**广东海洋大学学生实验报告书（学生用表）**

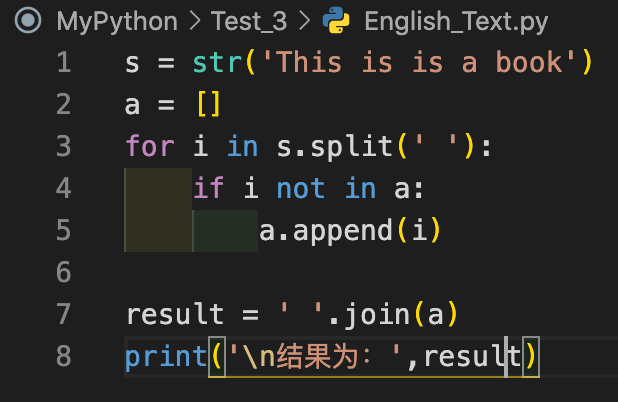
**GDOU-B-11-112**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实验名称 | Python字符串和正则表达式 | | | | 课程名称 | | 高级程序设计语言 | | | | 课程号 | |  |
| 学院(系) | 数学与计算机学院 | | 专业 | 物联网工程 | | | | | 班级 | | 物联1181 | | |
| 学生姓名 | 陈用林 | 学号 | 201811672102 | | | 实验地点 | | 二教A501 | | 实验日期 | | 2020.10.19 | |

1. **实验内容：**

**1、有一段英文文本，其中有单词连续重复了2次，编写程序检查重复的单词并只保留一个。例如，文本内容为：”This is is a book.”，程序输出为：”This is a book.”。**

**（1）程序代码：**



s = str('This is is a book')

a = []

for i in s.split(' '):

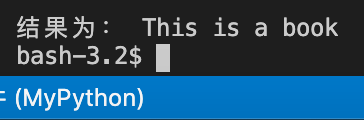
if i not in a:

a.append(i)

result = ' '.join(a)

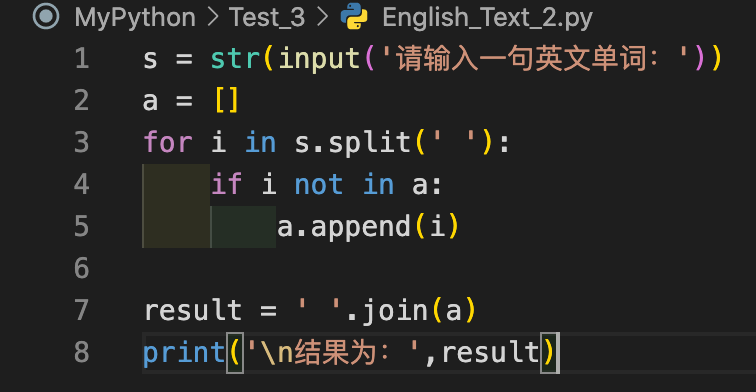
print('\n结果为：',result)

**（2）运行结果（截图）：**



**2、编写程序实现：输入任意英文句子，检查该句子中重复的单词，去掉该句子中任意重复的单词。**

**（1）程序代码：**



s = str(input('请输入一句英文单词：'))

a = []

for i in s.split(' '):

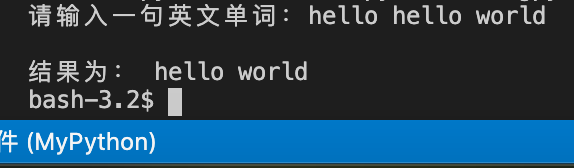
if i not in a:

a.append(i)

result = ' '.join(a)

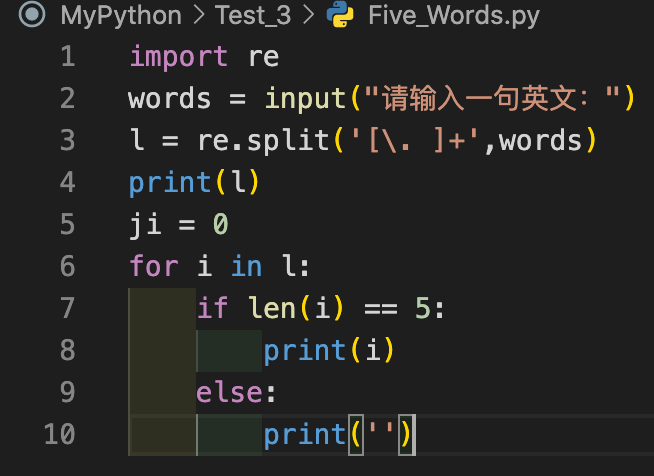
print('\n结果为：',result)

**（2）运行结果（截图）：**



**3、编写程序，用户输入一段英文，然后输出这段英文中所有长度为5个字母的单词。**

**（1）程序代码：**



import re

words = input("请输入一句英文：")

l = re.split('[\. ]+',words)

print(l)

ji = 0

for i in l:

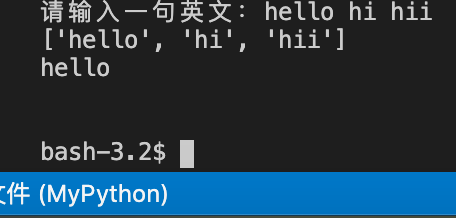
if len(i) == 5:

print(i)

else:

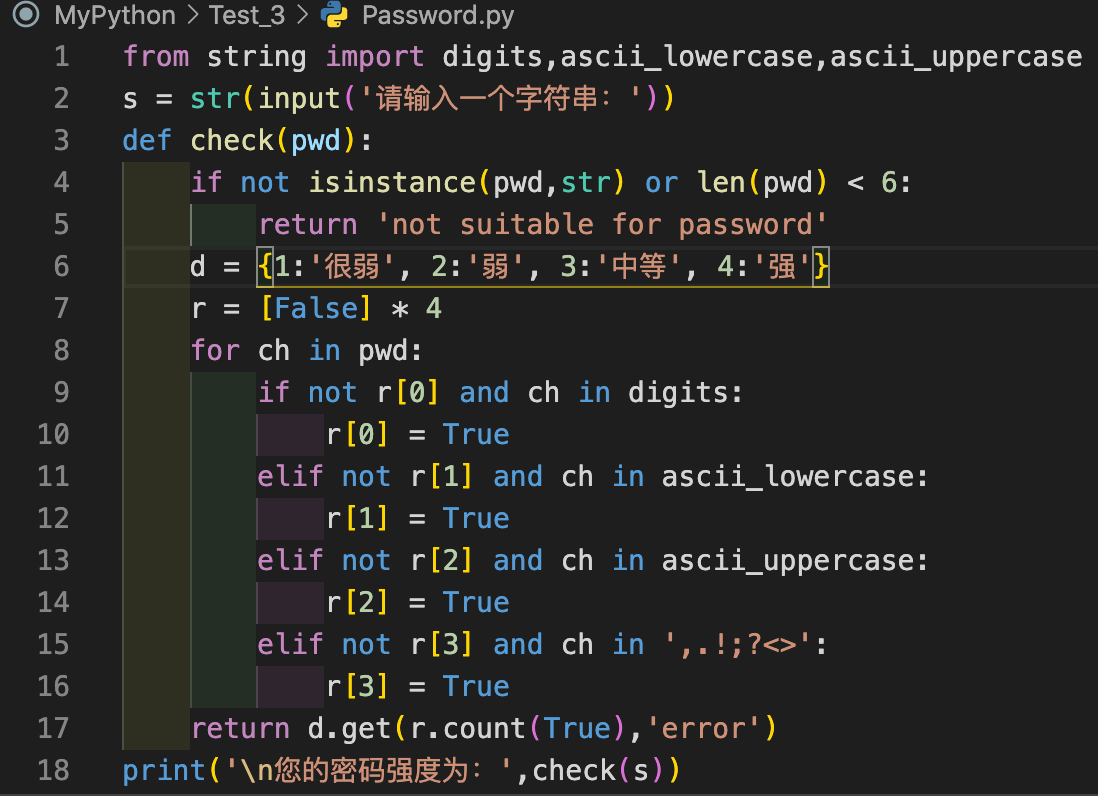
print('')

**（2）运行结果（截图）：**



**4、编写程序，输入一个字符串，输出该字符所为密码时的安全强度。**

**（1）程序代码：**



from string import digits,ascii\_lowercase,ascii\_uppercase

s = str(input('请输入一个字符串：'))

def check(pwd):

if not isinstance(pwd,str) or len(pwd) < 6:

return 'not suitable for password'

d = {1:'很弱', 2:'弱', 3:'中等', 4:'强'}

r = [False] \* 4

for ch in pwd:

if not r[0] and ch in digits:

r[0] = True

elif not r[1] and ch in ascii\_lowercase:

r[1] = True

elif not r[2] and ch in ascii\_uppercase:

r[2] = True

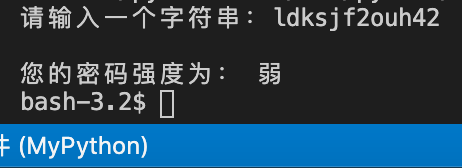
elif not r[3] and ch in ',.!;?<>':

r[3] = True

return d.get(r.count(True),'error')

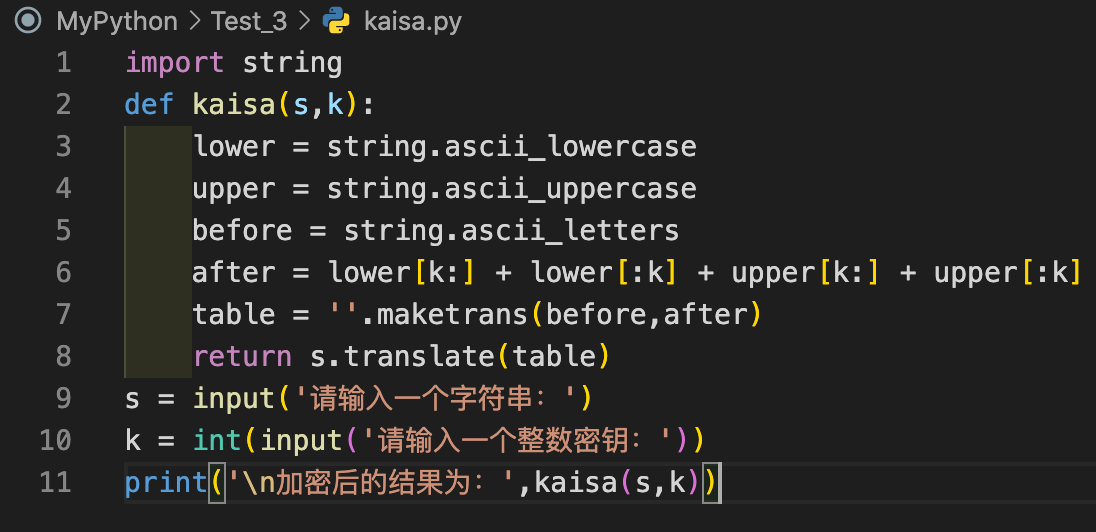
print('\n您的密码强度为：',check(s))

**（2）运行结果（截图）：**



**5、编写程序，输入要给字符串作为待加密的明文，然后输入一个整数作为凯撒加密算法的密钥，最后输出该字符串使用该密钥加密后的密文。**

**（1）程序代码：**



import string

def kaisa(s,k):

lower = string.ascii\_lowercase

upper = string.ascii\_uppercase

before = string.ascii\_letters

after = lower[k:] + lower[:k] + upper[k:] + upper[:k]

table = ''.maketrans(before,after)

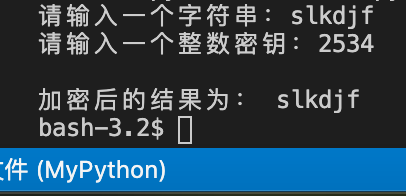
return s.translate(table)

s = input('请输入一个字符串：')

k = int(input('请输入一个整数密钥：'))

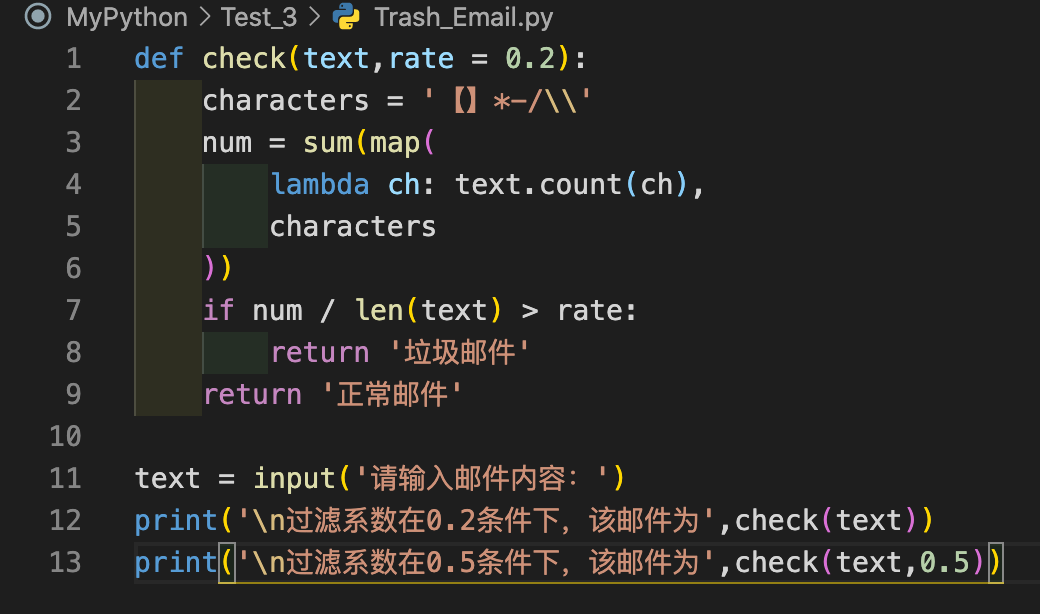
print('\n加密后的结果为：',kaisa(s,k))

**（2）运行结果（截图）：**



**6、编程实现垃圾邮件快速识别，要求从键盘输入任意的邮件标题，输出判断结果为“正常邮件”或“垃圾邮件”。**

**（1）程序代码：**



def check(text,rate = 0.2):

characters = '【】\*-/\\'

num = sum(map(

lambda ch: text.count(ch),

characters

))

if num / len(text) > rate:

return '垃圾邮件'

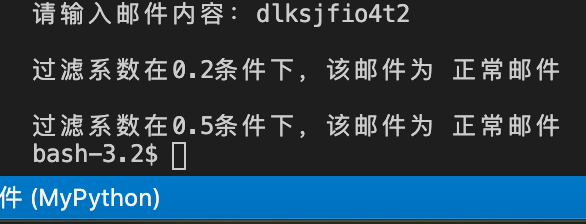
return '正常邮件'

text = input('请输入邮件内容：')

print('\n过滤系数在0.2条件下，该邮件为',check(text))

print('\n过滤系数在0.5条件下，该邮件为',check(text,0.5))

**（2）运行结果（截图）：**



1. **实验总结：**

**（本次实验的知识点小结；实验体会；未解决的问题；改进的方法）**

通过本次试验，基本掌握了Python字符串和正则表达式的基本概念和使用方法。