# Assignment #1: 拉齐大家Python水平

Updated 0940 GMT+8 Feb 19, 2024

2024 spring, Complied by 李佳霖、心理与认知科学学院

#### 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 <a href="https://typoraio.cn">https://typoraio.cn</a>, 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, <a href="https://pku.instructure.com">https://pku.instructure.com</a>, 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: macOS Sonoma 14.2.2 (c)

Python编程环境: Vscode 1.86.2

### 1. 题目

### 20742: 泰波拿契數

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

思路: 用一个列表用于存放结果, 接受输入后直接在列表中查找结果

```
# 泰波拿契數
n = int(input())
res = [0,1,1]
for i in range(100):
    res.append(res[-1]+res[-2]+res[-3])
print(res[n])
```

代码运行截图 <mark>(至少包含有"Accepted")</mark>

#43939293提交状态 查看 提交 统计

状态: Accepted

```
      源代码
      #: 43939293

      m = int(input())
      题目: 20742

      res = [0,1,1]
      提交人: 李佳霖2000013713

      for i in range(100):
      内存: 3596kB

      res.append(res[-1]+res[-2]+res[-3])
      时间: 21ms

      print(res[n])
      语言: Python3

      提交时间: 2024-02-20 16:06:01
```

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

基本信息

提问

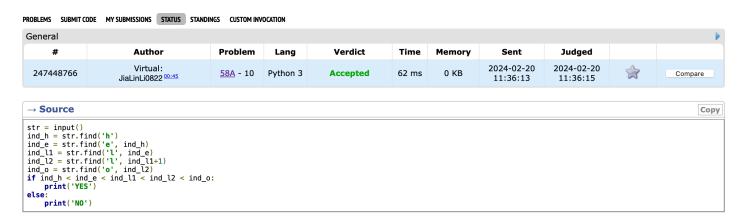
### 58A. Chat room

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

思路:用str.find函数查找每一个字母的位置。然后保证字母按顺序排列即可。

```
# Chat room
str = input()
ind_h = str.find('h')
ind_e = str.find('e', ind_h)
ind_l1 = str.find('l', ind_e)
ind_l2 = str.find('l', ind_l1+1)
ind_o = str.find('o', ind_l2)
if ind_h < ind_e < ind_l1 < ind_l2 < ind_o:
    print('YES')
else:
    print('NO')</pre>
```

### 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:

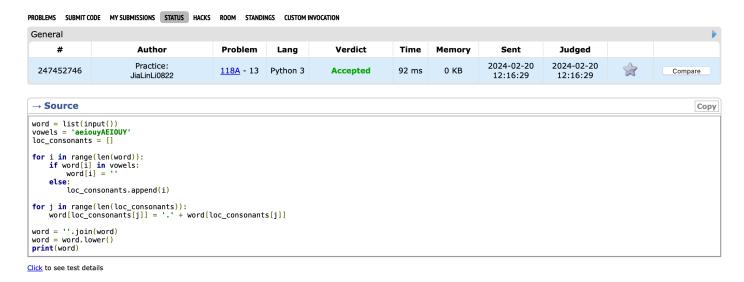
```
# String Task
word = list(input())
vowels = 'aeiouyAEIOUY'
loc_consonants = []

for i in range(len(word)):
   if word[i] in vowels:
```

```
word[i] = ''
else:
    loc_consonants.append(i)

for j in range(len(loc_consonants)):
    word[loc_consonants[j]] = '.' + word[loc_consonants[j]]

word = ''.join(word)
word = word.lower()
print(word)
```



## 22359: Goldbach Conjecture

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

思路:感觉这道题更多的涉及一些数学技巧,例如为什么只用找到平方根前:如果一个数 n可以被分解成两个因子 a 和 b,其中  $a \leq b$ ,那么这两个因子中必然有一个小于或等于  $\sqrt{n}$ ,而另一个大于或等于  $\sqrt{n}$ 

```
# Goldbach Conjecture
def is_prime(n):
    if n <= 1:
        return False
    for i in range(2, int(n**0.5) + 1): #只用找到平方根前就可以了</pre>
```

```
if n % i == 0:
    return False
return True

def find_primes(sum):
    for i in range(2, sum // 2 + 1): # only need to find the prime up to N/2
        if is_prime(i) and is_prime(sum - i):
            return i, sum - i
    return None

sum = int(input())
result = find_primes(sum)
print(result[0], result[1])
```

#43940888提交状态

查看 提交 统计 提问

#### 状态: Accepted

```
基本信息
源代码
                                                                                 #: 43940888
                                                                               题目: 22359
 def is prime(n):
                                                                              提交人: 李佳霖2000013713
     if n <= 1:
                                                                               内存: 3568kB
         return False
     for i in range(2, int(n**0.5) + 1):
                                                                               时间: 18ms
         if n % i == 0:
                                                                               语言: Python3
             return False
                                                                            提交时间: 2024-02-20 17:42:32
     return True
 def find_primes(sum):
     for i in range(2, sum // 2 + 1):
         if is_prime(i) and is_prime(sum - i):
            return i, sum - i
     return None
 sum = int(input())
 result = find_primes(sum)
 print(result[0], result[1])
©2002-2022 POJ 京ICP备20010980号-1
                                                                                               English 帮助 关于
```

### 23563: 多项式时间复杂度

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

思路:先通过字符串拆分,把项数和幂次存放到两个list当中,之后通过循环的方式找到最大的幂次并输出

· · · --

```
# 多项式时间复杂度
n = input().split('+')
items = []
num = []
power = []
# Put the numbers and powers into separate lists
for i in range(len(n)):
    items.append(n[i].split('n^'))
    if items[i][0] == '':
        num.append(1)
    else:
        num.append(int(items[i][0]))
    power.append(int(items[i][1]))
# Find the maximum power
maxp = -99
while True:
   maxp = max(power)
    if num[power.index(maxp)] !=0:
        break
    else:
        power[power.index(maxp)] = -99
print('n^' + str(maxp), end='')
```

#**43959754提交状态** 查看 提交 统计 提问

基本信息

### 状态: Accepted

```
源代码
                                                                                  #: 43959754
                                                                                题目: 23563
 n = input().split('+')
                                                                               提交人: 李佳霖2000013713
 items = []
                                                                                内存: 3644kB
 num = []
                                                                                时间: 22ms
 power = []
                                                                                语言: Python3
 # Put the numbers and powers into separate lists
                                                                             提交时间: 2024-02-22 15:56:56
 for i in range(len(n)):
     items.append(n[i].split('n^'))
     if items[i][0] == '':
        num.append(1)
        num.append(int(items[i][0]))
     power.append(int(items[i][1]))
 # Find the maximum power
 maxp = -99
 while True:
     maxp = max (power)
     if num[power.index(maxp)] !=0:
         power[power.index(maxp)] = -99
 print('n^' + str(maxp), end='')
©2002-2022 POJ 京ICP备20010980号-1
                                                                                                English 帮助 关于
```

### 24684: 直播计票

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

思路:通过字典来统计每一个元素的次数,通过max函数找到最大的频次,最后找出所有投票相同的元素并输出。

```
mumbers = list(map(int, input().split()))
choices = set(numbers)
frequency = {}
for i in range(len(choices)):
    frequency[list(choices)[i]] = 0

for i in range(len(numbers)):
    frequency[numbers[i]] += 1

max_frequency = max(frequency.values())
```

```
# 按要求输出所有得票最多的选项

print_list = [option for option, freq in frequency.items() if freq == max_frequency]

print_list = sorted(print_list)

print_list = [str(_) for _ in print_list]

print(' '.join(print_list))
```

状态: Accepted

```
基本信息
                                                                            #: 43960336
源代码
                                                                          题目: 24684
 # 读取输入
                                                                         提交人: 李佳霖2000013713
 numbers = list(map(int, input().split()))
                                                                          内存: 15332kB
 # 初始化选项出现次数的字典
                                                                          时间: 84ms
 frequency = {}
                                                                          语言: Python3
                                                                       提交时间: 2024-02-22 16:32:39
 # 统计每个选项的出现次数并找到最大的得票数
 max\_frequency = 0
 for number in numbers:
    frequency[number] = frequency.get(number, 0) + 1
    max frequency = max(max frequency, frequency[number])
 # 构建得票最多的选项列表(按编号从小到大排序)
 print_list = [str(option) for option in sorted(frequency.keys()) if free
 # 输出结果
 print(' '.join(print_list))
©2002-2022 POJ 京ICP备20010980号-1
                                                                                         English 帮助 关于
```

## 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"数算pre每日选做"、CF、LeetCode、洛谷等网站题目。

大二的时候选过闫老师的计算概论,这次有幸在毕业之前又修到了闫老师的数算课,一切都是熟悉的味道。因为许久没有写python代码,已经对语法产生了一些陌生感,这次作业算是帮我回顾了一下python基本的语法知识。因为大四课程压力比较小,所以可以放在这门课的时间也相应增加,自己也会在Leetcode上多刷一些题目,进一步巩固一下自己的python知识。