 1
```

Zadania niegraficzne sprawdzane są automatycznie z wykorzystaniem platformy konkursowej.

```

1 def licz(klocki):
2     wynik = 0
3     warstwa = 1
4     while warstwa <= klocki:
5         wynik += 1
6         klocki -= warstwa
7         warstwa += 1
8     return wynik
9
10 x = int(input())
11 print(licz(x))

```

Sprawdź

|   | Dane wejściowe | Oczekiwane | Otrzymane |   |
|---|----------------|------------|-----------|---|
| ✓ | 6<br>21        | 3<br>6     | 3<br>6    | ✓ |
| ✓ | 12<br>31       | 4<br>7     | 4<br>7    | ✓ |
| ✓ | 911<br>4268    | 42<br>91   | 42<br>91  | ✓ |
| ✓ | 1<br>8514      | 1<br>129   | 1<br>129  | ✓ |
| ✓ | 1378<br>9871   | 52<br>140  | 52<br>140 | ✓ |

Przeszedł wszystkie testy! ✓

Rysunek 3. Walidacja zadania na platformie konkursowej

## 4. Algorytmika

Zadania algorytmiczne na 2 i 3 etapie są zróżnicowane, tak aby z jednej strony każdy z uczestników był w stanie uzyskać punkty, a z drugiej wyłonić uczniów najzdolniejszych. Najtrudniejsze zadanie często poszerza podstawę programową przedmiotu, znaczenie ma także złożoność obliczeniowa zastosowanego algorytmu. Rysunek 4. przedstawia takie zadanie z II etapu tegorocznego konkursu.

Franek interesuje się kryptografią. Dowiedział się, że liczby półpierwsze są wykorzystywane w kryptografii. Liczba półpierwsza to liczba, która jest iloczynem dwóch liczb pierwszych. Na przykład 65 jest liczbą półpierwszą ( $65 = 5 * 13$ ). Franek zastanawia się, jak dużo jedynek występuje w rozwinięciu binarnym czynników liczby półpierwszej. Na przykład czynniki liczby 65 zawierają łącznie 5 jedynek ( $5_{10} = 101_2$ ,  $13_{10} = 1101_2$ ).

Pomóż Frankowi i napisz program, który wczyta dwie liczby naturalne **a** i **b** oraz znajdzie liczbę półpierwszą w przedziale **[a, b]** o największej liczbie jedynek w rozwinięciu binarnym czynników tej liczby. Jeśli jest kilka takich liczb o największej liczbie jedynek, program powinien znaleźć największą z nich.

Postaraj się tak napisać program, żeby Franek nie czekał długo na wynik.

#### Wejście:

Dwie liczby naturalne **a** i **b** oddzielone spacją,  $4 \leq a$ ,  $a < b$ ,  $b \leq 10\,000\,000$ . Przedział **[a, b]** zawiera co najmniej jedną liczbę półpierwszą.

#### Wyjście:

Liczba naturalna – największa liczba półpierwsza w przedziale **[a, b]** o maksymalnej liczbie cyfr 1 w rozwinięciu binarnym jej czynników.

#### Przykłady:

|              | Przykład 1                                             | Przykład 2                       | Przykład 3                                                         |
|--------------|--------------------------------------------------------|----------------------------------|--------------------------------------------------------------------|
| Wejście      | 4 100                                                  | 22 25                            | 1000 2000                                                          |
| Wyjście      | 93                                                     | 25                               | 1982                                                               |
| Uzasadnienie | $93 = 3 * 31$<br>$3_{10} = 11_2$ , $31_{10} = 11111_2$ | $25 = 5 * 5$<br>$5_{10} = 101_2$ | $1982 = 2 * 991$<br>$2_{10} = 10_2$ ,<br>$991_{10} = 1111011111_2$ |

**Rysunek 4.** Zadanie Liczby półpierwsze LOGIA 26 2025/2026 – II etap

Zadanie polega na znalezieniu liczby półpierwszej, a więc dwóch liczb pierwszych (mogą być takie same) o maksymalnej sumie jedynek w ich zapisie binarnym, ale tak, aby ich iloczyn mieścił się w przedziale **[a, b]**. Nie musimy więc sprawdzać wszystkich liczb z przedziału. Zamiast tego możemy generować liczby półpierwsze jako iloczyny liczb pierwszych.

Jednym z kluczowych etapów rozwiązania jest szybkie wyznaczenie zbioru liczb pierwszych. W tym celu warto zastosować algorytm sita Eratostenesa. Kolejnym ważnym elementem jest wyznaczenie liczby jedynek w zapisie binarnym dla liczb pierwszych. Warto wyliczyć dla każdej liczby pierwszej raz liczbę jedynek i ją zapamiętać. W tym celu można zmodyfikować nieznacznie algorytm sita Eratostenesa. Oto zapis funkcji w pseudokodzie.

```

funkcja sito(n)
  p ← lista wartości 0 i 1 indeksowana od 0 do n
  # wartości początkowe listy p:
  # 1 dla indeksów nieparzystych,
  # 0 dla parzystych, dla 1: 0, dla 2: 1
  d ← 3
  dopóki d * d ≤ n wykonuj
    jeżeli p[d]=1 to
      dla i od d*d do n z krokiem d wykonuj
        p[i] ← 0
      d ← d + 2
  dla i od 3 do n z krokiem 2 wykonuj
    jeżeli p[i]=1 to
      p[i] ← liczba jedynek w zapisie binarnym i
  zwróć p i zakończ

```

Analizy liczb półpierwszych parzystych, czyli postaci  $2 \cdot i$ , gdzie  $i$  jest liczbą pierwszą, można dokonać na początku. Dzięki temu w dalszej części algorytmu można rozważać już tylko iloczyny nieparzystych liczb pierwszych, a więc zwiększać indeksy pętli o 2. Zapis rozwiązania w pseudokodzie może być następujący:

```

n ← b div 2
p ← sito(n)
# maks – maksymalna liczba jedynek,
# x – liczba półpierwsza
jeżeli a < 6 to
  maks ← 2; x ← 4 #2*2, dwie jedynki
w przeciwnym przypadku
  maks ← 0; x ← 0
dla i z zakresu od 3 do n z krokiem 2 wykonuj
  jeżeli p[i]>0 oraz 2 * i ≥ a oraz p[i] + 1 > maks to
    maks ← p[i] + 1; x ← 2 * i
i ← 3
dopóki i * i ≤ b wykonuj
  jeżeli p[i]>0 to
    j ← i
    dopóki i * j ≤ b wykonuj
      jeżeli p[j]>0 oraz i*j ≥ a to
        jeżeli p[i] + p[j] > maks lub
          p[i] + p[j] = maks oraz i * j > x to
          maks ← p[i] + p[j]
          x ← i * j
        j ← j + 2
    i ← i + 2

```

Zadanie było sprawdzane automatycznie na platformie konkursowej. Testy obejmowały zarówno krótsze przedziały na mniejszych liczbach, jak długie obejmujące większe liczby, tak aby rozwiązania wolniejsze, np. sprawdzające oddzielnie pierwszość liczby otrzymały częściowe punkty.

## 5. Podsumowanie

Strona internetowa konkursu [7] zawiera wszystkie informacje dotyczące konkursu oraz archiwalne zasoby. Wśród nich znajdują się m.in. omówienia wybranych zadań oraz treści zadań wszystkich poprzednich konkursów [1]. Stanowi to doskonały zbiór zadań dla nauczycieli informatyki. Znajdują się tam zadania o różnym stopniu trudności zarówno do pracy z uczniami początkującymi jak i uczniami uzdolnionymi startującymi z sukcesem w konkursach i olimpiadach.

## Literatura

1. Bank zadań konkursu LOGIA, <https://logia.oeiizk.edu.pl/bankzadan>, ostatni dostęp 29.05.2026 roku
2. Borowiecka A. i inni, *Konkursy informatyczne LOGIA i miniLOGIA 2002/03-2007/08*, OEIiZK, Warszawa 2008
3. Borowiecka A. i inni, *Konkursy informatyczne LOGIA i POLLOGIA 2002/03-2009/10*, OEIiZK, Warszawa 2010
4. Borowiecka A. i inni, *Konkursy informatyczne miniLOGIA i Liga Zadaniowa 2002/03-2009/10*, OEIiZK, Warszawa 2010
5. Borowiecki M., Samulska A. *Mazowiecki Przedmiotowy Konkurs Informatyczny LOGIA – wczoraj i dziś*, Informatyka w Edukacji, Toruń 2025
6. Samulska A., *Przedmiotowy Konkurs Informatyczny LOGIA w roku szkolnym 2023/24*, kwartalnik W cyfrowej szkole 3/2024, OEIiZK, Warszawa 2024
7. Strona konkursu LOGIA, <https://logia.oeiizk.edu.pl>, ostatni dostęp 29.05.2026 roku