

Assignment #1: 拉齐大家Python水平

Updated 0940 GMT+8 Feb 19, 2024

2024 spring, Complied by 刘子暄 环境科学与工程学院

说明:

- 1) 数算课程的先修课是计概,由于计概学习中可能使用了不同的编程语言,而数算课程要求 Python语言,因此第一周作业练习Python编程。如果有同学坚持使用C/C++,也可以,但是建议也要会Python语言。
- 2)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3月1日导入选课名单后启用。作业写好后,保留在自己手中,待3月1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。

编程环境

操作系统: Windows 11

Python编程环境: PyCharm Community Edition 2023.3

1. 题目

20742: 泰波拿契數

http://cs101.openjudge.cn/practice/20742/

思路:

代码

```
def tribonacci(n: int) -> int:
    dp = {i:0 for i in range(n+1)}
    dp[1] = dp[2] = 1
    if n <= 2:
        return dp[n]
    for i in range(3, n + 1):
        dp[i] = dp[i - 1] + dp[i - 2] + dp[i - 3]
    return dp[n]

n = int(input())
print(tribonacci(n))</pre>
```

代码运行截图 (至少包含有"Accepted")



58A. Chat room

greedy/strings, 1000, http://codeforces.com/problemset/problem/58/A

思路:

代码

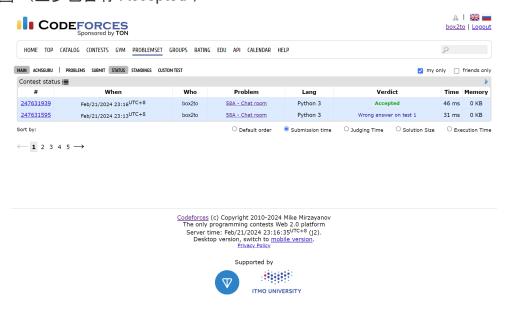
```
str0 = input()
str0 = str0.lower()

a = "hello"
lis = [0]*5
co1 = 0

for i in str0:
    if i == a[co1]:
        lis[co1] += 1
        co1 += 1
    if co1 == 5:
        break

if sum(lis) == 5:
    print('YES')
else:
    print('NO')
```

代码运行截图 (至少包含有"Accepted")



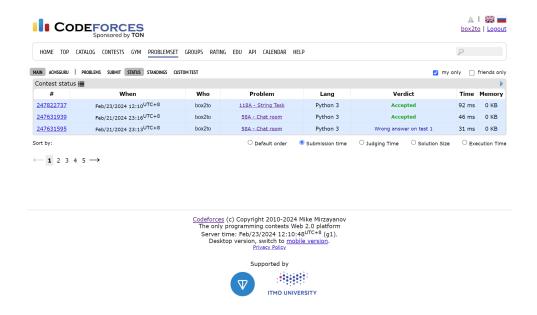
118A. String Task

implementation/strings, 1000, http://codeforces.com/problemset/problem/118/A

思路:将输入数据以列表形式存储后遍历并删除重复元素、输出时添加符号

代码

代码运行截图 (AC代码截图,至少包含有"Accepted")



(Ai)

22359: Goldbach Conjecture

http://cs101.openjudge.cn/practice/22359/

思路:找出n以下的所有质数,循环嵌套(循环时第二个循环注意限制范围,同时将相同数去除)

```
def prime(n):
    for i in range(2,int(n**0.5) + 1):
        if n % i == 0:
            return None
    return n
n = int(input())
lis = []
for i in range(n):
    if prime(i) != None:
        lis.append(i)
count = []
for i1 in range(len(lis)):
   for i2 in range(i1,len(lis)):
        if lis[i1] + lis[i2] == n and lis[i1] != lis[i2]:
            count.append(lis[i1])
            count.append(lis[i2])
for i3 in count:
    print(i3,end=' ')
```

代码运行截图 (AC代码截图, 至少包含有"Accepted")



23563: 多项式时间复杂度

http://cs101.openjudge.cn/practice/23563/

思路:

代码

```
lis = [x for x in input().split('+')]

comp = []
for i in lis:
    if i[0] != '0':
        for s in range(len(i)):
        if i[s] == 'n':
            a = i[s+2:]
            comp.append(int(a))

print(f"n^{max(comp)}")
```

代码运行截图 (AC代码截图,至少包含有"Accepted")



Ai

24684: 直播计票

http://cs101.openjudge.cn/practice/24684/

思路:

```
s = [int(x) for x in input().split()]
ss = []
counts = []
co = []

for i in s:
    if i not in ss:
        ss.append(i)

s1 = sorted(ss)

for i1 in s1:
    counts.append(s.count(i1))
for i2 in range(len(counts)):
    if counts[i2] == max(counts):
        co.append(s1[i2])

for i3 in co:
    print(i3,end=' ')
```

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

####A.chat room

我的思路是想要将输入信息中重复字符删除之后再去跟"hello"做顺序正确的对应,但是实现过程感觉太麻烦了,就去看了优秀答案

```
s = input()
s = s.lower()
dp = [0]*5
data = 'hello'
cnt = 0
for c in s:
    if c == data[cnt]:
       dp[cnt] += 1
        cnt += 1
    if cnt == 5:
        break
if sum(dp) == 5:
   print('YES')
else:
   print('NO')
```

```
import re
s = input()
r = re.search('h.*e.*l.*l.*o', s)
print(['YES', 'NO'][r==None])
```

答案中第一解法 (我的理解)

1.输入,统一大小写(大小写不影响实际结果,这个特殊情况例题中没有,我就没注意,提醒自己)

2.遍历输入数据,与"hello"——对应。

我的思路和这个其实完全一致,但是没有想到简洁的实现方法,原因是我在纠结是否需要在发现 不符情况后直接中断判断,但是这实际上是多余的,提醒自己

该答案将需要统计的内容分为两部分:hello是否被全部对应(从要查的到hello)(但是这里的 break是用来防止out of range的,没有强制退出的需要),hello有没有被多次对应(这里用一维 表对应位次来计数很对,因为一维表出现了顺序,同时又计算了重复与否,牛的)以及顺序对应(索引用法,和我想的是一样的)

第二解法引入了re库,使用search方法 re.search(pattern, string, flags=0) pattern模式串(原来的串) string原始串(需要被判断的串) flag调整匹配方式的,这里暂时不需要

特殊元素用法

模式	描述
٨	匹配字符串的开头
\$	匹配字符串的末尾。
	匹配任意字符,除了换行符,当re.DOTALL标记被指定时,则可以匹配包括换行符的任意字符。
[]	用来表示一组字符,单独列出:[amk] 匹配 'a', 'm'或'k'
[^]	不在[]中的字符: [^abc] 匹配除了a,b,c之外的字符。
re*	匹配0个或多个的表达式。
re+	匹配1个或多个的表达式。

(re表示正则表达式)

原程序中'h.8'就表示匹配任意字符+不限数量

(print ([a,b][r==none]) 是判断语句,后条件满足则返回b,负责返回a)

####总结

但凡有字符串匹配类的,可以沿用第一种小修小改,或者第二种暴力破解

####B.string task

做题时发现for循环遍历是以列表下标作为循环顺序的,而每一次remove都在缩短列表长度,使得for循环会漏掉被删除元素的下一位

解决方法是以列表元素个数为循环次数对列表多次循环,因为发生列表更新的总量不会超过元素数,保证每次更新的列表都被循环过,不出现遗漏。