

Assign #3: Oct Mock Exam暨选做题目满百

1. 题目

E28674: 《黑神话：悟空》之加密

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

思路：

放在一个库里，解码，读取，提前把k标准化算是一个简化流程吧

代码

```
small =
['a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t',
'u','v','w','x','y','z']
big =
['A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T',
'U','V','W','X','Y','Z']
k = int(input())
string = input()
result = []
while k > 26:
    k -= 26
for i in range(len(string)):
    for j in range(26):
        if string[i] == small[j]:
            num = j - k
            if num < 0:
                num += 26
            result.append(small[num])
        elif string[i] == big[j]:
            num = j - k
            if num < 0:
                num += 26
            result.append(big[num])
print(''.join(result))
```

代码运行截图 == (至少包含有"Accepted") ==

状态: Accepted

源代码

```
small = ['a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r',
big = ['A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S
k = int(input())
string = input()
result = []
while k > 26:
    k -= 26
for i in range(len(string)):
    for j in range(26):
        if string[i] == small[j]:
            num = j - k
            if num < 0:
                num += 26
            result.append(small[num])
        elif string[i] == big[j]:
            num = j - k
            if num < 0:
                num += 26
            result.append(big[num])
print(''.join(result))
```

E28691: 字符串中的整数求和

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

思路: 直接提取, 相加, 技术含量为0

代码

```
string_1, string_2 = input().split()
print(int(string_1[:2]) + int(string_2[:2]))
```

代码运行截图 == (至少包含有"Accepted") ==

状态: Accepted

源代码

```
string_1, string_2 = input().split()
print(int(string_1[:2]) + int(string_2[:2]))
```

M28664: 验证身份证号

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

思路：放列表里处理处理然后求余数，难点是手动输入好累

代码

```
n = int(input())
for i in range(n):
    idn = input()
    index = [7,9,10,5,8,4,2,1,6,3,7,9,10,5,8,4,2]
    test = [['1', 0], ['0', 1], ['x', 2], ['9', 3], ['8', 4], ['7', 5], ['6', 6],
            ['5', 7], ['4', 8], ['3', 9], ['2', 10]]
    num = [int(_) for _ in idn[:17]]
    pro = [index[m] * num[m] for m in range(17)]
    res = sum(pro) % 11
    flag = True
    for j in range(11):
        if res == test[j][1] and idn[17] == test[j][0]:
            flag = False
    print('YES' if flag == False else 'NO')
```

代码运行截图 == (AC代码截图，至少包含有"Accepted") ==

状态: Accepted

源代码

```
n = int(input())
for i in range(n):
    idn = input()
    index = [7,9,10,5,8,4,2,1,6,3,7,9,10,5,8,4,2]
    test = [['1', 0], ['0', 1], ['X', 2], ['9', 3], ['8', 4], ['7', 5], ['6', 6],
            ['5', 7], ['4', 8], ['3', 9], ['2', 10]]
    num = [int(_) for _ in idn[:17]]
    pro = [index[m] * num[m] for m in range(17)]
    res = sum(pro) % 11
    flag = True
    for j in range(11):
        if res == test[j][1] and idn[17] == test[j][0]:
            flag = False
    print('YES' if flag == False else 'NO')
```

M28678: 角谷猜想

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

思路：之前做过，不赘述

代码

```
n = int(input())
while n > 1:
    if n % 2 == 0:
        n = n//2
        print('%d/2=%d' %(2 * n, n))
    else:
        n = 3 * n + 1
        print('%d*3+1=%d' %((n - 1)//3, n))
print('End')
```

代码运行截图 == (AC代码截图, 至少包含有"Accepted") ==

状态: Accepted

源代码

```
n = int(input())
while n > 1:
    if n % 2 == 0:
        n = n//2
        print('%d/2=%d' %(2 * n, n))
    else:
        n = 3 * n + 1
        print('%d*3+1=%d' %((n - 1)//3, n))
print('End')
```

M28700: 罗马数字与整数的转换

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

思路: (优化过后和标答一致)

分两步, 就数字转罗马: 辗转相减, 加到列表里

罗马转数字: 分割是难点, 需要分类讨论

同样, 手输字典好累

代码

```
#调整后和标答基本一致, 就放标答了

# 定义罗马数字和整数的映射关系
roman_to_int_map = {
```

```

    'I': 1, 'V': 5, 'X': 10, 'L': 50, 'C': 100, 'D': 500, 'M': 1000
}

# 定义整数到罗马数字的映射列表（从大到小顺序）
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')
]

# 罗马数字转整数
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

# 整数转罗马数字
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)

# 主函数，判断输入是罗马数字还是整数
def main():
    # 输入处理
    input_data = input().strip()

    # 判断输入是整数还是罗马数字
    if input_data.isdigit():
        # 输入是整数
        num = int(input_data)
        print(int_to_roman(num))
    else:
        # 输入是罗马数字
        print(roman_to_int(input_data))

# 调用主函数
if __name__ == "__main__":
    main()

```

代码运行截图 == (AC代码截图, 至少包含有"Accepted") ==

状态: Accepted

源代码

```
# 定义罗马数字和整数的映射关系
```

***T25353: 排队 (选做)**

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

思路: 折腾了快四个小时都没自主解决, 看了标答也半懂半不懂的, 心累

代码

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

==如果作业题目简单, 有否额外练习题目, 比如: OJ“计概2024fall每日选做”、CF、LeetCode、洛谷等网站题目。==

就平日练习进度而言: 每日练习坚持跟着做, 有时做贪心会有些吃力, 但是语法还在进步, 开始掌握缩短简化语句的乐趣

就本次考试而言: 卡了一下时间, 第一题18分钟, 第二题2分钟, 第三题15分钟, 第四题5分钟, 第五题30分钟。

本次考试运用大量字典, 平常练习能不用字典就不用的我在没有搜索的条件下花了一点时间凭记忆熟悉字典语法, 这几道题思路上都还算简单, 除了第五题一开始罗马转数字上卡了一下, 都没啥大问题

第六题真的不会做, 太难了.....中间断断续续试了快四个小时, 各种地方出岔子, 最后凭个人努力也没能ac, 慢慢来吧。