Self Numbers 성공 다국어

☆ 영어

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
1 초	256 MB	156306	76946	60079	48.497%

문제

In 1949 the Indian mathematician D.R. Kaprekar discovered a class of numbers called self-numbers. For any positive integer n, define d(n) to be n plus the sum of the digits of n. (The d stands for digitadition, a term coined by Kaprekar.) For example, d(75) = 75 + 7 + 5 = 87. Given any positive integer n as a starting point, you can construct the infinite increasing sequence of integers n, d(n), d(d(n)), d(d(d(n))), ... For example, if you start with 33, the next number is 33 + 3 + 3 = 39, the next is 39 + 3 + 9 = 51, the next is 51 + 5 + 1 = 57, and so you generate the sequence

33, 39, 51, 57, 69, 84, 96, 111, 114, 120, 123, 129, 141, ...

The number n is called a generator of d(n). In the sequence above, 33 is a generator of 39, 39 is a generator of 51, 51 is a generator of 57, and so on. Some numbers have more than one generator: for example, 101 has two generators, 91 and 100. A number with no generators is a self-number. There are thirteen self-numbers less than 100: 1, 3, 5, 7, 9, 20, 31, 42, 53, 64, 75, 86, and 97.

Write a program to output all positive self-numbers less than 10000 in increasing order, one per line.

Code

```
constrs = set()
for i in range(10000):
    constr = i + int(i / 1000) + (int(i / 100) % 10) + (int(i / 10) % 10) + (i % 10)
    constrs.add(constr)
    i += 1
for i in range(1, 10001):
    if i not in constrs:
        print(i)
```

The purpose of the problem is to write a code which prints all the self numbers less than 10000. Self numbers are numbers n that don't have a generator defined as d(n) = n + tens digit of n + ones digit of n. The challenging part of the ocde was to find an expression which can distinguish self numbers from 10000 integers and to implement it in a efficient way.