



# Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

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

### Treść zadania

Staś układa na stole sześciennie klocki tej samej wielkości. Na każdym widnieje jedna z wielkich liter alfabetu łacińskiego. Klocki układane są rzędami (wierszami). Każdy rząd rozpoczyna się przy lewej krawędzi stołu. Pierwszy rząd zawiera jeden klocek, drugi – dwa, trzeci – trzy, ..., itd. Ostatni rząd może być krótszy. Napisz jednoparametrową funkcję **kolit**, której parametrem jest niepuste słowo długości co najwyżej 1000, zawierające litery widniejące na kolejnych klockach. Wynikiem funkcji jest liczba tych kolumn w układance Stasia, w których wszystkie litery są identyczne.

Przykłady:

Python:

wynikiem `kolit('ABCDEFGH')` jest **1**,

wynikiem `kolit('ALAMAKRABY')` jest **2**.

Logo:

wynikiem `kolit "ABCDEFGH"` jest **1**,

wynikiem `kolit "ALAMAKRABY"` jest **2**.

W pierwszym przykładzie warunek zadania spełnia ostatnia kolumna, w drugim przykładzie druga i ostatnia.

### Omówienie rozwiązania

Warto rozpisać ułożenie liter w kolumnach, a tak naprawdę numery (pozycje) liter w słowie układających się w kolumny. Na przykładach z treści zadania ustawienie jest następujące:

A B C D E F G H	1 2 3 4 5 6 7 8
A L A M A K R A B Y	1 2 3 4 5 6 7 8 9 10

Rozpiszmy pozycje liter trochę szerzej i zauważmy pewne zależności związane z położeniem litery w danym wierszu i kolumnie.

	1	2	3	4	5	6
1	1					
2	2	3				
3	4	5	6			
4	7	8	9	10		
5	11	12	13	14	15	
6	16	17	18	19	20	21



## Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Będziemy musieli sprawdzić poszczególne litery w każdej kolumnie, a więc musimy wiedzieć, jaki jest numer pierwszej litery w danej kolumnie oraz jak wyznaczyć numer kolejnej litery w tej kolumnie. Pierwsze litery kolejnych kolumn mają odpowiednio numery: 1, 3, 6, 10, 15, 21, .... Łatwo można zauważyć, że numer litery rozpoczynającej kolejną kolumnę otrzymamy dodając do numeru pierwszej litery poprzedniej kolumny numer bieżącej (dla drugiej kolumny mamy  $3=1+2$ , dla trzeciej  $6=3+3$ , dla czwartej  $10=6+4$ , itd.). Natomiast numer kolejnej litery w obrębie danej kolumny uzyskamy dodając numer aktualnego wiersza.

Należy policzyć liczbę kolumn, w których wszystkie litery są takie same. Potrzebny więc będzie licznik kolumn oraz pomocnicze zmienne określające numer kolumny, numer wiersza, numer pierwszej litery w danej kolumnie, numer badanej litery w danej kolumnie. Rozwiązanie możemy zapisać w dwóch pętlach, zewnętrzna przebiega po kolumnach oraz wewnętrzna sprawdzająca litery aktualnej kolumny. Zapis algorytmu może być następujący:

1. licznik kolumn  $\leftarrow 0$
2. numer kolumny  $\leftarrow 1$
3. początkowy numer litery  $\leftarrow 1$
4. dopóki początkowy numer litery  $\leq$  długość słowa
  - numer wiersza  $\leftarrow$  numer kolumny
  - numer litery  $\leftarrow$  początkowy numer litery + numer wiersza
  - dopóki numer litery  $\leq$  długość słowa i
    - słowo[początkowy numer litery] = słowo[numer litery]
    - numer wiersza  $\leftarrow$  numer wiersza + 1
    - numer litery  $\leftarrow$  numer litery + numer wiersza
  - jeśli numer litery  $>$  długość słowa
    - licznik kolumn  $\leftarrow$  licznik kolumn + 1
  - numer kolumny  $\leftarrow$  numer kolumny + 1
  - początkowy numer litery  $\leftarrow$  początkowy numer litery + numer kolumny
5. wynik licznik kolumn

Zwróćmy uwagę, że długość ostatniego wiersza (rzędu) może być krótsza, ale nie ma to znaczenia, ponieważ sprawdzamy, czy numer litery mieści się w obrębie słowa. Jeśli porównywanie liter zakończy się w sytuacji, że numer litery będzie większy od długości słowa, oznacza to, że sprawdziliśmy wszystkie litery w danej kolumnie i wszystkie były takie same. Powiększamy wtedy licznik kolumn.

### Rozwiązanie w języku Python

```
1. def kolut(slowo):
2.     ilek = 0
3.     nrk = 1
4.     ipocz = 1
5.     while ipocz <= len(slowo):
6.         nrw = nrk
7.         i = ipocz + nrw
8.         while i <= len(slowo) and slowo[ipocz-1] == slowo[i-1]:
9.             nrw += 1
10.            i += nrw
11.            if i > len(slowo):
12.                ilek += 1
13.            nrk += 1
14.            ipocz += nrk
15.     return ilek
```



## Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

W Pythonie indeksowanie (numerowane) rozpoczyna się zawsze od zera, dlatego od numerów porównywanych liter zostało odjęte 1.

### Rozwiązanie w języku Logo

```
1. oto kolit :słowo
2.   niech "ilek 0
3.   niech "nrk 1
4.   niech "ipocz 1
5.   dopóki [ :ipocz <= długość :słowo ]
6.     [
7.       niech "nrw :nrk
8.       niech "i :ipocz + :nrw
9.       niech "dobrze "prawda
10.      dopóki [ i :dobrze :i <= długość :słowo ]
11.        [
12.          niech "dobrze (element :ipocz :słowo) = element :i :słowo
13.          zwiększ "nrw
14.          (zwiększ "i :nrw)
15.        ]
16.      jeśli :dobrze [ zwiększ "ilek ]
17.      zwiększ "nrk
18.      (zwiększ "ipocz :nrk)
19.    ]
20.   wynik :ilek
21. już
```

Warunki logiczne w Logo są zawsze wyliczane „do końca”, tzn. jeśli pierwszy element koniunkcji daje wartość fałsz, to i tak, mimo że znamy już wartość całego wyrażenia, wyliczany jest drugi element. W związku z tym warunek

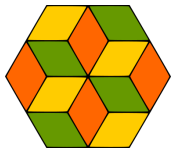
numer litery  $\leq$  długość słowa i `słowo[początkowy numer litery] = słowo[numer litery]`

dla numeru litery większego od długości słowa powodowałby błąd. Dlatego wprowadzona została pomocnicza zmienna logiczna `dobrze` i porównanie liter słowa odbywa się wewnątrz pętli, kiedy wiemy, że numer litery jest nie większy od długości słowa.

### Testy

Testowanie rozwiązania warto rozpocząć od krótkich słów, dla których jest łatwo ręcznie policzyć wynik.

Python	Logo	
<code>kolit('AAAAAA')</code>	<code>kolit "AAAAAA</code>	3
<code>kolit('AAAABA')</code>	<code>kolit "AAAABA</code>	2
<code>kolit('X')</code>	<code>kolit "X</code>	1
<code>kolit('OPQRSTUVWXYZ')</code>	<code>kolit "OPQRSTUVWXYZ</code>	0
<code>kolit('QQAQBVQCVGQDVHXQEVIXKQFVJXLZ')</code>	<code>kolit "QQAQBVQCVGQDVHXQEVIXKQFVJXLZ</code>	4



## Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

<pre>kolit('AABABCABCDABCDEABCDEFABCDEFGA BCDEFGHABCDEFGHIABCDEFGHIABCDEFGHI JKABCDEFGHIJKLABCDEFGHIJKLMABCDEFGHI JKLMNABCDEFGHIJKLMNOABCDEFGHIJKLMN OPABCDEFGHIJKLMNOPQABCDEFGHIJKLMN OPQRABCDEFGHIJKLMNOPQRSABCDEFGHIJK LMNOPQRSTABCDEFGHIJKLMNOPQRSTUABC DEFGHIJKLMNOPQRSTUABCDEFGHIJKLMNO PQRSTUUVWABCDEFGHIJKLMNOPQRSTUUVW ABCDEFGHIJKLMNOPQRSTUUVWXYABCDEFGH IJKLMNOPQRSTUUVWXYZABCDEFGHIJKLMNOP QRSTUUVWXYZAABCDEFGHIJKLMNOPQRSTU VWXYZAAABCDEFGHIJKLMNOPQRSTUUVWXYZA AAABCDEFGHIJKLMNOPQRSTUUVWXYZAAAAA BCDEFGHIJKLMNOPQRSTUUVWXYZAAAAAABC DEFGHIJKLMNOPQRSTUUVWXYZAAAAAABCD EFGHIJKLMNOPQRSTUUVWXYZAAAAAABCD EFGHIJKLMNOPQRSTUUVWXYZAAAAAABCD EFGHIJKLMNOPQRSTUUVWXYZAAAAAABCD DEFGHIJKLMNOPQRSTUUVWXYZAAAAAABCD BCDEFGHIJKLMNOPQRSTUUVWXYZAAAAAAB AAABCDEFGHIJKLMNOPQRSTUUVWXYZAAAAA AAAAAABCDEFGHIJKLMNOPQRSTUUVWXYZA AAAAAABCDABCDEFGHIJKLMNOPQRSTU WXYZAAAAAABCDABCDEFGHIJKLMNOP QRSTUUVWXYZAAAAAABCDABCDEFGHI JKLMNOPQRSTUUVWXYZAAAAAABCDABCDEF GHIJKLMNOPQRSTUUVWXYZAAAAAABCD AAAAAABCDABCDEFGHIJKLMNOPQRSTU VWXYZAAAAAABCDABCDEFGHIJKLMNOP YZAAAAAABCDABCDEFGHIJKLMN OPQRSTUUVWXYZAAAAAABCDABCDEF ZZZZZZZ')</pre>	<pre>kolit "AABABCABCDABCDEABCDEFABCDEFGA BCDEFGHABCDEFGHIABCDEFGHIABCDEFGHI JKABCDEFGHIJKLABCDEFGHIJKLMABCDEFGHI JKLMNABCDEFGHIJKLMNOABCDEFGHIJKLMN OPABCDEFGHIJKLMNOPQABCDEFGHIJKLMN OPQRABCDEFGHIJKLMNOPQRSABCDEFGHIJK LMNOPQRSTABCDEFGHIJKLMNOPQRSTUABC DEFGHIJKLMNOPQRSTUABCDEFGHIJKLMNO PQRSTUUVWABCDEFGHIJKLMNOPQRSTUUVW ABCDEFGHIJKLMNOPQRSTUUVWXYABCDEFGH IJKLMNOPQRSTUUVWXYZABCDEFGHIJKLMNOP QRSTUUVWXYZAABCDEFGHIJKLMNOPQRSTU VWXYZAAABCDEFGHIJKLMNOPQRSTUUVWXYZA AAABCDEFGHIJKLMNOPQRSTUUVWXYZAAAAA BCDEFGHIJKLMNOPQRSTUUVWXYZAAAAAABC DEFGHIJKLMNOPQRSTUUVWXYZAAAAAABCD EFGHIJKLMNOPQRSTUUVWXYZAAAAAABCD EFGHIJKLMNOPQRSTUUVWXYZAAAAAABCD DEFGHIJKLMNOPQRSTUUVWXYZAAAAAABCD BCDEFGHIJKLMNOPQRSTUUVWXYZAAAAAAB AAABCDEFGHIJKLMNOPQRSTUUVWXYZAAAAA AAAAAABCDEFGHIJKLMNOPQRSTUUVWXYZA AAAAAABCDABCDEFGHIJKLMNOPQRSTU WXYZAAAAAABCDABCDEFGHIJKLMNOP QRSTUUVWXYZAAAAAABCDABCDEFGHI JKLMNOPQRSTUUVWXYZAAAAAABCDABCDEF GHIJKLMNOPQRSTUUVWXYZAAAAAABCD AAAAAABCDABCDEFGHIJKLMNOPQRSTU VWXYZAAAAAABCDABCDEFGHIJKLMNOP YZAAAAAABCDABCDEFGHIJKLMN OPQRSTUUVWXYZAAAAAABCDABCDEF ZZZZZZZ"</pre>	34
---	---	----