

# Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

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

### Treść zadania

Logolandia ma tylko jedną owalną ulicę, przy której stoją domy podzielone na osiedla. Dwa domy stojące obok siebie są w jednym osiedlu, jeśli różnica ich numerów jest nie większa niż 3. Dom o największym numerze jest w tym samym osiedlu, co dom o numerze 1, 2 lub 3 (o ile istnieje). W Logolandii stoją co najmniej 2 domy, ale nie więcej niż 10000. Największy numer domu jest nie większy niż 30 000. Numery domów nie powtarzają się.

Napisz jednoparametrową funkcję **maksos**, której parametrem jest uporządkowana rosnąco lista numerów domów. Wynikiem jest liczba domów w osiedlu składającym się z największej liczby domów.

Przykłady:

Język	Wywołanie	Wynik	Uzasadnienie
Python	<code>maksos([1,4,7,13,14,15,20])</code>	4	osiedle 20-1-4-7
Logo	<code>maksos [1 4 7 13 14 15 20]</code>		
Python	<code>maksos([1,4,7,11,13,14,15,16,20])</code>	5	osiedle 11-13-14-15-16
Logo	<code>maksos [1 4 7 11 13 14 15 16 20]</code>		

### Omówienie rozwiązania

Rozwiązanie polega na przeglądaniu kolejnych par elementów oraz sprawdzaniu, czy domy z rozpatrywanej pary należą do tego samego osiedla. Dokładniej, zmiennej `licznik` przypisujemy wartość 1 – początkową wielkość pierwszego osiedla, a zmiennej `naj` – wielkość najdłuższego dotąd znalezionej osiedla. Jeżeli dwa sąsiednie domy są dostatecznie blisko – należą do tego samego osiedla, czyli ich numery nie różnią się więcej niż 3, zwiększamy licznik o 1, w przeciwnym przypadku mamy do czynienia z końcem osiedla. Wtedy sprawdzamy, czy nie znaleźliśmy większego osiedla od zapamiętanego, jeżeli tak, to uaktualniamy wartość zmiennej `naj` i ustawiamy `licznik` na 1. W ten sposób przeglądamy wszystkie pary sąsiednich domów.

Musimy jeszcze rozpatrzyć przypadek, gdy do jednego osiedla należą domy o najwyższych i najniższych numerach, czyli osiedle jest „zawijane”. W tym celu sprawdzamy, czy pierwszy dom należy do tego samego osiedla co ostatni. Jeśli tak, to rozpatrujemy to osiedle. Tak długo, dopóki kolejne pary domów są blisko siebie, zwiększamy wartość zmiennej `licznik`. Kończymy, gdy dojdziemy do końca listy lub gdy napotkamy dom z nowego osiedla.

Jeżeli ostatnio rozpatrywane osiedle jest największe, to musimy jeszcze uaktualnić wartość zmiennej `naj`.

Może się zdarzyć, że wszystkie domy należą do tego samego osiedla, wtedy wynikiem funkcji jest długość listy.



## Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

### Rozwiązanie w języku Python

```
1. def maksos(lista):
2.     licznik = 1
3.     naj = 0
4.
5.     # zliczanie osiedli bez zawijania
6.     for i in range(1, len(lista)):
7.         if lista[i]-lista[i-1] <= 3:
8.             licznik += 1
9.         else:
10.            if licznik > naj:
11.                naj = licznik
12.            licznik = 1
13.
14.    # zawijanie
15.    if lista[0] < 3:
16.        licznik += 1
17.        i = 1
18.        while i < len(lista) and lista[i] - lista[i-1] <= 3:
19.            licznik += 1
20.            i += 1
21.    # ostatnie osiedle największe
22.    if licznik > naj:
23.        naj = licznik
24.
25.    # wszystkie domy w jednym osiedlu
26.    if naj == 2 * len(lista):
27.        return len(lista)
28.
29.    return naj
```



# Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

## Rozwiązanie w języku Logo

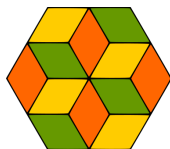
```
1. oto maksos :lista
2.   niech "licznik 1
3.   niech "naj 0
4.
5.   ; zliczanie osiedli bez zawijania
6.   powtórz długość bp :lista
7.   [
8.     jeżeli (element npw+1 :lista) - (element npw :lista) <= 3
9.       [zwiększ "licznik]
10.    [jeżeli :licznik > :naj [niech "naj :licznik]
11.    niech "licznik 1
12.  ]
13. ]
14.
15. ; zawijanie
16. jeżeli pierw :lista < 3
17. [ zwiększ "licznik
18.   niech "ii 1
19.   dopóki [ :ii < długość :lista]
20.   [
21.     jeżeli (element npw+1 :lista) - (element npw :lista) > 3
22.       [niech "ii długość :lista]
23.       [zwiększ "licznik
24.       zwiększ "ii]
25.   ]
26. ]
27.
28. ; ostatnie osiedle największe
29. jeżeli :licznik > :naj [niech "naj :licznik]
30.
31.
32. ; wszystkie domy w jednym osiedlu
33. jeżeli :naj = (2 * długość :lista)
34.   [wy długość :lista]
35.
36. wy :naj
37. już
```

## Testy

Rozwiązanie testujemy dla różnych przypadków. Najprostszy wydaje się test dla najdłuższego osiedla w środku listy. Sprawdzamy też przypadki, kiedy najdłuższe osiedle jest na końcu listy oraz zawijane. Wśród testów nie powinno zabraknąć przykładu, gdy wszystkie domy są w jednym osiedlu.

Dla języka Python

Wywołanie	Wynik
<code>maksos([4,7,10,13,16,19,22,25,28,34,37,40,43,46,49,52,55,59,60,63])</code>	9
<code>maksos([4,7,10,13,16,19,22,25,28,34,37,40,43,46,49,52,55,59,60,63,64,65,66,67,68,69,70,100])</code>	10
<code>maksos([10,20]+[i for i in range(100,1000,1)])</code>	900
<code>maksos([1,2,3,4,7,13,14,15,19,20])</code>	7



## Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

<code>maksos([1,2,3,4,7,10,13,14,15,19,20])</code>	11
<code>maksos([1,2])</code>	2
<code>maksos([1,20])</code>	2
<code>maksos([1,10,20])</code>	2
<code>maksos([i for i in range(1,101,1)]+[i for i in range(200,300,1)] +[i for i in range(400,500,1)])</code>	200
<code>maksos([i for i in range(1,101,1)])</code>	100

Dla języka Logo

Wywołanie	Wynik
<code>maksos [4 7 10 13 16 19 22 25 28 34 37 40 43 46 49 52 55 59 60 63]</code>	9
<code>maksos [4 7 10 13 16 19 22 25 28 34 37 40 43 46 49 52 55 59 60 63 64 65 66 67 68 69 70 100]</code>	10
<code>maksos zd [10 20] generuj 900 [:%+1] 100</code>	900
<code>maksos [1 2 3 4 7 13 14 15 19 20]</code>	7
<code>maksos [1 2 3 4 7 10 13 14 15 19 20]</code>	11
<code>maksos [1 2]</code>	2
<code>maksos [1 20]</code>	2
<code>maksos [1 10 20]</code>	2
<code>maksos (zd (generuj 100 [:%+1] 1) (generuj 100 [:%+1] 200) (generuj 100 [:%+1] 400))</code>	200
<code>maksos generuj 100 [:%+1] 1</code>	100