

# Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

## Zadanie Liczby Lucasa – LOGIA 20 (2019/20), etap 3

### Treść zadania

Dwie początkowe liczby Lucasa to **2** i **1**, a kolejne tworzone są jako suma dwóch poprzednich. Wobec tego trzecia liczba równa się **3**, czwarta **4**, piąta **7**, ... a dziesiąta **76**. Napisz funkcję **lucas(n)**, której wynikiem są dwie ostatnie cyfry n-tej liczby Lucasa, przy czym jeżeli liczba Lucasa jest jednocyfrowa lub przedostatnią cyfrą jest **0**, to wynikiem jest ostatnia cyfra. Parametr może przyjmować wartości od **1** do **10 000 000**. Postaraj się, by na wynik nie trzeba było czekać zbyt długo.

#### Przykłady:

Wynikiem **lucas(9)** jest **47**.

Wynikiem **lucas(35)** jest **43**, ponieważ trzydziesta piąta liczba Lucasa wynosi 12752043.

Wynikiem **lucas(45)** jest **7**, ponieważ czterdziesta piąta liczba Lucasa wynosi 1568397607.

### Omówienie rozwiązania

Najprostsze rozwiązanie to obliczanie kolejnej liczby Lucasa jako sumy dwóch poprzednich. Ponieważ dwie pierwsze liczby dane są z definicji, to obliczenia możemy zacząć od trzeciej liczby. Na końcu obliczamy resztę z dzielenia otrzymanego wyniku przez 100, otrzymując liczbę jedno lub dwucyfrową, ponieważ takie były wymagania zadania.

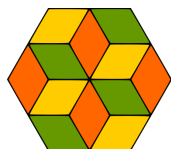
Testowanie takiego zadania staje się trudne dla dużych liczb, praktycznie powyżej 1 000 000 nie można się doczekać na wynik, a w treści zadania czytamy, że parametr może przyjmować wartości do 10 000 000.

Celem usprawnienia liczenia można już przy liczeniu sumy ograniczać się tylko do dwóch ostatnich cyfr, szybkość działania staje się zadawalająca, wynik dla maksymalnej wartości parametru pojawia się praktycznie natychmiast.

Jeśli ktoś spotkał się z ciągiem Fibonacciego, w którym podobnie jak w ciągu Lucasa kolejny element jest równy sumie dwóch poprzednich, to może przypuszczać, że każdy tego typu ciąg jest okresowy jeśli bierzemy pod uwagę jego ostatnie cyfry lub dwie ostatnie. Powstaje tylko pytanie, ile wynosi ten okres, czyli kiedy dwie ostatnie cyfry będą się powtarzać. Bardzo szybko można to sprawdzić w arkuszu kalkulacyjnym, wystarczy wpisać odpowiednią formułę i przeciągnąć obliczenia. Można też samemu napisać odpowiednią funkcję, która znajduje ten okres i wynikiem jej jest lista liczb, które powtarzają się jako dwie ostatnie cyfry ciągu Lucasa. Obliczając kolejne dwie ostatnie cyfry liczb Lucasa wstawiamy je do listy wyników. Kończymy, gdy dwa ostatnio wstawione elementy znajdują się w takiej samej kolejności na liście, jak dwa pierwsze.

Poniżej znajduje się lista dwóch ostatnich cyfr kolejnych liczb Lucasa. Widać, że od 61-wszego elementu wartości zaczynają się powtarzać.

[2, 1, 3, 4, 7, 11, 18, 29, 47, 76, 23, 99, 22, 21, 43, 64, 7, 71, 78, 49, 27, 76, 3, 79, 82, 61, 43, 4, 47, 51, 98, 49, 47, 96, 43, 39, 82, 21, 3, 24, 27, 51, 78, 29, 7, 36, 43, 79, 22, 1, 23, 24, 47, 71, 18, 89, 7, 96, 3, 99, 2, 1, 3, 4, 7, .....]



# Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Ostateczne obliczenie n-tej liczby Lucasa działa w czasie stałym, polega na wybraniu odpowiedniego elementu z listy danych.

## Rozwiązanie w języku Python

### Metoda 1

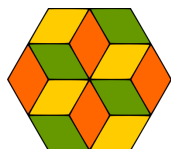
```
1. def lucas1(n):
2.     a = 2
3.     b = 1
4.     if n == 1:
5.         return a
6.     if n == 2:
7.         return b
8.     for i in range(n - 2):
9.         a, b = b, (a + b)
10.    return b % 100
```

### Metoda 2

```
1. def lucas2(n):
2.     a = 2
3.     b = 1
4.     if n == 1:
5.         return a
6.     if n == 2:
7.         return b
8.     for i in range(n - 2):
9.         a, b = b, (a + b) % 100
10.    return b
```

### Metoda 3

```
1. def okres():
2.     a = 2
3.     b = 1
4.     tab = [a, b]
5.     while True:
6.         a, b = b, (a + b) % 100
7.         if a in tab and b in tab and tab.index(a) + 1 == tab.index(b):
8.             return tab[:-1]
9.         tab = tab + [b]
10.    return tab
11. # print(okres())
12.
13. def lucas3(n):
14.     tab = okres()
15.    return tab[(n - 1) % 60]
```



## Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

### Testy

Chcąc sprawdzić prawidłowość wyników warto przeprowadzić następujące testy:

lucas(1)	2
lucas(2)	1
lucas(11)	23
lucas(44)	29
lucas(999)	3
lucas(100044)	79
lucas(200046)	11
lucas(687654)	71
lucas(897895)	18
lucas(8090386)	36
lucas(10000000)	24

Do wygodnego testowania warto wykorzystać pętlę, w której danymi będą kolejne liczby z listy.

Podczas wypisywania wyników, wartość danej (**x**) jest oddzielona znakiem tabulacji ("**\t**") od wyniku funkcji **lucas(x)**.

```
1. testy = [1, 2, 11, 44, 999, 100044, 200046, 687652, 897895, 8090386, 10000000]
2.
3. for x in testy:
4.     print(x, "\t", lucas(x))
```

Jeśli zainteresowały cię liczby Lucasa, obejrzyj film *Przedziwne liczby Lucasa. Rodzina złotych ciągów*.  
<https://www.wykop.pl/link/2655845/przedziwne-liczby-lucasa-rodzina-zlotych-ciagow>