**长沙学院计算机科学与工程学院**

**实验报告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 课程名称 | Python程序开发基础 | 班级 | 20软件01 | 实验日期 | 2023.3 |
| 姓名 | 胡锦华 | 学号 | B20200304125 | 指导教师 | 周景 |
| 成绩 |  | | | | |

### 一. 实验名称

Python进阶实验

### 二. 实验目的及要求

第一部分（2学时）：

掌握列表、元组、字典、集合以及其他常见数据结构的使用方法 。

第二部分（2学时）：

掌握列表、元组、字典、集合以及其他常见数据结构的使用方法 。

第三部分（2学时）：

结合循环语句，掌握列表、元组、字典、集合的推导、遍历以及数据类型转换、用户输入等用法。

### 三. 实验环境

Python 3.6+

### 四. 实验练习

**第一部分：**

练习4-1：比萨 想出至少三种你喜欢的比萨，将其名称存储在一个列表中，再使用for 循环将每种比萨的名称打印出来。

修改这个for 循环，使其打印包含比萨名称的句子，而不仅仅是比萨的名称。对于每种比萨，都显示一行输出，下面是一个例子。

I like pepperoni pizza.

在程序末尾添加一行代码，它不在for 循环中，指出你有多喜欢比萨。输出应包含针对每种比萨的消息，还有一个总结性句子，下面是一个例子。

I really love pizza!

练习4-2：动物 想出至少三种有共同特征的动物，将其名称存储在一个列表中，再使用for 循环将每种动物的名称打印出来。修改这个程序，使其针对每种动物都打印一个句子，下面是一个例子。

A dog would make a great pet.

在程序末尾添加一行代码，指出这些动物的共同之处，如打印下面这样的句子。

Any of these animals would make a great pet!

练习4-9：立方解析 使用列表解析生成一个列表，其中包含前10个整数的立方。

练习4-10：切片 选择你在本章编写的一个程序，在末尾添加几行代码，完成如下任务。

* 打印消息“The first three items in the list are:”，再使用切片来打印列表的前三个元素。
* 打印消息“Three items from the middle of the list are:”，再使用切片来打印列表的中间三个元素。
* 打印消息“The last three items in the list are:”，再使用切片来打印列表的末尾三个元素。

练习4-11：你的比萨，我的比萨 在你为完成练习4-1而编写的程序中，创建

比萨列表的副本，并将其赋给变量friend\_pizzas ，再完成如下任务。

在原来的比萨列表中添加一种比萨。

在列表friend\_pizzas 中添加另一种比萨。

核实有两个不同的列表。为此，打印消息“My favorite pizzas are:”，再使用一个for 循环来打印第一个列表；打印消息“My friend'sfavorite pizzas are:”，再使用一个for 循环来打印第二个列表。核实新增的比萨被添加到了正确的列表中。

练习4-12：使用多个循环 在本节中，为节省篇幅，程序foods.py的每个版本都没有使用for 循环来打印列表。请选择一个版本的foods.py，在其中编写两个for 循环，将各个食品列表打印出来。

练习4-13：自助餐 有一家自助式餐馆，只提供五种简单的食品。请想出五种

简单的食品，并将其存储在一个元组中。

* 使用一个for 循环将该餐馆提供的五种食品都打印出来。
* 尝试修改其中的一个元素，核实Python确实会拒绝你这样做。
* 餐馆调整了菜单，替换了它提供的其中两种食品。请编写一个这样的代码块：给元组变量赋值，并使用一个for 循环将新元组的每个元素都打印出来。

**第二部分：**

练习5-11：序数 序数表示位置，如1st和2nd。序数大多以th结尾，只有1、2和3例外。在一个列表中存储数字1～9。遍历这个列表。在循环中使用一个if-elif-else 结构，以打印每个数字对应的序数。输出内容应为"1st 2nd 3rd 4th 5th 6th 7th 8th 9th" ，但每个序数都独占一行。

练习5-13：自己的想法 与刚拿起本书时相比，现在你是一名能力更强的程序员了。鉴于你对如何在程序中模拟现实情形有了更深入的认识，可以考虑使用程序来解决一些问题了。随着编程技能不断提高，你可能想解决一些问题，请将这方面的想法记录下来。想想你可能想编写的游戏、想研究的数据集以及想创建的Web应用程序。

练习6-1：人 使用一个字典来存储一个熟人的信息，包括名、姓、年龄和居住的城市。该字典应包含键first\_name 、last\_name 、age 和city 。将存储在该字典中的每项信息都打印出来。

练习6-2：喜欢的数 使用一个字典来存储一些人喜欢的数。请想出5个人的名字，并将这些名字用作字典中的键；找出每个人喜欢的一个数，并将这些数作

为值存储在字典中。打印每个人的名字和喜欢的数。为了让这个程序更有趣，通过询问朋友确保数据是真实的。

练习6-3：词汇表 Python字典可用于模拟现实生活中的字典。为避免混淆，我们将后者称为词汇表。

* 想出你在前面学过的5个编程术语，将其用作词汇表中的键，并将它们的含
* 义作为值存储在词汇表中。以整洁的方式打印每个术语及其含义。为此，可先打印术语，在它后面加

上一个冒号，再打印其含义；也可在一行打印术语，再使用换行符（\n ）插入一个空行，然后在下一行以缩进的方式打印其含义。

练习6-4：词汇表2 现在你知道了如何遍历字典，可以整理为完成练习6-3而编写的代码，将其中的一系列函数调用print() 替换为一个遍历字典中键和值的循环。确定该循环正确无误后，再在词汇表中添加5个Python术语。当你再次运行这个程序时，这些新术语及其含义将自动包含在输出中。

练习6-10：喜欢的数2 修改为完成练习6-2而编写的程序，让每个人都可以有多个喜欢的数，然后将每个人的名字及其喜欢的数打印出来。

**第三部分：**

练习7-2：餐馆订位 编写一个程序，询问用户有多少人用餐。如果超过8位，就打印一条消息，指出没有空桌；否则指出有空桌。

练习7-4：比萨配料 编写一个循环，提示用户输入一系列比萨配料，并在用户输入'quit' 时结束循环。每当用户输入一种配料后，都打印一条消息，指出我们会在比萨中添加这种配料。

练习7-5：电影票 有家电影院根据观众的年龄收取不同的票价：不到3岁的观众免费；3～12岁的观众收费10美元；超过12岁的观众收费15美元。请编写一个循环，在其中询问用户的年龄，并指出其票价。

练习7-6：三种出路 以不同的方式完成练习7-4或练习7-5，在程序中采取如下做法。

* 在while 循环中使用条件测试来结束循环。
* 使用变量active 来控制循环结束的时机。
* 使用break 语句在用户输入'quit' 时退出循环。

练习7-8：熟食店 创建一个名为sandwich\_orders 的列表，在其中包含各种三明治的名字，再创建一个名为finished\_sandwiches 的空列表。遍历列表sandwich\_orders ，对于其中的每种三明治，都打印一条消息，如I made your tuna sandwich ，并将其移到列表finished\_sandwiches中。所有三明治都制作好后，打印一条消息，将这些三明治列出来。

练习7-9：五香烟熏牛肉卖完了 使用为完成练习7-8而创建的列表sandwich\_orders ，并确保'pastrami' 在其中至少出现了三次。在程序开头附近添加这样的代码：打印一条消息，指出熟食店的五香烟熏牛肉（pastrami）卖完了；再使用一个while 循环将列表sandwich\_orders 中的'pastrami' 都删除。确认最终的列表finished\_sandwiches 未包含'pastrami' 。

### 五. 实验内容

**前往**[**codewars.com**](http://www.codewars.com)**注册账号，完成下面的习题(或者自选题目)并且在题目的页面提交，通过题目的测试用例来获得分数，并将完成情况在表格中登记。**

**20软件1班：【腾讯文档】20软件1班Codewars登记**

[**20软件1班Codewars账号登记**](https://docs.qq.com/sheet/DWXhEV2lNZVVhVUFP?tab=BB08J2)

**20软件2班：【腾讯文档】20软件2班Codewars登记**

[**20软件2班Codewars账号登记**](https://docs.qq.com/sheet/DWVJVV3Z2YXlnWWxR?tab=BB08J2)

1. **3和5的倍数**

**如果我们列出所有低于 10 的 3 或 5 倍数的自然数，我们得到 3、5、6 和 9。这些数的总和为 23. 完成一个函数，使其返回小于某个整数的所有是3 或 5 的倍数的数的总和。此外，如果数字为负数，则返回 0。**

**注意：如果一个数同时是3和5的倍数，应该只被算一次。**

**代码提交地址：**

[**https://www.codewars.com/kata/514b92a657cdc65150000006**](https://www.codewars.com/kata/514b92a657cdc65150000006)

1. **重复字符的编码器**

**本练习的目的是将一个字符串转换为一个新的字符串，如果新字符串中的每个字符在原字符串中只出现一次，则为"("，如果该字符在原字符串中出现多次，则为")"。在判断一个字符是否是重复的时候，请忽略大写字母。**

**例如:**

"din" => "((("

"recede" => "()()()"

"Success" => ")())())"

"(( @" => "))(("

**代码提交地址:**

[**https://www.codewars.com/kata/54b42f9314d9229fd6000d9c**](https://www.codewars.com/kata/54b42f9314d9229fd6000d9c)

**提示：**

1. **首先使用列表解析得到一个列表，元素全部是3或者5的倍数。**
2. **使用sum函数可以获取这个列表所有元素的和。**
3. **元音统计**

**返回给定字符串中元音的数量（计数）。**

**对于这个Kata，我们将考虑a、e、i、o、u作为元音（但不包括y）。**

**输入的字符串将只由小写字母和/或空格组成。**

**代码提交地址：**

[**https://www.codewars.com/kata/54ff3102c1bad923760001f3**](https://www.codewars.com/kata/54ff3102c1bad923760001f3)

1. **从随机三元组中恢复秘密字符串**

**有一个不为你所知的秘密字符串。给出一个随机三个字母的组合的集合，恢复原来的字符串。**

**这里的三个字母的组合被定义为三个字母的序列，每个字母在给定的字符串中出现在下一个字母之前。"whi "是字符串 "whatisup "的一个三个字母的组合。**

**作为一种简化，你可以假设没有一个字母在秘密字符串中出现超过一次。**

**对于给你的三个字母的组合，除了它们是有效的三个字母的组合以及它们包含足够的信息来推导出原始字符串之外，你可以不做任何假设。特别是，这意味着秘密字符串永远不会包含不出现在给你的三个字母的组合中的字母。**

**测试用例：**

**secret = "whatisup"**

**triplets = [**

**['t','u','p'],**

**['w','h','i'],**

**['t','s','u'],**

**['a','t','s'],**

**['h','a','p'],**

**['t','i','s'],**

**['w','h','s']**

**]**

**test.assert\_equals(recoverSecret(triplets), secret)**

**代码提交地址：**

[**https://www.codewars.com/kata/53f40dff5f9d31b813000774/train/python**](https://www.codewars.com/kata/53f40dff5f9d31b813000774/train/python)

**提示：**

1. **利用集合去掉triplets中的重复字母，得到字母集合letters，最后的secret应该由集合中的字母组成，secrect长度也等于该集合。**

letters = {letter for triplet in triplets for letter in triplet }

length = len(letters)

1. **创建函数check\_first\_letter(triplets, first\_letter)，检测一个字母是不是secret的首字母，返回True或者False。**
2. **创建函数remove\_first\_letter(triplets, first\_letter), 从三元组中去掉首字母，返回新的三元组。**
3. **遍历字母集合letters，利用上面2个函数得到最后的结果secret。**
4. **去掉喷子的元音**

**喷子正在攻击你的评论区!**

**处理这种情况的一个常见方法是删除喷子评论中的所有元音(字母：a,e,i,o,u)，以消除威胁。**

**你的任务是写一个函数，接收一个字符串并返回一个去除所有元音的新字符串。**

**例如，字符串 "This website is for losers LOL!" 将变成 "Ths wbst s fr lsrs LL!".**

**注意：对于这个Kata来说，y不被认为是元音。**

**代码提交地址：**

[**https://www.codewars.com/kata/52fba66badcd10859f00097e**](https://www.codewars.com/kata/52fba66badcd10859f00097e)

**提示：**

1. **首先使用列表解析得到一个列表，列表中所有不是元音的字母。**
2. **使用字符串的join方法连结列表中所有的字母，例如：**

**last\_name = "lovelace"**

**letters = [letter for letter in last\_name ]**

**print(letters) # ['l', 'o', 'v', 'e', 'l', 'a', 'c', 'e']**

**name = ''.join(letters) # name = "lovelace"**

1. **停止逆转我的单词**

**编写一个函数，接收一个或多个单词的字符串，并返回相同的字符串，但所有5个或更多的字母单词都是相反的（就像这个Kata的名字一样）。传入的字符串将只由字母和空格组成。只有当出现一个以上的单词时，才会包括空格。**

**例如：**

spinWords( "Hey fellow warriors" ) => returns "Hey wollef sroirraw"

spinWords( "This is a test") => returns "This is a test"

spinWords( "This is another test" )=> returns "This is rehtona test"

代码提交地址：

<https://www.codewars.com/kata/5264d2b162488dc400000001>

**提示：**

1. **利用str的split方法可以将字符串分为单词列表**

**例如：**

**words = "hey fellow warrior".split()**

**# words should be ['hey', 'fellow', 'warrior']**

1. **利用列表推导将长度大于等于5的单词反转(利用切片word[::-1])**
2. **最后使用str的join方法连结列表中的单词。**
3. Find The Parity Outlier

给你一个包含整数的数组（其长度至少为3，但可能非常大）。该数组要么完全由奇数组成，要么完全由偶数组成，除了一个整数N。请写一个方法，以该数组为参数，返回这个 "离群 "的N。

例如：

[2, 4, 0, 100, 4, 11, 2602, 36]

Should return: 11 (the only odd number)

[160, 3, 1719, 19, 11, 13, -21]

Should return: 160 (the only even number)

代码提交地址：

<https://www.codewars.com/kata/5526fc09a1bbd946250002dc>

1. 数字之和/数字根

数字根是一个数字中所有数字的递归之和。

如果该值有一个以上的数字，就继续这样还原，直到产生一个个位数。输入的将是一个非负的整数。

例如：

16 --> 1 + 6 = 7

942 --> 9 + 4 + 2 = 15 --> 1 + 5 = 6

132189 --> 1 + 3 + 2 + 1 + 8 + 9 = 24 --> 2 + 4 = 6

493193 --> 4 + 9 + 3 + 1 + 9 + 3 = 29 --> 2 + 9 = 11 --> 1 + 1 = 2

代码提交地址：

<https://www.codewars.com/kata/541c8630095125aba6000c00>

1. 数组的差

你在这个卡塔中的目标是实现一个差值函数，从一个列表中减去另一个列表并返回结果。

它应该从列表a中删除所有在列表b中存在的值，并保持它们的顺序。

array\_diff([1,2],[1]) == [2] 。

如果一个值出现在b中，它的所有出现必须从另一个中删除。

array\_diff([1,2,2,2,3],[2]) == [1,3]

代码提交地址：

<https://www.codewars.com/kata/523f5d21c841566fde000009>

1. 建塔

建立一个金字塔形的塔，作为一个字符串的数组/列表，给定一个正整数的楼层数。一个塔块用 "\*"字符表示。

例如，一个有3层的塔看起来像这样。

[

" \* ",

" \*\*\* ",

"\*\*\*\*\*"

]

而一个有6层的塔楼看起来是这样的。

[

" \* ",

" \*\*\* ",

" \*\*\*\*\* ",

" \*\*\*\*\*\*\* ",

" \*\*\*\*\*\*\*\*\* ",

"\*\*\*\*\*\*\*\*\*\*\*"

]

代码提交地址：

<https://www.codewars.com/kata/576757b1df89ecf5bd00073b>

1. 检测Pangram

pangram是一个至少包含每个字母一次的句子。例如，"The quick brown fox jumps over the lazy dog "这个句子就是一个pangram，因为它至少使用了一次字母A-Z（大小写不相关）。

给定一个字符串，检测它是否是一个pangram。如果是则返回True，如果不是则返回False。忽略数字和标点符号。

代码提交地址：

<https://www.codewars.com/kata/545cedaa9943f7fe7b000048>

1. **哈梅林耳聋的老鼠**

**魔笛手被征召去演奏他的魔法曲调，把所有的老鼠哄出城去。但有些老鼠是聋子，而且走错了路!**

**卡塔任务**

**有多少只耳聋的老鼠？**

**P = 魔笛手**

**O~ = 老鼠向左走**

**~O = 老鼠向右走**

**例子：**

**ex1 ~O~O~O~O P has 0 deaf rats**

**ex2 P O~ O~ ~O O~ has 1 deaf rat**

**ex3 ~O~O~O~OP~O~OO~ has 2 deaf rats**

代码提交地址：

<https://www.codewars.com/kata/598106cb34e205e074000031>

1. 数独解决方案验证

数独背景

数独是一种在 9x9 网格上进行的游戏。游戏的目标是用 1 到 9 的数字填充网格的所有单元格，以便每一列、每一行和九个 3x3 子网格（也称为块）中的都包含数字 1 到 9。（更多信息请访问：http://en.wikipedia.org/wiki/Sudoku）

编写一个函数接受一个代表数独板的二维数组，如果它是一个有效的解决方案则返回 true，否则返回 false。数独板的单元格也可能包含 0，这将代表空单元格。包含一个或多个零的棋盘被认为是无效的解决方案。

棋盘总是 9 x 9 格，每个格只包含 0 到 9 之间的整数。

代码提交地址：

<https://www.codewars.com/kata/63d1bac72de941033dbf87ae>

1. 将零移到末尾

写一个算法，将一个数组的所有零移到最后，保留其他元素的顺序。

例如：

move\_zeros([1, 0, 1, 2, 0, 1, 3]) # returns [1, 1, 2, 1, 3, 0, 0]

代码提交地址：

<https://www.codewars.com/kata/52597aa56021e91c93000cb0>

1. 计算零钱的组合

编写一个函数，计算在给定一个硬币面额数组的情况下，你可以用多少种不同的方式来换取一定数量的钱。例如，如果你有面额为1和2的硬币，有3种方法可以找回4元钱：

1+1+1+1, 1+1+2, 2+2.

硬币的次序不需要考虑：

1+1+2 == 2+1+1

另外，假设你有无限多的硬币。

你的函数应该接受一个要兑换的金额和一个独特的硬币面额数组，例如：

count\_change(4, [1,2]) # => 3

count\_change(10, [5,2,3]) # => 4

count\_change(11, [5,7]) # => 0

代码提交地址：

<https://www.codewars.com/kata/541af676b589989aed0009e7>

### 六. 实验过程和结果

1. **3和5的倍数**

代码：

def solution(number):

    if number < 3:

        return 0

    sums = 0

    while number > 2:

        number-=1

        if number % 3 == 0 or number % 5 == 0:

            sums += number

    return sums

输出

print(solution(10)) # 23

1. **重复字符的编码器**

**代码：**

def duplicate\_encode(word):

    word = word.lower()

    words = ''

    for i in word:

        if word.count(i) > 1:

            words+=')'

        else:

            words+='('

    return words

**输出：**

print(duplicate\_encode('Wt@e@KrxnQ!VqbCX!Xb C))AoBkIDZ@IBYy(s@'))

# (()())()())()))))))()))(()))(()))))(()

1. **元音统计**

**代码：**

def get\_count(sentence):

    i = sentence.count('a')

    j = sentence.count('e')

    k = sentence.count('i')

    l = sentence.count('o')

    m = sentence.count('u')

    return i + j + k + l + m

**输出：**

print(get\_count("aeiouasafgvbfdb")) # 7

1. **从随机三元组中恢复秘密字符串**

**代码：**

def recoverSecret(triplets):

        l,l1,l2 = [],[],[]

        i = 0

        while triplets:

            i = int(len(l))

            for triplet in triplets:

                l1.append(triplet[0])

                l2.append(triplet[1])

                l2.append(triplet[2])

            for m in l1:

                if m not in l2:

                    l.append(m)

                    break

            l1,l2 = [],[]

            if i == int(len(l)):

                break

            for t in triplets:

                if l[-1] == t[0]:

                    t.pop(0)

                    t.append(0)

        return ''.join(x for x in l)

**输出：**

triplets = [

  ['t','u','p'],

  ['w','h','i'],

  ['t','s','u'],

  ['a','t','s'],

  ['h','a','p'],

  ['t','i','s'],

  ['w','h','s']

]

print(recoverSecret(triplets)) # whatisup

1. **去掉喷子的元音**

**代码：**

def disemvowel(string\_):

    for l in string\_:

        if l in 'aeiouAEIOU':

            string\_ = string\_.replace(l, '')

    return string\_

**输出：**

print(disemvowel("This website is for losers LOL!"))

# Ths wbst s fr lsrs LL!

1. **停止逆转我的单词**

**代码：**

def spin\_words(sentence):

    l = sentence.split()

    li = []

    for word in l:

        if len(word) >= 5:

            li.append(word[::-1])

        else:

            li.append(word)

    return ' '.join(x for x in li)

**输出：**

print(spin\_words("CodeWars")) # sraWedoC

print(spin\_words("Welcome")) # emocleW

print(spin\_words("function when the more and a the more five letter all words only the in in more string than letters letters"))

# noitcnuf when the more and a the more five rettel all sdrow only the in in more gnirts than srettel srettel

1. **Find The Parity Outlier**

**代码：**

def find\_outlier(integers):

    j,o = 0,0

    for i in integers:

        if i % 2 == 0:

            o+=1

        else:

            j+=1

        if o == 2:

            for ii in integers:

                if ii % 2 != 0:

                    return ii

        if j == 2:

            for ii in integers:

                if ii % 2 == 0:

                    return ii

    return None

**输出：**

print(find\_outlier([2, 4, 6, 8, 10, 3])) # 3

1. **数字之和/数字根**

**代码：**

def digital\_root(n):

    # n = str(n)

    # while len(n) > 1:

    #     sums = 0

    #     for i in range(len(n)):

    #         sums += int(n[i])

    #     n = str(sums)

    # return int(n)

    if n > 9:

        return n % 9

    else:

        return n

**输出：**

print(digital\_root(493193)) # 2

1. **数组的差**

**代码：**

def array\_diff(a, b):

    for i in range(len(b)):

        while b[i] in a:

            a.remove(b[i])

    return a

**输出：**

print(array\_diff([1,2,2], [2])) # [1]

1. **建塔**

**代码：**

def tower\_builder(n\_floors):

    l = []

    s = '\*'

    for i in range(n\_floors):

        while i > 0:

            s+='\*\*'

            i-=1

        l.append(s.center(n\_floors \* 2 - 1))

        s = '\*'

    return l

**输出：**

print(tower\_builder(3)) # [' \* ', ' \*\*\* ', '\*\*\*\*\*']

1. **检测Pangram**

**代码：**

def is\_pangram(s):

    s = s.lower()

    l = ['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']

    for i in l:

        if i not in s:

            return False

    return True

**输出：**

print(is\_pangram("The quick, brown fox jumps over the lazy dog!"))

# True

1. **哈梅林耳聋的老鼠**

**代码：**

def count\_deaf\_rats(town):

    s = town.replace(' ', '')

    l = s.index('P')

    i = 0

    sums = 0

    while i < l:

        if s[i:i+2] != '~O':

            sums += 1

        i += 2

    i += 1

    while i > l and i < len(s):

        if s[i:i+2] != 'O~':

            sums += 1

        i += 2

    return sums

**输出：**

print(count\_deaf\_rats("~O~O~O~OP~O~OO~")) # 2

1. **数独解决方案验证**

**代码：**

def validate\_sudoku(board):

    if 0 in board:

        return False

    r,r1,r2 = [],[],[]

    i = 0

    while i < 9:

        for j in range(9):

            r = board[j]

            for n in range(1, 10):

                if n not in r:

                    return False

            r1.append(r[i])

        for k in range(9):

            if k+1 not in r1:

                return False

        r1 = []

        i += 1

    i = 0

    while i < 9:

        for j in range(0, 9):

            r = board[j]

            r2.append(r[i])

            r2.append(r[i+1])

            r2.append(r[i+2])

            if j % 3 == 2:

                for a in range(9):

                    if a+1 not in r2:

                        return False

                r2 = []

        i += 3

    return True

**输出：**

print(validate\_sudoku([[1,3,2,5,7,9,4,6,8],

                  [4,9,8,2,6,1,3,7,5],

                  [7,5,6,3,8,4,2,1,9],

                  [6,4,3,1,5,8,7,9,2],

                  [5,2,1,7,9,3,8,4,6],

                  [9,8,7,4,2,6,5,3,1],

                  [2,1,4,9,3,5,6,8,7],

                  [3,6,5,8,1,7,9,2,4],

                  [8,7,9,6,4,2,1,5,3]])) # True

print(validate\_sudoku([[5,5,5,5,5,5,5,5,5],

                  [5,5,5,5,5,5,5,5,5],

                  [5,5,5,5,5,5,5,5,5],

                  [5,5,5,5,5,5,5,5,5],

                  [5,5,5,5,5,5,5,5,5],

                  [5,5,5,5,5,5,5,5,5],

                  [5,5,5,5,5,5,5,5,5],

                  [5,5,5,5,5,5,5,5,5],

                  [5,5,5,5,5,5,5,5,5]])) # False

1. 将零移到末尾

**代码：**

def move\_zeros(lst):

    l = len(lst)

    i,j = 0,0

    while i < l:

        if lst[i] == 0:

            lst.pop(i)

            lst.append(0)

            i -= 1

        j += 1

        if j == l:

            break

        i += 1

    return lst

**输出：**

print(move\_zeros([9, 0, 0, 9, 1, 2, 0, 1, 0, 1, 0, 3, 0, 1, 9, 0, 0, 0, 0, 9])) # [9, 9, 1, 2, 1, 1, 3, 1, 9, 9, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

### 七. 实验小结

通过本次实验，我发现 Python 是一种简单、易学、高效、可扩展的编程语言，具有广泛的应用领域和良好的发展前景。学习 Python 对我今后的学习和工作都有很大的帮助，它给了我跟Java完全不同的编程体验，拓展了我的思维。每一次写Python都在感叹它的简洁开放，尤其是类型库这一方面。在实验过程中，我也体会到了编程的乐趣和挑战，这将激励我更加努力地学习和研究编程技术。