

# Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

## Zadanie Duże liczby – LOGIA 20 (2019/20), etap 3

### Treść zadania

Ola bawi się liczbami – zarówno małymi jak i takimi, które mają kilkaset cyfr. Zapisuje wybraną liczbę na kartce i szuka największej liczby całkowitej, której kwadrat nie przekracza zapisanej liczby. Szybko znalazła algorytm dający poprawny wynik. Teraz zastanawia się, jak szukać takich liczb, które podniesione do innej jednocyfrowej potęgi nie przekraczają wybranej liczby. Pomóż Oli uporać się z problemem. Napisz funkcję **liczba(a, n)**. Pierwszy parametr **a** jest liczbą złożoną maksymalnie z 1000 cyfr, drugi **n** to liczba z zakresu od 2 do 7 oznaczająca potęgę, do której podnosimy poszukiwaną wartość. Postaraj się, by na wynik nie trzeba było czekać zbyt długo.

#### Przykłady:

Wynikiem **liczba(1024, 2)** jest 32, ponieważ  $32^2 = 1024$

Wynikiem **liczba(899, 2)** jest 29, ponieważ  $29^2 = 841 < 899$ , a  $30^2 = 900 > 899$ .

Wynikiem **liczba(11000, 3)** jest 22, ponieważ  $22^3 = 10648 < 11000$ , a  $23^3 = 12167 > 11000$ .

Wynikiem **liczba(90972061672647417382949994702882855264900000, 2)** jest  
9537927535510396054172.

### Omówienie rozwiązania

#### Pierwsze podejście

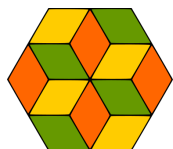
Można próbować wyznaczać wynik, korzystając ze wzoru:

$$a^{(1/n)}$$

Niestety, otrzymany wynik po zaokrągleniu w dół do najbliższej liczby całkowitej da poprawny wynik dla niewielu liczb. Wraz ze wzrostem liczby cyfr wynik jest obciążony coraz większym błędem. Na rezultat ma też wpływ wynik operacji  $1/n$  (zestawienie w tabeli poniżej).

a	n	$a^{(1/n)}$	$\text{int}(a^{(1/n)})$	poprawny wynik
1024	2	32.0	32	32
11000	3	22.23980090569315	22	22
117649	3	48.99999999999999	48	49
90972061672647417382949994702882855264900000	2	9.537927535510396e+21	9537927535510396338176	9537927535510396054172

Ponadto powyższe operacje są możliwe tylko dla liczb składających się z maksymalnie 308 cyfr – dla większych pojawia się błąd.



# Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

## Drugie podejście

Można zacząć od 0 i sprawdzać kolejne liczby. Taki algorytm zapiszemy w następujący sposób:

```
liczba = 0
dopóki liczban <= a
    liczba = liczba + 1
wynik liczba - 1
```

Niestety taka strategia zawodzi – dla liczb składających się z trzynastu cyfr na wynik trzeba czekać około sekundy. W tabeli poniżej znajdują się informacje dotyczące czasu działania algorytmu.

liczba	czas
$10^{12}$	0.6 s
$10^{16}$	53 s (ok. 1 minuta)
$10^{30}$	ponad 1 rok

## Trzecie podejście

Można wykorzystać wyszukiwanie binarne, gdzie w każdym kroku odrzucamy połowę kandydatów na poszukiwaną liczbę.

liczba	maksymalna liczba kroków prowadzących do uzyskania wyniku
$10^{10}$	34
$10^{100}$	333
$10^{1000}$	2322

Algorytm korzystający z wyszukiwania binarnego:

```
początek = 1
koniec = a
dopóki początek < koniec
    srodek = (początek + koniec) // 2
    jeśli srodekn <= a
        początek = srodek + 1
    w przeciwnym przypadku
        koniec = srodek
wynik początek - 1
```

Ta strategia prowadzi do sukcesu wystarczająco szybko. Dla osób dociekliwych polecamy sprawdzenie, jaki wpływ na czas działania programu ma operacja potęgowania dla dużych liczb ( $srodek^n$ ).



Funkcja realizująca opisany algorytm, zapisana w języku Python może wyglądać tak:

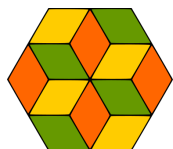
## Testy

Pierwsze dwie grupy testów przeznaczone są dla rozwiązania iteracyjnego, w którym sprawdzane są kolejne liczby począwszy od 0. Zastosowanie wzoru skutkuje błędnym wynikiem, przy czym rezultat różni się co najwyżej o 1 (wyjątek stanowi pierwszy test). Trzecia grupa testów przeznaczona jest dla rozwiązań, które liczą wynik ze wzoru, a następnie iteracyjnie sprawdzają kolejno liczby mniejsze (pierwszy test z grupy III) i większe (drugi test z grupy III). Czwarta i piąta grupa testów przeznaczona jest dla rozwiązań korzystających z metody połowienia. Tym razem zastosowanie strategii skutecznej w przypadku trzeciej grupy nie daje poprawnego wyniku w skończonym czasie (czwarta grupa) lub generuje błąd (piąta grupa).

[illegible]



- 4 -



## Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

4573544720121736348638787726807389838 4172108857236363790764951799147007809 7343121934911871912885179046366520362 7049286516584014763343896162912846127 2672346651206445680749732440468151101 1697421843792457096841164423814614175 2775054754738371757967919039436854903 6269244689867195928247321133716765621 1159587443057458278375999410626850882 1198589192560024747765306302143692221 7241641078063244513579355041303934589 9074669240304302683213037270525481363 6016266176961995704509915114144018569 000000, 4)	
---	--