

# Euler problem 35

于船长

书山有路勤为径，学海无涯苦作舟

# 本期内容

一. 题目讲解

二. 代码演示

## 一. 题目讲解

# 一. 题目讲解

Circular primes

The number, 197, is called a circular prime because all rotations of the digits: 197, 971, and 719, are themselves prime.

There are thirteen such primes below 100: 2, 3, 5, 7, 11, 13, 17, 31, 37, 71, 73, 79, and 97.

How many circular primes are there below one million?

---

圆周素数

197被称为圆周素数，因为将它逐位轮转所得到的数：197, 971, 和 719 都是素数。

小于100的圆周素数有十三个： 2, 3, 5, 7, 11, 13, 17, 31, 37, 71, 73, 79, 和 97。

小于一百万的圆周素数有多少个？

# 一. 题目讲解

- 1、如何完成一个正整数  $N$  的循环右移
- 2、如何用最少的计算次数获得  $N$  可以循环右移多少次
- 3、其实第2个问题，等价于求  $N$  的10进制表示的数字位数

# 一. 题目讲解

按照如下公式操作，既完成一次 N 的循环右移：

$$N \rightarrow \left\lfloor \frac{N}{10} \right\rfloor + (N \% 10) * dh$$

其中，dh 是数字 N 十进制表示中的最高位位权

例如：

N = 123, dh=100

N = 9853, dh=1000

## 二. 代码演示

## 二. 代码演示

```
#include <stdio.h>
#include <math.h>
#define MAX_N 1000000

int prime[MAX_N + 5] = {0};
int is_prime[MAX_N + 5] = {0, 1};
void init_prime() {
    for (int i = 2; i <= MAX_N; i++) {
        if (!is_prime[i]) prime[++prime[0]] = i;
        for (int j = 1; j <= prime[0]; j++) {
            if (i * prime[j] > MAX_N) break;
            is_prime[prime[j] * i] = 1;
            if (i % prime[j] == 0) break;
        }
    }
    return ;
}

int is_val(int n) {
    int digit = floor(log10(n)) + 1, tmp = n;
    for (int i = 0; i < digit; i++) {
        tmp = tmp % 10 * (int)pow(10, digit - 1) + tmp / 10;
        if (is_prime[tmp]) return 0;
    }
    return 1;
}
```

```
int main() {
    init_prime();
    int sum = 0;
    for (int i = 1; i <= prime[0]; i++) {
        sum += is_val(prime[i]);
    }
    printf("%d\n", sum);
    return 0;
}
```