

# Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

## Zadanie Naszyjnik – LOGIA 21 (2020/21), etap 3

### Treść zadania

Marcin układa naszyjniki z kolorowych koralików tak, aby były symetryczne. Kolory koralików od lewej i prawej strony powinny być takie same. Każdy naszyjnik składa się z co najwyżej 10 koralików. Marcin koduje układ koralików liczbą całkowitą dodatnią złożoną z cyfr od 1 do 9 (bez 0). Kolejne cyfry oznaczają następujące kolory: 1 – czerwony, 2 – niebieski, 3 – zielony, 4 – niebieski, 5 – biały, 6 – zielony, 7 – czerwony, 8 – zielony, 9 – niebieski. Pomóż Marcinowi układać naszyjniki i napisz program, który wczyta liczbę i wypisze tę samą liczbę, gdy jest ona poprawnym kodem naszyjnika, a w przypadku niepoprawnego kodu znajdzie najmniejszą liczbę większą będącą poprawnym kodem naszyjnika.

#### Wejście:

Liczba całkowita z zakresu od 1 do 9 999 999 999 nie zawierająca zer.

#### Wyjście:

Najmniejsza liczba większa lub równa od liczby na wejściu poprawnie kodująca naszyjnik.

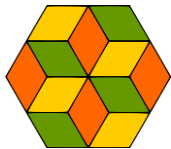
	<i>Przykład 1</i>	<i>Przykład 2</i>	<i>Przykład 3</i>
Wejście	<b>237</b>	<b>1241</b>	<b>7719668571</b>
Wyjście	<b>239</b>	<b>1241</b>	<b>7719669111</b>

### Omówienie rozwiązania

Zadanie można rozwiązać w następujący sposób: po wczytaniu danej, należy zamienić ją na napis przypisując poszczególnym cyfrom pierwszą literę koloru, a następnie sprawdzić, czy napis jest palindromem. Jeśli tak, to należy wypisać wczytaną liczbę, a w przeciwnym przypadku zwiększyć liczbę o 1 i ponownie zamienić ją na napis. Trzeba jednak uważać, czy zwiększając o 1 nie otrzymujemy liczby zawierającej 0. Wtedy należy pierwszą cyfrę 0 od lewej oraz wszystkie cyfry za nią zamienić na 1. Postępowanie należy kontynuować tak długo, aż nie osiągniemy palindromu. Przy czym zmianę liczby na napis dokonujemy zgodnie z poniższą tabelą:

Cyfra	Znak
1	c
2	n
3	z
4	n
5	b
6	z
7	c
8	z
9	n

Sprawdzenie, czy napis jest palindromem najprościej można wykonać odwracając napis i porównując pierwotny napis z odwróconym.



# Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

## Rozwiązanie w języku Python

Przy implementacji warto zdefiniować funkcje pomocnicze: `zamien(dana)`, której argumentem jest napis złożony z cyfr, a wynikiem napis złożony z liter oznaczających kolory oraz `znajdz(dana)`, której argumentem jest również napis złożony z cyfr, a wynikiem szukany kod naszyjnika. Warto zwrócić uwagę, że w języku Python wczytujemy dane jako napis, aby uzyskać liczbę całkowitą trzeba zastosować funkcję `int()`. Zapis `x != x[::-1]` porównuje napis `x` z napisem odwróconym `x[::-1]`.

```
1. def zamien(dana):
2.     pom = ""
3.     for i in dana:
4.         if i in "17":
5.             pom += "c"
6.         if i in "249":
7.             pom += "n"
8.         if i in "368":
9.             pom += "z"
10.        if i in "5":
11.            pom += "b"
12.    return pom
13.
14. def znajdz(dana):
15.     x = zamien(dana)
16.     while x != x[::-1]:
17.         dana = str(int(dana) + 1)
18.         for i in range(len(dana)):
19.             # gdy 0, zamieniamy na 1 daną cyfrę i wszystkie po niej
20.             if dana[i] == "0":
21.                 dana = dana[:i] + (len(dana) - i) * "1"
22.                 break
23.         x = zamien(dana)
24.     return dana
25.
26. print(znajdz(input()))
```

## Testy

Przy testowaniu należy uwzględnić przypadki, w których już na wejściu mamy poprawny naszyjnik oraz takie, w których będziemy szukać kolejnego kodu. Należy też zadbać o różne sekwencje cyfr. Wywołujemy program dla następujących testów.

Wejście	Wyjście
67371	67373
7824486	7824487
2	2
671	673
9898	9922
11999	12121
999	999
564519774	564525235
1224369942	1224369947
4336558865	4336558869
7328756152	7328773231
9751162391	9751171512