### 数字形式转换 I

获得用户输入的一个正整数输入，输出该数字对应的中文字符表示。‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‬

0到9对应的中文字符分别是：零一二三四五六七八九‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪

template = "零一二三四五六七八九"

s = input()for c in s:

print(template[eval(c)], end="")

print()中增加end=""参数表示输出后不增加换行，多个print()可以连续输出。

### 三次方格式化

获得用户输入的一个数字，可能是整数或浮点数，a，计算a的三次方值，并打印输出。‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

输出结果采用宽度20个字符、居中输出、多余字符采用减号(-)填充。‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

如果结果超过20个字符，则以结果宽度为准。‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

****输入示例****‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

10

****输出示例****‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

--------1000--------

a = eval(input()) print("{:-^20}".format(pow(a, 3)))

### 恺撒密码

恺撒密码是古罗马恺撒大帝用来对军事情报进行加解密的算法，它采用了替换方法对信息中的每一个英文字符循环替换为字母表序列中该字符后面的第三个字符，即，字母表的对应关系如下：‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

原文：A B C D E F G H I J K L M N O P Q R S T U V W X Y Z‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

密文：D E F G H I J K L M N O P Q R S T U V W X Y Z A B C‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

对于原文字符P，其密文字符C满足如下条件：C=(P+3) mod 26‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

上述是凯撒密码的加密方法，解密方法反之，即：P=(C-3) mod 26‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

假设用户可能使用的输入包含大小写字母a~zA~Z、空格和特殊符号，请编写一个程序，对输入字符串进行恺撒密码加密，直接输出结果，其中空格不用进行加密处理。使用input()获得输入。‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

****输入示例****‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

python is good

****输出示例****‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

sbwkrq lv jrrg

str=input()

for i in str:

if 'a'<= i <= 'z':

print(chr(ord('a')+(ord(i)-ord("a")+3)%26),end="")

elif 'A'<=i<='Z':

print(chr(ord('A')+(ord(i)-ord('A')+3)%26),end="")

else:

print(i,end="")

### 三位水仙花数

"水仙花数"是指一个三位整数，其各位数字的3次方和等于该数本身。‪‬‪‬‪‬‪‬‪‬‮‬‫‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

例如：ABC是一个"3位水仙花数"，则：A的3次方＋B的3次方＋C的3次方 = ABC。‪‬‪‬‪‬‪‬‪‬‮‬‫‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

请按照从小到大的顺序输出所有的3位水仙花数，请用"逗号"分隔输出结果。‪‬‪‬‪‬‪‬‪‬‮‬‫‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

输出仅表示格式，不表示对错。‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

****输出示例****‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

111,222

sum=""

for i in range(100,1000):

a=i//100

b=(i-a\*100)//10

c=(i-a\*100-b\*10)

if a\*\*3+b\*\*3+c\*\*3==i:

stri=str(i)

sum+=stri

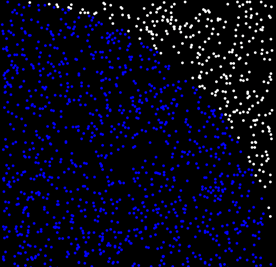
sum+=","

print("{}".format(sum[:-1]))

### 实例6：圆周率的计算

这是"实例"题，与课上讲解实例相同，请作答检验学习效果。‪‬‪‬‪‬‪‬‪‬‮‬‫‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

求解圆周率可以采用蒙特卡罗方法，在一个正方形中撒点，根据在1/4圆内点的数量占总撒点数的比例计算圆周率值。‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

请以123作为随机数种子，获得用户输入的撒点数量，编写程序输出圆周率的值，保留小数点后6位。‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

****输入示例****‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

1024

****输出示例****‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

3.218750

import random

import time

DARTS = eval(input())

hits = 0.0

start = time.perf\_counter()

random.seed(123)

for i in range(1,DARTS + 1):

x,y=random.random(),random.random()

dists=pow(x\*\*2+y\*\*2,0.5)

if(dists<=1.0):

hits+=1

pi = 4\*(hits/DARTS)

print("{:.6f}".format(pi))

### 用户登录的三次机会

给用户三次输入用户名和密码的机会，要求如下：‪‬‪‬‪‬‪‬‪‬‮‬‫‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

1）如输入第一行输入用户名为‘Kate’,第二行输入密码为‘666666’，输出‘登录成功！’，退出程序；‪‬‪‬‪‬‪‬‪‬‮‬‫‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

2）当一共有3次输入用户名或密码不正确输出“3次用户名或者密码均有误！退出程序。”。‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

****输入示例1****‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

Kate666666

****输出示例1****‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

登录成功！

****输入示例2****‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

kate123

alice456

john111111

****输出示例2****

import sys

for i in range(3):

for i in range(2):

if input()=="Kate":

if input()=="666666":

print("登录成功！")

sys.exit()

print("3次用户名或者密码均有误！退出程序。")

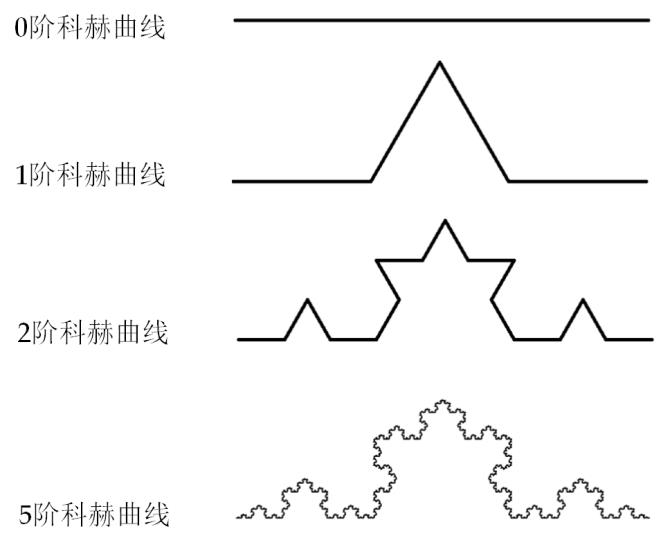
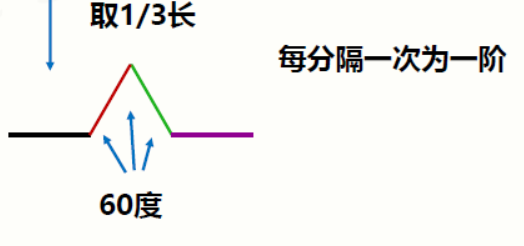
# 实例8：科赫雪花小包裹

## ****描述****

这是"实例"题，与课上讲解实例相同，请作答检验学习效果。‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

科赫曲线，也叫雪花曲线。绘制科赫曲线。‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

 ‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

                  ‪‬‪‬‪‬‪‬‪‬‮‬‫‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

 ‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

请补充编程模板中代码，完成功能：获得用户输入的整数N，作为阶，绘制N阶科赫曲线。

import turtle

def koch(size, n):

if n == 0:

turtle.fd(size)

else:

for angle in [0,60,-120,60]:

turtle.left(angle)

koch(size/3,n-1)

def main(level):

turtle.setup(600, 600)

turtle.penup()

turtle.goto(-200, 100)

turtle.pendown()

turtle.pensize(2)

koch(400, level)

turtle.hideturtle()

turtle.done()

try:

level = eval(input("请输入科赫曲线的阶: "))

main(level)

except:

print("输入错误")

### 任意累积

请根据编程模板补充代码，计算任意个输入数字的乘积。‪‬‪‬‪‬‪‬‪‬‮‬‫‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

注意，仅需要在标注...的地方补充一行或多行代码。‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

****输入示例1****‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

1,2,3,4

****输出示例1****‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

24

def multiply\_numbers(\*args):

result =1

for num in args:

result \*= num

return result

input\_str = input("")

numbers = [int(num) for num in input\_str.split(',')]

result = multiply\_numbers(\*numbers)

print( result)

### 汉诺塔实践

汉诺塔问题大家都清楚，这里不再赘述。‪‬‪‬‪‬‪‬‪‬‮‬‫‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

请补充编程模板中代码，完成如下功能：‪‬‪‬‪‬‪‬‪‬‮‬‫‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

有三个圆柱A、B、C，初始时A上有N个圆盘，N由用户输入给出，最终移动到圆柱C上。‪‬‪‬‪‬‪‬‪‬‮‬‫‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

每次移动步骤的表达方式示例如下：[STEP 10] A->C。其中，STEP是步骤序号，宽度为4个字符，右对齐。‪‬‪‬‪‬‪‬‪‬‮‬‫‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

请编写代码，获得输入N后，输出汉诺塔移动的步骤。‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

****输入格式：一个整数****‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

****输入示例1****‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

4

****输出格式：每个步骤一行，每行参考格式如下：[STEP 10] A->C****‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

****输出示例1****‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

[STEP 1] A->C[STEP 2] A->B[STEP 3] C->B[STEP 4] A->C[STEP 5] B->A[STEP 6] B->C[STEP 7] A->C

**平均代码量** 12 行

def hanoi(n, source, auxiliary, target):

if n == 1:

print("[STEP {:>3}] {}->{}".format(hanoi.step, source, target))

hanoi.step += 1

else:

hanoi(n-1, source, target, auxiliary)

print("[STEP {:>3}] {}->{}".format(hanoi.step, source, target))

hanoi.step += 1

hanoi(n-1, auxiliary, source, target)

hanoi.step = 1 # 初始化步骤计数

n = int(input()) # 输入汉诺塔的盘数

hanoi(n, 'A', 'B', 'C') # 调用汉诺塔函数

### 人名独特性统计

编程模板中给出了一个字符串，其中包含了含有重复的人名，请去重后给出独特性人名的统计。‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‭‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‭‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‭‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

输出模板中字符串共有多少个独特人名。‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‭‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‭‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‭‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

****输出示例****‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

10

s = '''双儿 洪七公 赵敏 赵敏 逍遥子 鳌拜 殷天正 金轮法王 乔峰 杨过 洪七公 郭靖

杨逍 鳌拜 殷天正 段誉 杨逍 慕容复 阿紫 慕容复 郭芙 乔峰 令狐冲 郭芙

金轮法王 小龙女 杨过 慕容复 梅超风 李莫愁 洪七公 张无忌 梅超风 杨逍

鳌拜 岳不群 黄药师 黄蓉 段誉 金轮法王 忽必烈 忽必烈 张三丰 乔峰 乔峰

阿紫 乔峰 金轮法王 袁冠南 张无忌 郭襄 黄蓉 李莫愁 赵敏 赵敏 郭芙 张三丰

乔峰 赵敏 梅超风 双儿 鳌拜 陈家洛 袁冠南 郭芙 郭芙 杨逍 赵敏 金轮法王

忽必烈 慕容复 张三丰 杨逍 令狐冲 黄药师 袁冠南 杨逍 完颜洪烈 殷天正

李莫愁 阿紫 逍遥子 乔峰 逍遥子 完颜洪烈 郭芙 杨逍 张无忌 杨过 慕容复

逍遥子 虚竹 双儿 乔峰 郭芙 黄蓉 李莫愁 陈家洛 杨过 忽必烈 鳌拜 王语嫣

洪七公 韦小宝 阿朱 梅超风 段誉 岳灵珊 完颜洪烈 乔峰 段誉 杨过 杨过 慕容复

黄蓉 杨过 阿紫 杨逍 张三丰 张三丰 赵敏 张三丰 杨逍 黄蓉 金轮法王 郭襄

张三丰 令狐冲 郭芙 韦小宝 黄药师 阿紫 韦小宝 金轮法王 杨逍 令狐冲 阿紫

洪七公 袁冠南 双儿 郭靖 鳌拜 谢逊 阿紫 郭襄 梅超风 张无忌 段誉 忽必烈

完颜洪烈 双儿 逍遥子 谢逊 完颜洪烈 殷天正 金轮法王 张三丰 双儿 郭襄 阿朱

郭襄 双儿 李莫愁 郭襄 忽必烈 金轮法王 张无忌 鳌拜 忽必烈 郭襄 令狐冲

谢逊 梅超风 殷天正 段誉 袁冠南 张三丰 王语嫣 阿紫 谢逊 杨过 郭靖 黄蓉

双儿 灭绝师太 段誉 张无忌 陈家洛 黄蓉 鳌拜 黄药师 逍遥子 忽必烈 赵敏

逍遥子 完颜洪烈 金轮法王 双儿 鳌拜 洪七公 郭芙 郭襄'''

ls = s.split()

lenth= set(ls)

print(len(lenth))

# ****字典翻转输出****

## ****描述****

读入一个字典类型的字符串，反转其中键值对输出。‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

即，读入字典key:value模式，输出value:key模式。‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

## ****输入格式****

用户输入的字典格式的字符串，如果输入不正确，提示：输入错误。‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

****输出格式****‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

给定字典d，按照print(d)方式输出‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

## ****输入输出示例****

|  |  |  |
| --- | --- | --- |
|  | 输入 | 输出 |
| 示例 1 | {"a": 1, "b": 2} | {1: 'a', 2: 'b'} |

s = input()

d = eval(s)

e = {}

try:

for k in d:

e[d[k]] =k

print(e)

except:

print("输入错误")

### 《沉默的羔羊》之最多单词

附件是《沉默的羔羊》中文版内容，请读入内容，分词后输出长度大于等于2且出现频率最多的单词。‪‬‪‬‪‬‪‬‪‬‮‬‫‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

如果存在多个单词出现频率一致，请输出按照Unicode排序后最大的单词。‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

****输入格式：文件****‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

****输出格式：字符串****‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

****输出示例****‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‭‬‫‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‮‬‪‬‪‬‪‬‪‬‪‬‮‬‪‬‪‬‪‬‪‬‪‬‪‬‪‬‮‬‫‬‪‬

羔羊

import jieba

txt = open("沉默的羔羊.txt","r",encoding="utf-8").read()

words = jieba.lcut(txt)

counts = {}

for word in words:

if len(word)==1:

continue

else:

counts[word] = counts.get(word,0)+1

items = list(counts.items())

items.sort(key=lambda x:x[1],reverse=True)

word,count = items[0]

print("{}".format(word))