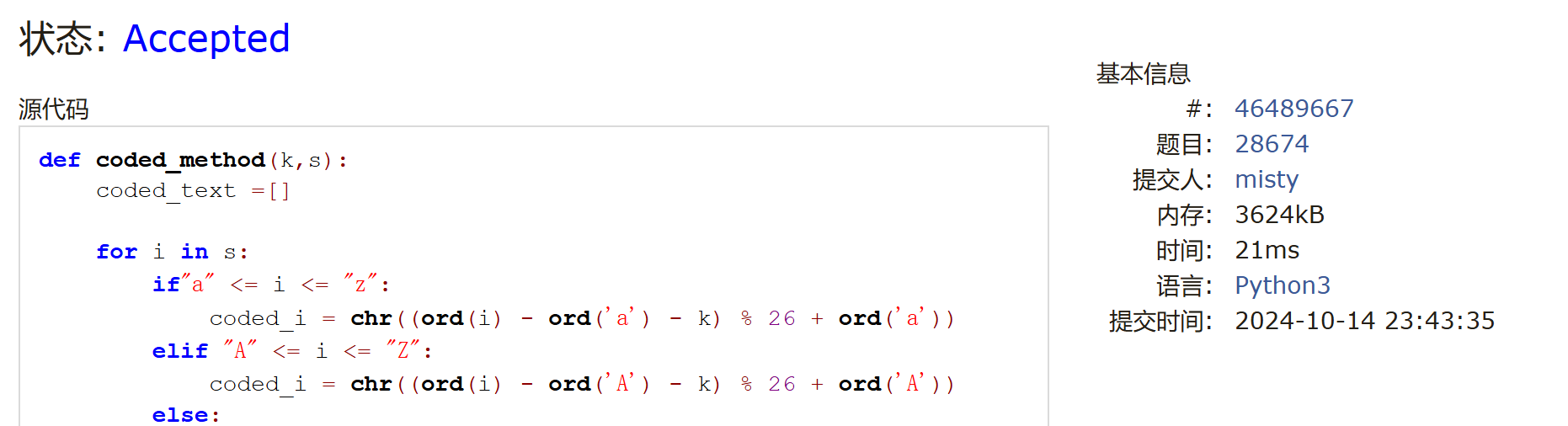
1. 题目

E28674:《黑神话：悟空》之加密

<http://cs101.openjudge.cn/practice/28674/>

**代码**  
def coded\_method(k,s):  
 coded\_text =[]  
  
 for i in s:  
 if"a" <= i <= "z":  
 coded\_i = chr((ord(i) - ord('a') - k) % 26 + ord('a'))  
 elif "A" <= i <= "Z":  
 coded\_i = chr((ord(i) - ord('A') - k) % 26 + ord('A'))  
 else:  
 coded\_i = i  
 coded\_text.append(coded\_i)  
  
 return "".join(coded\_text)  
  
k = int(input())  
s = input()  
  
print(coded\_method(k,s))

代码运行截图 ==（至少包含有"Accepted"）==



备注：题干中提到的加密方法是a往后找k个字母，如a找3个到d。但是由于是破译，示例中是向前找k个，这涉及到代码中究竟是加还是减k，做题时曾在这里卡住。

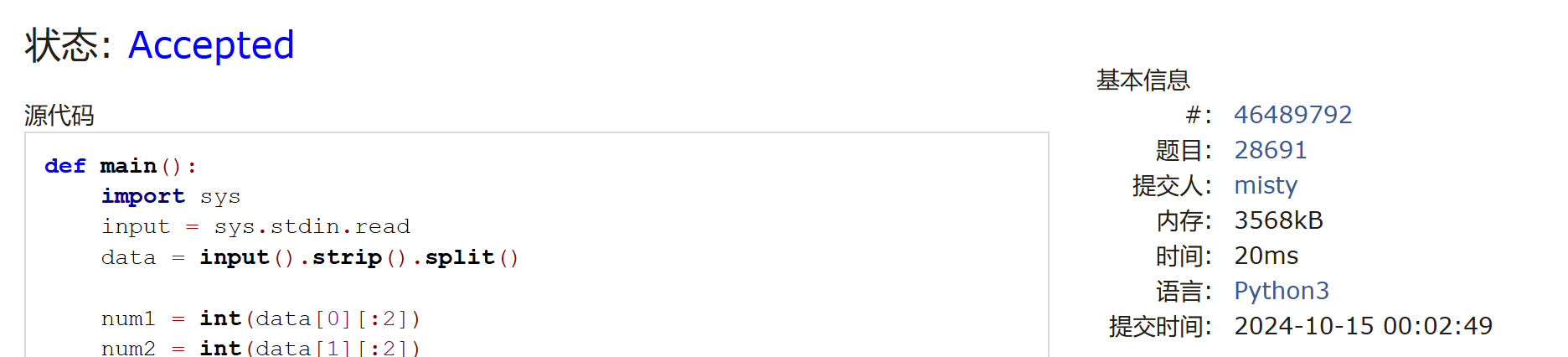
E28691: 字符串中的整数求和

<http://cs101.openjudge.cn/practice/28691/>

**代码**

def main():  
 import sys  
 input = sys.stdin.read  
 data = input().strip().split()  
  
 num1 = int(data[0][:2])  
 num2 = int(data[1][:2])  
  
 result = num1 + num2  
 print(result)  
  
main()

代码运行截图 ==（至少包含有"Accepted"）==



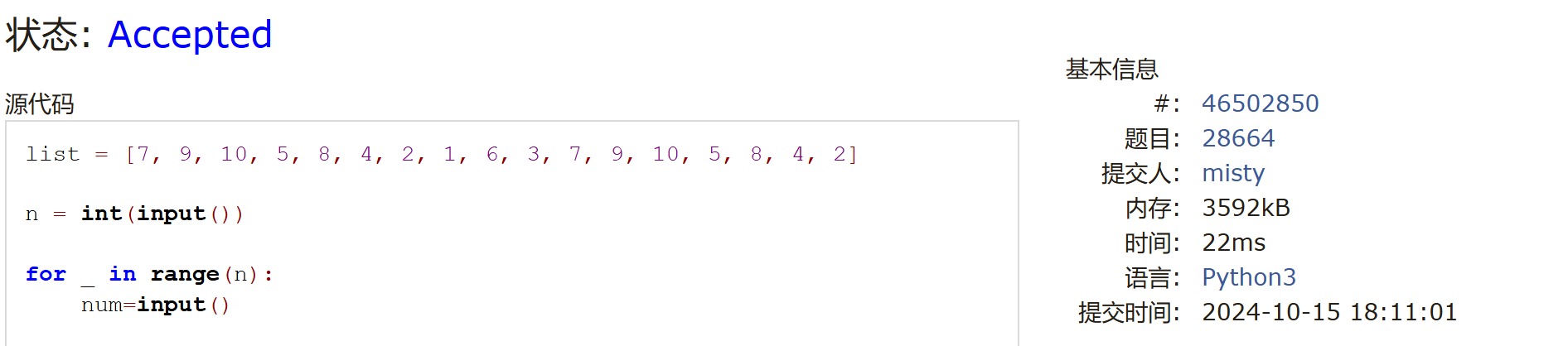
M28664: 验证身份证号

<http://cs101.openjudge.cn/practice/28664/>

**代码**

list = [7, 9, 10, 5, 8, 4, 2, 1, 6, 3, 7, 9, 10, 5, 8, 4, 2]  
  
n = int(input())  
  
for \_ in range(n):  
 num=input()  
  
 if len(num) != 18:  
 print("NO")  
 continue  
  
 x = sum(int(num[i]) \* list[i] for i in range(17)) % 11  
 x = (12 - x) % 11  
  
 if x == 10:  
 x = "X"  
  
 if num[17] == str(x):  
 print("YES")  
 else:  
 print("NO")

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==



M28678: 角谷猜想

<http://cs101.openjudge.cn/practice/28678/>

**代码**

def collatz\_sequence(n):

if n == 1:

print("End")

return

while n != 1:

if n % 2 == 1:

next\_n = 3 \* n + 1

print(f"{n}\*3+1={next\_n}")

else:

next\_n = n // 2

print(f"{n}/2={next\_n}")

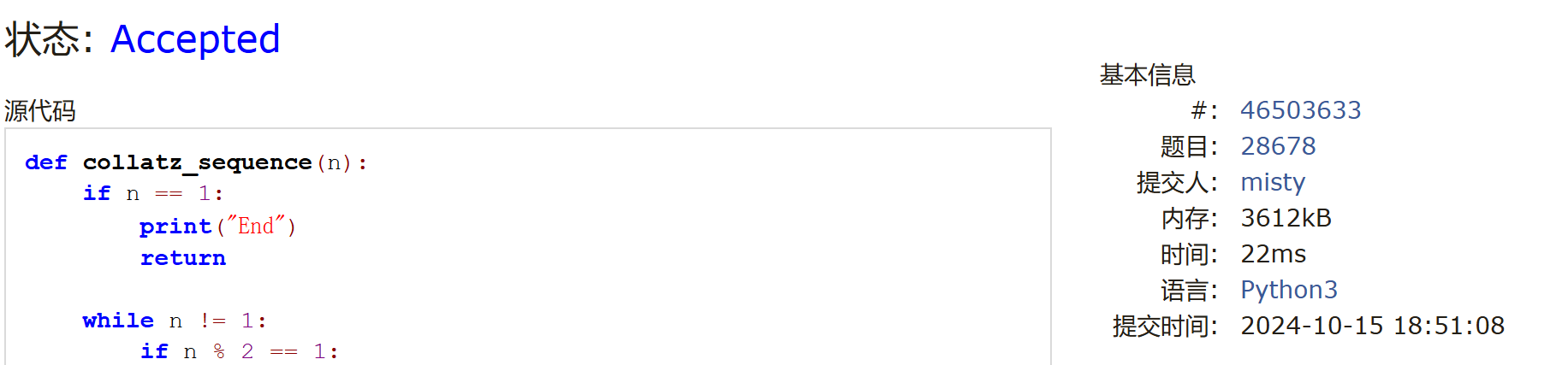
n = next\_n

print("End")

n = int(input())

collatz\_sequence(n)

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==



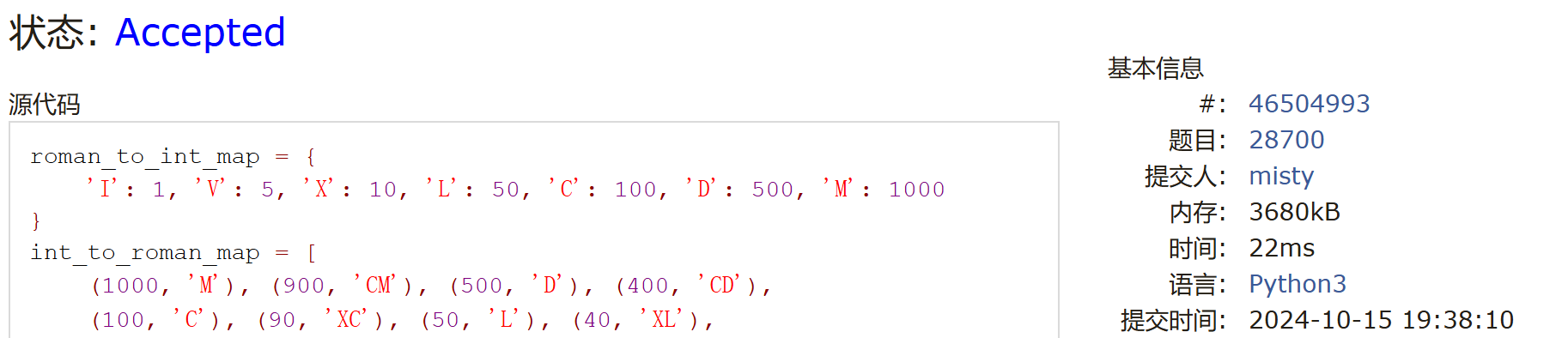
M28700: 罗马数字与整数的转换

<http://cs101.openjudge.cn/practice/28700/>

**代码**

roman\_to\_int\_map = {  
 'I': 1, 'V': 5, 'X': 10, 'L': 50, 'C': 100, 'D': 500, 'M': 1000  
}  
int\_to\_roman\_map = [  
 (1000, 'M'), (900, 'CM'), (500, 'D'), (400, 'CD'),  
 (100, 'C'), (90, 'XC'), (50, 'L'), (40, 'XL'),  
 (10, 'X'), (9, 'IX'), (5, 'V'), (4, 'IV'), (1, 'I')  
]  
  
def roman\_to\_int(s):  
 total = 0  
 prev\_value = 0  
 for char in s:  
 value = roman\_to\_int\_map[char]  
 if value > prev\_value:  
 total += value - 2 \* prev\_value # 处理特殊情况，如IV, IX  
 else:  
 total += value  
 prev\_value = value  
 return total  
  
def int\_to\_roman(num):  
 result = []  
 for value, symbol in int\_to\_roman\_map:  
 while num >= value:  
 result.append(symbol)  
 num -= value  
 return ''.join(result)  
  
def main():  
 input\_data = input().strip()  
  
 if input\_data.isdigit():  
 num = int(input\_data)  
 print(int\_to\_roman(num))  
 else:  
 print(roman\_to\_int(input\_data))  
  
main()

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==



\*T25353: 排队 （选做）

<http://cs101.openjudge.cn/practice/25353/>

思路：

代码

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

学习总结和收获

1. 第3和第5题卡了好久，第三题runtime error好多回没有找出问题所在，第五题写了好长自然花的时间很久。
2. 建立映射时注意格式
3. 第3题与同学讨论时，发现if逻辑关系在该断开的时候没有断开会导致wrong answer，但是使用样例输入跑通并未检查出来，因为样例输入对应的输出并未包含全部情况的输出结果。

