

Assignment 1

02733 判断闰年

思路：简单的分类讨论，判断4整除 → 判断100整除 → 判断400整除

坑点在于题给出3000的时间限制，所以是否被3200整除不需要判断

代码：

```
`a=int(input('what is the year'))`

if a%4==0:`

    `if a%100==0:`

        •   `if a%400==0:`

        •   `\\#if a%3200==0:  cheat:3200>3000`

        •   `\\#    print('N')`

        •   `\\#else:`

        •   `    print('Y')`

        •   `else:`

        •   `    print('N')`

    `else:`

        •   `print('Y')`

`else:`

    `print('N')`
```

状态: Accepted

源代码

```
a=int(input())

if a%4==0:
    if a%100==0:
        if a%400==0:
            print("Y")
        else:
            print("N")
    else:
        print("Y")
else:
    print("N")
```

©2002-2022 POJ 京ICP备20010980号-1

自评：做的第一道题，明显不熟悉，一开始没有看出3200的坑，耗时约一小时

02750：鸡兔同笼

思路：整除的分类讨论，不绕脑子

代码：

```
feet=int(input())

head_max=feet//2

head_min=feet//4 #a//b=int(a/b)

if feet%2 == 0:

    if feet%4 == 0:

        • print(str(head_min)+' '+str(head_max))

        • \#print('{} {}'.format(head_min,head_max)) --by official answer

        • \#print(f'{head_min} {head_max}') --by yyx

        • \#print("%d %d" % (head_min,head_max)) #--by yyx

    else:

        • print(str(head_min+1)+' '+str(head_max))

else:

    print('0 0')
```

状态: Accepted

源代码

```
feet=int(input())
head_max=feet//2
head_min=feet//4 #a//b=int(a/b)
if feet%2 == 0:
    if feet%4 == 0:
        print(str(head_min)+' '+str(head_max))
    else:
        print(str(head_min+1)+' '+str(head_max))
else:
    print('0 0')
```

@2002-2022 B01 吉ICP备20010080号-1

自评：思路想了不到一分钟就出来了，难点在于如何把东西写的更简洁，一开始整数型和字符串转换非常不熟练，求助同学和网络后有所改善，自做+学习共一小时左右

主要学习了：

| a//b = int(a/b)

| format函数使用

| %控制符使用： %s 是一个占位符，表示一个字符串； %d 是一个占位符，表示一个整数（修改版在备注中体现）

50A Domino Piling

思路：简单整除

代码：

```
a, b =input().split()

space=int(a)*int(b)

print(int(space/2))
```

Contest status							
#	When	Who	Problem	Lang	Verdict	Time	Memory
280571598	Sep/10/2024 20:25UTC+8	lyralee	50A - Domino piling	Python 3	Accepted	216 ms	0 KB

自评：解决思路+语法调整20分钟，明显提速，上网学习了split函数，一开始没有意识到split处理后变量还是字符串形式

1A Theatre Square

思路：简单整除

代码：

```
n,m,a = input().split()
```

```

n = int(n)

m = int(m)

a = int(a)

n1 = n//a

m1 = m//a

if n%a == 0 and m%a == 0:

    print(n1*m1)

elif n%a != 0 and m%a == 0:

    print((n1+1)*m1)

elif n%a == 0 and m%a != 0:

    print(n1*(m1+1))

else:

    print((n1+1)*(m1+1))

```

#	When	Who	Problem	Lang	Verdict	Time	Memory
280672344	Sep/11/2024 09:46 UTC+8	lyralee	1A - Theatre Square	Python 3	Accepted	77 ms	0 KB

自评：耗时20分钟，还是整数字符串傻傻分不清，但是对速度提升较满意

112A

思路：官方：直接比较

本人：先把字符串分割标化处理 ➡ 转换为ascii码 ➡ 数字比大小

代码（个人）：

```

\#prepare for string_1

string_1 = input().lower()

split_1 = list(string_1)

ascii_1 = [ord(char) for char in split_1 ]


\#prepare for string_2

string_2 = input().lower()

split_2 = list(string_2)

```

```
ascii_2 = [ord(char) for char in split_2]
```

```
num=0
```

```
\#compare
```

```
for i in range(len(string_1)):
```

```
    if ascii_1[i] < ascii_2[i]:
```

```
        • print('-1')
```

```
        • break
```

```
    elif ascii_1[i] > ascii_2[i]:
```

```
        • print('1')
```

```
        • break
```

```
    else:
```

```
        • num += 1
```

```
if num == len(string_1):
```

```
    print('0')
```

280675686	Sep/11/2024 10:54 ^{UTC+8}	lyralee	112A - Petva and Strings	Python 3	Accepted	154 ms	0 KB
-----------	------------------------------------	---------	--------------------------	----------	----------	--------	------

自评：努力了，但是蠢蠢的，做的时候因为好多语法没见过，痛苦的差点想要放弃，好不容易做完一看官方答案：原来如此简单啊！更加痛苦了！

耗时五十分钟百分之九十九是因为自己的愚蠢，但是好歹是学了点东西：

l Python ascii 码转换 ord(i)

(其中i为单个字符！)

l 读取字符串长度 len(string)

l list函数 分割字符串成单个字母

或者列表循环 char for char in string

```
l range(0,i)=range(i)
```

231A team

思路：加就完了！

代码：

```

number = int(input())

n=0

for i in range(number):

    a,b,c = [int(s) for s in input().split()]

    if a + b + c > 1:

        n += 1

print(n)

```

ID	WHEN	WHO	PROBLEM	LANG	VERDICT	TIME	MEMORY
280719857	Sep/11/2024 19:10UTC+8	lyralee	231A - Team	Python 3	Accepted	154 ms	0 KB

自评：耗时十分钟，第一次写出和答案差不多的解答，很是激动

个人补充

01003 Hangover

思路：读取 ➡ 判断是否是0.00 ➡ 执行循环操作

代码：

```

while True:

    lengths = float(input())

    if lengths == 0.00:

        \# standard comparison: if math.isclose(n, 0.00, rel_tol=1e-5) : (import math
        first)

        break

    n = 0

    sum = 0

    while True:

        n += 1

        sum += 1/(n+1)

        if sum > lengths:

            break

    print(n, 'card(s)')

```

自评：耗时半个小时左右，循环套循环把自己整不会了

02808 树

思路：（一开始错误版）总-每次扣除数，没有考虑到重合

发现错误后改用集合，并集计算后总元素数-扣除元素数

代码：

```
L,M = input().split()

tree = {number for number in range(int(L)+1)}

cut = set()

for i in range(int(M)):

    a,b = input().split()

    area = {j for j in range(int(a), int(b)+1)}

    cut = cut.union(area)

result = len(tree)-len(cut)

print(result)
```

#46053805提交状态

状态: **Accepted**

源代码

```
L,M = input().split()
tree = {number for number in range(int(L)+1)}
cut = set()
for i in range(int(M)):
    a,b = input().split()
    area = {j for j in range(int(a), int(b)+1)}
    cut = cut.union(area)
result = len(tree)-len(cut)
print(result)
```

自评：耗时二十分钟，一开始错误思路的做法耗时五分钟不到，后来百度学习集合

几个要点：



Plus: 创建空集应该用set(), 不然会和字典混淆

02701

思路: 排除尾号7的 排除头是7的 (坑点) 排除7整除的

代码:

```
n = int(input())

result=[]

for i in range(n+1):

    if i%10 != 7 and i%7 != 0 and i//10 != 7:

        result.append(i**2)

print(sum(result))
```

状态: Accepted

源代码

```
n = int(input())
result=[]
for i in range(n+1):
    if i%10 != 7 and i%7 != 0 and i//10 != 7:
        result.append(i**2)
print(sum(result))
```

自评: 耗时十分钟, 一开始掉坑里了, 可恶!

03143 质数

思路：先定义质数，然后进入循环来判断

代码：

```
import math

def is_prime(n):

    for i in range(2, math.floor(math.sqrt(n)) + 1):

        if n % i == 0:

            return False

    return True

n = int(input())

if n < 6 or n % 2 != 0:

    print('Error!')

else:

    for j in range(2, n//2 + 1):

        if is_prime(j) and is_prime(n-j):

            print('%d=%d+%d' % (n, j, n-j))
```

状态: Accepted

源代码

```
import math

def is_prime(n):
    for i in range(2, math.floor(math.sqrt(n)) + 1):
        if n % i == 0:
            return False
    return True

n = int(input())
if n < 6 or n % 2 != 0:
    print('Error!')
else:
    for j in range(2, n//2 + 1):
        if is_prime(j) and is_prime(n-j):
            print('%d=%d+%d' % (n, j, n-j))
```

自评：一开始循环没套清楚，没有进行质数定义，于是获得评价：

你真的看懂你在干什么了吗

你看第八行，If I more, J等于等于0啊，那么只要这个I。不能被2整除，也就是 $M2 \neq 0$ ，它就可以进入第10行的else啊，直接进入第十行的else。Else完了之后， $K=N-I$ 第12的循环，也不知道在做什么东西，也不知道为什么是根号J，那个J为什么很很迷很迷好吧。完了之后啊，只要再一次的啊，只要这个K不能被2整除，也就是他是个奇数的话，那么就可以直接进入第15行的，Else, 并且把这个 $100=35+65$ 这种东西打印出来

还有我们说啊，判断素数那这是不是一个可以复用的啊，这样一个流程啊，就是去循环从二到他根号这个 $N+1$ 之间这个范围，它是否可以有可以有它的因数没有的话，就是素数这个流程是不是可以复用啊？那我们是不是可以把它写成一个函数的形式，就不用这样多层嵌套，然后你还嵌套错了这种问题

你这个不知道为什么错的时候，你就按照这个程序运行的逻辑，你去一行一行的对啊，跟人的逻辑去比对啊。对到某一行发现跟你想象的不对的时候，你就知道为什么你写错了

后来进行修改，耗时一个小时+（没有仔细计算过）

学习了：def函数使用

19944 今天星期几

思路：字符串读取 → 年份月份处理 → 计算

一开始想把年份分开来处理的，忽略了00的情况

代码：

```
n = int(input())

for i in range(n):

    data = input()

    year = int(data[:4])

    m = int(data[4:6])

    d = int(data[6: ])

    if m < 3:

        year -= 1

        m += 12

    c = int(str(year)[:2])

    y = int(str(year)[2:4])
```

```

calendar = (y + c//4 + y//4 - 2*c + (26*(m+1))//10 + d -1) % 7

d =
{0: 'Sunday', 1: 'Monday', 2: 'Tuesday', 3: 'Wednesday', 4: 'Thursday', 5: 'Friday', 6: 'Saturday'}

print(d[calendar])

```

状态: Accepted

源代码

```

n = int(input())

for i in range(n):
    data = input()
    year = int(data[:4])
    m = int(data[4:6])
    d = int(data[6:])
    if m < 3:
        year -= 1
        m += 12
    c = int(str(year)[2:])
    y = int(str(year)[2:4])
    calendar = (y + c//4 + y//4 - 2*c + (26*(m+1))//10 + d -1) % 7
    d = {0: 'Sunday', 1: 'Monday', 2: 'Tuesday', 3: 'Wednesday', 4: 'Thursday', 5: 'Friday', 6: 'Saturday'}
    print(d[calendar])

```

自评: 学习了字符串分割[:], 以及字典读取

语法还不算熟练 26*(m+1)一开始按数学思维没有加乘号

02689 大小写互换

思路: 全大写/全小写后比较决定保留

代码:

```

string = input()

ori = list(string)

low = list(string.lower())

up = list(string.upper())

result = []

for i in range(len(string)):

    if ori[i] == low[i]:

        result.append(up[i])

```

```
else:

    result.append(low[i])

print(''.join(result))
```

状态: Accepted

源代码

```
string = input()
ori = list(string)
low = list(string.lower())
up = list(string.upper())
result = []
for i in range(len(string)):
    if ori[i] == low[i]:
        result.append(up[i])
    else:
        result.append(low[i])
print(''.join(result))
```

自评: 五分钟, 简单题

Join函数巩固

(upd:后来才发现 一个swapcase秒杀啊!)

19449 密码

思路: (没思路于是直接看答案) 为合并有效内容 把中间为空格的###删去, 在进行计算

状态: Accepted

源代码

```
n = int(input())
entity = 0
for i in range(n):
    ask = input().replace('###', ' ')
    result = ask.count('###')//2
    entity += result
print(entity)
```

03248

思路: 直接套用math自带函数

代码:

```
[import math]()

while True:

    try:

        • a,b = [int(number) for number in input().split()]

        • print(math.gcd(a,b))

    except EOFError:

        • break
```

状态: Accepted

源代码

```
import math
while True:
    try:
        a,b = [int(number) for number in input().split()]
        print(math.gcd(a,b))
    except EOFError:
        break
```

自评：一开始一直超时，看了标答后发现是测试集无法判断何时结束，于是添加框架结构

```
while(True):

    try:

        • ...

    except EOFError:

        • break
```

21532 数学密码

思路：排除暴力计算，最好的就是直接找到输入数字的最大因数，且输入数/最大因数>6

一开始超时，经过提醒发现从后往前筛选更加高效，排除比较增加过程

代码：

```
import math

n = int(input())

for i in range(n//6,0,-1):

    if n % i == 0:

        • print(i)

        • break
```

自我评价：

（几乎）零基础学生，一开始在语法上面花了很大功夫，但是没有看教材，主打一个跟着题目学习，做完作业做每日一题，遇到不会的语法就是求助百度和高中同学（笑

感觉作业还是以考察逻辑思维为主（比如数学密码，在了解了基本简化原理后还要想到从后往前才能避免超时），虽然目前还不熟练，但是可以感受到逐渐提速的过程，很有成就感。