实验三 Python列表

班级： 21计科02

学号： B20210202314

姓名： 朱华畅

Github地址：https://github.com/Lakzhu/python_project

CodeWars地址：https://www.codewars.com/users/Lak朱

实验目的

1. 学习Python的简单使用和列表操作
2. 学习Python中的if语句

实验环境

1. Git
2. Python 3.10
3. VSCode
4. VSCode插件

实验内容和步骤

第一部分

Python列表操作

完成教材《Python编程从入门到实践》下列章节的练习：

- 第3章 列表简介
- 第4章 操作列表
- 第5章 if语句

第二部分

在Codewars网站注册账号，完成下列Kata挑战：

第一题：3和5的倍数（Multiples of 3 or 5）

难度： 6kyu

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。

提示：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。 使用sum函数可以获取这个列表所有元素的和.

代码提交地址： https://www.codewars.com/kata/514b92a657cdc65150000006

第二题： 重复字符的编码器（Duplicate Encoder）

难度： 6kyu

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

---

第三题：括号匹配（Valid Braces）

难度：6kyu

写一个函数，接收一串括号，并确定括号的顺序是否有效。如果字符串是有效的，它应该返回True，如果是无效的，它应该返回False。 例如：

```
"(){}[]" => True
"([{}])" => True
 "(}" => False
 "[(])" => False
"[({})](]" => False
```

提示： python中没有内置堆栈数据结构，可以直接使用`list`来作为堆栈，其中`append`方法用于入栈，`pop`方法可以出栈。

代码提交地址 https://www.codewars.com/kata/5277c8a221e209d3f6000b56

第四题： 从随机三元组中恢复秘密字符串(Recover a secret string from random triplets)

难度： 4kyu

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

提示：

- 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，`secret`长度也等于该集合。

```
letters = {letter for triplet in triplets for letter in triplet }
length = len(letters)
```

- 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。
- 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。
- 遍历字母集合letters，利用上面2个函数得到最后的结果`secret`。

第五题： 去掉喷子的元音（Disemvowel Trolls）

难度： 7kyu

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

提示：

- 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。
- 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"
```

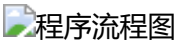
第三部分

使用Mermaid绘制程序流程图

安装VSCode插件：

- Markdown Preview Mermaid Support
- Mermaid Markdown Syntax Highlighting

使用Markdown语法绘制你的程序绘制程序流程图（至少一个），Markdown代码如下：

 程序流程图

显示效果如下：

```
flowchart LR
    A[Start] --> B{Is it?}
    B -->|Yes| C[OK]
    C --> D[Rethink]
    D --> B
    B ---->|No| E[End]
```

查看Mermaid流程图语法-->点击这里

使用Markdown编辑器（例如VScode）编写本次实验的实验报告，包括实验过程与结果、实验考查和实验总结，并将其导出为 **PDF格式** 来提交。

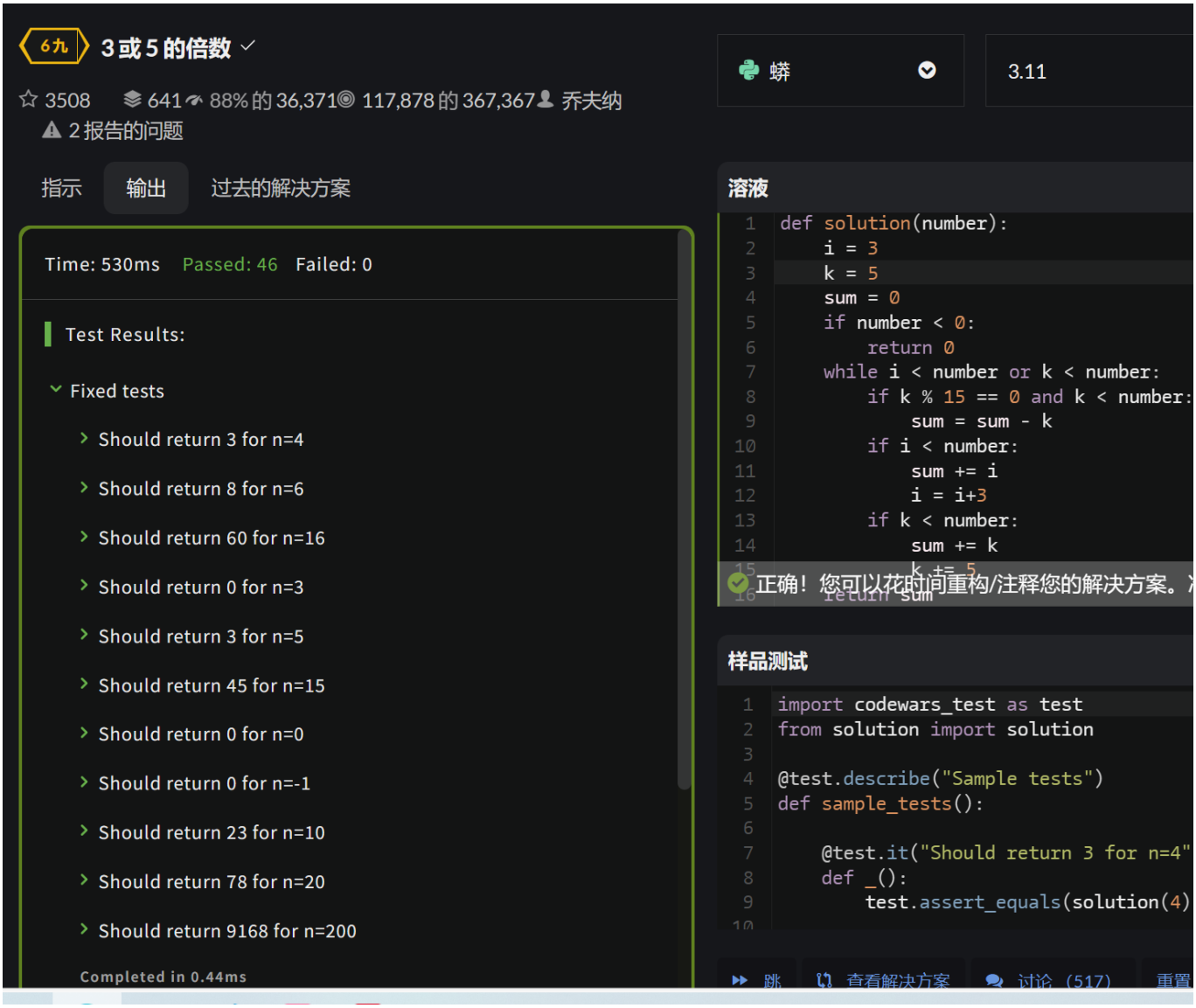
实验过程与结果

请将实验过程与结果放在这里，包括：

- 第一部分 Python列表操作和if语句
- 第二部分 Codewars Kata挑战
- 第三部分 使用Mermaid绘制程序流程图

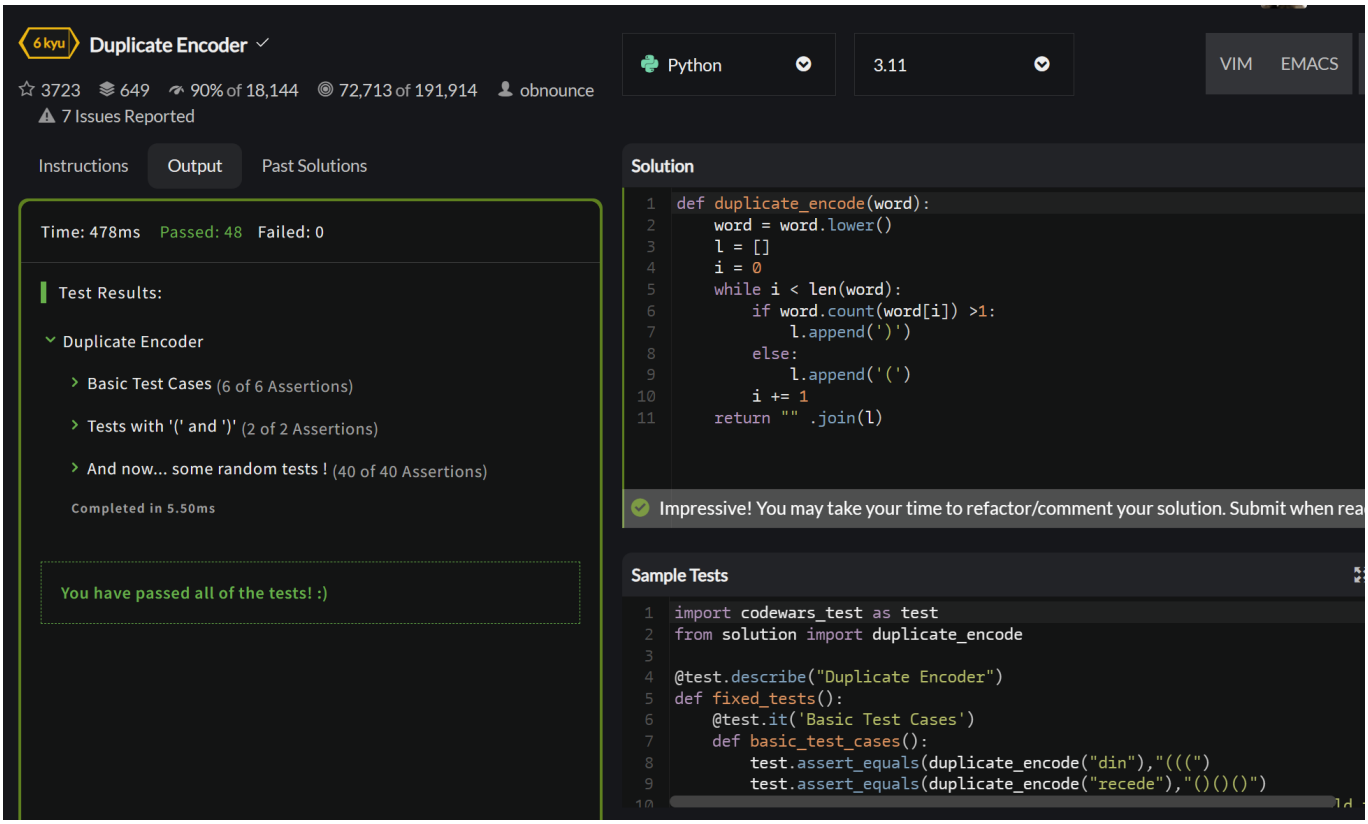
第一题：3和5的倍数（Multiples of 3 or 5）

```python
def solution(number):
    i = 3
    k = 5
    sum = 0
    if number < 0:
        return 0
    while i < number or k < number:
        if k % 15 == 0 and k < number:
            sum = sum - k
        if i < number:
            sum += i
            i = i + 3
        if k < number:
            sum += k
            k = k + 5
    return sum
```



第二题： 重复字符的编码器（Duplicate Encoder）

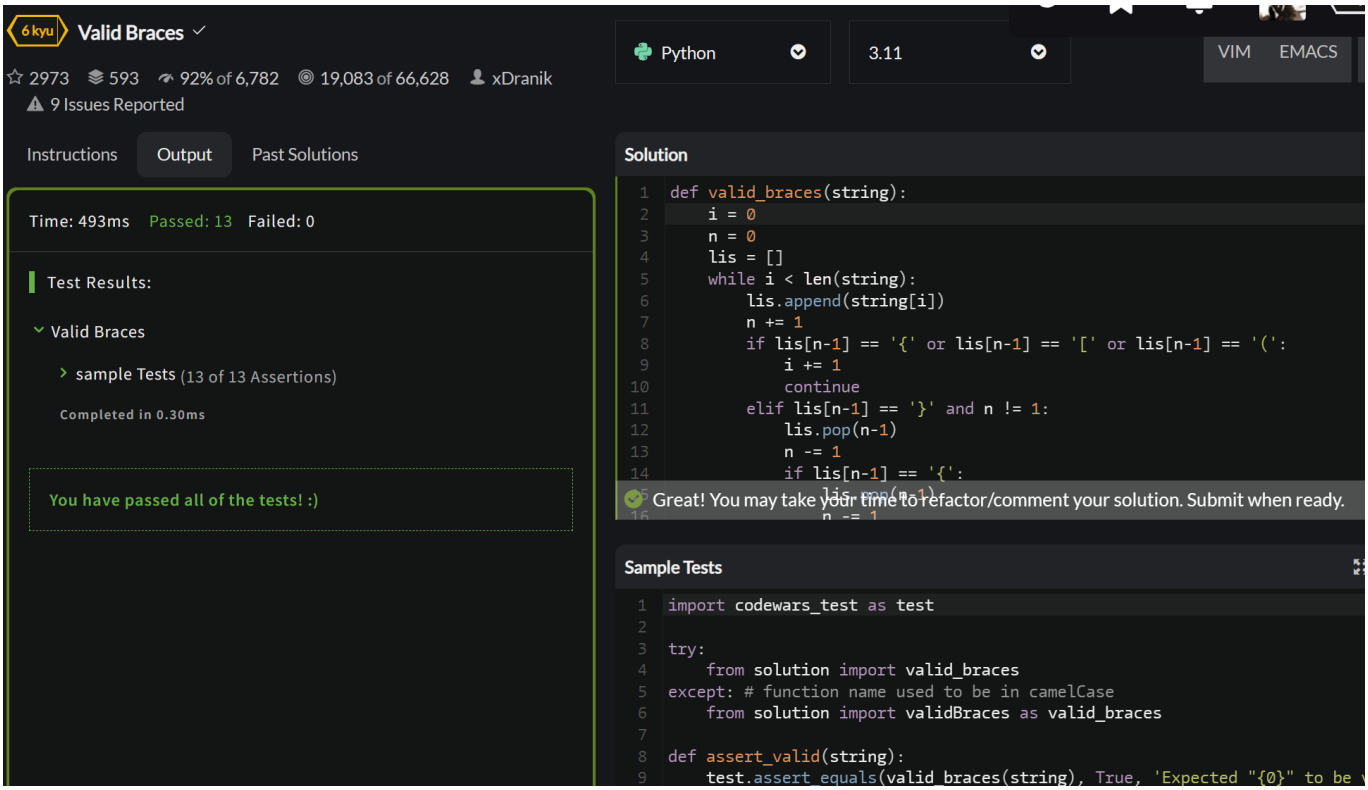
```python
def duplicate_encode(word):
    word = word.lower()
    l = []
    i = 0
    while i < len(word):
        if word.count(word[i]) >1:
            l.append(')')
        else:
            l.append('(')
        i += 1
    return "" .join(l)
```




```mermaid
flowchart LR
    A[word] --> B[将word转换为小写]
    B --> C[定义列表 l]
    C --> D[定义i = 0]
    D --> E{ i 是否小于 word的长度?}
    E ---->|YES| F{统计word【i】字母出现的次数是否大于1}
    E ---->|NO| G[返回空字符串]
    F ---->|YES| H[将字母) 存入列表l中]
    F ---->|NO| I[将字母 (存入列表l中]
    H --> J[i++]
    K --> J[i++]
    J --> E
```

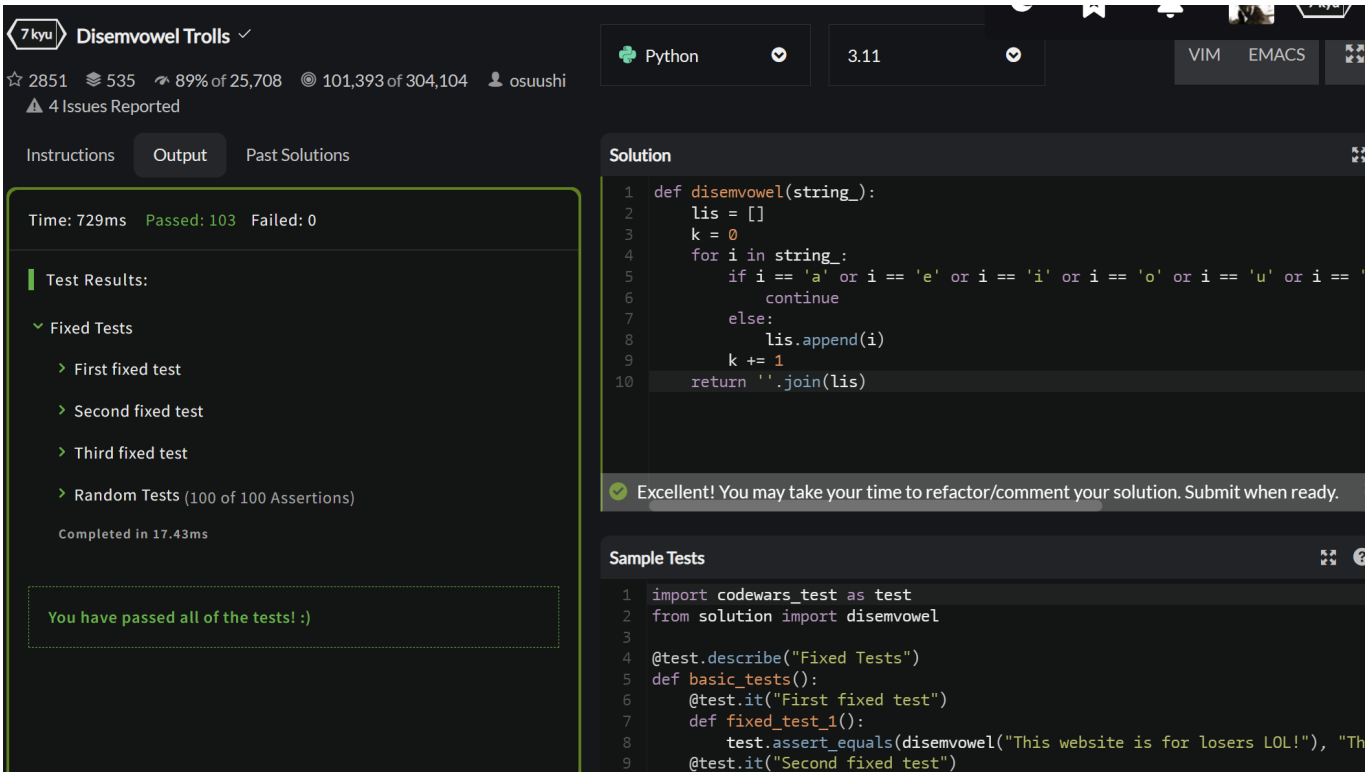
第三题：括号匹配（Valid Braces）

```
def valid_braces(string):
    i = 0
    n = 0
    lis = []
    while i < len(string):
        lis.append(string[i])
        n += 1
        if lis[n-1] == '{' or lis[n-1] == '[' or lis[n-1] == '(':
            i += 1
            continue
        elif lis[n-1] == '}' and n != 1:
            lis.pop(n-1)
            n -= 1
            if lis[n-1] == '{':
                lis.pop(n-1)
                n -= 1
            else: return False
        elif lis[n-1] == ']' and n != 1:
            lis.pop(n-1)
            n -= 1
            if lis[n-1] == '[':
                lis.pop(n-1)
                n -= 1
            else: return False
        elif lis[n-1] == ')' and n != 1:
            lis.pop(n-1)
            n -= 1
            if lis[n-1] == '(':
                lis.pop(n-1)
                n -= 1
            else: return False
        i += 1
    if '(' in lis or '{' in lis or '[' in lis:
        return False
    return True
```



第五题： 去掉喷子的元音（Disemvowel Trolls）

```
def disemvowel(string_):
    lis = []
    k = 0
    for i in string_:
        if i == 'a' or i == 'e' or i == 'i' or i == 'o' or i == 'u' or i == 'A' or
i == 'E' or i == 'I' or i == 'O' or i == 'U':
            continue
        else:
            lis.append(i)
        k += 1
    return ''.join(lis)
```



实验考查

请使用自己的语言并使用尽量简短代码示例回答下面的问题，这些问题将在实验检查时用于提问和答辩以及实际的操作。

1. Python中的列表可以进行哪些操作？ 添加元素：使用append()方法在列表末尾添加元素，或使用insert()方法在特定位置插入元素。

   删除元素：使用remove()方法通过值删除元素，或使用del关键字通过索引删除元素。

   访问元素：使用索引访问列表中的元素

   切片：使用切片操作符[]来获取列表的子集.

   合并列表：使用+操作符将两个列表合并为一个新的列表。

   列表长度：使用len()函数获取列表的长度。

   排序和反转：使用sort()方法对列表进行排序，使用reverse()方法将列表中的元素反转。

   元素计数和查找：使用count()方法计算列表中特定元素的出现次数，使用index()方法查找元素的索引位置。

   列表复制：使用copy()方法复制一个列表。

2. 哪两种方法可以用来对Python的列表排序？这两种方法有和区别？

   sort()方法：这是列表对象的一个方法，用于原地对列表进行排序。例如，my_list.sort()会将列表my_list按升序重新排列。该方法对原始列表进行修改，不返回新的排序后的列表。

   sorted()函数：这是一个内置函数，用于对列表进行排序并返回一个新的排序后的列表。例如，sorted_list = sorted(my_list)会创建一个新的列表sorted_list，其中包含了对原始列表my_list进行排序后的

元素。

3. 如何将Python列表逆序打印?

   使用内置函数 reversed () 内置函数 reversed () 可以将一个可迭代对象的元素逆序输出。

4. Python中的列表执行哪些操作时效率比较高？哪些操作效率比较差？是否有类似的数据结构可以用来替代列表?

   列表的访问效率比其他数据结构要高，因为列表中的元素是连续存储的。列表的插入和删除效率比较低，因为需要重新分配内存并移动元素。

5. 阅读《Fluent Python》Chapter 2. An Array of Sequence - Tuples Are Not Just Immutable Lists小节（p30-p35）。总结该小节的主要内容。

实验总结

总结一下这次实验你学习和使用到的知识，例如：编程工具的使用、数据结构、程序语言的语法、算法、编程技巧、编程思想。

> 这次实验让我熟悉了基本的python语言，练习了python语法的使用，如while循环，for循环，if条件语句.....学习了列表的使用，学习了列表的基本操作，提升了python语言的编程能力。