

Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Zadanie Liczby – LOGIA 20 (2019/20), etap 2

Treść zadania

Basia szuka liczb, których suma cyfr jest równa iloczynowi tych cyfr np. 123. Napisz funkcję `ile(lista)`, której parametrem jest niepusta lista dodatnich liczb naturalnych, a wynikiem liczba znalezionych przez Basię liczb z podanej listy. Długość listy jest nie większa niż 1 000, a każda z liczb jest nie większa niż 1 000 000.

Przykłady:

Wynikiem `ile([7, 13, 1122, 111, 52, 52111])` jest **2**, ponieważ są dwie takie liczby 7 i 52111.

Wynikiem `ile([11, 1000, 123])` jest **1**, ponieważ jest jedna taka liczba 123.

Wynikiem `ile([321, 152, 2141, 4211])` jest **3**, ponieważ są trzy takie liczby 321, 2141, 4211.

Omówienie rozwiązania

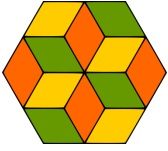
Ustawiamy wartość licznika na 0. Następnie przeglądamy kolejne liczby z listy i dla każdej z nich liczymy sumę cyfr i ich iloczyn. Gdy wyliczone wartości są równe, to zwiększamy licznik. Wynikiem jest wartość licznika.

Obliczanie sumy i iloczynu cyfr liczby można zrealizować przez wyodrębnianie cyfr od prawej strony, czyli najmniej znaczącej cyfry. Ostatnia cyfra liczby to wyniki reszty z dzielenia (%) liczby przez 10, a liczba bez ostatniej cyfry to wynik dzielenia całkowitego (//) liczby przez 10. Wobec tego sumę cyfr danej liczby można policzyć następująco:

1. Przypisz zmiennej `pom` wartość 0.
2. Dopóki `liczba` jest większa od 0 wykonuj:
 - Wartość zmiennej `pom` zwiększ o `liczba % 10`
 - Zmiennej `liczba` przypisz wartość `liczba // 10`
3. Wynikiem jest wartość zmiennej `pom`.

Analogicznie postępujemy przy obliczaniu iloczynu, z tym że wartością początkową zmiennej `pom` jest 1. Wartość zmiennej `pom` mnożymy w pętli przez wyodrębnioną cyfrę.

1. Przypisz zmiennej `pom` wartość 1.
2. Dopóki `liczba` jest większa od 0 wykonuj:
 - Pomnóż wartość zmiennej `pom` przez `liczba % 10`
 - Zmiennej `liczba` przypisz wartość `liczba // 10`
3. Wynikiem jest wartość zmiennej `pom`.



Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Rozwiązanie w języku Python

Rozwiązanie składa się z dwóch funkcji pomocniczych `suma(liczba)` i `iloczyn(liczba)` oraz głównej funkcji podanej w treści zadania `ile(lista)`.

```
1. def suma(liczba):
2.     pom = 0
3.     while liczba > 0:
4.         pom = pom + liczba % 10
5.         liczba = liczba // 10
6.     return pom
7.
8. def iloczyn(liczba):
9.     pom = 1
10.    while liczba > 0:
11.        pom = pom * (liczba % 10)
12.        liczba = liczba // 10
13.    return pom
14.
15. def ile(lista):
16.     licznik = 0
17.     for x in lista:
18.         if suma(x) == iloczyn(x):
19.             licznik += 1
20.     return licznik
```

Można też jednocześnie liczyć sumę i iloczyn. Wtedy rozwiązanie przedstawia się następująco:

```
1. def czy_rowne(liczba):
2.     suma = 0
3.     iloczyn = 1
4.     while liczba > 0:
5.         x = liczba % 10
6.         suma += x
7.         iloczyn *= x
8.         liczba = liczba // 10
9.     return suma == iloczyn
10.
11. def ile(lista):
12.     licznik = 0
13.     for x in lista:
14.         if czy_rowne(x):
15.             licznik += 1
16.     return licznik
```

Testy

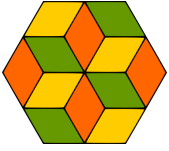
Wywołujemy funkcję `ile` dla różnych testów. W testach powinniśmy uwzględnić sytuację, gdy na liście znajdują się liczby szukane przez Basię oraz gdy nie ma takich liczb.

```
ile([512, 4121, 1234, 1124, 36, 45, 123, 4121])
```

4

```
ile([237, 1412, 2141, 1142, 213, 456, 1421, 7890, 123, 2114, 22, 4121, 312, 9, 4112, 7, 1214, 8, 6790, 231, 1241, 2411, 321, 1124, 132])
```

21



Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

```
ile([9385, 111126, 21122, 446, 31311, 4823, 7907, 3124, 31131, 25111, 51112, 111162, 21151, 6857, 31113, 33111, 51121, 21115, 5689, 51211, 52111, 8347, 15211, 3668, 3906])
```

15

```
ile([1, 2, 3, 4, 5, 6, 7, 8, 9, 22, 123, 132, 213, 231, 312, 321, 1124, 1142, 1214, 1241, 1412, 1421, 2114, 2141, 2411, 4112, 4121, 4211, 11125, 11133, 11152, 11215, 11222, 11251, 11313, 11331, 11512, 11521, 12115, 12122, 12151, 12212, 12221, 12511, 13113, 13131, 13311, 15112, 15121, 15211, 21115, 21122, 21151, 21212, 21221, 21511, 22112, 22121, 2456, 22211, 25111, 31113, 31131, 31311, 33111, 51112, 51121, 51211, 52111, 111126, 111162, 111216, 111261, 111612, 111621, 112116, 135, 112161, 112611, 116112, 116121, 116211, 121116, 121161, 121611, 126111, 161112, 161121, 161211, 162111, 211116, 211161, 211611, 216111, 261111, 611112, 611121, 611211, 612111, 23578, 621111])
```

98

oraz

```
ile([i for i in range(10**4 - 1000, 10**4)])
```

0

```
ile([i for i in range(10**4 - 1000, 10**4)] + [1])
```

1

Do generowania dwóch ostatnich testów zostały wykorzystane listy składane. Zwiększa to czytelność, a przede wszystkim upraszcza zapis.