



Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Zadanie Liczby lustrzane – LOGIA 19 SP (2018/19), etap 2

Treść zadania

Liczby lustrzane to takie dwie liczby, które są swoim lustrzanym odbiciem, np.: 125 i 521. Adam zauważył, że bardzo często ich suma jest liczbą palindromiczną, czyli taką, którą czyta się tak samo od lewej i prawej. Na przykład $125 + 521 = 646$. Jeżeli otrzymana suma nie jest liczbą palindromiczną, wtedy Adam dodaje do niej jej liczbę lustrzaną i ponownie sprawdza, czy nie jest liczbą palindromiczną. Wykonuje maksymalnie n prób, chyba że wcześniej otrzyma liczbę palindromiczną.

Zdefiniuj dwuparametrową funkcję **lustro**, której pierwszym parametrem jest dodatnia liczba całkowita nie większa niż 100 000, a drugim maksymalna liczba prób wykonywanych przez Adama (od 1 do 8). Wynikiem jest liczba palindromiczna powstała według reguły Adama lub -1, gdy nie da się jej uzyskać w podanej liczbie prób.

Przykłady:

wynikiem **lustro(125,3)** jest **646**

wynikiem **lustro(91,2)** jest **121**

wynikiem **lustro(91,1)** jest **-1**

Omówienie rozwiązania

Rozwiązanie zadania polega na powtórzeniu, co najwyżej n razy, obliczenia sumy dwóch liczb, pierwszą z nich jest liczba podana jako parametr, drugą jej lustrzane odbicie. Jeśli otrzymana suma jest liczbą palindromiczną, czyli taką, którą czyta się tak samo od lewej i prawej, to jej wartość jest wynikiem działania funkcji. W przeciwnym wypadku, czyli wtedy, gdy otrzymana suma nie jest liczbą palindromiczną należy sprawdzić, czy liczba wykonanych operacji dodawania jest równa drugiemu parametrowi n :

- jeśli tak, to wynikiem działania funkcji jest -1,
- jeśli nie, to powtarzamy wszystko od początku, zamiast parametru do obliczeń bierzemy ostatnio otrzymaną sumę.



Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Rozwiązanie w języku Python

Operacja dodawania musi być wykonana przynajmniej jeden raz, co wynika z treści zadania. Liczba powtórzeń tej operacji nie jest znana, dlatego w takim przypadku warto zastosować pętlę **while**, w której obliczamy sumę tak długo, dopóki nie otrzymamy w wyniku liczby palindromicznej lub liczba powtórzeń wyniesie **n**. Pamiętajmy, że **return** kończy działanie funkcji niezależnie od liczby powtórzeń.

```
1. def czy_palindrom(liczba):
2.     return str(liczba) == str(liczba)[::-1]
3.
4. def l2(liczba):
5.     return int(str(liczba)[::-1])
6.
7. def lustro(liczba,n):
8.     i=0
9.     nowa = liczba + l2(liczba)
10.    while i<n:
11.        if czy_palindrom(nowa):
12.            return nowa
13.        else:
14.            nowa = nowa + l2(nowa)
15.            i+=1
16.    return -1
```



Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Testy

Podczas testowania warto sprawdzić następujące przypadki:

- dla danego parametru wynikiem jest liczba palindromiczna w pierwszym dodawaniu
- brak wyniku dla określonej drugim parametrem liczby prób
- wynik w postaci liczby palindromicznej dla dokładnie n prób
- wynik w postaci liczby palindromicznej dla liczby prób mniejszej niż wartość drugiego parametru

Przykładowe wywołanie testu:

```
print(lustro(323, 1))
```

Wywołanie – Python	Wynik
lustro(323, 1)	646
lustro(42, 1)	66
lustro(298, 1)	-1
lustro(987, 3)	-1
lustro(98319, 5)	-1
lustro(98319, 6)	189444981
lustro(98319, 7)	189444981
lustro(4, 5)	8
lustro(5, 2)	11
lustro(66656, 2)	355553
lustro(9899, 8)	22399322