

Assignment #2: 语法练习

Updated 0126 GMT+8 Oct 5, 2024

2024 fall, Compiled by ==胡杨 元培学院==

说明:

1) 请把每个题目解题思路 (可选), 源码Python, 或者C++ (已经在Codeforces/Openjudge上AC), 截图 (包含Accepted), 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) 课程网站是Canvas平台, <https://pku.instructure.com>, 学校通知9月19日导入选课名单后启用。作业写好后, 保留在自己手中, 待9月20日提交。

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”区有上传的md或者doc附件。

4) 如果不能在截止前提交作业, 请写明原因。

1. 题目

263A. Beautiful Matrix

<https://codeforces.com/problemset/problem/263/A>

思路: 相当于是计算5*5的矩阵中数字1到 (2, 2) 的位置要移动几步 (从0开始编索引)。首先的难点是如何input一个5*5的矩阵XD, 计算距离可以用数字1的索引数- (2, 2) 求绝对值

大致用时: 10min (因为做每日题的时候做过)

题解启示

1. 可以直接读取一个矩阵

```
matrix=[[int(x) for x in input().split()]for _ in range(5)]
```

或

```
matrix=[list(map(int,input().split())) for _ in range(5)]
```

也可以一行一行的读取输入

```
for i in range(5):  
    s = input().split()
```

2. 可以用 `matrix[i].index(1)` 读取矩阵第i行中数字1所在位置的索引

3. abs() 绝对值计算

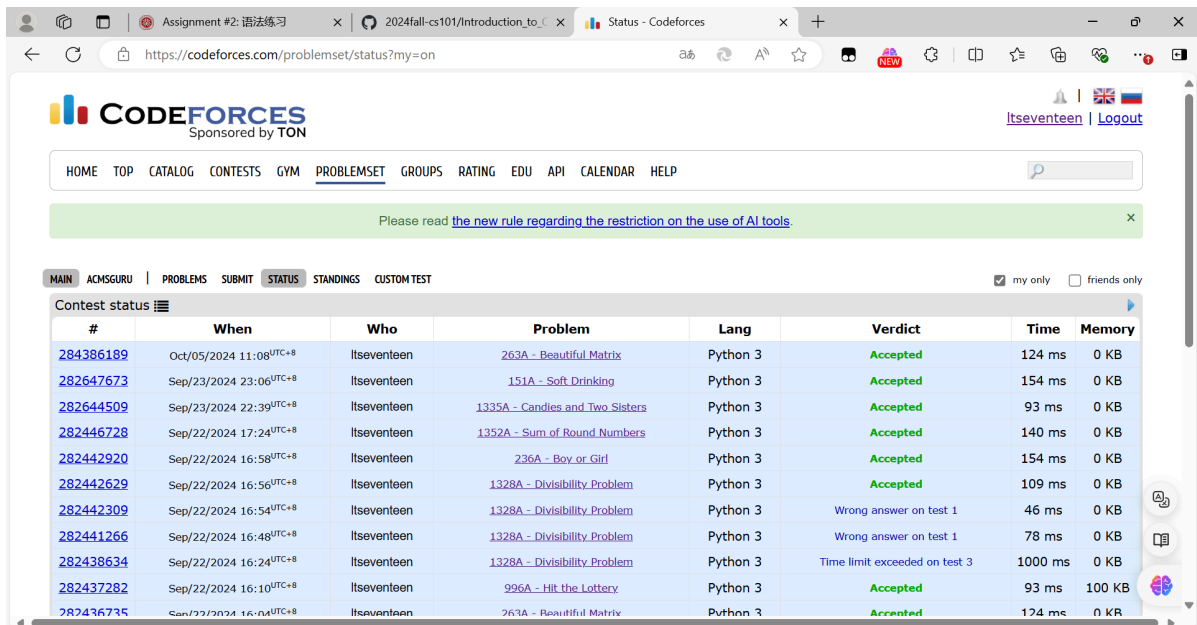
代码

```
#输入5*5的矩阵
matrix=[[int(x) for x in input().split()]for _ in range(5)]

#确定1的为位置
for i in range(5):
    for j in range(5):
        if matrix[i][j]==1:
            move=abs(i-2)+abs(j-2)#计算移动的步数

print(move)
```

代码运行截图 == (至少包含有"Accepted") ==



The screenshot shows the Codeforces website with the contest status page for problem 1328A. The table lists submissions with columns for ID, When, Who, Problem, Lang, Verdict, Time, and Memory. Several submissions are marked as 'Accepted'.

#	When	Who	Problem	Lang	Verdict	Time	Memory
284386189	Oct/05/2024 11:08 UTC+8	Itseventeen	263A - Beautiful Matrix	Python 3	Accepted	124 ms	0 KB
282647673	Sep/23/2024 23:06 UTC+8	Itseventeen	151A - Soft Drinking	Python 3	Accepted	154 ms	0 KB
282644509	Sep/23/2024 22:39 UTC+8	Itseventeen	1335A - Candies and Two Sisters	Python 3	Accepted	93 ms	0 KB
282446728	Sep/22/2024 17:24 UTC+8	Itseventeen	1352A - Sum of Round Numbers	Python 3	Accepted	140 ms	0 KB
282442920	Sep/22/2024 16:58 UTC+8	Itseventeen	236A - Boy or Girl	Python 3	Accepted	154 ms	0 KB
282442629	Sep/22/2024 16:56 UTC+8	Itseventeen	1328A - Divisibility Problem	Python 3	Accepted	109 ms	0 KB
282442309	Sep/22/2024 16:54 UTC+8	Itseventeen	1328A - Divisibility Problem	Python 3	Wrong answer on test 1	46 ms	0 KB
282441266	Sep/22/2024 16:48 UTC+8	Itseventeen	1328A - Divisibility Problem	Python 3	Wrong answer on test 1	78 ms	0 KB
282438634	Sep/22/2024 16:24 UTC+8	Itseventeen	1328A - Divisibility Problem	Python 3	Time limit exceeded on test 3	1000 ms	0 KB
282437282	Sep/22/2024 16:10 UTC+8	Itseventeen	996A - Hit the Lottery	Python 3	Accepted	93 ms	100 KB
282436735	Sep/22/2024 16:04 UTC+8	Itseventeen	263A - Beautiful Matrix	Python 3	Accepted	124 ms	0 KB

1328A. Divisibility Problem

<https://codeforces.com/problemset/problem/1328/A>

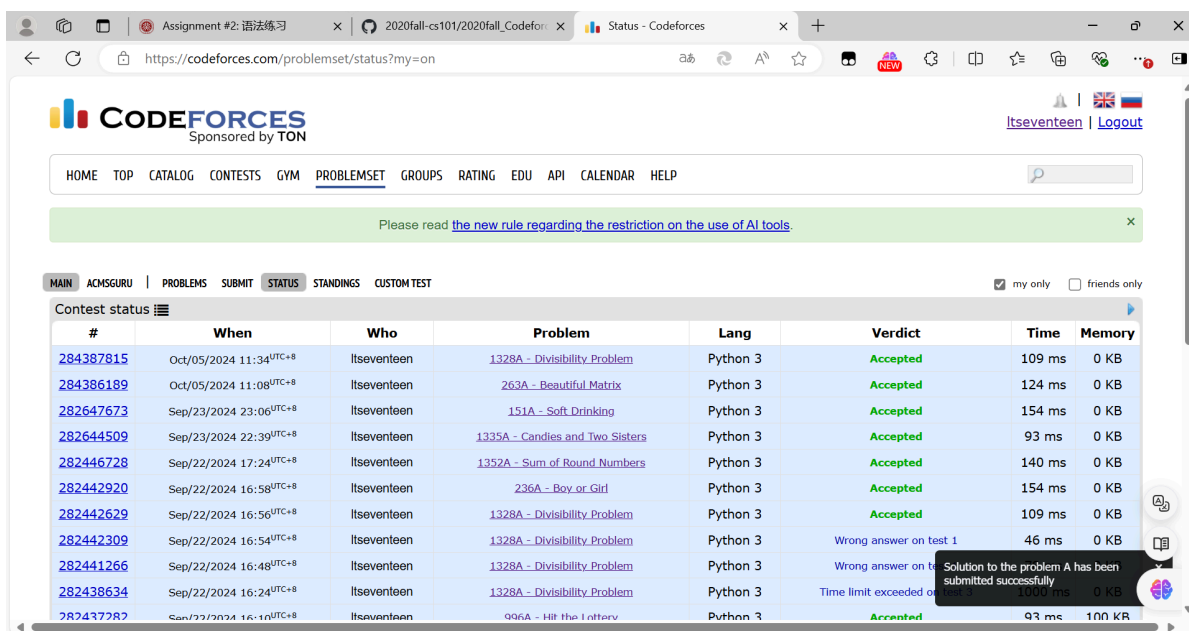
思路：本来想把a一次加一用for循环，直到a能够被b整除，但是发现会超时。如果a不能被b整除，可以计算a%b所得的余数x，所需要的步数即为(b-x)；如果a本来就可以被b整除，则步数是0 (if语句)

大概用时：10min (好像也做过)

代码

```
#读取输入
t=int(input())
for _ in range(t):#每次读取一组a,ba
    a,b=map(int,input().split())
    if a%b!=0:#看a是否能被b整除
        move=b-(a%b)
    else:
        move=0
    print(move)
```

代码运行截图 == (至少包含有"Accepted") ==



427A. Police Recruits

<https://codeforces.com/problemset/problem/427/A>

思路：在事件的列表中，如果第i项是一个正数，那么它可以用来抵消第i+1项及以后的-1，直到它变成0
或许可以尝试弄一个计数器，从第一个事件开始读取，如果x>0，police+x，如果x<0，考虑police的正负，如果police>0，police-1；如果police=0，police不变，并且untreated_crime+1

大致用时：25min（真·自己写的）

题解启示

- 1.可以直接读取列表中的内容，不一定非要用索引
- 2.灵活运用and,or等逻辑联系词可以减少延时

注意：or返回第一个真值（>0的正数），否则返回最后一个值

代码

```
n=int(input())
events=list(map(int,input().split()))

#初始化计数器
police=0
untreated_crime=0

#读取每一个事件，不是招人就是犯罪
for x in range(n):
    if 0<events[x]<=10:#招人
        police+=events[x]
    elif events[x]==-1:
        if police>0:#有人就治理犯罪
            police-=1
        elif police==0:#没有人就继续犯罪
            untreated_crime+=1

#输出结果
print(untreated_crime)
```

代码运行截图 == (AC代码截图，至少包含有"Accepted") ==

284390131	Oct/05/2024 12:06 ^{UTC+8}	Itseventeen	427A - Police Recruits	Python 3	Accepted	109 ms	8900 KB
-----------	------------------------------------	-------------	------------------------	----------	----------	--------	---------

(顺便我发现图床用不起的原因好像是我截的图太大了???)

02808: 校门外的树

<http://cs101.openjudge.cn/practice/02808/>

思路：题干让我想到了用集合取并集的方法，让我去现学一下。然后发现集合不完全是像数学里那么用的（.....）

可以先建一个长度为L的列表，每一项都是1.后面哪一段要修铁路就把那里面的1改成0.改过的再改一次也还是0.最后计数列表中1的数量

大概耗时：25min（集合碰壁过后看了一下题解）

题解启示

1. `alist.count(item)` 可以返回该列表中 `item` 的个数
2. 用平常的思考方式无解时，或许可以考虑用计算机模拟现实情况直接遍历？

代码

```
#先读取路的长度和建路的段数
L,M=map(int,input().split())
#建立种树列表
trees=[1]*(L+1)

#每一段铁路砍树
for _ in range(M):
    a,b=map(int,input().split())#读取起止点
    for i in range(a,b+1):#遍历 (a,b)段里的所有树, 注意不含结束点, 所以要是b+1
        trees[i]=0#砍掉树

#统计剩余的树的数量
print(trees.count(1))
```

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

状态: Accepted

源代码

```
#先读取路的长度和建路的段数
L,M=map(int,input().split())
#建立种树列表
trees=[1]*(L+1)

#每一段铁路砍树
for _ in range(M):
    a,b=map(int,input().split())#读取起止点
    for i in range(a,b+1):#遍历 (a,b)段里的所有树, 注意不含结束点, 所以要是b+1
        trees[i]=0#砍掉树

#统计剩余的树的数量
print(trees.count(1))
```

基本信息

#: 46315895
题目: 02808
提交人: 胡杨
内存: 3640kB
时间: 47ms
语言: Python3
提交时间: 2024-10-05 14:44:34

sy60: 水仙花数II

<https://sunnywhy.com/sfbj/3/1/60>

思路: 遍历闭区间[a,b]上的每一个数, 看它是不是水仙花数。一个三位数x的百位可以表示为x//100, 十位可以表示为x%100//10, 个位可以表示为x%100%10//1。输出时可以尝试用join整理格式?

大致用时: 25min (自己写的)

题解启示

- 1.AI似乎总是很喜欢先def包装一个函数, 再直接用它读取输入并输出结果
- 2.join可以用于整理字符串输出

基本语法:

```
str.join(iterable)
```

- `str`：这是作为**分隔符**的字符串，它将用于连接序列中的各个元素。
- `iterable`：这是一个**可迭代对象**，比如列表、元组、集合等，其中的**所有元素都必须是字符串类型**，否则会引发 `TypeError`。

代码

```
#读取[a,b]
a,b=map(int,input().split())
flower_number=[]#初始化列表

#遍历[a,b]中的所有整数,从小到大读取,判断是不是水仙花数
for i in range(a,b+1):
    x,y,z=i//100,i%100//10,i%100%10//1
    if i==x**3+y**3+z**3:
        flower_number.append(str(i))

#输出结果
if len(flower_number)==0:
    print('NO')
else:
    result=' '.join(flower_number)#用空格分隔输出结果,注意join后面的可迭代对象的每一个元素必须是字符串
    print(result)
```

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

水仙花数II

通过数 2687 提交数 9734 难度 简单 显示标签 ☆

题目描述

如果一个三位数 n 的各位数字的立方和等于 n , 那么称 n 为水仙花数。例如 $153 = 1^3 + 5^3 + 3^3$, 因此153是水仙花数。

给定两个正整数 a 、 b , 输出在闭区间 $[a, b]$ 内的所有水仙花数。

输入描述

两个正整数 a 、 b ($100 \leq a \leq b \leq 999$)。

输出描述

```
4
5 #遍历[a,b]中的所有整数,从小到大读取,判断是不是水仙花数
6 for i in range(a,b+1):
7     x,y,z=i//100,i%100//10,i%100%10//1
8     if i==x**3+y**3+z**3:
9         flower_number.append(str(i))
10
11 #输出结果
12 if len(flower_number)==0:
13     print('NO')
14 else:
15     result=' '.join(flower_number)#用空格分隔输出结果
16     print(result)
```

测试输入

提交结果

历史提交

完美通过

[查看题解](#)

100% 数据通过测试

运行时长: 0 ms

01922: Ride to School

<http://cs101.openjudge.cn/practice/01922/>

思路：N=0时结束输入，让我想到使用while N>0。然后大脑宕机了，找个同学教我想（.....）

其实我们并不需要关心Charley到达的过程，他必定和0点及以后出发的某个同学一起到（0点以前出发的同学，要么追不上，要么追上了但是Charley的速度本来就比他大，不会跟他一起到达）。只需要求出0点以后出发的同学谁到达时间最早，即为Charley的到达时间

感觉本题主要难在思路，如何简化Charley到达的问题

大致用时：50min

题解启示

1. `while true` 死循环，可以实现循环读取每个N，然后根据N是否为0决定是否终止循环

```
while True:
    n = int(input())
    if n == 0:
        break
```

2. `float('inf')` 浮点型正无穷大，`int('inf')` 整型正无穷大

代码

```
import math

#定义到达时间函数
def charley_arrival_time(riders):
    distance = 4.5 #公里数
    charley_time=float('inf')#Charley的到达时间初始化为无穷大

    for speed,start_time in riders:
        if start_time>=0:#如果出发时间大于等于0，则Charley有可能跟随
            riders_time=distance/speed*3600+start_time#计算每个骑手的到达时间
            charley_time=min(charley_time,riders_time)#更新Charley的到达时间

    return math.ceil(charley_time)#返回Charley的到达时间，向上取整

#读取输入，人数以及每个人的速度和出发时间
while True:#死循环，目的是读取输入的每个N后根据N是否为零判断是否退出循环
    N=int(input())
    if N==0:
        break

    riders=[]#初始化每个人速度和出发时间列表
    for _ in range(N):
        speed,start_time=map(int,input().split())
        riders.append((speed,start_time))
    print(charley_arrival_time(riders))#调用到达时间函数并打印结果
```

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

#46318725提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
import math

#定义到达时间函数
def charley_arrival_time(riders):
    distance = 4.5 #公里数
    charley_time=float('inf') #Charley的到达时间初始化为无穷大

    for speed,start_time in riders:
        if start_time>=0: #如果出发时间大于等于0, 则Charley有可能跟随
            riders_time=distance/speed*3600+start_time #计算每个骑手的到达时间
```

基本信息

#: 46318725

题目: 01922

提交人: 胡杨

内存: 5056kB

时间: 47ms

语言: Python3

提交时间: 2024-10-05 17:40:41

2. 学习总结和收获

新学到的知识点总结出现在每道题的“题解启示”板块，这里就不复制一遍了。另外在做每日题的时候建立了“每日题解偷师”文档，记录从题解中学到的内容。通过做题学习似乎是唯一一种我可以学进去语法的方法（之前尝试了看讲义看网站看书看课，都记不住一点），但是这样学会的知识似乎过于零散而不成体系，不知道怎么成体系学习，感觉我只会用我用过的东西，基础不牢有一堆空缺。

每日题做到了9.1，由于我前面疯狂做题屯了一堆解析没看，然后集中看解析信息量过大看崩溃了于是进度搁置.....完了做不完了X(

作业题目对我而言并不简单，特别是最后一题是在同学的指导下做的。我肯定对于原来连print都不会打的状态长进了许多，但这似乎远远不够，我不知道我进步的速度是否太慢了，而同时起点又太低，做的练习速度太慢，而数量也太少太少。我从来没有接触过类似的学科，不知道怎么学才是对的，每天看到群里同学们的学习总结，大家都从零基础小白变得熟悉编程语言了，成为大师了，做完选做题了，觉得作业简单了，似乎我是最差的那一个，于是每天都被焦虑折磨:(