



Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Zadanie Własny System Kodów Paskowych – miniLOGIA 16 (2017/18), etap 3

Treść zadania

Michał przygotowuje Własny System Kodów Paskowych (WSKP) i dlatego szuka liczb, których suma cyfr jest większa od **a** i mniejsza od **b**. Przegląda kolejne liczby począwszy od 1 do 100000 (włącznie). Napisz trójparametrową funkcję **jaka**. Pierwszym parametrem funkcji jest lewy koniec badanego przedziału (**a**), drugim prawy koniec (**b**). Wynikiem funkcji jest **n**-ta kolejna liczba (określona przez trzeci parametr), której suma cyfr jest większa niż **a** i mniejsza niż **b**. Jeśli taka liczba nie istnieje, to wynikiem jest -1. Przyjmij, że trzeci parametr **n** wynosi co najwyżej 100, natomiast $a < b < 40$.

Przykłady:

Python:

wynikiem **jaka(1, 10, 4)** jest 5,

wynikiem **jaka(1, 3, 100)** jest -1.

Logo:

wynikiem **jaka 1 10 4** jest 5,

wynikiem **jaka 1 3 100** jest -1.

W pierwszym przykładzie kolejne liczby spełniające warunek to 2, 3, 4, 5. W drugim przykładzie istnieje tylko 15 liczb spełniających warunek.

Omówienie rozwiązania

Rozwiązanie zadania polega na przeglądaniu kolejnych liczb począwszy od 1. Dla każdej liczby **x** liczymy sumę cyfr i jeśli mieści się w podanym przedziale $a < \text{suma}$ oraz $\text{suma} < b$, to zwiększamy licznik o 1. Jeśli licznik jest równy podanej liczbie **n**, to kończymy działanie funkcji. Wynikiem jest wartość zmiennej **x**. W przeciwnym przypadku rozpatrujemy kolejną liczbę. Postępowanie kontynuujemy tak długo, aż nie osiągniemy podanego progu 100000.

Warto zdefiniować pomocniczą funkcję `suma_cyfr`, której wynikiem dla danej liczby jest suma jej cyfr. Sumę liczymy, dodając kolejno wartości ostatniej cyfry.

Rozwiązanie w języku Python

```
1. def suma_cyfr(liczba):
2.     suma = 0
3.     while liczba != 0:
4.         suma += liczba % 10
5.         liczba = liczba // 10
6.     return suma
7.
8. def jaka(a, b, n):
9.     licznik = 0
10.    x = 1
11.    while x <= 100000:
12.        suma = suma_cyfr(x)
13.        if a < suma and suma < b:
14.            licznik += 1
15.            if licznik == n:
16.                return x
17.        x += 1
18.    return -1
```



Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Rozwiązanie w języku Logo

```
1. oto suma_cyfr :liczba
2.   niech "suma 0
3.   dopóki [:liczba > 0]
4.     [
5.       niech "suma :suma + reszta :liczba 10
6.       niech "liczba ilorazc :liczba 10
7.     ]
8.   wy :suma
9. już
10.
11. oto jaka :a :b :n
12.   niech "licznik 0
13.   niech "x 1
14.   dopóki [:x <= 100000]
15.     [ niech "suma suma_cyfr :x
16.       jeśli i :a < :suma :suma < :b
17.         [ zwiększ "licznik
18.           jeśli :licznik = :n [wy :x]
19.         ]
20.       zwiększ "x
21.     ]
22.   wy -1
23. już
```

Testy

Testowanie rozwiązania warto rozpocząć od przykładów, których wyniki można sprawdzić ręcznie np. `jaka(1, 3, 5)`. Dla tego przykładu liczby spełniające warunek tzn. takie, których suma cyfr wynosi 2, to 2, 11, 20, 101, 110,... Czyli wynikiem jest liczba 110. Potem wykonujemy kolejne testy zmieniając zakres sumy i numer kolejnej liczby. Nie można zapomnieć o przypadkach, w których nie istnieje n -ta liczba w podanym przedziale np. `jaka(2, 4, 60)` oraz o warunkach brzegowych.

Python	Logo	
<code>jaka(1, 3, 5)</code>	<code>jaka 1 3 5</code>	110
<code>jaka(21, 23, 20)</code>	<code>jaka 21 23 20</code>	985
<code>jaka(3, 17, 34)</code>	<code>jaka 3 17 34</code>	43
<code>jaka(23, 39, 58)</code>	<code>jaka 23 39 58</code>	2598
<code>jaka(2, 4, 60)</code>	<code>jaka 2 4 60</code>	-1
<code>jaka(1, 4, 90)</code>	<code>jaka 1 4 90</code>	-1
<code>jaka(1, 40, 100)</code>	<code>jaka 1 40 100</code>	103
<code>jaka(20, 40, 100)</code>	<code>jaka 20 40 100</code>	1589
<code>jaka(25, 30, 100)</code>	<code>jaka 25 30 100</code>	4896
<code>jaka(28, 32, 100)</code>	<code>jaka 28 32 100</code>	7967