



链滴

Reciprocal cycles

作者: [taujong](#)

原文链接: <https://ld246.com/article/1577802418746>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



Project Euler 第 26 题

<!-- more -->

Question

A unit fraction contains 1 in the numerator. The decimal representation of the unit fractions with denominators 2 to 10 are given:

$$1/2 = 0.5$$

$$1/3 = 0.(3)$$

$$1/4 = 0.25$$

$$1/5 = 0.2$$

$$1/6 = 0.1(6)$$

$$1/7 = 0.(142857)$$

$$1/8 = 0.125$$

$$1/9 = 0.(1)$$

$$1/10 = 0.1$$

Where 0.1(6) means 0.166666..., and has a 1-digit recurring cycle. It can be seen that 1/7 has a 6-digit recurring cycle.

Find the value of $d < 1000$ for which $1/d$ contains the longest recurring cycle in its decimal fraction part.

Analysis

1. 如果被除数是 2 的整数次幂或者 5 的整数次幂，那么结果是有限小数，可以排除出去。
2. 按照除法的流程，如果除数小于被除数，那就乘 10，直到除数大于被除数，之后取余作为新的除数。
3. 在被除数不变的情况下，如果除数在若干次运算后再次出现，就说明循环周期已经出现，循环小数长度就是两次相同除数出现之间循环的次数。

Program

```
cycle = -1
num_list = list(range(2, 1000))
result = 0

def get_recurring_cycle(divisor, dividend):
    divisors = []
    while divisor not in divisors:
        divisors.append(divisor)
        divisor *= 10
        divisor %= dividend

    return len(divisors) - divisors.index(divisor)

tmp = 2
while tmp < 1000:
    num_list.remove(tmp)
    tmp *= 2

tmp = 5
while tmp < 1000:
    num_list.remove(tmp)
    tmp *= 5

for num in num_list:
    tmp = get_recurring_cycle(1, num)
    if tmp > cycle:
        cycle = tmp
        result = num

print(result)
```