

Assign #3: Oct Mock Exam 暨选做题目满百

Updated 1537 GMT+8 Oct 10, 2024

2024 fall, Compiled by Hongfei Yan==（汤伟杰，信息管理系）==

说明：

- 1) Oct月考：AC6==（4）==。考试题目都在“题库（包括计概、数算题目）”里面，按照数字题号能找到，可以重新提交。作业中提交自己最满意版本的代码和截图。
- 2) 请把每个题目解题思路（可选），源码 Python, 或者 C++/C（已经在 Codeforces/Openjudge 上 AC），截图（包含 Accepted, 学号），填写到下面作业模版中（推荐使用 typora <https://typoraio.cn>，或者用 word）。AC 或者没有 AC，都请标上每个题目大致花费时间。
- 3) 提交时候先提交 pdf 文件，再把 md 或者 doc 文件上传到右侧“作业评论”。Canvas 需要有同学清晰头像、提交文件有 pdf、作业评论有 md 或者 doc。
- 4) 如果不能在截止前提交作业，请写明原因。

1. 题目

E28674: 《黑神话：悟空》之加密

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

思路：想到了用 ascii 表来计算，k 只需要 mod26 即可，然后小写字母如果减 k 小于 a 的 97，就从 z 的 120 开始减剩下的直到得出答案，大写字母同理，要用到 string 里面的 `ascii_lowercase` 和 `ord()`,`chr()`函数。

但是计算调整用时较长，大概用 25min

代码

```
import string
kk=int(input())
k=kk%26
s=input()
ans=''
for i in s:
    if i in string.ascii_lowercase:
        if ord(i)-k<97:
            rest=k-ord(i)+97
            ans+=chr(122-rest+1)
        else:
            ans+=chr(ord(i)-k)
    else:
        if ord(i)-k<65:
            rest=k-ord(i)+65
            ans+=chr(90-rest+1)
        else:
```

```
        ans+=chr(ord(i)-k)
print(ans)
```

代码运行截图 ==（至少包含有"Accepted"）==

#46399992提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```
import string
kk=int(input())
k=kk%26
s=input()
ans=''
for i in s:
    if i in string.ascii_lowercase:
        if ord(i)-k<97:
            rest=k-ord(i)+97
            ans+=chr(122-rest+1)
        else:
            ans+=chr(ord(i)-k)
    else:
        if ord(i)-k<65:
            rest=k-ord(i)+65
            ans+=chr(90-rest+1)
        else:
            ans+=chr(ord(i)-k)
print(ans)
```

基本信息

#: 46399992
题目: E28674
提交人: 24n2400016635
内存: 3904kB
时间: 23ms
语言: Python3
提交时间: 2024-10-10 15:27:51

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关](#)

E28691: 字符串中的整数求和

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

思路: 直接取出前两个字符相加, 用 2min

代码

```
a,b=input().split()
na=int(a[:2])
nb=int(b[:2])
print(na+nb)
```

代码运行截图 ==（至少包含有"Accepted"）==

状态: **Accepted**

源代码

```
a,b=input().split()
na=int(a[:2])
nb=int(b[:2])
print(na+nb)
```

基本信息

#: 46400082
题目: E28691
提交人: 24n2400016635
内存: 3576kB
时间: 21ms
语言: Python3
提交时间: 2024-10-10 15:29:35

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

M28664: 验证身份证号

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

思路：因为要对于数字乘系数，想到了提前用字典存系数，后面直接用。把最后一位的计算结果和对照结果也用字典储存起来，方便判断是否合理。在第 5 题里面也有用字典的想法，比较方便，但是字典打起来有点费时间。

用 15min

代码

```
d={0:7,1:9,2:10,3:5,4:8,5:4,6:2,7:1,8:6,9:3,10:7,11:9,12:10,13:5,14:8,15:4,16:2}
ans={0:1,1:0,2:'X',3:9,4:8,5:7,6:6,7:5,8:4,9:3,10:2}
n=int(input())
for _ in range(n):
    s=input()
    cnt=0
    for i in range(0,17):
        cnt+=int(s[i])*int(d[i])
    mo_cnt=cnt%11
    if str(ans[mo_cnt])==str(s[-1]):
        print('YES')
    else:
        print('NO')
```

代码运行截图 ==（AC 代码截图，至少包含有"Accepted"）==

#46400461提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```
d={0:7,1:9,2:10,3:5,4:8,5:4,6:2,7:1,8:6,9:3,10:7,11:9,12:10,13:5,14:8,15:4,16:2}
ans={0:1,1:0,2:'X',3:9,4:8,5:7,6:6,7:5,8:4,9:3,10:2}
n=int(input())
for _ in range(n):
    s=input()
    cnt=0
    for i in range(0,17):
        cnt+=int(s[i])*int(d[i])
    mo_cnt=cnt%11
    if str(ans[mo_cnt])==str(s[-1]):
        print('YES')
    else:
        print('NO')
```

基本信息

#: 46400461
题目: M28664
提交人: 24n2400016635
内存: 3592kB
时间: 19ms
语言: Python3
提交时间: 2024-10-10 15:38:27

M28678: 角谷猜想

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

思路：用 while 循环直到 n 减小到 1，中间不断用 f-string 来输出过程的计算式

用 8min

代码

```
n=int(input())
while n>1:
    if n%2==0:
```

```

        print(f'{n}/2={n//2}')
        n//=2
    else:
        print(f'{n}*3+1={n*3+1}')
        n=n*3+1
print('End')

```

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

#46400566提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```

n=int(input())
while n>1:
    if n%2==0:
        print(f'{n}/2={n//2}')
        n//=2
    else:
        print(f'{n}*3+1={n*3+1}')
        n=n*3+1
print('End')

```

基本信息

#: 46400566
 题目: M28678
 提交人: 24n2400016635
 内存: 3600kB
 时间: 19ms
 语言: Python3
 提交时间: 2024-10-10 15:40:57

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

M28700: 罗马数字与整数的转换

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

思路: 刚开始想打表, 把两者之间的关系全部用两个字典搞出来, 但是太耗时间放弃这个思路;

然后想到可以用阿拉伯数字可能对应的字母用字典储存, 这样转化为罗马数字就十分简单了, 但是转化为阿拉伯数字一开始想到的是对字母遍历并不断分类, 但是这样代码又长又易出错, 导致时间浪费完了;

考试结束时候看到测试数据才到 3999, 就想到完全可以把 1 到 3999 的数字全部转化为罗马数字, 用字典保存; 再把这个字典的 key,value 变成新字典的 value,key 就得到了两种互相转化的更好方法。

考试用时 1h, 思路没找到最优

代码

```

d={1:['I','X','C','M'],2:['II','XX','CC','MM'],3:['III','XXX','CCC','MMM'],
  4:['IV','XL','CD','MMMM'],5:['V','L','D','M'*5],
  6:['VI','LX','DC','M'*6],7:['VII','LXX','DCC','M'*7],
  8:['VIII','LXXX','DCCC','M'*8],9:['IX','XC','CM','M'*9]}
ans_ll={};ans_nn={} #分别为阿拉伯转罗马, 罗马转阿拉伯的字典
#把 1 到 3999 全部转化为罗马数字并用字典保存
for p in range(1,4000):
    ans_l = ''
    s=str(p)
    cnt=1
    for i in s:

```

```

        j = int(i)
        if j != 0:
            ans_l += d[j][len(s) - cnt]
            cnt += 1
        else:
            pass
            cnt += 1
        ans_ll[p] = ans_l
#我把键值互换就得到了另一种转化的字典，简单多了
for key,value in ans_ll.items():
    ans_nn[value]=key

j=input()

#numbers
if j[0] in str(d.keys()):
    print(ans_ll[int(j)])
#luoma
else:
    print(ans_nn[j])

```

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

#46409406提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```

d={1:['I','X','C','M'],2:['II','XX','CC','MM'],3:['III','XXX','CCC','MMM'],
  4:['IV','XL','CD','MMMM'],5:['V','L','D','M'*5],
  6:['VI','LX','DC','M'*6],7:['VII','LXX','DCC','M'*7],
  8:['VIII','LXXX','DCCC','M'*8],9:['IX','XC','CM','M'*9]}
ans_ll={};ans_nn={}
for p in range(1,4000):
    ans_l = ''
    s=str(p)
    cnt=1
    for i in s:
        j = int(i)
        if j != 0:
            ans_l += d[j][len(s) - cnt]
            cnt += 1
        else:
            pass
            cnt += 1
    ans_ll[p] = ans_l
for key,value in ans_ll.items():
    ans_nn[value]=key

j=input()

#numbers
if j[0] in str(d.keys()):
    print(ans_ll[int(j)])
#luoma
else:
    print(ans_nn[j])

```

基本信息

#: 46409406
 题目: 28700
 提交人: 24n2400016635
 内存: 4348kB
 时间: 29ms
 语言: Python3
 提交时间: 2024-10-10 17:57:10

*T25353: 排队 (选做)

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

思路：真没思路。补：参考思路后尝试代码如下

代码

```
#TIME LIMIT EXCEEDED!!!!!!!
from collections import deque

n,d=map(int,input().split())
q=deque(list(map(int,input().split())))
ans,curr=[],[];mi=q[0];ma=q[0]
while q:
    curr=[] #储存每一层的数
    for i in q.copy(): #用 q 的 copy 列表以防止在对列表数据处理时出现遍历出错，但后果是时间复
        if i<mi:          #杂度很大
            mi=i
        if i>ma:
            ma=i
        q.popleft()      #O(1)
        if i+d<=ma and i-d>=mi:
            curr.append(i)
        else:
            q.append(i)
    ans+=sorted(curr)
print(ans)
```

代码运行截图 ==（AC 代码截图，至少包含有"Accepted"）==

TLE

#46437647提交状态

查看提交统计提问

状态: Time Limit Exceeded

源代码

```
from collections import deque

n,d=map(int,input().split())
q=deque(list(map(int,input().split())))
ans,curr=[],[];mi=q[0];ma=q[0]
while q:
    curr=[]
    for i in q.copy():
        if i<mi:
            mi=i
        if i>ma:
            ma=i
        q.popleft()
        if i+d<=ma and i-d>=mi:
            curr.append(i)
        else:
            q.append(i)
    ans+=sorted(curr)
print(ans)
```

基本信息

#: 46437647

题目: 25353

提交人: 24n2400016635

内存: 3624kB

时间: 29931ms

语言: Python3

提交时间: 2024-10-12 09:36:26

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。==

①身份证号和数字转换两道题我都明显感觉到了字典的使用（但是感觉字典输入的时候很痛苦很费时）。同时也要注意题目数据的范围提示，比如罗马数字那题，整数最大才 3999，那就完全可以把从 1 到 3999 的所有互换结果保存起来，直接输出答案。

②但是最后排队的题我完全不知道考的是什么知识点，更不用说怎么下手去做了.....不过收获了经验就是做题一定不能慌，简单的题要冷静分析才能保证代码一遍过，否则会很影响心态。

③在 oj 集合加法里面，初步想到遍历两个列表但是肯定超时，想到用二分查找但是查找后还需要用 `count()` 结果还是超时，问了 gpt 知道了有

```
from bisect import bisect_left, bisect_right
```

从而可以完成在特定列表找到特定连续相同元素的区间范围，优化了算法；最后看老师提供的答案发现 `Counter` 函数原来是最快速的。

补：④在参考群里面排队的思路后先自己写但是超时，比较答案之后发现问题是答案采用一系列布尔值储存列表中的元素是否已经使用过，而避免了 `pop()` 和 `append()` 等已经我想的 `copy()` 等操作，这个方法值得学习。