

Assignment #1: 拉齐大家Python水平

Updated 0940 GMT+8 Feb 19, 2024

2024 spring, Compiled by 尹柚鑫 光华管理学院 2100015878

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

==（请改为同学的操作系统、编程环境等）==

操作系统：windows11

Python编程环境：Python 3.11 Jupyter notebook

1. 题目

20742: 泰波拿契數

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

耗时：5 mins

思路：使用递归的想法计算，只需要计算一个n，不需要记忆计算过程

代码

```
#
index_tmp=int(input())
def tabo(index):
    if index==1:
        return(int(1))
    elif index==2:
        return(int(1))
    elif index==3:
        return(int(2))
    else:
        return tabo(index-1)+tabo(index-2)+tabo(index-3)
print(tabo(index_tmp))
```

代码运行截图 == (至少包含有"Accepted") ==

状态: Accepted

源代码

```
index_tmp=int(input())
def tabo(index):
    if index==1:
        return(int(1))
    elif index==2:
        return(int(1))
    elif index==3:
        return(int(2))
    else:
        return tabo(index-1)+tabo(index-2)+tabo(index-3)
print(tabo(index_tmp))
```

基本信息

#: 43985656
题目: 20742
提交人: 尹柚鑫(2100015878)
内存: 3612kB
时间: 424ms
语言: Python3
提交时间: 2024-02-25 15:57:30

58A. Chat room

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

耗时: 7 mins

思路: 使用递归的方法, 依次查找'hello'的每个字母是否出现

代码

```
#
string=input()
def searchfunc(string,i=0):
    string_list=list(string)
    hello_list=list('hello')
    ishello=True
    while i<=4:
        if hello_list[i] in string_list:
            index=string_list.index(hello_list[i])
            i=i+1
            return(searchfunc(string_list[index+1:],i))
        else:
            ishello=False
```

```

        return 'NO'

    if i==5:
        return 'YES'

print(searchfunc(string))

```

代码运行截图 == (至少包含有"Accepted") ==

Contest status							
#	When	Who	Problem	Lang	Verdict	Time	Memory
248206470	Feb/25/2024 16:37 UTC+8	YouxinYin	58A - Chat room	Python 3	Accepted	46 ms	0 KB

118A. String Task

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

耗时: 5 mins

思路: 按照题目要求依次对字符串进行操作即可

代码

```

#
string=input()
v_list=["A","O","Y","E","U","I","a","o","y","e","u","i"]
new_string_list=''.join([word for word in string if word not in v_list])
new_string_list=new_string_list.lower()
result_string = ''.join(['.' + char for char in new_string_list])
print(result_string)

```

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

#	When	Who	Problem	Lang	Verdict	Time	Memory
248315471	Feb/26/2024 15:39 UTC+8	YouxinYin	118A - String Task	Python 3	Accepted	92 ms	0 KB

22359: Goldbach Conjecture

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

耗时: 5 mins

思路: 从0到n遍历, 之后用最朴素的判断素数的方法即可

代码

```

#
def is_prime(number):
    if number <= 1:
        return False

```

```

for i in range(2, int(number**0.5) + 1):
    if number % i == 0:
        return False
return True

def find_prime(sum):
    for i in range(2, sum//2+1):
        if is_prime(i) and is_prime(sum-i):
            return i, sum-i

int_sum=int(input())
first,second=find_prime(int_sum)
print(f"{first} {second}")

```

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

#43990544提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```

def is_prime(number):
    if number <= 1:
        return False
    for i in range(2, int(number**0.5) + 1):
        if number % i == 0:
            return False
    return True

def find_prime(sum):
    for i in range(2, sum//2+1):
        if is_prime(i) and is_prime(sum-i):
            return i, sum-i

int_sum=int(input())
first,second=find_prime(int_sum)
print(f"{first} {second}")

```

基本信息

#: 43990544
 题目: 22359
 提交人: 尹柚鑫(2100015878)
 内存: 3808kB
 时间: 22ms
 语言: Python3
 提交时间: 2024-02-26 15:51:21

©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/>

耗时: 7mins

思路: 把原有字符串按照"^"和"+"分割, 之后进行处理

代码

```
#
import re
input_string = input()
result = re.split('\^+', input_string)

for i in range(len(result)-2,-1,-2):
    if result[i]=='0n':
        del result[i+1]
        del result[i]
jie_list=[int(i) for i in result if "n" not in i]
print(f"n^{int(max(jie_list))}")
```

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

状态: Accepted

源代码

```
import re
input_string = input()
result = re.split('\^+', input_string)

for i in range(len(result)-2,-1,-2):
    if result[i]=='0n':
        del result[i+1]
        del result[i]
jie_list=[int(i) for i in result if "n" not in i]
print(f"n^{int(max(jie_list))}")
```

基本信息

#: 43991032
题目: 23563
提交人: 尹柚鑫(2100015878)
内存: 3852kB
时间: 25ms
语言: Python3
提交时间: 2024-02-26 17:16:22

24684: 直播计票

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

耗时: 10mins

思路: 使用词典计数

代码

```
#
vote=input()
vote_list=vote.split(' ')
dic={}
for i in vote_list:
    if i in dic:
        dic[i]=dic[i]+1
    else:
        dic[i]=1
max_value=max(dic.values())
final_list=[key for key,value in dic.items() if value==max_value]
final_list.sort(key=int)
string=' '.join(final_list)
```

```
print(string)
```

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

状态: **Accepted**

源代码

```
vote=input()
vote_list=vote.split(' ')
dic={}
for i in vote_list:
    if i in dic:
        dic[i]=dic[i]+1
    else:
        dic[i]=1
max_value=max(dic.values())
final_list=[key for key,value in dic.items() if value==max_value]
final_list.sort(key=int)
string=' '.join(final_list)
print(string)
```

基本信息

#: 43991150
题目: 24684
提交人: 尹柚鑫(2100015878)
内存: 11376kB
时间: 49ms
语言: Python3
提交时间: 2024-02-26 17:40:11

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。==

我是光华的学生, 之前只上过计概C, 在算法类的课程上学的很少, 数算B的课程对我也有一定的挑战, 我想在这里记录写每次作业的过程中, 遇到的比较好的编程思路或者是python小技巧。

1.这次作业前两题都可以用递归很简单就解决, 我觉得写递归一定要注意三个点, 第一, 大规模的题解建立在小规模题解的基础上(这样才能用递归解决), 第二, 想清楚大规模题解怎么样变成更小规模的, 第三, 想清楚结束条件是什么。

2.第三题, 让我想到, 如果题目解答需要我建立一个26个英文字母大小写的列表或者词典, 可以采用ASCII码, 可以方便快捷地得到这样的列表, 不用手动输入

3.列表解析式写起来非常方便快捷, 可以多用, 下面这两个都可以参考

```
[word for word in string if word not in v_list]
```

```
['.' + char for char in new_string_list]
```

4.print(f'{first} {second}'), 这种格式化输出字符串的方式挺渐变

5.复习简单的正则表达式

```
import re
input_string = input()
result = re.split('[^+]', input_string)
```