第一次习题课!

喵喵

2022.07.18

白盒

说明:
https://physics-data.meow.plus/faq/rules/whitebox/

不要随意修改给出的代码模板!

Git 配置

- user <user@hostname.localdomain>
- Blah Blah <john@apple.com>
- Your Name <you@example.com>

Git 配置

- user <user@hostname.localdomain>
- Blah Blah <john@apple.com>
- Your Name <you@example.com>
- Jiajie Chen <c@jia.je> ?

Git 配置

- user <user@hostname.localdomain>
- Blah Blah <john@apple.com>
- Your Name <you@example.com>
- Jiajie Chen <c@jia.je> ?

```
> git config --global user.name "Meow"
> git config --global uer.email "meow@impl.cat"
# Or
> vim ~/.gitconfig
```

git 配置 (Cont.)

`core.editor`: 可以配置编辑器
`alias.cmd_here`: 自定义指令

代码风格

PEP8: Python Enhancement Proposal 8: Style Guide for Python Code

代码风格

PEP8: Python Enhancement Proposal 8: Style Guide for Python Code

- autopep8
- 编辑器内置格式化

代码风格

PEP8: Python Enhancement Proposal 8: Style Guide for Python Code

- autopep8
- 编辑器内置格式化

空行、空格：

- 逻辑单元之间加空行分隔
- 运算符两侧**推荐加空格！**(E226)

autopep8 can't help you with...

```
zheshiyigebianliang = 1
```

autopep8 can't help you with...

```
zheshiyigebianliang = 1
这是一个变量 = 1.5
```

autopep8 can't help you with...

```
zheshiyigebianliang = 1
这是一个变量 = 1.5
arrayofelement = 2
```

autopep8 can't help you with...

```
zheshiyigebianliang = 1
这是一个变量 = 1.5
arrayofelement = 2
Σ = 3
```

autopep8 can't help you with...

```
zheshiyigebianliang = 1
这是一个变量 = 1.5
arrayofelement = 2
Σ = 3

var_count = 1
elems = 2
tot_sum = 3
"snake_case"
```

Besides...

```
if d.__contains__(k) == True:
    print("It's true!")
```

Besides...

```
if d.__contains__(k) == True:
    print("It's true!")


if a == b:
    return True
else:
    return False
```

Besides...

```
if d.__contains__(k) == True:
    print("It's true!")

if a == b:
    return True
else:
    return False

# l list
result = []
for i in range(len(l)):
    func1(l[i])
for i in range(len(l)):
    result.push(func2(l[i]))
```

Besides... (Cont.)

```
if k in d:
    print("It's true!")

return a == b

# l list
for elem in l:
    do_something_with(l)
result = [func(elem) for elem in l]
```

One more thing...

注释

One more thing...

注释

- Docstring
- TODO / FIXME

poly

```
print(sum([int(input()) * x ** i for i in range(n + 1)]))
```

Note：无需过度追求 one-liner!

exgcd

白盒问题：

- 不使用递归：试图降低复杂度很好，但题目中有明确要求
- 错误递归（在 exgcd 中使用 gcd）
- 其他：

exgcd

白盒问题：

- 不使用递归：试图降低复杂度很好，但题目中有明确要求
- 错误递归（在 exgcd 中使用 gcd）
- 其他：

```
tmp = a
a = b
b = tmp
```

exgcd

白盒问题：

- 不使用递归：试图降低复杂度很好，但题目中有明确要求
- 错误递归（在 exgcd 中使用 gcd）
- 其他：

```
tmp = a
a = b
b = tmp


g = exgcd(b, a % b)[2]
x = exgcd(b, a % b)[1]
y = exgcd(b, a % b)[0] - (a // b) * exgcd(b, a % b)[1]
```

如何减少递归?

- 记忆化（保存以前的结果）
- 使用 functools.lru_cache 修饰器透明消除递归

```
@lru_cache(maxsize=None)
def fib(n):
    if n < 2:
        return n
    return fib(n-1) + fib(n-2)

>>> [fib(n) for n in range(16)]
[0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610]

>>> fib.cache_info()
CacheInfo(hits=28, misses=16, maxsize=None, currsize=16)
```

Planning

1 艹 + 4 口 -> 1 苗
1 苗 + 1 口 -> 1 喵
2 喵 -> 1 喵喵
Q:
喵喵 114514

Planning

1 艹 + 4 口 -> 1 苗
1 苗 + 1 口 -> 1 喵
2 喵 -> 1 喵喵
Q:
喵喵 114514

“线性规划?”

Planning

```
1 艹 + 4 口 -> 1 苗
1 苗 + 1 口 -> 1 喵
2 喵 -> 1 喵喵
Q:
喵喵 114514
```

“线性规划?”

- scipy.linalg
- 自制迭代算法

测试情况

增加了两个隐藏黑盒测例，共 $10 \times 8 = 80$ 分。

- 最后一行末尾没有 `'\n'`：使用 `s.strip()` 而非 `s[-1]`
- `3 A1 -> 1 A2, 3 A2 -> 1 A3, ...`：检验浮点精度问题

Why `Fraction`?

$$0.1 + 0.2 = ?$$

Why Fraction?

$0.1 + 0.2 = ?$

```
>>> 0.1 + 0.2
0.30000000000000004
```

Why Fraction?

$0.1 + 0.2 = ?$

```
>>> 0.1 + 0.2
0.30000000000000004
```

$(1/49) \times 49 = ?$

Why Fraction?

$0.1 + 0.2 = ?$

```
>>> 0.1 + 0.2
0.30000000000000004
```

$(1/49) \times 49 = ?$

```
>>> 1 / 49 * 49
0.9999999999999999
```

Why Fraction?

$0.1 + 0.2 = ?$

```
>>> 0.1 + 0.2
0.30000000000000004
```

$(1/49) \times 49 = ?$

```
>>> 1 / 49 * 49
0.9999999999999999
```

小数存储 (IEEE 754 浮点数) $=$ " 科学计数法"

Why Fraction?

$0.1 + 0.2 = ?$

```
>>> 0.1 + 0.2
0.30000000000000004
```

$(1/49) \times 49 = ?$

```
>>> 1 / 49 * 49
0.9999999999999999
```

小数存储 (IEEE 754 浮点数) $=$ " 科学计数法"

`Fraction(1 / 3)` v.s. `Fraction(1, 3)`

以 # 开始的行

```
if row != "# 一些注释":
    if row != "# 二些注释":
        print("PROFIT?")
```

以 # 开始的行

```
if row != "# 一些注释":
    if row != "# 二些注释":
        print("PROFIT?")
```

“三些注释”?

以 # 开始的行

```
if '#' in line:
    print("PROFIT?")
```

以 # 开始的行

```
if '#' in line:
    print("PROFIT?")
```

```
2 Q + 1 # -> 1 Q#Q
2 Q#Q + 1 # -> 1 Q#Q#Q#Q
```

编辑器配置

学术诚信

```
num1=input()
num2=input()
num3=int(num1)+int(num2)
print (num3)
```

```
Author: [redacted]@tsinghua.com>
Date:   Wed Jul 13 21:35:42 2022 +0800


修改了aplusb.py和HONOR-CODE.md文件，实现了两个数的加法。


读入两个数字（一行一个），输出一行数字，为前面两个数字之和。过程无参考，无交流。
```

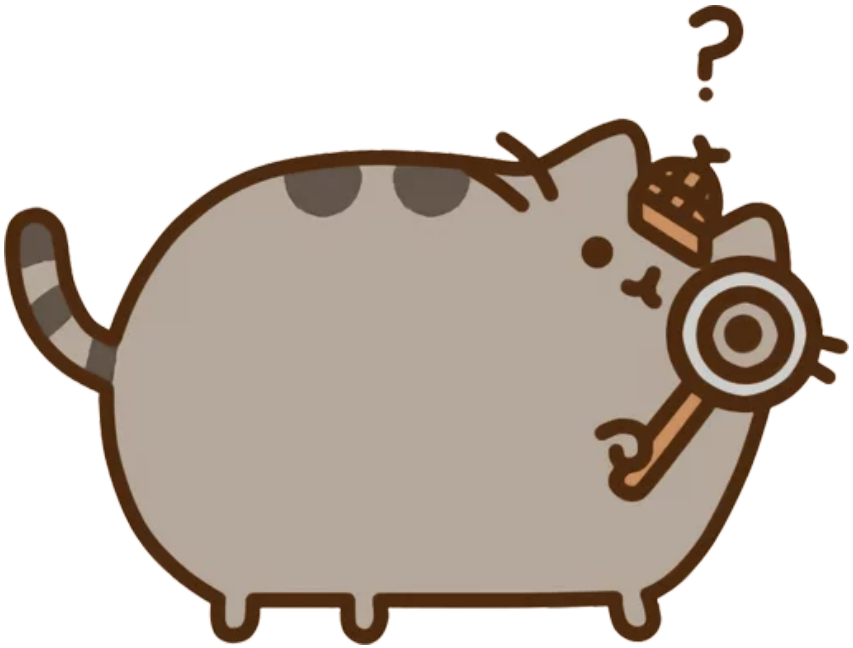
```
sum=0
n=int(input())
x=int(input())
j=[]
for i in range(n+1):
  j.append(input())
  sum=sum+int(j[i])*(x**i)
print (sum)
```

```
a=input()
b=input()
c=int(a)+int(b)
print (c)
```

```
Author: [redacted]@example.com>
Date:   Wed Jul 13 23:34:04 2022 +0800

    aplusb.py:实现了a和b的加法

    修改
aplusb.py，使得其读入两个数字（一行一个），输出一行数字，为前面两个数字之和。

    感谢：老师的指导。
```

```
n=int(input())
x=int(input())
out=0
q=[]
for p in range(n+1):
    q.append(input())
    out+=int(q[p])*(x**p)
print (out)
```

That's All!



Question time!