

Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Zadanie Trzy mnożniki – LOGIA 21 (2020/21), etap 1

Treść zadania

Mat Multip poszukuje największego iloczynu trzech kolejnych cyfr danej liczby **n**. Napisz program, który pobierze z wejścia liczbę **n**, znajdzie szukaną wartość i wypisze na wyjściu wynik.

Wejście:

Liczba całkowita dodatnia **n** nie zawierająca zer, $111 \leq n \leq 10^{20}-1$.

Wyjście:

Liczba całkowita – maksymalna wartość iloczynu trzech kolejnych cyfr liczby **n**.

	Przykład 1	Przykład 2	Przykład 3
Wejście	7451298	8812266123	54321
Wyjście	144	72	60

Omówienie rozwiązania

Zadanie polega na znalezieniu wartości ekstremalnej (tu maksimum) w nieposortowanym ciągu liczb. Algorytm realizujący to zadanie pobiera kolejno liczby z ciągu i w kolejnych iteracjach porównuje je z aktualną wartością ekstremalną. Przed rozpoczęciem iteracji ustala się wartość ekstremalną w zmiennej **m** równą wartości wybranego elementu ciągu (np. pierwszy) lub wartości, o której wiemy, że dowolny element z ciągu jest ekstremalny w porównaniu z **m**. Przykładowo szukając wartości największej wśród ciągu liczb dodatnich za wartość początkową można ustawić $m = 0$.

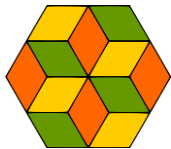
W pseudokodzie szukanie wartości maksymalnej w **k** elementowej tablicy **a** zawierającej liczby dodatnie mogłoby wyglądać tak:

```
1.  m ← 0
2.  i ← 0
3.  dopóki i < k:
4.    jeśli a[i] > m:
5.      m ← a[i]
6.    i ← i + 1
7.  wynik m
```

W pierwszych dwóch liniach ustawiamy najmniejszą możliwą wartość maksymalną **m** oraz zmienną **i** numerującą kolejno przeglądane elementy tablicy **a**.

W zadaniu nie mamy podanego ciągu **a**, w którym szukamy wartości maksymalnej, lecz znamy sposób jego konstruowania. Bierzemy kolejne trzy cyfry z podanej liczby **n** i mnożymy je aby otrzymać element ciągu, w którym szukamy ekstremum. Zapisanie wszystkich wyliczonych wartości ciągu przez utworzenie tablicy **a** nie jest potrzebne. Wystarczy, że będziemy z liczby **n** „wyluskiwać” po jednej cyfrze i zapisywać w zmiennych **a**, **b**, **r** tylko trzy ostatnie cyfry w danym momencie.

```
1.  m ← 0
2.  a ← 0
3.  b ← 0
```



Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

```
4.   dopóki n > 0:
5.       r ← n mod 10
6.       n ← n div 10
7.       jeśli a * b * r > m:
8.           m ← a * b * r
9.       a ← b
10.      b ← r
11.  wynik m
```

Linie o numerach 7 – 8 można zastąpić przypisaniem:

```
m ← max(m, a * b * r)
```

gdzie funkcja **max** zwraca wartość większą z dwóch podanych argumentów.

Rozwiązanie w języku Python

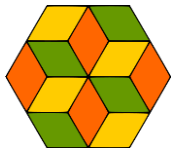
Wykorzystując właściwość Pythona pozwalającą na przypisanie kilku zmiennym wartości oraz wbudowaną funkcję **divmod**, która dla podanych dwóch argumentów zwraca parę liczb będącą wynikiem działań **div** oraz **mod** mamy następujące rozwiązanie:

```
1. def mat_mull(n):
2.     a, b, m = 0, 0, 0
3.     while n > 0:
4.         n, r = divmod(n, 10)
5.         a, b, m = b, r, max(m, a * b * r)
6.     return m
7.
8. z = int(input())
9. print(mat_mull(z))
```

Interesująca może być analiza rozwiązania w wersji rekurencyjnej:

```
1. def mat_mull(n, a = 0, b = 0, m = 0):
2.     if n==0:
3.         return m
4.     else:
5.         n, r = divmod(n, 10)
6.         return mat_mull(n, b, r, max(m, a * b * r))
7.
8. z = int(input())
9. print(mat_mull(z))
```

Zachowano nazwy zmiennych z wersji iteracyjnej.



Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Testy

Testy zostały tak dobrane, by sprawdzić wybrane wartości graniczne w danych wejściowych, m. in. najmniejszą i największą długość liczby ze specyfikacji, umieszczenie największej wartości na początku lub na końcu ciągu.

Grupa testów	Wejście	Wynik
I	123	6
	987	504
II	981299	162
	123999321	729
III	999998	729
	8888889	576
IV	9934512	243
	11919119	81
V	99123456789123456789	504
	11111111111111111112	2