



# Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

## Zadanie Neony – LOGIA 18 (2017/18), etap 2

### Treść zadania

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 10000.

Przykłady:

Język	Wywołanie	Wynik	Uzasadnienie
Python	<code>neon([10, 4, 5, 7, 1, 4, 1])</code>	24	najwyższa ocena będzie uzyskana przy wyborze pierwszego i przedostatniego słupa, $24 = 10 + 4 + 5 \cdot 2$
Logo	<code>neon [10 4 5 7 1 4 1]</code>		
Python	<code>neon([1, 10, 1])</code>	13	$13 = 1 + 10 + 1 \cdot 2$ , też $13 = 10 + 1 + 1 \cdot 2$
Logo	<code>neon [1 10 1]</code>		

### Omówienie rozwiązania

Pierwszym nasuwającym się rozwiązaniem jest rozpatrzenie każdej pary słupów i wybranie tej, która uzyska najwyższą ocenę. Dokładniej, przeglądamy daną listę i dla każdego elementu, który jest dalej na liście niż rozpatrywany słup, liczymy ocenę. Najlepszą ocenę zapamiętujemy. Takie rozwiązanie wymaga dla listy długości  $n$ ,  $n \cdot (n-1 + n-2 + \dots + 1)$  porównań. Jest to rozwiązanie o złożoności kwadratowej. Można jednak zaproponować szybsze rozwiązanie.

Drugi algorytm zakłada przeglądanie listy element po elemencie, z tym że pamiętamy dwie wartości – **naj** – najlepszą dotychczasową ocenę i **pom** – najlepszą ocenę, która uwzględnia odległość do bieżącego słupa. W danym kroku porównujemy te wartości i jeśli druga z nich jest większa, to uaktualniamy wartość maksymalną. Ponadto ustalamy nową wartość zmiennej **pom** jako maksymalną z dwóch liczb: dotychczasowej wartości i oceny neonu, który rozpoczyna się w rozpatrywanym słupie.

Przeanalizujmy to na przykładzie:

Dane: lista 2, 3, 4, 10, 2, 1

Obrót pętli	0	1	2	3	4	5
naj	0	7	10	18	18	18
pom	2	4	6	10	12	14

Najpierw **naj** ma wartość 0 (mniejszą od najmniejszego możliwego wyniku), a **pom** wartość 2 – pierwszego elementu listy. Rozpatrujemy drugi element listy. Dodajemy do wartości zmiennej **pom** odległość między słupami, czyli 2. Zmiennej **naj** przypisujemy  $4 + 3 = 7$ , zmienna **pom** pozostaje bez zmiany (4 jest większe od 3, wysokości aktualnego słupa). Rozpatrujemy trzeci element listy: zmiennej **naj** przypisujemy 10, bo  $\max(7, 6 + 4) = 10$ , zmienna **pom** ma wartość 6 ( $\max(6, 4) = 6$ ). W następnym kroku zmienna **naj** otrzymuje wartość  $18 = \max(10, 8 + 10)$ , a **pom**  $10 = \max(10, 8)$ . W przedostatnim kroku **naj**  $= \max(18, 12 + 2)$ , **pom**  $= \max(12, 2)$ , w ostatnim kroku **naj**  $= \max(18, 14 + 1)$ .



## Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Wynikiem funkcji jest wartość **naj** po analizie ostatniego elementu listy, w omawianym przykładzie liczba 18.

### Rozwiązanie w języku Python

W omawianym rozwiązaniu rozpatrujemy listę począwszy od drugiego elementu, w języku Python zapisujemy to następująco: `lista[1:]`, czyli nazwa zmiennej oraz w nawiasie kwadratowym dwa indeksy oddzielone dwukropkiem: pierwszy wskazujący od jakiej wartości zaczynamy – 1 (listy numerujemy od 0), drugi wskazujący do jakiej wartości. Przy czym, jeśli elementy listy mają być brane do końca, to wartość indeksu można pominąć.

```
1. def neon(lista):
2.     naj = 0
3.     pom = lista[0]
4.     for x in lista[1:]:
5.         pom = pom + 2
6.         naj = max(naj, pom + x)
7.         pom = max(pom, x)
8.     return naj
```

Warto też zwrócić uwagę, że w rozwiązaniu skorzystano z wbudowanej funkcji `max`, której wynikiem jest maksymalna wartość liczb podanych jako parametr.

### Rozwiązanie w języku Logo

Ponieważ w pętli przeglądamy listę od drugiego elementu, odwołujemy się do elementu o indeksie `npw+1` – numer powtórzenia +1.

```
1. oto neon :lista
2.   niech "naj 0
3.   niech "pom pierw :lista
4.   powtórz (długość :lista) - 1
5.   [
6.     niech "x element npw+1 :lista
7.     (zwiększ "pom 2)
8.
9.     jeśli :pom + :x > :naj
10.    [niech "naj :pom + :x]
11.
12.    jeśli :x > :pom
13.    [niech "pom :x]
14.  ]
15.
16.  wy :naj
17.  już
```



# Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

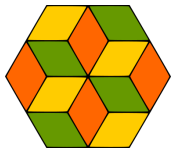
## Testy

Testowanie rozwiązania zaczynamy od przypadków, których wynik można wyznaczyć w pamięci. Pierwsze dwa testy obejmują bardzo proste przykłady, kolejne dwa trochę trudniejsze. Przykład trzeci i czwarty bierze te same wartości jako podstawę optymalnego rozwiązania, ale w innym położeniu na liście. W piątym przykładzie jest testowana poprawność dla przypadków, które niewiele się różnią od siebie, by sprawdzić zdefiniowane warunki. Kolejne przykłady są dla losowych danych z wybranych przedziałów, a ostatni test dla wartości maksymalnych.

Dla języka Python

Wywołanie	Wynik
<code>neon([10,20])</code>	32
<code>neon([10,2,3,5,20])</code>	38
<code>neon([2,200,4,3,2,94,1])</code>	302
<code>neon([2,2,5,200,4,3,2,94,1])</code>	302
<code>neon([1,2,3,200,4,3,2,1,1,1,1,1,1,1,1,2,3,90,4,3,2])</code>	320
<code>neon([9001,9000,8999,8998,8997,8996,8995,8994,8993,8992,8991,8990,8989,8988,8987,8986,8985,8984,8983,8982,8981,8980,8979,8978,8977,8976,8975,8974,8973,8972,8971,8970,8969,8968,8967,8966,8965,8964,8963,8962,8961,8960,8959,8958,8957,8956,8955,8954,8953,8952,8951,8950,8949,8948,8947,8946,8945,8944,8943,8942,8941,8940,8939,8938,8937,8936,8935,8934,8933,8932,8931,8930,8929,8928,8927,8926,8925,8924,8923,8922,8921,8920,8919,8918,8917,8916,8915,8914,8913,8912,8911,8910,8909,8908,8907,8906,8905,8904,8903,8902])</code>	18101
<code>neon([28,65,19,5,55,81,46,18,24,8,55,80,63,30,62,70,13,85,56,21])</code>	190
<code>neon([500,288,622,701,425,810,355,274,495,422,875,407,152,882,310,334,482,541,276,805,858,870,120,706,716,639,565,212,916,716,241,622,382,315,543,522,405,498,460,186,588,416,525,852,368,180,647,439,709,144])</code>	1828
<code>neon([9896,1183,6511,375,4949,9803,1836,3,6465,5925])</code>	19709
<code>neon([10000 for i in range(500)])</code>	20998

W ostatnim teście generowana jest lista długości 500, gdzie wartość każdego elementu wynosi 10000.



## Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Dla języka Logo

Wywołanie	Wynik
neon [10 20]	32
neon [10 2 3 5 20]	38
neon [2 200 4 3 2 94 1]	302
neon [2 2 5 200 4 3 2 94 1]	302
neon [1 2 3 200 4 3 2 1 1 1 1 1 1 1 1 2 3 90 4 3 2]	320
neon [9001 9000 8999 8998 8997 8996 8995 8994 8993 8992 8991 8990 8989 8988 8987 8986 8985 8984 8983 8982 8981 8980 8979 8978 8977 8976 8975 8974 8973 8972 8971 8970 8969 8968 8967 8966 8965 8964 8963 8962 8961 8960 8959 8958 8957 8956 8955 8954 8953 8952 8951 8950 8949 8948 8947 8946 8945 8944 8943 8942 8941 8940 8939 8938 8937 8936 8935 8934 8933 8932 8931 8930 8929 8928 8927 8926 8925 8924 8923 8922 8921 8920 8919 8918 8917 8916 8915 8914 8913 8912 8911 8910 8909 8908 8907 8906 8905 8904 8903 8902]	18101
neon [28 65 19 5 55 81 46 18 24 8 55 80 63 30 62 70 13 85 56 21]	190
neon [500 288 622 701 425 810 355 274 495 422 875 407 152 882 310 334 482 541 276 805 858 870 120 706 716 639 565 212 916 716 241 622 382 315 543 522 405 498 460 186 588 416 525 852 368 180 647 439 709 144]	1828
neon [9896 1183 6511 375 4949 9803 1836 3 6465 5925]	19709
neon (generuj 500 [10000] 10000)	20998

W ostatnim teście generowana jest lista długości 500, gdzie wartość każdego elementu wynosi 10000.