

Assignment #1: 自主学习

Updated 0110 GMT+8 Sep 10, 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

02733: 判断闰年

<http://cs101.openjudge.cn/practice/02733/>

思路：年份有是或者不是闰年两种情况，所以可以判断需要使用if语句，满足条件Y，不满足N。因为不会所以先根据感觉写了一下然后让AI改一改bug现学了一下语法（x）。是闰年则需满足能被4整除且不能被100整除or能被400整除且不能被3200整除。因为测试数据小于3000，所以被3200整除的情况可以不考虑！

大致用时：20min

题解启示：可以先排除能被4整除但不是闰年的所有特殊情况，print("N")，之后能被4整除的就都是闰年了

代码

```
#判断闰年
#能被4整除的大多是闰年，但能被100整除而不能被400整除的年份不是闰年， 能被3200整除的也不是闰年，
如1900年是平年，2000年是闰年，3200年不是闰年。

a=int(input())
if (a%4==0 and a%100!=0) or (a%400==0):
    print("Y")
else:
    print("N")
```

代码运行截图

#46088618提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
#判断闰年
#能被4整除的大多是闰年，但能被100整除而不能被400整除的年份不是闰年， 能被3200整除的

a=int(input())
if (a%4==0 and a%100!=0) or (a%400==0):
    print("Y")
else:
    print("N")
```

基本信息

#: 46088618

题目: 02733

提交人: 胡杨

内存: 3588kB

时间: 22ms

语言: Python3

提交时间: 2024-09-15 18:47:51

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

02750: 鸡兔同笼

<http://cs101.openjudge.cn/practice/02750/>

思路：首先，如果输入数字为奇数，则不可能，输出0 0；如果输入数为偶数，则不管鸡和兔子各有几个，这种情况都是可能存在的。输入数为偶数时，此时动物最多的情况是笼子里全是鸡，即可用a/2算出；动物最少的情況是全是兔子，但a//4不一定整除。当a//4==0时，a//4的结果即为动物最少的情況，全是兔子；当a//4!=0时，此时动物最少的情況时一堆兔子里有一只鸡，即a//4+1.python默认的輸出方式就是隔一个空格，不用特別修改。

大致用时：15min

代码

```
#鸡兔同笼

a=int(input())
if a%2==0:
    if a%4==0:
        b=int(a//4)
        c=int(a//2)
        print(b,c)
    else:
        b=int(a//4+1)
        c=int(a//2)
        print(b,c)
else:
    print(0,0)
```

代码运行截图

状态: **Accepted**

源代码

```
#鸡兔同笼

a=int(input())
if a%2==0:
    if a%4==0:
        b=int(a//4)
        c=int(a//2)
        print(b,c)
    else:
        b=int(a//4+1)
        c=int(a//2)
        print(b,c)
else:
    print(0,0)
```

基本信息

#: 46089116
题目: 02750
提交人: 胡杨
内存: 3616kB
时间: 24ms
语言: Python3
提交时间: 2024-09-15 19:30:31

50A. Domino piling

greedy, math, 800, <http://codeforces.com/problemset/problem/50/A>

思路：棋盘有两种情况：长和宽中存在一条的格子数为偶数（N或M为偶数），或者长和宽都是奇数（MN都是奇数）。当MN有一个为偶数时，可以用多米诺骨牌中长度为2的边填满它，另一边长度为1，也就是竖着铺，必定可以填满整个棋盘。当MN都是奇数时，可以先按前面的方法填棋盘，最后一排把多米诺骨牌横过来，最后只剩一个空格。

ps.map函数语法为map(function,iterable)，将函数function作用于列表或元组等iterable中的所有对象

大致耗时：20min

题解启示：M, N = [int(x) for x in input().split()]和M,N= map(int,input().split())作用一致。int(x) for x in ...即将...中得到的数转成整型

代码

```
#
M,N= map(int,input().split())
a = M * N // 2
print(a)
```

代码运行截图

#	When	Who	Problem	Lang	Verdict	Time	Memory
281460221	Sep/16/2024 17:08 ^{UTC+8}	itseventeen	50A - Domino tiling	Python 3	Accepted	184 ms	0 KB

1A. Theatre Square

math, 1000, <https://codeforces.com/problemset/problem/1/A>

思路：填广场的是正方形所以不用考虑旋转的情况，只要保证 $xa \geq n, ya \geq m$ 就行。

大致耗时：10min

题解启示：**ceil () 函数可以向上取整**，可以简化对于是否整除的讨论（整除向下取整不用加一，不整除向下取整要加一）。用法是先**import math**,然后**math.ceil()**

代码

```
#
n,m,a=map(int,input().split())
if n%a!=0 and m%a!=0:
    x=n//a+1
    y=m//a+1
    print(x*y)
elif n%a!=0 and m%a==0:
    x=n//a+1
    y=m//a
    print(x*y)
elif n%a==0 and m%a!=0:
    x=n//a
    y=m//a+1
    print(x*y)
else:
    x=n//a
    y=m//a
    print(x*y)
```

代码运行截图

Contest status

#	When	Who	Problem	Lang	Verdict	Time	Memory
281461656	Sep/16/2024 17:23 UTC+8	ltseventeen	1A - Theatre Square	Python 3	Accepted	77 ms	0 KB
281460221	Sep/16/2024 17:08 UTC+8	ltseventeen	50A - Domino piling	Python 3	Accepted	184 ms	0 KB

Sort by:

☐ Default order ☒ Submission time ☐ Judging Time ☐ Solution Size ☐ Execution Time

← 1 2 3 4 5 →

112A. Petya and Strings

implementation, strings, 1000, <http://codeforces.com/problemset/problem/112/A>

思路：测试输入的字符串需要调整大小写，我选择全部lower小写。查询维基百科了解字典顺序感觉好复杂没看懂，问了一下AI原来字符串大小比较python是自带的，可以直接进行大小比较。

大概耗时：15min

题解启示：（）表判断，返回布尔值，**布尔值可以直接加减**，**true=1, false=0**，可以简化代码如下：

```
i=input;a=i().lower();b=i().lower()
print((a>b) - (a<b))
```

代码

```
#读取输入的两个字符串并转化为小写
str1=input().lower()
str2=input().lower()

#判断两个字符串的大小关系，字典顺序在python里是内置自带的，根据ASCII码进行比较
if str1<str2:
    print(-1)
elif str1>str2:
    print(1)
else:
    print(0)
```

代码运行截图

Contest status							
#	When	Who	Problem	Lang	Verdict	Time	Memory
281463217	Sep/16/2024 17:38 UTC+8	Itseventeen	112A - Petya and Strings	Python 3	Accepted	186 ms	600 KB
281461656	Sep/16/2024 17:23 UTC+8	Itseventeen	1A - Theatre Square	Python 3	Accepted	77 ms	0 KB
281460221	Sep/16/2024 17:08 UTC+8	Itseventeen	50A - Domino piling	Python 3	Accepted	184 ms	0 KB

Sort by:

☐ Default order☒ Submission time☐ Judging Time☐ Solution Size☐ Execution Time

← 1 2 3 4 5 →

231A. Team

bruteforce, greedy, 800, <http://codeforces.com/problemset/problem/231/A>

思路：感觉这道题的主要难点是输入，怎么读取题数以及每道题的三个0或1，然后遍历每道题看是不是有大于等于2个1就可以了。题数可以用`n=int(input())`解决，至于每道题的0和1怎么读取我显然是不会的，于是复制题目让AI教我，学习了一下使用map函数，`map(function,iterable)`，注意用逗号连接函数和后面要作用的值！

大概用时：30min

题解启示：`print()`也很耗时间，如果要多次输出，可以在`print()`里套循环，减少调用次数。但是这种方法比较绕，不好理解语句，目前暂不考虑。

代码

```
n=int(input())#输入题目数量

solved_problems=0#记录已解决题目的数量

for i in range(n):#遍历题目（0-（n-1））
    a,b,c=map(int,input().strip().split())#每排的0或1,map的作用是将输入的字符串转化为int类型，语法为map（函数，序列）
    if a+b+c>=2:
        solved_problems+=1#如果题目有两个以上1，则加1
    else:
        solved_problems+=0#否则不加，也可以用continue！！3

print(solved_problems)#输出已解决题目的数量
```

代码运行截图

Contest status							
#	When	Who	Problem	Lang	Verdict	Time	Memory
281752367	Sep/18/2024 22:46 ^{UTC+8}	ltseventeen	231A - Team	Python 3	Accepted	154 ms	0 KB
281463217	Sep/16/2024 17:38 ^{UTC+8}	ltseventeen	112A - Petya and Strings	Python 3	Accepted	186 ms	600 KB
281461656	Sep/16/2024 17:23 ^{UTC+8}	ltseventeen	1A - Theatre Square	Python 3	Accepted	77 ms	0 KB
281460221	Sep/16/2024 17:08 ^{UTC+8}	ltseventeen	50A - Domino piling	Python 3	Accepted	184 ms	0 KB

Sort by:

☐ Default order ☒ Submission time ☐ Judging Time ☐ Solution Size ☐ Execution Time

← 1 2 3 4 5 →

2. 学习总结和收获

一些学到的常用语句总结：

1.输入：

- `a=int(input())` #将输入的数据转化为整型
- `a,b,c=map(int,input().split())`
- `a,b,c=[int(x) for x in input().split()]` #这两句的作用一致，都是将输入的字符串按空格切分并且转化为整型。第一句使用map函数，语法为map(function,iterable)，第二句使用for循环，语法为for _ in ...

注意：#不要写split(")，换成split()，具体语法如下图



在 Python 中，split() 方法不写参数时，默认会按照空白字符（whitespace）进行切分。这里的空白字符不仅包括空格 (' '), 还包括制表符 ('\t')、换行符 ('\n') 等所有的空白符。

此外，默认情况下，split() 会自动忽略字符串开头和结尾的空白符，并且不会把相邻的多个空白符当作单独的分隔符，而是会将它们视为一个分隔符。

举个例子：

python

Copy

```
text = "  Hello  world\tthis is\nPython  "
words = text.split()
print(words)
```

输出将是：

Copy

```
['Hello', 'world', 'this', 'is', 'Python']
```

可以看到，多个连续的空白符（如多个空格、制表符、换行符）都被视为一个分隔符。



#不要在input里面加输入提示语（输入提示语用于和人交互，但OJ平台不需要交互）

2.if语句：

```
if 条件1:
    #如果条件1为真，执行这段代码
elif 条件2:
    #如果条件1为假，条件2为真，执行这段代码
else:
    #如果条件1和条件2都为假，执行这段代码
```

注意：if后直接跟条件，可以用（）和and/or等词语使条件更明确，**句尾的“:”不要忘记**

3.for循环:

后面是AI教的各种用法，copy过来以防忘记

基本用法

遍历列表

```
fruits = ["apple", "banana", "cherry"]
for fruit in fruits:
    print(fruit)
```

遍历字符串

```
message = "Hello, world!"
for char in message:
    print(char)
```

使用 range() 函数

range() 函数生成一个整数序列，常用于需要根据索引进行操作的场景。

生成简单的整数序列

```
for i in range(5): # 生成 0 到 4 的整数
    print(i)
```

指定起始值和结束值

```
for i in range(1, 6): # 生成 1 到 5 的整数
    print(i)
```

指定步长

```
for i in range(0, 10, 2): # 生成 0 到 9 的偶数
    print(i)
```

遍历字典

字典的 for 循环可以遍历键、值或键值对。

遍历键

```
person = {"name": "Alice", "age": 30, "city": "New York"}
for key in person:
    print(key)
```

遍历值

```
for value in person.values():
    print(value)
```

遍历键值对

```
for key, value in person.items():
    print(f"{key}: {value}")
```

ps. **格式化字符串 (f-string)**: `f"{key}: {value}"` 是一种格式化字符串, 前面的 `f` 表示这是一个格式化字符串, 可以直接在字符串中嵌入变量。

使用 `enumerate()` 函数

`enumerate()` 函数可以在遍历序列时同时获取元素的索引和值。

```
fruits = ["apple", "banana", "cherry"]
for index, fruit in enumerate(fruits):
    print(f"Index: {index}, Fruit: {fruit}")
```

其输出为

```
Index: 0, Fruit: apple
Index: 1, Fruit: banana
Index: 2, Fruit: cherry
```

使用 `zip()` 函数

`zip()` 函数可以将多个序列合并为一个元组序列, 常用于并行遍历多个序列。

```
names = ["Alice", "Bob", "Charlie"]
ages = [25, 30, 35]
for name, age in zip(names, ages):
    print(f"{name} is {age} years old")
```

嵌套 `for` 循环

嵌套 `for` 循环用于处理多层结构的数据。

```
matrix = [
    [1, 2, 3],
    [4, 5, 6],
    [7, 8, 9]
]

for row in matrix:
    for element in row:
        print(element, end=" ")
    print() # 换行
```

其输出为:

```
1 2 3
4 5 6
7 8 9
```

else 子句

for 循环可以有一个 else 子句，当循环正常结束（即没有被 break 语句中断）时执行。

```
fruits = ["apple", "banana", "cherry"]
for fruit in fruits:
    if fruit == "orange":
        break
else:
    print("Orange not found")
```

列表推导式

列表推导式是一种简洁的创建列表的方法，内部使用了 for 循环。

```
squares = [x**2 for x in range(10)]
print(squares)
#output:[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
```

4.ceil函数：

可以用来向上取整

```
import math
math.ceil()
```

5.布尔值可以直接加减，True=1，False=0

其他收获和总结：

零基础第一次编程，基本上都是在AI的指导下完成的，经常出现知道思路但是不知道如何用计算机语言表达或者语法错误的情况。我认为这种情况还是需要多加练习熟能生巧，将自学与做题相结合。自己看书和看网站都相当痛苦并且一点也没记住，但是好像在做题中反而掌握了一些内容。决定后面尝试一下**站看网课加做题**的学习方法，但是又担心遗漏掉什么知识（.....）但是能每天学会一点新东西已经很不错了！不要焦虑！先活着再说（.....）

其他的练习：每日题02689, 02701, 4A

菜菜的，做得也少少的，学的也慢慢的，别的课还有好多，好崩溃