

Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Zadanie Zakodowane liczby – LOGIA 23 (2022/23), etap 3

Treść zadania

Ola używa do kodowania liczb małych liter alfabetu łacińskiego. Litera **a** oznacza wartość 0, **b** – 1, ... **z** – 25. Pozycja litery w kodzie jest związana z wagą. Pierwsza pozycja ma wagę 1, druga $1 \cdot 2 = 2$, trzecia $1 \cdot 2 \cdot 3 = 6$, czwarta $1 \cdot 2 \cdot 3 \cdot 4 = 24$, itd.

Obliczając wartość liczby na podstawie jej kodu sumujemy iloczyny wartości liczbowej litery i jej wagi. Na przykład **acb**, to $0 \cdot 1 + 2 \cdot 1 \cdot 2 + 1 \cdot 1 \cdot 2 \cdot 3 = 0 + 4 + 6 = 10$, gdyż **a** – 0, **c** – 2, a **b** – 1 oraz **a** jest w kodzie na pozycji pierwszej, **c** na drugiej, a **b** na trzeciej.

Litera **a** może wystąpić na dowolnej pozycji oprócz ostatniej, a każda z liter od **b** do **z** może wystąpić w kodzie jedynie od pozycji odpowiadającej jej wartości liczbowej. Na przykład **z** może być jedynie na pozycji 25, litera **c** od pozycji 2 do 25. Niepoprawnymi kodami są: **cb** – litera **c** może wystąpić jedynie od pozycji 2, **ba** – litera **a** nie może wystąpić na końcu.

Poniższa tabela przedstawia zapis kolejnych liczb w opisanym systemie.

Liczba dziesiętna	Kod liczby	Wyjaśnienie
1	b	$1 \cdot 1 = 1$
2	ab	$0 \cdot 1 + 1 \cdot 2 = 2$
3	bb	$1 \cdot 1 + 1 \cdot 2 = 3$
4	ac	$0 \cdot 1 + 2 \cdot 2 = 4$
5	bc	$1 \cdot 1 + 2 \cdot 2 = 5$
6	aab	$0 \cdot 1 + 0 \cdot 2 + 1 \cdot 6 = 6$
7	bab	$1 \cdot 1 + 0 \cdot 2 + 1 \cdot 6 = 7$
8	abb	$0 \cdot 1 + 1 \cdot 2 + 1 \cdot 6 = 8$
9	bbb	$1 \cdot 1 + 1 \cdot 2 + 1 \cdot 6 = 9$
10	acb	$0 \cdot 1 + 2 \cdot 2 + 1 \cdot 6 = 10$
...		
119	bcde	$1 \cdot 1 + 2 \cdot 2 + 3 \cdot 6 + 4 \cdot 24 = 119$

Pomóż Oli i napisz program, który wczyta liczbę i wypisze jej kod opisany w treści zadania.

Wejście

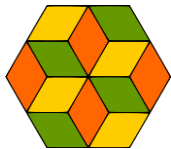
Liczba całkowita z zakresu od 1 do 403291461126605635583999999.

Wyjście

Kod liczby z wejścia opisany w treści zadania.

Przykłady:

Wejście	3	10	119
Wyjście	bb	acb	bcde



Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Omówienie rozwiązania

Problem w zadaniu sprowadza się do znalezienia zapisu liczb w systemie silniowym z użyciem liter reprezentujących cyfry.

System silniowy to system pozycyjny w którym mnożniki pozycji są wartościami silni. Na n -tej pozycji korzystamy z cyfr z przedziału od 0 do n .

Pozycja w liczbie	n	...	4	3	2	1
cyfry	0- n		0-4	0-3	0-2	0-1
mnożnik	$n!$		$4!$	$3!$	$2!$	$1!$

Liczba $321_1 = 3 * 3! + 2 * 2! + 1 * 1! = 18 + 4 + 1 = 23$

Każda liczba naturalna ma unikatową reprezentację w systemie silniowym.

Zauważmy, że różnica względem klasycznego systemu silniowego (oprócz reprezentacji cyfr przez litery) dotyczy kolejności zapisu liter – najmniej znacząca jest po lewej stronie, a najbardziej znacząca po prawej.

Pozycja w kodzie	1	2	3	4	...	25
litery	a-b	a-c	a-d	a-e		a-z
mnożnik	$1!$	$2!$	$3!$	$4!$		$25!$

Kod bcd odpowiada liczbie $23 = 1 * 1! + 2 * 2! + 3 * 3!$

Do utworzenia kodu liczby można zastosować jedno z dwóch podejść: metodę zachłanną lub wyznaczając reszty z dzielenia.

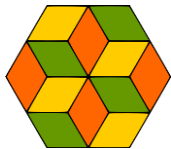
Metoda zachłanna polega na znalezieniu największej silni mieszczącej się w danej liczbie. Wyznamy w ten sposób długość kodu liczby.

```
dlugosc ← 1
silnia ← 1
dopóki liczba div silnia > dlugosc
    dlugosc ← dlugosc + 1
    silnia ← silnia * dlugosc
```

Kolejny krok to zastosowanie metody zachłannej do wyznaczenia poszczególnych cyfr.

```
wynik ← ""
dopóki dlugosc >= 1 wykonuj
    cyfra ← liczba div silnia
    znak ← odpowiednik(cyfra)
    wynik ← znak + wynik
    liczba ← liczba - cyfra * silnia
    silnia ← silnia div dlugosc
    dlugosc ← dlugosc - 1
```

W rozwiązaniu oprócz cyfr będziemy operować na odpowiadającym im literom.



Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Rozwiązanie w języku Python

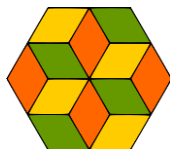
```
1 def kod(liczba):
2     dlugosc = 1
3     silnia = 1
4     while liczba // silnia > dlugosc:
5         dlugosc = dlugosc + 1
6         silnia = silnia * dlugosc
7     wynik = ''
8     while dlugosc >= 1:
9         cyfra = liczba // silnia
10        wynik = chr(cyfra + 97) + wynik
11        liczba = liczba - cyfra * silnia
12        silnia = silnia // dlugosc
13        dlugosc = dlugosc - 1
14    return wynik
15
16 liczba = int(input())
17 print(kod(liczba))
```

Konwertując liczbę na zapis w systemie silniowym można też posłużyć się resztami z dzielenia. Tym razem kolejność liter pozwala na ich natychmiastowe wypisanie.

```
i ← 2
dopóki liczba > 0 wykonuj
    cyfra ← liczba mod i
    wypisz odpowiednik(cyfra)
    liczba ← liczba div i
    i ← i + 1
```

Rozwiązanie w języku Python

```
1 def kod(liczba):
2     i = 2
3     while liczba > 0:
4         cyfra = liczba % i
5         print(chr(cyfra + 97), end = '')
6         liczba = liczba // i
7         i = i + 1
8
9 liczba = int(input())
10 kod(liczba)
```



Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Testy

Grupa testów	Silnia	Test	Wynik	Uwagi
I	$4! - 1$	23	bcd	liczby do 1000 bez silni
	$5! + 1$	121	baaab	
	losowa	535	babce	
II	$6!$	720	aaaaab	wartości silni lub jej wielokrotności
	$8!$	40320	aaaaaaab	
	$2 \cdot 9!$	725760	aaaaaaaac	
III	losowa	123456789	bbdcechbabd	losowe liczby z zakresu od 10^8 do 10^9
	losowa	987654321	bbbdabdgbiac	
	losowa	1000000000	accbfeegfabc	
IV	$4! + 2 \cdot 7! + 16!$	20922789898104	aaabaacaaaaaaaab	liczby z pełnego zakresu wartość - suma trzech silni
	$1! + 4 \cdot 4! + 17!$	355687428096097	baaeaaaaaaaaaaaaab	
	$5! + 5 \cdot 6! + 25!$	1551121004330985984003720	aaaabfaaaaaaaaaaaaaaaaab	
V	$1! + 2 \cdot 3! + 5 \cdot 5! + 2 \cdot 7! + 9!$	373573	bacafacab	liczby z pełnego zakresu
	$26! - 1$	403291461126605635583999999	bcdefghijklmnopqrstuvwxyz	
	$0 \cdot 1! + 1 \cdot 2! + \dots + 24 \cdot 25!$	387132773011805081755059686	abcdefghijklmnopqrstuvwxyz	