*# Assign #3: Oct Mock Exam暨选做题目满百*

Updated 1537 GMT+8 Oct 10, 2024

2024 fall, Complied by Hongfei Yan==（付耀贤 信息管理系 2400016634）==

Oct⽉考：AC 4

*## 1. 题目*

*### E28674:《黑神话：悟空》之加密*

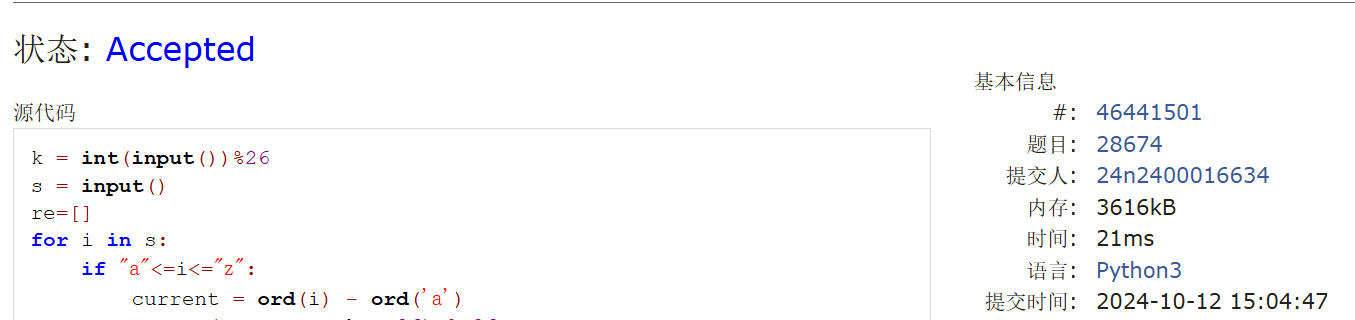
http://cs101.openjudge.cn/practice/28674/

思路：遇到第一题，由于还没进入状态和不熟练ord和chr，思维比较停滞。后来把下一道简单题目做完后，冷静下来，用了创建了26字母大小写字符串索引的方法位移，有惊无险地AC了，课后又重写了一遍使用ord和chr的算法。

代码

k = int(input())%26  
s = input()  
re=[]  
for i in s:  
 if "a"<=i<="z":  
 current = ord(i) - ord('a')  
 new = (current - k + 26) % 26  
 n = chr(new + ord('a'))  
 re.append(n)  
 else:  
 current = ord(i) - ord('A')  
 new = (current - k + 26) % 26  
 n = chr(new + ord('A'))  
 re.append(n)  
print(\*re,sep="")

代码运行截图 ==（至少包含有"Accepted"）==



*### E28691: 字符串中的整数求和*

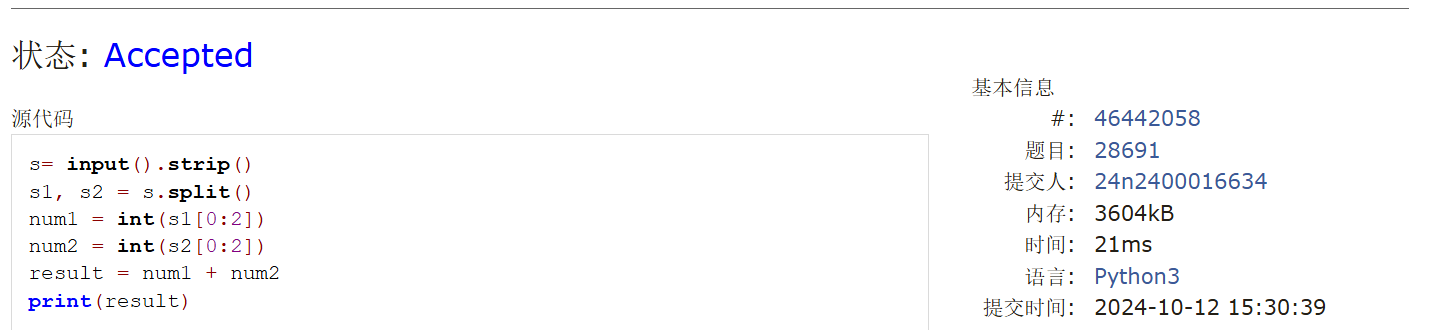
http://cs101.openjudge.cn/practice/28691/

思路：比较简单，起到稳定情绪的重要作用

代码

s= input().strip()  
s1, s2 = s.split()  
num1 = int(s1[0:2])  
num2 = int(s2[0:2])  
result = num1 + num2  
print(result)

代码运行截图 ==（至少包含有"Accepted"）==



*### M28664: 验证身份证号*

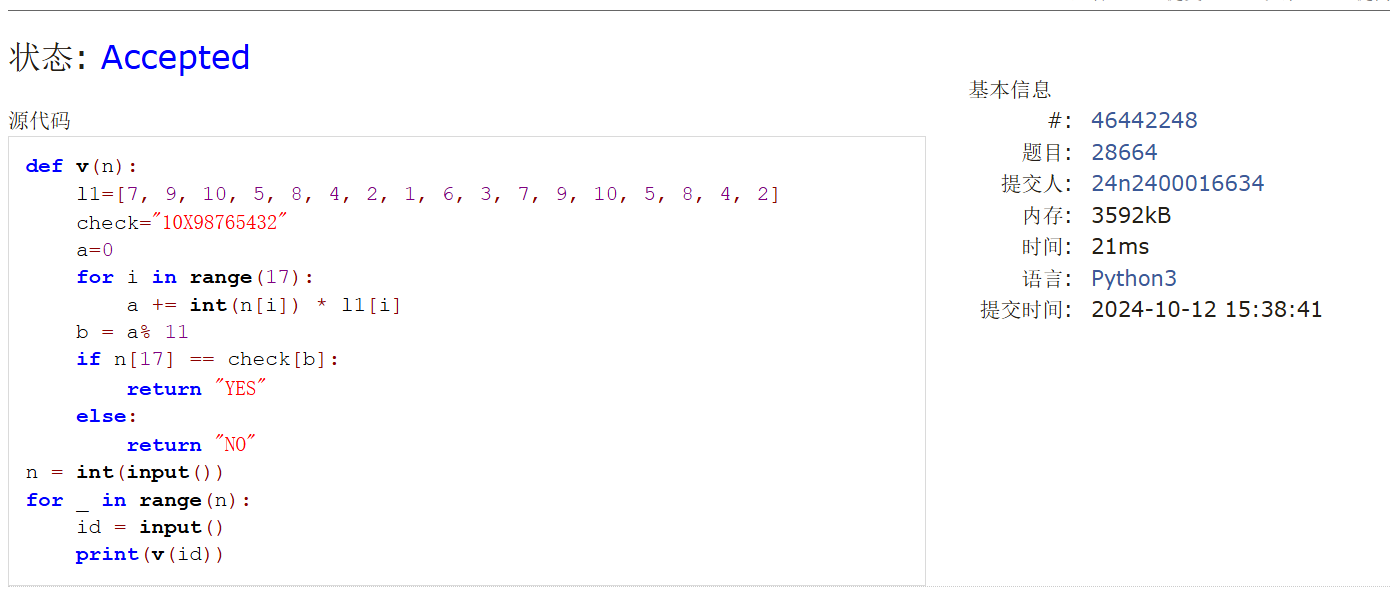
http://cs101.openjudge.cn/practice/28664/

思路：整体思路不难，在循环结构中忘记重置变量，酿成大错，还好最后迷途知返。

代码

def v(n):  
 l1=[7, 9, 10, 5, 8, 4, 2, 1, 6, 3, 7, 9, 10, 5, 8, 4, 2]  
 check="10X98765432"  
 a=0  
 for i in range(17):  
 a += int(n[i]) \* l1[i]  
 b = a% 11  
 if n[17] == check[b]:  
 return "YES"  
 else:  
 return "NO"  
n = int(input())  
for \_ in range(n):  
 id = input()  
 print(v(id))

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/

思路：这道题练习中曾出现过，主要是格式的输出，题目难度不大，考场上很快AC

代码：

n=int(input())  
while n!=1:  
 if n%2!=0:  
 print("{}\*3+1={}".format(n,n\*3+1))  
 n=n\*3+1  
 else:  
 print("{}/2={}".format(n,n//2))  
 n=n//2  
else:  
 print("End")

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*

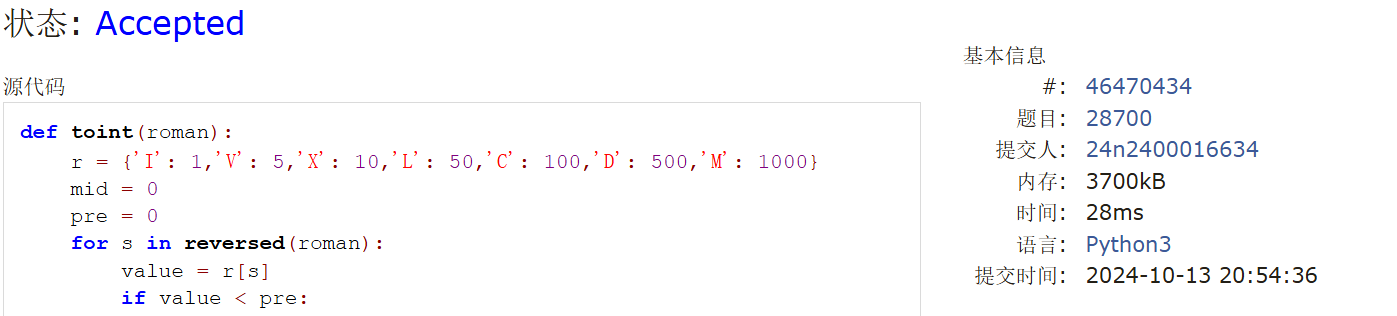
http://cs101.openjudge.cn/practice/28700/

思路：考场上是真不会做。课下借助课程群和AI的提示慢慢想到该怎么做了，只能说考场上做不出来是可以接受的，争取以后遇到这类题能在考场上有思路。

*##### 代码*

def toint(roman):  
 r = {'I': 1,'V': 5,'X': 10,'L': 50,'C': 100,'D': 500,'M': 1000}  
 mid = 0  
 pre = 0  
 for s in reversed(roman):  
 value = r[s]  
 if value < pre:  
 mid -= value  
 else:  
 mid += value  
 pre = value  
 return mid  
  
def toroman(num):  
 v = [  
 (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')  
 ]  
 roman = []  
 for value, symbol in v:  
 while num >= value:  
 roman.append(symbol)  
 num -= value  
 return ''.join(roman)  
  
s = input()  
if s.isdigit():  
 number = int(s)  
 print(toroman(number))  
else:  
 print(toint(s))

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==



*## 2. 学习总结和收获*

零基础刚起飞，考场上惊险地AC了4道，应该是很满意了。开始遇到第一题，由于还没进入状态，思维比较停滞。后来把下一道简单题目做完后，冷静下来，创建了大小写两个字符串索引位移，有惊无险地AC了，课后又重写了一遍使用ord和chr的算法。最大的堵点是身份证号验证，由于忘了在循环结构中重置变量，一直提示WA，在考场有一种很强烈的无力感，还好最后迷途知返。最后两道不是我能在考场上AC的题目，罗马数字在提示下能勉强做出来；排队问题到现在都搞不懂，遂暂时放弃。总体觉得，考试时比较紧张，思维有时候不能舒展开来，要不断模拟考场心态；一些基本功不扎实，一个小地方想不出来就可能导致整题崩盘，下一步要注重基本语法的巩固。

每日选做的难度上升到1100＋的时候感觉进入一个新阶段，经常遇到题目读不懂或者超时，独立写不出能AC的代码。需要不断学习优化时间的算法和数据结构，然后举一反三；同时尽力提高自己的思维能力，争取能处理一些稍复杂的问题。