西安电子科技大学

安全前沿讨论班（I） 课程实验报告

实验名称 Python 入门实践案例

学　院　网络与信息安全学院　　　　班 级 1618019

姓　名　曹寅峰　　　　学 号 16020610025

实验日期 2019 年 4 月 9 日

指导教师评语： 指导教师： ______年____月____日

一、实验目的

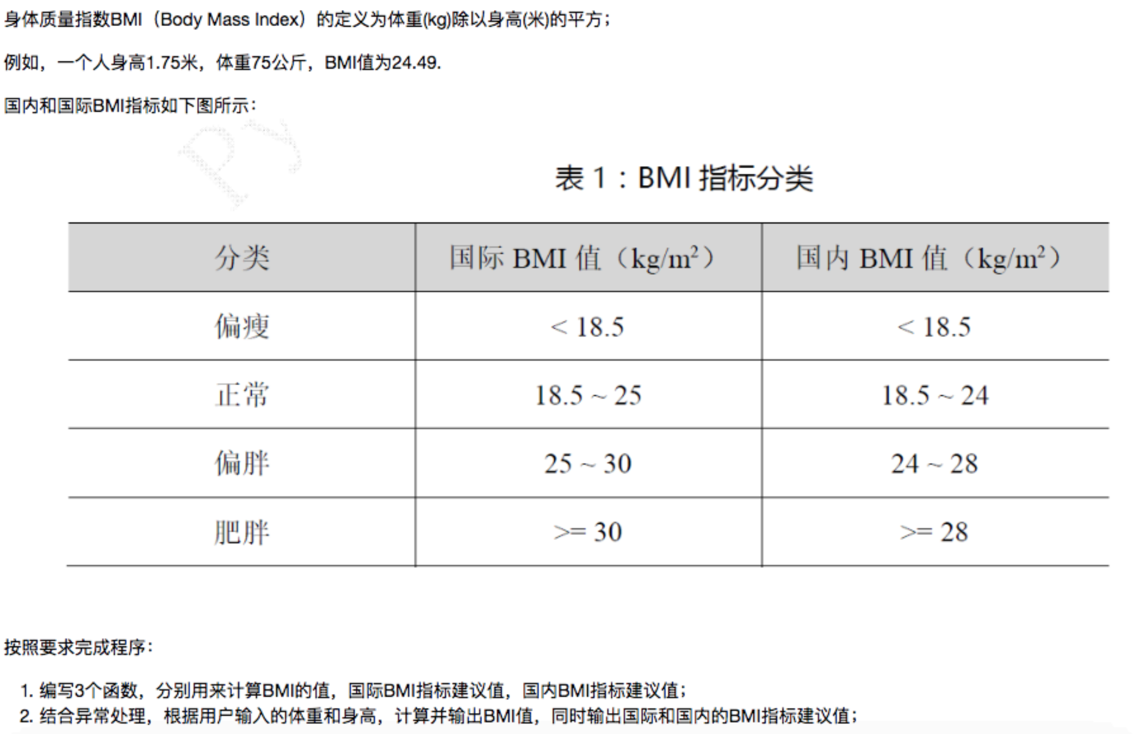
练习，回顾 python 的基本语法，控制流等，并在最后实现一个简单的拼写检查器

二、实验环境

Python3.5 ,Jupyter notebook

三、实验基本原理及步骤

1. BMI 计算



身体质量指数BMI（Body Mass Index）的定义为体重(kg)除以身高(米)的平方；

例如，一个人身高1.75米，体重75公斤，BMI值为24.49.

国内和国际BMI指标如下图所示：

表 1：BMI 指标分类

分类	国际 BMI 值（kg/m^2）	国内 BMI 值（kg/m^2）
偏瘦	< 18.5	< 18.5
正常	18.5 ~ 25	18.5 ~ 24
偏胖	25 ~ 30	24 ~ 28
肥胖	>= 30	>= 28

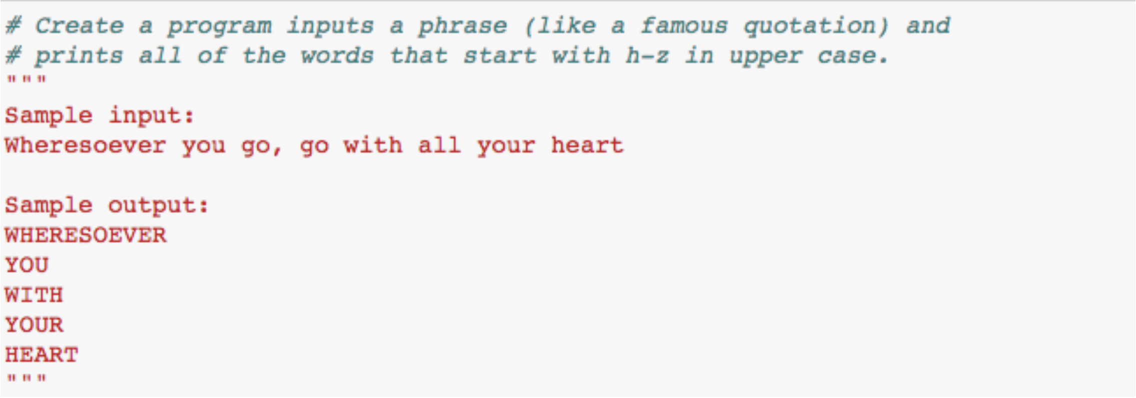
按照要求完成程序：

1. 编写3个函数，分别用来计算BMI的值，国际BMI指标建议值，国内BMI指标建议值；
2. 结合异常处理，根据用户输入的体重和身高，计算并输出BMI值，同时输出国际和国内的BMI指标建议值；

根据题意，通过计算 BMI 后用 if 判断 BMI 范围即可

```
def computeBMI(height,weight):
    """compute BMI value based on height and weight"""
    res=weight/(height*height)
    return res
def internationalBMI(BMI):
    """return international BMI index, the input parameter is BMI value"""
    res=''
    if BMI>=30:
        res='肥胖'
    elif BMI>=25:
        res='偏胖'
    elif BMI>18.5:
        res='正常'
    else:
        res='偏瘦'
    return res
def domesticBMI(BMI):
    """return domestic BMI index, the input parameter is BMI value"""
    res=''
    if BMI>=38:
        res='肥胖'
    elif BMI>=24:
        res='偏胖'
    elif BMI>18.5:
        res='正常'
    else:
        res='偏瘦'
    return res
```

2. 字符串输出



```
# Create a program inputs a phrase (like a famous quotation) and
# prints all of the words that start with h-z in upper case.
"""
Sample input:
Wheresoever you go, go with all your heart

Sample output:
WHERESOEVER
YOU
WITH
YOUR
HEART
"""
```

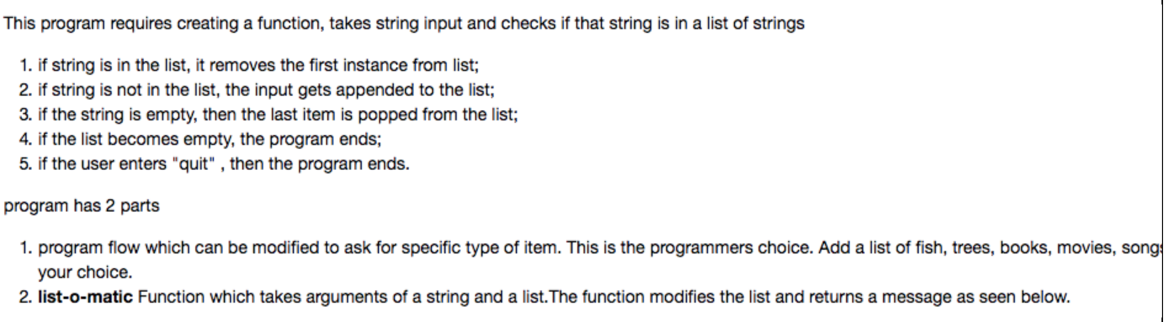
题中要求保留一段英文中首字母为 h-z 的单词

总体思路为：

1. **将输入字符串分割成单词列表**
2. **利用 ord 函数判断每个单词的首字母是否在范围内，如果在范围内，添加到一个新列表中，防止引用出错**

```
oristr=input('input your string')
strlist=list(oristr.lower().split(' '))
reslist=[]
3print(strlist)
for word in strlist:
    if ord(word[0])<ord('h') or ord(word[0])>ord('z'):

        continue
    else:
        reslist.append(word)
for word in reslist:
    print(word.upper())
```

3. 字符串操作



This program requires creating a function, takes string input and checks if that string is in a list of strings

1. if string is in the list, it removes the first instance from list;
2. if string is not in the list, the input gets appended to the list;
3. if the string is empty, then the last item is popped from the list;
4. if the list becomes empty, the program ends;
5. if the user enters "quit" , then the program ends.

program has 2 parts

1. program flow which can be modified to ask for specific type of item. This is the programmers choice. Add a list of fish, trees, books, movies, songs your choice.
2. **list-o-matic** Function which takes arguments of a string and a list.The function modifies the list and returns a message as seen below.

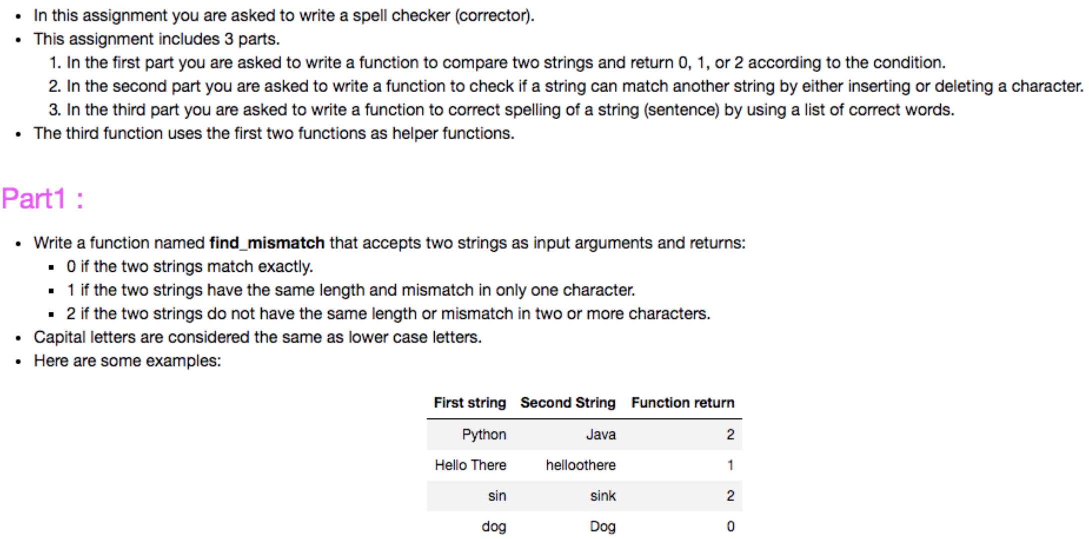
题中要求输入一个列表和一个字符串，对列表进行操作，字符串如果在原来列表中存在则弹出一个，不存在则加入，空输入弹出最后一个，quit 退出程序

总体思路为：

1. **利用 if 判断输入的字符串是否为 quit，是则直接 return**
2. **如果不是 quit，则按照要求操作字符串**
3. **用循环结构，所以函数不需要有返回值，直接打印结果即可**

```
def checklist(givenlist,oristr):
    if oristr.strip()=='':
        tmp=[]
        tmp.extend(givenlist)
        givenlist.pop()
        print('{} pop from list'.format(tmp[-1]))
        return 0
    if oristr=='quit':
        print('Goodbye!')
        return 0
    if oristr in givenlist:
        givenlist.remove(oristr)
        print('1 instance of {} removed from list'.format(oristr))
        return 0
    if oristr not in givenlist:
        givenlist.append(oristr)
        print('1 instance of {} appended to list'.format(oristr))
        return 0

    return 0
givenlist=['cat', 'goat', 'cat']
while(1):
    print('look at all the animals',givenlist)
    oristr=input('enter the name of an animal:')
    if oristr=='quit':
        print('Goodbye!')
        break
    checklist(givenlist,oristr)
    print('\n')
```

4. 单词检查 Part1



- In this assignment you are asked to write a spell checker (corrector).
- This assignment includes 3 parts.
  1. In the first part you are asked to write a function to compare two strings and return 0, 1, or 2 according to the condition.
  2. In the second part you are asked to write a function to check if a string can match another string by either inserting or deleting a character.
  3. In the third part you are asked to write a function to correct spelling of a string (sentence) by using a list of correct words.
- The third function uses the first two functions as helper functions.

Part1：

- Write a function named **find_mismatch** that accepts two strings as input arguments and returns:
  - 0 if the two strings match exactly.
  - 1 if the two strings have the same length and mismatch in only one character.
  - 2 if the two strings do not have the same length or mismatch in two or more characters.
- Capital letters are considered the same as lower case letters.
- Here are some examples:

First string	Second String	Function return
Python	Java	2
Hello There	helloothere	1
sin	sink	2
dog	Dog	0

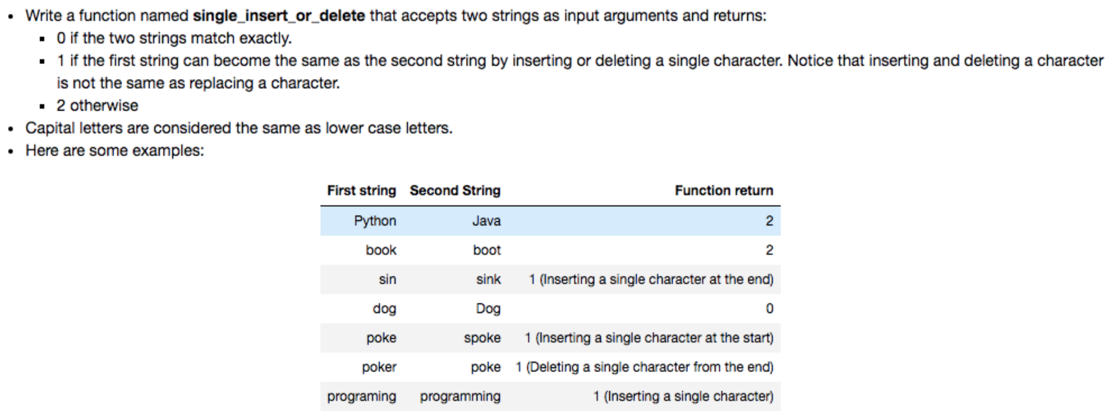
第一部分要求我们完成第一个拼写检查函数，输入两个字符串（不区分大小写）如果完全相同返回 0，长度相同但有一个字母不同返回 1，其他返回 2

总体思路为：

1. **先从简单的判断，都转换为小写字母，如果完全相同，返回 0**
2. **对于 1 的情况，比较长度后遍历对比每一位字母，记录不同字母的数量，如果数量为 1 则返回 1**
3. **其他情况返回 2**

```
def find_mismatch(str1,str2):
    str1=str1.lower()
    str2=str2.lower()
    if str1==str2:
        return 0
    if len(str1)==len(str2):
        count=0
        for i in range(len(str1)):
            if str1[i]!=str2[i]:
                count=count+1
        i=i+1
        if count==1:
            return 1
    else:
        return 2
```

5. 单词拼写 Part2



- Write a function named **single_insert_or_delete** that accepts two strings as input arguments and returns:
  - 0 if the two strings match exactly.
  - 1 if the first string can become the same as the second string by inserting or deleting a single character. Notice that inserting and deleting a character is not the same as replacing a character.
  - 2 otherwise
- Capital letters are considered the same as lower case letters.
- Here are some examples:

First string	Second String	Function return
Python	Java	2
book	boot	2
sin	sink	1 (Inserting a single character at the end)
dog	Dog	0
poke	spoke	1 (Inserting a single character at the start)
poker	poke	1 (Deleting a single character from the end)
programing	programming	1 (Inserting a single character)

和上一部分类似，只是返回 1 的判断条件不同，变为通过删减或者增加一个字母可以和另一个单词相同

总体思路为：

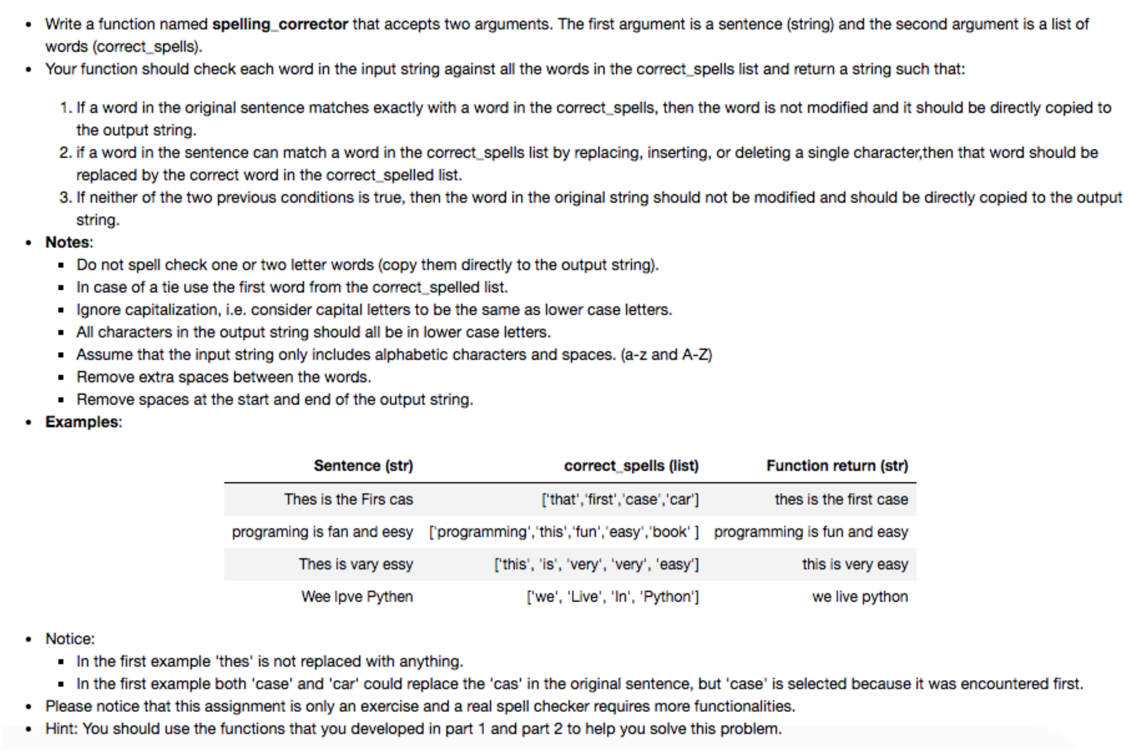
1. 先从简单的判断，都转换为小写字母，如果完全相同，返回 0

2. 对于 1 的情况，我们先找出较长的单词（较短的单词可能需要遍历字母表），然后逐个删去里面的一个字母，和另一个单词对比，相同则返回 1

3. 其他情况返回 2

```
# Type your function here
def single_insert_or_delete(str1,str2):
    str1=str1.lower()
    str2=str2.lower()
    if str1==str2:
        return 0
    flag=0
    if len(str1)<len(str2):
        str1,str2=str2,str1
    i=0
    for char in str1:
        resstr=str1[:i]+str1[i+1:]
        i=i+1
        if resstr==str2:
            return 1

    else:
        return 2
```

6. 单词拼写 Part3



- Write a function named **spelling_corrector** that accepts two arguments. The first argument is a sentence (string) and the second argument is a list of words (correct_spells).
- Your function should check each word in the input string against all the words in the correct_spells list and return a string such that:
  1. If a word in the original sentence matches exactly with a word in the correct_spells, then the word is not modified and it should be directly copied to the output string.
  2. if a word in the sentence can match a word in the correct_spells list by replacing, inserting, or deleting a single character,then that word should be replaced by the correct word in the correct_spelled list.
  3. If neither of the two previous conditions is true, then the word in the original string should not be modified and should be directly copied to the output string.
- **Notes**:
  - Do not spell check one or two letter words (copy them directly to the output string).
  - In case of a tie use the first word from the correct_spelled list.
  - Ignore capitalization, i.e. consider capital letters to be the same as lower case letters.
  - All characters in the output string should all be in lower case letters.
  - Assume that the input string only includes alphabetic characters and spaces. (a-z and A-Z)
  - Remove extra spaces between the words.
  - Remove spaces at the start and end of the output string.
- **Examples**:

Sentence (str)	correct_spells (list)	Function return (str)
Thes is the Firs cas	['that','first','case','car']	thes is the first case
programing is fan and eesy	['programming','this','fun','easy','book']	programming is fun and easy
Thes is vary essy	['this', 'is', 'very', 'very', 'easy']	this is very easy
Wee lpve Pythen	['we', 'Live', 'In', 'Python']	we live python

- Notice:
  - In the first example 'thes' is not replaced with anything.
  - In the first example both 'case' and 'car' could replace the 'cas' in the original sentence, but 'case' is selected because it was encountered first.
- Please notice that this assignment is only an exercise and a real spell checker requires more functionalities.
- Hint: You should use the functions that you developed in part 1 and part 2 to help you solve this problem.

第三部分结合上面两个函数（纠正函数），当返回为 1 时，利用正确的单词列表替换字符串，达到纠正的效果

总体思路为：

1. 首先一定要保证前两个纠正函数没写错

2. 将字符串处理成列表

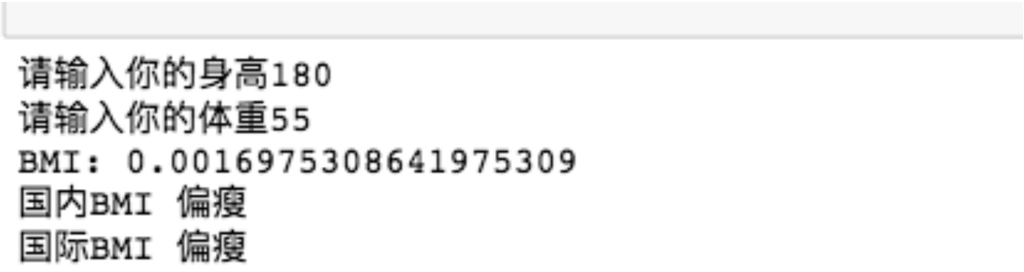
3. 利用 if or，如果有任意一个纠正函数为 1，则用正确单词替换列表中的单词

4. 将列表转换回字符串

```
# Type your function here
def spelling_corrector(oristr,correct_spell):
    strlist=list(oristr.lower().split(' '))
    res=''
    count=0
    for word in strlist:
        for cword in correct_spell:
            word=word.lower()
            cword=cword.lower()
            if single_insert_or_delete(word,cword)==1 or
find_mismatch(word,cword)==1:
                strlist[count]=cword
                break
        count=count+1
    return ' '.join(strlist)
```

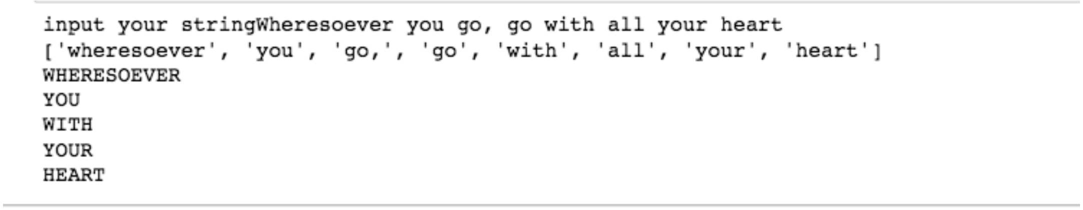
四、实验结果分析及回答问题（或测试环境及测试结果）

1. BMI 计算



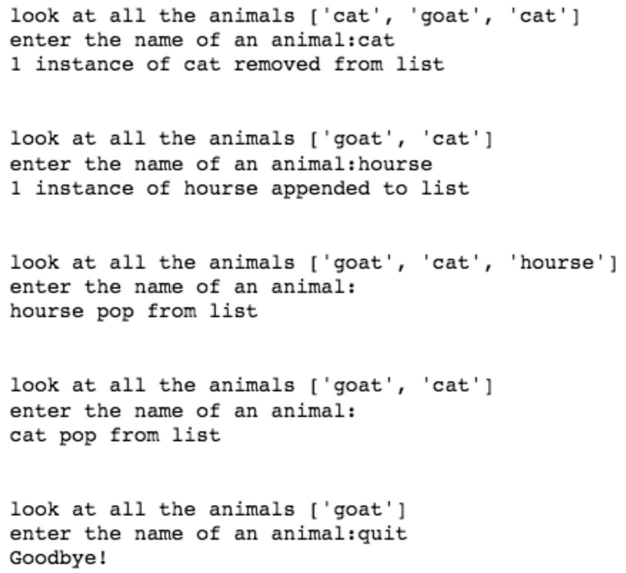
```
请输入你的身高180
请输入你的体重55
BMI: 0.0016975308641975309
国内BMI  偏瘦
国际BMI  偏瘦
```

2. 输出字符串



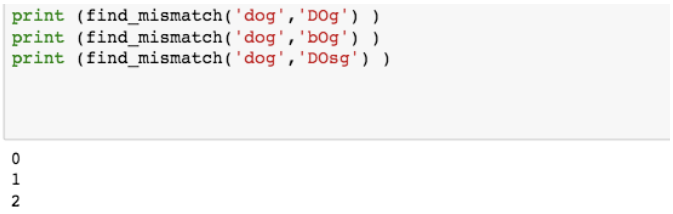
```
input your stringWheresoever you go, go with all your heart
['wheresoever', 'you', 'go,', 'go', 'with', 'all', 'your', 'heart']
WHERESOEVER
YOU
WITH
YOUR
HEART
```

3. 字符串操作



```
look at all the animals ['cat', 'goat', 'cat']
enter the name of an animal:cat
1 instance of cat removed from list


look at all the animals ['goat', 'cat']
enter the name of an animal:hourse
1 instance of hourse appended to list


look at all the animals ['goat', 'cat', 'hourse']
enter the name of an animal:
hourse pop from list


look at all the animals ['goat', 'cat']
enter the name of an animal:
cat pop from list


look at all the animals ['goat']
enter the name of an animal:quit
Goodbye!
```

4. 拼写检查 Part1

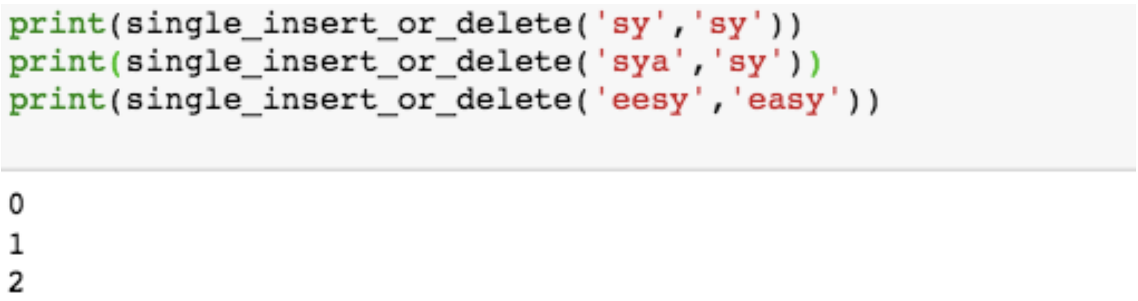


```
print (find_mismatch('dog','DOg') )
print (find_mismatch('dog','bOg') )
print (find_mismatch('dog','DOsg') )
```

```
0
1
2
```

这里注意要逐个对比，不能用 not in 否则 eesy 和 easy，由于有 2 个 e 无法正确比较

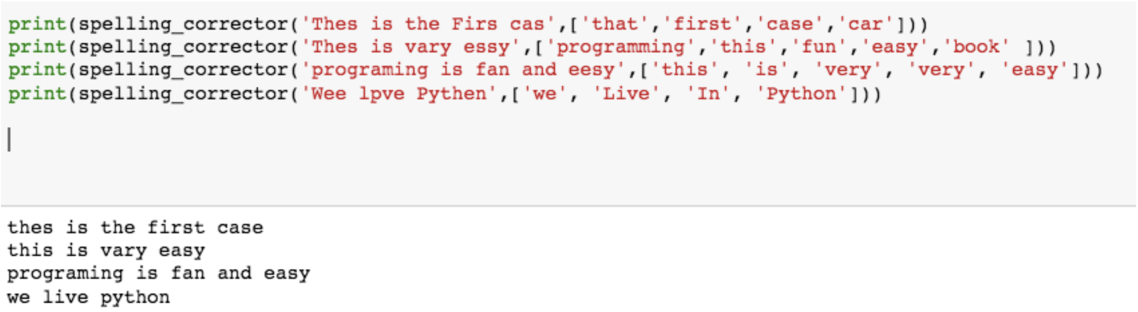
5. 单词检查 Part2



```
print(single_insert_or_delete('sy','sy'))
print(single_insert_or_delete('sya','sy'))
print(single_insert_or_delete('eesy','easy'))
```

```
0
1
2
```

6. 单词检查 part3



```
print(spelling_corrector('Thes is the Firs cas',['that','first','case','car']))
print(spelling_corrector('Thes is vary essy',['programming','this','fun','easy','book' ]))
print(spelling_corrector('programing is fan and eesy',['this', 'is', 'very', 'very', 'easy']))
print(spelling_corrector('Wee lpve Pythen',['we', 'Live', 'In', 'Python']))
```

```
thes is the first case
this is vary easy
programing is fan and easy
we live python
```

经过测试，实验结果均和样例相同，说明代码功能正确